



# Universidad Austral de Chile

Facultad de Ciencias de la Ingeniería  
Escuela de Ingeniería Civil Electrónica

## “DISEÑO E IMPLEMENTACIÓN DE EXPERIENCIAS DOCENTES PARA EL SERVICIO DE VOZ SOBRE IP, MEDIANTE LA UTILIZACIÓN DE LA PLATAFORMA ASTERISK IPBX”

Trabajo para optar al título de:  
Ingeniero en Electrónica

Profesor Patrocinante:  
Sr. Pedro Rey Clericus.  
Ingeniero Electrónico,  
Licenciado en Ciencias de la Ingeniería,  
Diplomado en Ciencias de la Ingeniería.

ALEX HANS HUNTER MENA  
VALDIVIA - CHILE  
2007



Universidad Austral de Chile

Escuela Electricidad y Electrónica

UNIVERSIDAD AUSTRAL DE CHILE  
FACULTAD DE CIENCIAS DE LA INGENIERÍA  
ESCUELA DE ELECTRICIDAD Y ELECTRÓNICA

**DISEÑO E IMPLEMENTACIÓN DE EXPERIENCIAS DOCENTES PARA EL SERVICIO  
DE VOZ SOBRE IP, MEDIANTE LA UTILIZACIÓN DE LA PLATAFORMA  
ASTERISK IPBX**

Trabajo de Titulación para optar al  
Título de Ingeniero en Electrónica

PROFESOR PATROCINANTE:  
Sr. PEDRO REY CLERICUS

ALEX HANS HUNTER MENA

VALDIVIA 2007

---

**Profesor Patricinante: Sr. Ing. PEDRO REY CLERICUS**

---

**Profesor Informante: Sr. Ing. FRANKLIN CASTRO ROJAS**

---

**Profesor Informante: Sr. Ing. ROLAND GLÖCKLER**

**Fecha de examen de titulación: Valdivia \_\_\_\_\_ de \_\_\_\_\_ de \_\_\_\_\_.**

Agradezco a Dios por acompañarme en cada momento de mi vida, por darme la fuerza y perseverancia para seguir adelante, por alumbar mi camino y escucharme cada vez que lo he necesitado.

Doy las gracias a mi familia, en especial a mis padres, Alejandro y Angelica, por el gran esfuerzo y dedicación que han depositado en mí, por creer siempre que podría lograr cumplir esta meta tan importante en mi vida, por luchar junto en el cumplimiento de mis sueños. Espero poder hacerlos sentir orgullosos con este logro y salir adelante cada día con el mismo esfuerzo que ellos lo han hecho siempre. Valoraré por siempre esta gran oportunidad que me han dado.

Gracias a Macarena, mi pareja, ya que hizo cambiar mi vida para siempre, por acompañarme en los buenos y en los malos momentos, por ayudarme a madurar y entregarme todo su amor y apoyo cada día. Que Dios nos acompañe siempre para que logremos todo lo que hemos proyectado juntos.

Sinceramente quiero dar las gracias a todo el cuerpo docente de la facultad, en especial a los señores Pedro Rey C., Franklin Castro R., Julio Zarecht O., Nestor Fierro M., Renato Loaiza y Roland Glöckler, por que de cada uno de ellos logre aprender cosas importantes de la vida y sobre todo a mirar más allá en mi futuro como profesional.

Agradezco a Alvaro por su tiempo y paciencia para enseñarme sus conocimientos en Linux, a mis colegas, con quienes estude esta hermosa carrera sin límites y por las amistades que logramos hacer en esta etapa tan especial e inolvidable de nuestra vida.

<b>ÍNDICE</b>	IV
<b>PORTADA</b> .....	I
<b>COMISIÓN REVISORA</b> .....	II
<b>AGRADECIMIENTOS</b> .....	III
<b>INDICE</b> .....	IV
<b>RESUMEN</b> .....	IX
<b>SUMMARY</b> .....	X
<b>1. CAPITULO I. INTRODUCCIÓN</b> .....	1
1.1. Objetivo General.....	2
1.2. Objetivos Específicos.....	2
<b>2. CAPITULO II. REDES DE COMPUTADORES</b> .....	3
2.1. Jerarquía de Protocolos.....	3
2.2. Clasificación de Servicios.....	4
2.3. Modelos de Referencia.....	5
2.3.1. Modelo OSI.....	5
2.3.2. Modelo TCP/IP.....	8
2.4. Elementos de una Red.....	9
2.5. Protocolos de Internet.....	11
2.6. Aplicaciones en Tiempo Real.....	12
2.6.1. Condiciones de Red y características de tráfico en tiempo real.....	12
2.6.2. Calidad de Servicio.....	13
2.6.3. Arquitectura de servicios integrados.....	14
2.6.4. Protocolo de reserva de recurso.....	14
2.6.5. Protocolo de tiempo real (RTP).....	14
2.6.6. Estandarización de sistemas de Multimedia.....	17
2.7. Codificación de Voz.....	17

<b>3. CAPITULO III. CODIFICACIÓN DE VOZ APLICADA A REDES DE PAQUETES..</b>	<b>21</b>
3.1. Codificadores tipo Waveform.....	21
3.1.1. Codificación PCM.....	22
3.1.2. Codificación ADPCM.....	23
3.1.3. Otras tecnologías.....	25
3.2. Codificadores tipo Voice Coder.....	26
3.2.1. Codificadores CELP.....	26
3.3. Calidad de Codificadores.....	29
3.3.1. Métodos Subjetivos.....	30
3.3.2. Métodos Objetivos.....	32
3.4. Codificación de Voz en Redes de Paquetes.....	34
3.4.1. Parámetros de codificación.....	34
3.4.2. Influencia del Retardo.....	36
3.4.3. Influencia del Jitter.....	38
3.4.4. Influencia de las Pérdidas de Paquetes.....	39
3.4.5. Influencia del Eco.....	40
<b>4. CAPITULO IV. VOZ SOBRE IP (VOIP).....</b>	<b>41</b>
4.1. Estado del Arte.....	41
4.1.1. Evolución Histórica hacia la VoIP.....	41
4.1.2. Antecedentes de desarrollo de VoIP.....	41
4.2. Principales componentes de la VOIP.....	48
4.3. Encapsulamiento de una trama VoIP.....	49
4.4. Funcionamiento de una red VoIP.....	49
4.5. Protocolos de Señalización.....	52
4.5.1. H.323.....	52
4.5.2. Protocolo SIP.....	53
4.5.3. MGCP-MEGACO.....	62
4.5.4. IAX.....	62
4.6. Tipos de Arquitecturas.....	63
4.6.1. Arquitectura centralizada.....	63

	VI
4.6.2. Arquitectura distribuida.....	64
4.7. Centrales Telefónicas.....	65
4.7.1. Funcionalidades.....	66
4.7.2. Interfaces de acceso a la red pública.....	68
4.7.3. IPBX.....	70
<b>5. Capítulo V. ASTERISK.....</b>	<b>71</b>
5.1. Asterisk IPBX.....	71
5.2. El Rol de Digium.....	72
5.3. El proyecto Zapata.....	73
5.4. Reducción de Costos Utilizando Asterisk.....	73
5.5. Limitaciones de la arquitectura de Asterisk.....	74
5.6. Arquitectura de Asterisk.....	74
5.6.1. Canales.....	75
5.6.2. Codecs y Conversores de CODEC.....	76
5.6.3. Protocolos.....	77
5.6.4. Aplicaciones.....	77
5.7. Escenarios de uso de Asterisk.....	78
5.7.1. Visión General.....	78
5.8. Telefonía con Asterisk.....	79
5.8.1. Sistema PABX 1x1.....	79
5.8.2. Ampliación de PABX usando un banco de canales.....	80
5.8.3. Intercomunicación de sitios remotos con un sitio central. ....	81
<b>6. CAPITULO VI. IMPLEMENTACIÓN.....</b>	<b>83</b>
6.1. Consideraciones sobre la instalación de Asterisk.....	83
6.1.1. Sistemas en producción.....	83
6.1.2. Consideraciones sobre la red.....	83
6.1.3. Escenario y especificaciones.....	84
6.2. Metodología para el desarrollo de las experiencias de laboratorio.....	86
6.2.1. Hipótesis de Trabajo.....	86
6.2.2. Perfil de los Alumnos.....	86

	VII
6.2.3. Infraestructura Necesaria.....	87
6.2.4. Estructura de las Guías de Laboratorio.....	89
6.2.5. Planificación de las Experiencias.....	89
<b>7. CAPITULO VII. CONCLUSIONES.....</b>	<b>91</b>

<b>BIBLIOGRAFÍA.....</b>	<b>95</b>
--------------------------	-----------

<b>ANEXO I. Guías de Laboratorio.....</b>	<b>101</b>
<b>ANEXO II. Guía de ayuda para experiencias de laboratorio.....</b>	<b>123</b>
<b>ANEXO III. Comandos de Linux.....</b>	<b>155</b>

## **INDICE DE FIGURAS**

Figura 1. Esquema genérico de una estructura de red.....	4
Figura 2. Capas del modelo OSI.....	6
Figura 3. Capas y equivalencia de los modelos de referencia OSI y TCP/IP.....	8
Figura 4. Protocolos implementados usando el modelo de referencia TCP/IP.....	9
Figura 5. Cabecera del protocolo RTP.....	15
Figura 6. Procesamiento básico de codificación de voz.....	21
Figura 7. Cuantización uniforme de una señal.....	22
Figura 8. Diagrama en bloques simplificado del codificador ITU-T G.72.....	24
Figura 9. Diagrama en bloques simplificado del decodificador ITU-T G.726.....	25
Figura 10. Esquema básico de codificadores CELP.....	27
Figura 11. Esquema básico de decodificadores CELP.....	29
Figura 12. Ejemplo de una trama VoIP sobre una red LAN y WAN.....	49
Figura 13. Flujo de una llamada VoIP.....	50
Figura 14. Router, Gateway de VoIP.....	50
Figura 15. PABX y Router en VoIP.....	50
Figura 16. Distribución de los protocolos VoIP dentro del modelo OSI.....	51
Figura 17. Establecimiento y liberación de sesión SIP.....	58
Figura 18. Arquitectura de ASTERISK.....	74
Figura 19. Esquema híbrido de comunicaciones IP y TDM.....	78

Figura 20. Ejemplo de un PABX de una troncal y una línea analógica.....	80
Figura 21. Ampliación de Canales VOIP.....	80
Figura 22. Funcionalidad de un Gateway.....	81
Figura 23. Escenario Global de VoIP y Telefonía IP.....	85
Figura 24. Esquema de red utilizado para trabajo de tesis.....	87

## **INDICE DE TABLAS**

Tabla 1. Escala de 5 puntos de MOS.....	31
Tabla 2. Codificadores más utilizados en el servicio de Telefonía y Voz sobre IP.....	33
Tabla 3. Calidad e interactividad de una llamada, según el retardo de transmisión.....	36
Tabla 4. Guía de requisitos del sistema para Asterisk.....	84
Tabla 5. Detalle de Equipos utilizados.....	88
Tabla 6. Planificación de experiencias de laboratorio.....	89

## **RESUMEN**

El presente proyecto se orientó al desarrollo de una plataforma de VoIP, con base en el diseño e implementación de un sistema de comunicación de voz sobre IP a través de una red LAN conectada por protocolo TCP/IP, mediante el uso de herramientas de software de código abierto. La plataforma de VoIP que se propone es abierta, flexible, escalable y está basada en librerías abiertas. Además sigue el principio de que todo servicio multimedia puede ser accesible mediante el protocolo H.323 o SIP, sea cual sea la naturaleza del servicio.

Por lo tanto, el principal objetivo es presentar una solución que pueda servir para mejorar los sistemas de comunicación y de mensajería de voz, ya sea en un entorno residencial o de una pequeña organización. Para este propósito se tuvieron en cuenta los análisis previos en cuanto a los fundamentos teóricos y prácticos de todos los aspectos relacionados con el tema para posteriormente formular la propuesta de diseño de la aplicación.

## **SUMMARY**

This project is devoted to the development of a Voice over Internet Protocol (VoIP) platform, with base on the design and implementation of a communication of voice on IP system through a LAN network connected by TCP/IP protocol, by means of the use of open code software tools. The implemented VoIP platform is open source, flexible and scalable, and is based on open source libraries. In addition it follows the principle that all multimedia service can be accessible by means of the H.323 or SIP protocols, whatever the nature of the service.

Therefore, the main objective of this work is to present a solution that can serve to improve the communication systems of voice mail boxes in home or small organizations. For this purpose previous analysis were taken into account related to theoretical and practical foundations of all the aspects of the subject to formulate later the proposal of design of the application.

## 1. CAPITULO I. INTRODUCCIÓN

La convergencia de los servicios telefónicos en las redes de datos está marcando el inicio de la unificación de los principales servicios de una organización en una sola red y la facilidad para el usuario de manejar todos sus recursos desde su computador. Así mismo, permite el manejo de la tecnología en la penetrante y amplia expansión de las redes IP en las redes de área local (LAN) y las redes de área amplia (WAN).

La investigación propuesta, busca mediante la aplicación de la teoría y los conceptos de protocolos base para los servicios en tiempo real de transmisión de voz VoIP, poder desarrollar una solución de software de código abierto que sirva para la transmisión de voz a través de una red LAN.

En la actualidad se utilizan productos basados en los protocolos RTP (Real Time Protocol) y RTCP (Real Time Control Protocol) para proveer servicios multimedia de voz sobre redes IP, pero la mayoría tiene un costo por el software de éstos, sin embargo, en Linux se encuentran algunas aplicaciones de VoIP bajo licencia GPL (General Public License), las cuales se pueden implementar de forma gratuita.

Teniendo en cuenta los anteriores aspectos por medio de esta tesis, se pretende desarrollar una herramienta de transmisión de voz VoIP sobre una red LAN con base en un software de código abierto, de forma de generar experiencias de laboratorio para los alumnos de la carrera de Ingeniería Civil Electrónica de la Universidad Austral de Chile. Para ello los usuarios se comunicarán con la asistencia de un equipo servidor usando computadores que deben tener un software de cliente para la aplicación mencionada. Por lo tanto, lo que se pretende es desarrollar una propuesta sencilla y de fácil manejo, con el fin de que pueda ser implementada en cualquier red LAN, ya que de esta forma permitirá a los alumnos comprender en forma práctica tanto el funcionamiento como la implementación de esta tecnología, crear sus propios criterios y a la vez generar ideas que permitan desarrollar nuevas aplicaciones.

### **1.1. OBJETIVO GENERAL**

- Generar experiencias de laboratorios orientadas a la Implementación un sistema de comunicación basado en voz sobre IP a través de una red LAN conectada por protocolo TCP/IP de pequeña escala y baja inversión.

### **1.2. Objetivos Específicos**

- Analizar tecnologías y estándares para la transmisión de voz sobre IP.
- Determinar qué tipo de aplicaciones se pueden implementar sobre conexiones de voz sobre IP.
- Definir la metodología que más se acomode para la implementación del proyecto, objeto de investigación.
- Especificar los requerimientos de los diversos elementos que interactúan y hacen posible la comunicación de voz sobre IP.
- Crear experiencias de laboratorio que permitan al alumno conocer, comprender y aplicar, el como opera un servicio de VoIP de baja escala.

## 2. CAPITULO II. REDES DE COMPUTADORES

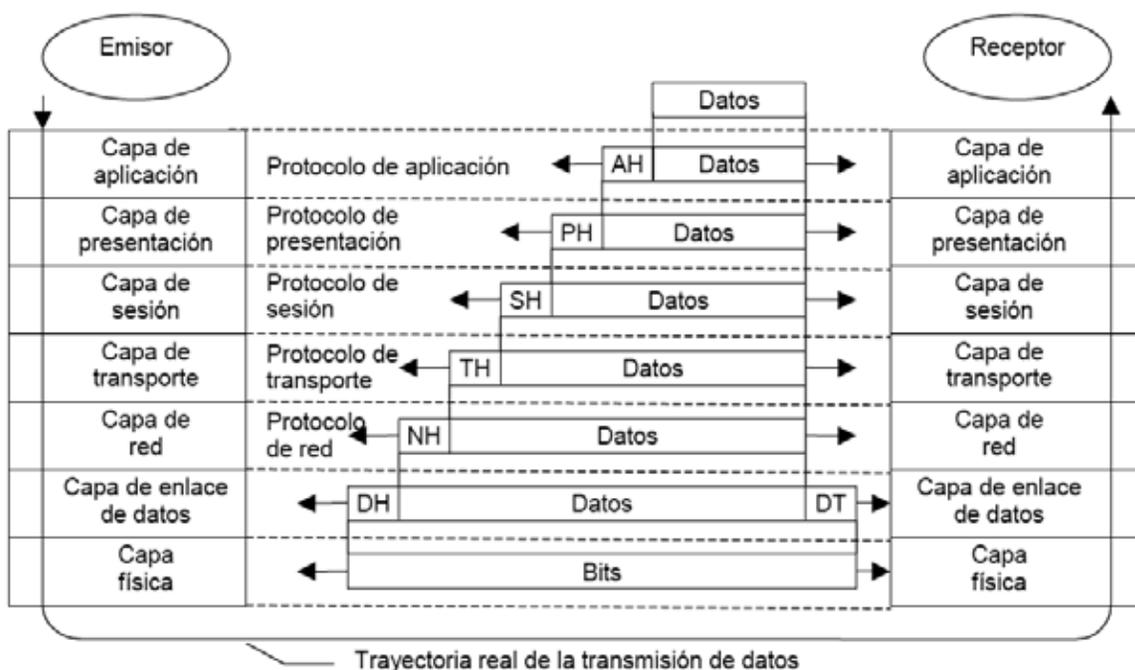
La estructura de las redes de computadores varía dependiendo de su tamaño, tecnología y topología. Las redes de propiedad privada dentro de un sólo edificio o instalaciones próximas entre sí, son llamadas redes de área local (*Local Area Network*, LAN). Se usan ampliamente para conectar computadores personales y estaciones de trabajo en oficinas de compañías y fábricas. Dos de las principales redes de difusión son las topologías de buses y anillos normadas por los estándares IEEE 802.3 y IEEE 802.5 respectivamente. Para abarcar grupos de oficinas corporativas, ciudades o redes privadas o públicas de mayor tamaño se utilizan las redes de área metropolitana (*Metropolitan Area Network*, MAN). Finalmente para conectar países o continentes se utilizan redes de área amplia (*Wide Area Network*, WAN), cuya función es principalmente interconectar redes usando líneas de transmisión y elementos de conmutación.

La capacidad de transmisión, el retardo, el costo y las facilidades de instalación y mantenimiento de una red quedan determinadas por el medio de transmisión que se utilice. Se pueden usar diferentes medios físicos para interconectar equipos como par trenzado, cable coaxial, fibra óptica y sistemas inalámbricos dependiendo de las características propias que se requieran de la red.

### 2.1. Jerarquía de protocolos

Para reducir la complejidad de diseño, las redes de computadores han sido implementadas en su mayoría utilizando el concepto de capas o niveles. En este esquema, cada capa realiza una determinada función que sirve para ofrecer ciertos servicios a las capas siguientes. Tanto el número como la funcionalidad de los niveles varían de red en red.

Cada capa interactúa sólo con la capa del mismo nivel de la otra máquina. Esta interacción no es física sino que virtual, o sea, los datos se encapsulan hasta llegar a la capa correspondiente del otro *host*. Los protocolos son el conjunto de reglas y convenciones definidas para la interacción de cada nivel y que, en conjunto con el grupo de capas, definen una arquitectura de red. La figura 1 muestra un esquema genérico de una estructura de red.



**Figura 1.** Esquema genérico de una estructura de red. Los encabezados DH, NH, TH, SH, PH y DT contienen la información necesaria para que las capas se comuniquen entre sí.

## 2.2. Clasificación de servicios

Los servicios que se ofrecen entre capas pueden ser orientados o no a la conexión. Un servicio orientado a conexión es similar al sistema telefónico, en el cual antes de realizar una comunicación, es necesario establecer una conexión entre los usuarios. De manera similar, un servicio de red orientado a la conexión deberá establecer, utilizar y liberar la conexión. La transmisión actúa como un tubo en el cual los datos son enviados de una máquina a la otra ordenadamente (*First In First Out*, FIFO). Un servicio no orientado a la conexión es similar al sistema de correos, en donde lo único que se especifica es la dirección de destino. No se garantiza que el orden en que son recibidos los datos sea el mismo que el orden en el que fueron enviados, por lo que no se requiere establecer conexión.

Otra forma de clasificar los servicios es a través de la confiabilidad. Un servicio confiable es aquel en el cual se garantiza que los datos son transmitidos sin pérdidas. La implementación de sistemas confiables generalmente incorpora acuse de recibo para que

el transmisor retransmita mensajes perdidos. Este sistema introduce sobrecarga y retardos por lo que no es posible utilizarlo en algunos servicios.

Las aplicaciones de transferencia de archivos generalmente son implementadas como servicios confiables orientados a conexión. Sin embargo, para aplicaciones en tiempo real no es aceptable retardos ni sobrecarga de tráfico por lo que son implementados como servicios no confiables y no orientados a la conexión, salvo en aplicaciones implementadas sobre redes privadas corporativas, como es el caso de la telefonía IP.

### **2.3. Modelos de referencia**

Los modelos de red basados en capas, los más importantes son el modelo OSI (*Open Systems Interconnection*) y el TCP/IP. El primero, aceptado internacionalmente, es un modelo teóricamente completo que sirve para describir otras arquitecturas de red en forma abstracta. Por otro lado, el modelo TCP/IP, a pesar de carecer de ciertas funcionalidades, ha sido el esquema más implementado. Por estas razones, ambos modelos son analizados.

#### **2.3.1. Modelo OSI**

La organización de estándares internacionales (*International Standards Organization*, ISO) desarrolló un modelo de red basado en capas denominado modelo de referencia OSI. El modelo es robusto y abierto, lo que permite utilizarlo para explorar las especificaciones de red ya existentes y diseñar los estándares del futuro. En total, el modelo utiliza siete niveles para describir la red, siendo estos las capas de nivel físico, enlace de datos, red, transporte, sesión, presentación y aplicación, como se ve en la figura 2. A continuación se describe, brevemente, el funcionamiento de cada nivel.



**Figura 2.** Capas del modelo OSI

### Capa física

Esta capa es la encargada de la transmisión real de los bits de un *host* a otro utilizando un medio físico. Se definen interfaces mecánicas, eléctricas y de procedimiento. Define las técnicas para transferir una cadena de bits por un cable y los equipos a utilizar como tarjetas de red, tipo de cable, tipo de conectores etc. Entre los protocolos implementados en este nivel destacan IEEE 802, IEEE 802.2 y ISO 2110 ISDN. Los componentes de red utilizados son repetidores, multiplexadores, *hubs*, amplificadores, etc.

### Capa de enlace

Esta capa es la responsable que los datos transmitidos por el medio físico lleguen sin error al receptor. Para ello los bits son agrupados y delimitados en tramas o paquetes y por medio de acuse de recibo se asegura que lleguen correctamente a la capa de red. Es aquí donde se corrigen las tramas dañadas, perdidas y duplicadas. Se regula el tráfico, evitando que transmisores veloces saturen a transmisores lentos y, en medios físicos compartidos, se regula la asignación de recursos para conexiones dúplex y símplex. El nivel de enlace constituye el puente entre el hardware y el software. Entre los protocolos implementados en este nivel destacan IEEE 802.1, 802.2, 802.3, 802.4, 802.5 y 802.12. Los componentes de red utilizados son *bridge*, *switch*, ISDN *router*, etc.

## **Capa de red**

Esta capa debe interconectar subredes heterogéneas, resolviendo problemas como direccionamiento, congestión de datos, incompatibilidad en tamaños de paquetes, etc. Esta capa deberá traducir nombres y direcciones lógicas de red a direcciones físicas. Debe determinar rutas de modo interconectar subredes. Entre los protocolos implementados en este nivel destacan IP, ICMP, RIP, OSFP, IGMP, ARP y RARP. Los dos últimos tienen por función traducir las direcciones. Los componentes de red utilizados son *brouter*, *router*, *frame relay device*, *ATM switch*, etc.

## **Capa de transporte**

La función principal de la capa de transporte es aceptar datos de la capa de sesión, dividirlos en paquetes pequeños si es necesario, pasarlos a la capa de red y asegurarse que lleguen correctamente al *host* de destino. La funcionalidad de esta capa dependerá si el servicio es o no confiable y si es o no orientado a conexión. Puede ofrecer control de flujo, variando la tasa de transmisión en caso necesario y acuse de recibo, para asegurar que los paquetes lleguen correctamente. Entre los protocolos implementados en este nivel destacan TCP, SPX, UDP y ATP.

## **Capa de Sesión**

Esta capa debe establecer, mantener y terminar sesiones a través de la red entre equipos diferentes. Debe identificar los nombres de los equipos participantes de la conexión y negar el acceso al resto. En caso de interrupción en la conexión, esta capa debe reanudar la transmisión sin tener que enviar todo nuevamente. Debe permitir que el tráfico se envíe en ambas o en una sola dirección al mismo tiempo, dependiendo del servicio. Entre los protocolos implementados en este nivel destacan *NetBIOS* y *RPC*.

## **Capa de presentación**

La capa de presentación se ocupa de la sintaxis y la semántica de la información que se transmite permitiendo que tecnologías con estructuras de datos diferentes puedan

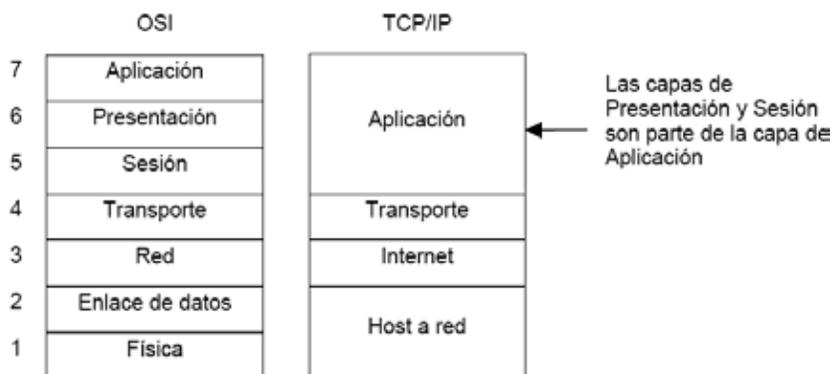
interactuar. Un ejemplo de esto constituyen los códigos ASCII y Unicode para representar cadenas de caracteres. Esta capa puede proporcionar conexión segura en una red, encriptando y comprimiendo los datos.

### Capa de aplicación

Esta capa es utilizada para aplicaciones específicas que son ejecutadas en la red. Proporciona la interfaz con la red que los usuarios utilizarán para acceder a los servicios de red. Entre los protocolos implementados en este nivel destacan DNS, FTP, TFTP, TELNET, etc.

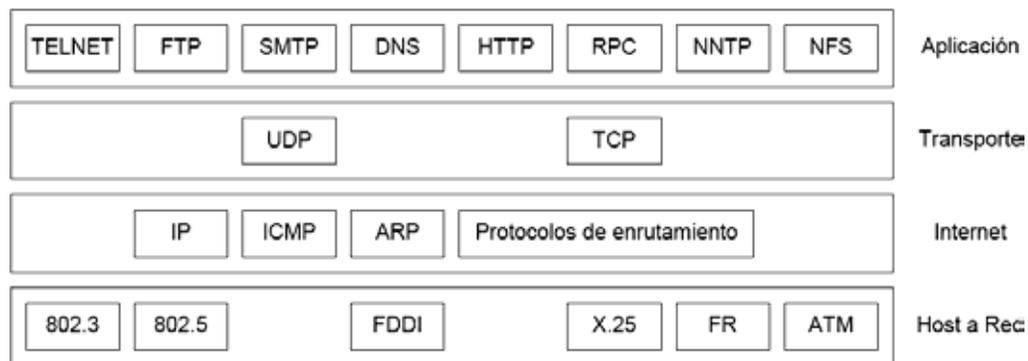
### 2.3.2. Modelo TCP/IP

El Departamento de Defensa de Estados Unidos patrocinó a mediados del siglo veinte el proyecto de investigación ARPANET, el cual buscaba conectar diferentes redes de computadores de universidades e instalaciones del gobierno. Producto de la diversidad de tecnologías utilizadas por estas redes, se hizo necesario la creación de un estándar que permitiera la interconexión de ellas. Se buscaba además que la red fuera robusta a ataques que dejaran fuera de servicio a alguno de sus nodos y que fuese flexible a numerosas aplicaciones que en esos años se proyectaban. Es así como en 1974 se define el modelo de referencia TCP/IP. La correspondencia entre el modelo OSI y el modelo TCP/IP se muestra en la figura 3.



**Figura 3.** Capas y equivalencia de los modelos de referencia OSI y TCP/IP

Las funciones de las capas internet, transporte y aplicación del modelo TCP/IP son análogas a las capas correspondientes del modelo OSI. Las capas de presentación y sesión no fueron implementadas en un principio pues muy pocas aplicaciones hacen uso de ellas. Sin embargo, debido al gran aumento de tráfico en tiempo real, se ha necesitado incorporar nuevos protocolos con similares funcionalidades. Bajo la capa de internet el modelo TCP/IP es flexible ya que sólo establece que se debe conectar a la red utilizando un protocolo que permita mandar paquetes, dejando abierta la posibilidad de interconectar tecnologías disímiles. Los protocolos más importantes de este modelo se muestran en la figura 4.



**Figura 4.** Protocolos implementados usando el modelo de referencia TCP/IP

## 2.4. Elementos de una Red

Para diseñar una red de computadores, se dispone de los siguientes dispositivos básicos: hubs, repetidores, bridges, switches y routers.

### Hub

Los hubs o concentradores, son usados para conectar múltiples usuarios a un dispositivo físico, el cual es conectado a la red. Los hubs o concentradores realizan la misma labor de los repetidores regenerando las señales que pasan a través de ellos, y operan en la capa física del modelo OSI (OSI capa 1). A diferencia de un repetidor que tiene pocas puertas, un hub repite las señales por varias puertas (de 5 a 24 puertas).

Debido a que operan en la capa física, donde en general no se considera direccionamiento, no poseen facultades para establecer dominios.

### **Bridge**

El bridge es empleado para separar lógicamente segmentos dentro de la misma red, y operan en la capa de datos del modelo OSI (OSI capa 2). Una de las principales funciones de un bridge es poder realizar la traducción de paquetes de tecnologías de capa 2.

### **Switch**

El switch es similar al bridge, pero usualmente tiene más puertas. El switch provee un único segmento de red por cada puerta (esto se conoce como microsegmentación) siendo capaz de separar dominios de colisión. De este modo los errores producidos por colisiones de paquetes en un segmento no son transmitidos a los otros. Actualmente los diseñadores de redes están reemplazando los hubs por switches de manera de incrementar las prestaciones y ancho de banda de una red, conservando las instalaciones de alambrado existentes. Al igual que los bridges, los switches operan en la capa de datos del modelo OSI (OSI capa 2).

### **Router**

Los routers son equipos diseñados para interconectar redes en el ámbito de la capa de red del modelo OSI (OSI capa 3). Dentro de sus capacidades está la de separar dominios de broadcast, lo que permite realizar un mejor empleo del ancho de banda en una red de gran extensión, de modo que los mensajes de difusión colectiva en una red sólo afectan a la red que los origina, sin utilizar los recursos de las redes vecinas. Los routers encaminan el tráfico de acuerdo al contenido del campo de direccionamiento de los paquetes de capa 3. Los routers son dependientes del protocolo. Las técnicas de diseño e implementación de las redes actuales utilizan routers y switches para la gestión del tráfico, debido a que además de presentar un mejor desempeño, ofrecen mecanismos de crecimiento más flexible y escalable. Las redes previas se construían utilizando bridges y hubs.

## 2.5. Protocolos de Internet

Los protocolos principales del modelo TCP/IP son el protocolo de Internet (*Internet Protocol*, IP), el protocolo de control de transmisión (*Transmisión Control Protocol*, TCP) y el protocolo de datagrama de usuario (*User Datagram Protocol*, UDP).

### Protocolo IP

Perteneciente a la capa de internet, el protocolo IP, permite enrutar información a través de diferentes tecnologías tanto de hardware como de software. Permite direccionar, fragmentar y corregir paquetes.

Dentro de los campos que posee este protocolo se encuentra uno que permite indicar a la subred el tipo de servicio que se desea. Son posibles diversas combinaciones entre confiabilidad y velocidad. En la práctica, este campo ha sido utilizado en redes con calidad de servicio para telefonía IP pertenecientes a redes privadas corporativas. Sin embargo, en redes públicas este campo ha sido ignorado, por lo que IP sólo ofrece un servicio no confiable y no orientado a la conexión.

### Protocolo TCP

TCP es un protocolo confiable orientado a la conexión que se implementa en la capa de transporte. Dado que el protocolo IP no es ni confiable ni orientado a conexión, es responsabilidad de TCP ofrecer estos servicios a la capa de aplicación. Debe asegurarse que los datos que fueron transmitidos sean recibidos sin error y en el orden correcto, retransmitiendo información perdida y almacenando y reordenando paquetes en caso necesario.

El protocolo TCP se implementó para adaptarse dinámicamente a las condiciones de red por lo que una de sus principales características es el control de flujo. Esta funcionalidad permite que el protocolo sea capaz de determinar la tasa de transmisión de paquetes, para no sobrecargar a receptores lentos. Además, si aumenta la tasa de pérdida de paquetes (*Packet Loss Rate*, PLR), el protocolo disminuye exponencialmente su tasa de transmisión, de modo de no saturar la red. Este mecanismo hace que la tasa de transmisión sea conservadora y discontinua.

## **Protocolo UDP**

A diferencia de TCP, el protocolo UDP no es un protocolo confiable ni orientado a conexión. Ofrece a la capa de aplicación un mecanismo para enviar paquetes IP sin establecer conexión. No existe retroalimentación sobre las condiciones de red por lo que no se retransmiten paquetes perdidos ni se realiza control de flujo. El protocolo no disminuirá su tasa de transmisión en presencia de un PLR (*Packet Loss Rate*) alto, por lo que en la práctica el tráfico UDP resulta ser muy agresivo. El protocolo no garantiza que los paquetes sean recibidos, mas aún pueden estos llegar en desorden o duplicados.

## **2.6. APLICACIONES EN TIEMPO REAL**

La capacidad de las redes en sistemas LAN y WAN ha aumentado considerablemente en estos últimos tiempos. Años atrás, los enlaces no superaban los 2 Mbps, mientras que hoy, las redes operan a 155 Mbps o más. Esta rápida introducción de redes de alta velocidad ha motivado el desarrollo de nuevas aplicaciones que han sido altamente asimiladas por los usuarios quienes han experimentado cambios, a nivel individual y de empresa, en la forma de operar. La demanda por utilizar estos sistemas de comunicación ha sido también consecuencia del desarrollo de computadores de alta capacidad y de programas con interfaces gráficas adecuadas y simples de usar.

### **2.6.1. Condiciones de Red y características de tráfico en tiempo real**

Las características más importantes de una red física son el ancho de banda y el retardo. El ancho de banda queda determinado por la capacidad de los enlaces físicos que se utilicen, mientras que el retardo dependerá de la tecnología utilizada, el largo de los enlaces y el número y la capacidad de los enrutadores que procesen los datos. Sin embargo, dada la estructura con que está diseñada la red Internet, con la congestión aparecen efectos negativos para la transmisión de datos. Cada enrutador está implementado con un sistema de buffer que permite almacenar y procesar cierto volumen de tráfico. Si la carga aumenta, los buffer se llenan produciendo pérdidas y retraso de paquetes. Dado que el enrutamiento es dinámico, existe la posibilidad que los paquetes tomen caminos diferentes provocando desorden en los datos y *jitter*, variación en el tiempo de llegada entre paquetes sucesivos.

Es posible clasificar dos tipos de tráfico en la red dependiendo del tipo de aplicación. Se denomina tráfico elástico al tráfico correspondiente a servicios que no se ven muy afectados por las condiciones de la red. Internet fue diseñado para este tipo de tráfico y, en volumen, corresponde a la mayoría de la carga transmitida. Ejemplo de estas aplicaciones son correos electrónicos, transferencia de archivos, etc. El protocolo de transporte ideal para estas aplicaciones es TCP pues ofrece un servicio confiable orientado a conexión. Por otro lado, se denomina tráfico inelástico al tráfico generado por servicios susceptibles a las condiciones de red como aplicaciones en tiempo real. Las propiedades de red deseables para este tipo de tráfico son: bajo *jitter*; baja latencia; poder integrar tráfico elástico e inelástico; ancho de banda constante; adaptación a los cambios dinámicos de la red; mínima utilización de requerimientos de buffer dentro de la red; baja carga adicional producto de encabezados de protocolos; baja carga computacional en componentes de red, etc. Estos requerimientos son difíciles de entregar en una red como Internet, pues ni TCP ni UDP ofrecen servicios adecuados para este tipo de tráfico.

### **2.6.2. Calidad de Servicio**

Dado que los protocolos TCP y UDP no son completamente apropiados para aplicaciones en tiempo real, es necesario desarrollar metodologías que permitan mejorar la calidad de servicio (*Quality of Service*, QoS) de estas aplicaciones. En los últimos años, se han instalado medios de transmisión de alta capacidad, como fibra óptica, que han permitido reducir los retardos considerablemente a niveles aceptables, por lo que los factores de red que deben ser analizados son el *jitter*, el desorden de paquetes y la pérdida de paquetes. Para los dos primeros, es posible dar robustez a la transmisión de voz usando *buffers* adaptivos. Sin embargo, solucionar el efecto de pérdida de paquetes en la aplicación implica un desafío mayor.

Se han planteado dos enfoques distintos para solucionar este problema. Uno de ellos es modificar la estructura de red de modo de proporcionar servicios diferenciados a aplicaciones distintas. El segundo enfoque es implementar en las aplicaciones mecanismos de recuperación de información a través de interpolación, retransmisión o redundancia.

### 2.6.3. Arquitectura de servicios integrados

La cantidad de tráfico correspondiente a aplicaciones de multimedia y de tiempo real presumiblemente seguirá aumentando por lo que las redes deberán ser capaces de dar soporte a estos servicios. Esto involucra la necesidad de controlar la congestión de tráfico y de proveer diferentes niveles de QoS a diferentes aplicaciones. En respuesta a estos requerimientos, se desarrolló una arquitectura de servicios integrados (*Integrated Services Architecture, ISA*), que implementó nuevos protocolos que funcionan en distintas capas. Las principales innovaciones se presentan a continuación.

### 2.6.4. Protocolo de reserva de recurso

El protocolo de reserva de recursos (*Resource Reservation Protocol, RSVP*), permite a los usuarios tanto reservar capacidad en la Internet como especificar los requerimientos de red en términos de ancho de banda y retardo. Este protocolo es implementado a nivel de enrutadores, entre los cuales se establece conexión para reservar recursos que permitan mantener la QoS sobre cierto nivel. Junto con el protocolo RSVP, la arquitectura ISA contempla otros componentes que deben ser implementados en los enrutadores. Uno de ellos es el bloque de admisión de control (*admission control*), el cual permite determinar si hay suficientes recursos disponibles para el tráfico que requiere esa QoS. Otro bloque es el de gestión (*management agent*), que permite incorporar control de normas en el uso de los recursos. Estos, junto a otros bloques, permiten a nivel de enrutadores ofrecer QoS a las aplicaciones.

Con estas herramientas es posible discriminar en enrutadores el servicio ofrecido a tráfico elástico y a tráfico inelástico. Para que estos protocolos puedan ser utilizados por los usuarios, todos los enrutadores involucrados en la transmisión deben tener implementado esta nueva estructura, lo que hoy en día sólo es posible en redes privadas.

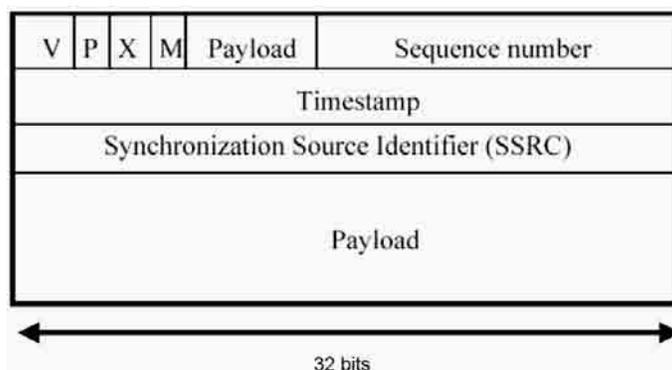
### 2.6.5. Protocolo de tiempo real (RTP)

El protocolo de tiempo real (*Real Time Protocol, RTP*), provee funcionalidades apropiadas para la transmisión en tiempo real de aplicaciones como video y audio sobre servicios entre dos (*unicast*) o entre varios (*multicast*) equipos. El protocolo RTP, no provee reserva de recursos ni garantiza QoS, pero permite monitorear la recepción de

datos y controlar e identificar el tipo de servicio ofrecido. Su diseño es independiente de los protocolos utilizados en la capa de red.

El RTP esta diseñado para ser implementado sobre un protocolo no orientado a conexión como UDP, brindando servicios de numeración de paquetes, realimentación de condiciones de red, etc. En el modelo de capas, este protocolo correspondería a capa de presentación y sesión del modelo OSI. Dependiendo del servicio implementado en la capa de aplicación, el protocolo ajustará su funcionamiento de modo de ofrecer las herramientas adecuadas.

Cada paquete RTP consiste en una cabecera que acompaña a la información que se transmite. La cabecera contiene números de secuencia, marcas de tiempo, y monitoreo de entrega. El formato de esta cabecera es el mostrado en la figura 5.



**Figura 5.** Cabecera del protocolo RTP

### **Tipo de información (Payload)**

El campo “payload” identifica el tipo de información que viaja en el paquete. Es un campo de 7 bits, lo que permite diferenciar hasta 128 tipos de información. En audio, este campo indica el tipo de codificación.

### **Número de secuencia (Sequence Number)**

El campo correspondiente al número de secuencia es de 16 bits. Con cada paquete enviado, el emisor incrementa en uno el número de secuencia. Esto permite al receptor detectar paquetes perdidos, o fuera de orden.

### **Marca de tiempo (Time Stamp)**

Este campo es de 32 bits. Indica el momento al que corresponde la primera muestra de la ventana de información que viaja en el paquete. Este campo es utilizado por el receptor, para reproducir las muestras con la misma cadencia con las que fueron obtenidas. Es a su vez útil para medir el "jitter". En audio, el campo "Time Stamp" se mide en unidades de 125  $\mu$ s.

### **Identificador del origen (SSRC - Synchronization Source Identifier)**

El campo correspondiente al SSRC es de 32 bits. Típicamente cada flujo en una sesión RTP tiene un identificador diferente. El origen establece este número, asegurando que no se repita.

### **Mecanismos de Recuperación Paquetes a nivel de aplicación**

Una alternativa diferente para mejorar la calidad de servicio de aplicaciones en tiempo real es dar robustez a la transmisión utilizando mecanismos de recuperación de información perdida. Esta metodología busca resolver el problema de calidad de servicio modificando sólo los extremos finales de la conexión, disminuyendo así los costos de implementación. Las técnicas más estudiadas han sido usar retransmisión, incluir información redundante e implementar técnicas de recuperación de error (*error recovery technique*).

De las técnicas de retransmisión, la más importante corresponde a la solicitud de repetición automática (*Automatic Repeat Request, ARQ*), que es un mecanismo basado en la retransmisión de paquetes que no fueron recibidos por el *host* de destino. La técnica de redundancia más conocida es la corrección de error hacia delante (*Forward Error Correction, FEC*), que es un mecanismo que permite recuperar uno o más paquetes de los datos previamente recibidos. Las técnicas de ocultación de error (*error concealment*) corresponden a algoritmos dependientes del tipo de aplicación que permiten recuperar datos perdidos en base a interpolación y predicción de parámetros etc.

Además se han desarrollado sistemas robustos ante pérdidas de paquetes como RAT (*Robust Audio Tool*), el cual utiliza transmisión de redundancia, y MDC (*Multiple Description Coding*), cuyo principio es codificar un trozo de voz en múltiples paquetes que

incluyen redundancia los cuales pueden ser decodificados en forma independiente. Las técnicas de ocultación de error son muy efectivas sólo cuando el PLR es pequeño en torno al 2%. Las técnicas FEC y ARQ incorporan sobrecarga de datos y aumentan considerablemente el retardo. Además, ninguno de estos métodos soluciona el problema de la QoS frente a ráfagas de pérdidas de paquetes.

#### **2.6.6. Estandarización de sistemas de Multimedia**

Grupos de investigación, como empresas y universidades, han invertido valiosos recursos en desarrollar y perfeccionar sistemas de multimedia, ya que el potencial que se espera de estas tecnologías es enorme. Las aplicaciones más importantes sin lugar a duda son la transmisión de voz, audio y video. Si cada una de estas tecnologías fuese desarrollada e implementada en forma independiente, sería muy complicado interconectar sistemas, por lo que se han creado estándares.

Una de las instituciones líderes en sistemas de multimedia es la ITU-T (*International Telecommunication Union, Telecommunication Standardization sector*). Esta institución está dividida en grupos de estudios, los cuales cubren una gama enorme de temas relacionados con el correcto funcionamiento de equipos y servicios del área de telecomunicaciones. Los tópicos incluyen sistemas y servicio de multimedia, operación de redes y servicios, principios de contabilidad y tarifas, redes de datos, etc.

Los estándares producidos por ITU-T son ordenados en series, siendo la G (*Transmission systems and media, digital systems and networks*) y la H (*Audiovisual and multimedia systems*) las más importantes para aplicaciones de multimedia sobre la Internet. La serie G incluye sistemas de codificación de voz, características de medios de transmisión, redes digitales etc. La serie H incluye sistemas y terminales para servicios audiovisuales, codificación de video, servicios adicionales para multimedia, etc.

## 2.7. Codificación de Voz

El desarrollo de tecnologías de compresión de voz ha contribuido enormemente en el crecimiento del área. La tasa de codificación de voz se ha reducido de 128 Kbps a 5.3 Kbps sin mayor pérdida perceptiva de calidad. Además VoIP ofrece numerosas ventajas económicas para las compañías que dan el servicio, ya que éstas pueden reducir la inversión en equipos, disminuir cargos de interconexión con empresas externas y ofrecer nuevas aplicaciones como videoconferencias y centros de llamados por Internet.

Las dos técnicas mayormente utilizadas en codificación de voz son los sistemas *waveform coding* (codificación de forma de onda) y *vocoder*. La primera tiene por objetivo reproducir la señal decodificada lo más parecido a la señal original. Los principales métodos son ITU-T G.711 (*Pulse Code Modulation*, PCM) y ITU-T G.726 (*Adaptive Differential Pulse Code Modulation*, ADPCM). Los *vocoders*, por otra parte, toman en consideración el modelo de producción de la voz para reproducir una señal perceptivamente igual a la original, o sea, las señales original y decodificada no tienen que ser iguales en forma, sino que deben ser escuchadas iguales. Los principales métodos son ITU-T G.729 (*Conjugate-Structure Algebraic Code Excited Linear Prediction*, CS-ACELP), ITU-T G.728 (*Low Delay Code Excited Linear Prediction*, LD-CELP) y ITU-T G.723.1 (*Dual rate speech coder for multimedia communication transmitting at 6.3 and 5.3 kbps*).

Para medir la calidad de voz en codificadores *waveform* se utilizan métodos objetivos como la razón señal a ruido (*Signal to Noise Ratio*, SNR), que pueden ser calculados directamente comparando la señal original con la decodificada. Sin embargo, para *vocoders* es necesario implementar métodos subjetivos. Se han desarrollado varias estrategias de medición como *Mean Opinión Score* (MOS), *opinionequivalent Q value* (Qop) y *Articulation Equivalent loss* (AEN). El más utilizado como medida de calidad es el estándar MOS, en el cual personas debidamente entrenadas evalúan secuencias equilibradas de sonidos, calificándolas en el rango de 5 (excelente, deterioro imperceptible) a 1 (muy molesto), en pruebas de tipo *absolute category rating* (ACR). Dado lo costoso y complicado de desarrollar este tipo de evaluaciones, se han desarrollado métodos objetivos que permiten predecir la calidad subjetiva de la voz como el estándar ITU-T P.862 (*Perceptual Evaluation of Speech Quality*, PESQ).

Uno de los problemas importantes que enfrenta VoIP es la capacidad de procesamiento necesario en equipos terminales para codificar y decodificar señales de voz, ya que algunos *vocoders* requieren ejecutar sobre 10 millones de instrucciones por segundo. Sin embargo, el desarrollo de DSP (*Digital Signal Processing*) ha permitido confeccionar procesadores apropiados para estos sistemas. Por otro lado, como cualquier otra aplicación en tiempo real, el tráfico de VoIP es inelástico, por lo que requiere condiciones de red apropiadas para su transmisión. Por otro lado, todos estos codificadores fueron desarrollados para redes con pérdidas y *jitter* cercanos a cero.

### 3. CAPITULO III. CODIFICACIÓN DE VOZ APLICADA A REDES DE PAQUETES

Uno de los campos del procesamiento de voz que ha sido ampliamente estudiado en las últimas décadas es la codificación digital de la voz. La idea principal es representar la señal de voz con un número mínimo de bits, sin perder considerablemente su calidad. Razones de almacenamiento y uso eficiente de canales de transmisión han motivado el desarrollo de nuevas tecnologías que han permitido codificar a muy bajas tasas.

Desde el punto de vista de la comunicación, la señal de voz es transformada en una secuencia de bits por el codificador, es transmitida a través de un canal y finalmente convertida nuevamente en una señal audible por el decodificador. En el proceso es necesario convertir la señal de analógica a digital para su posterior procesamiento.

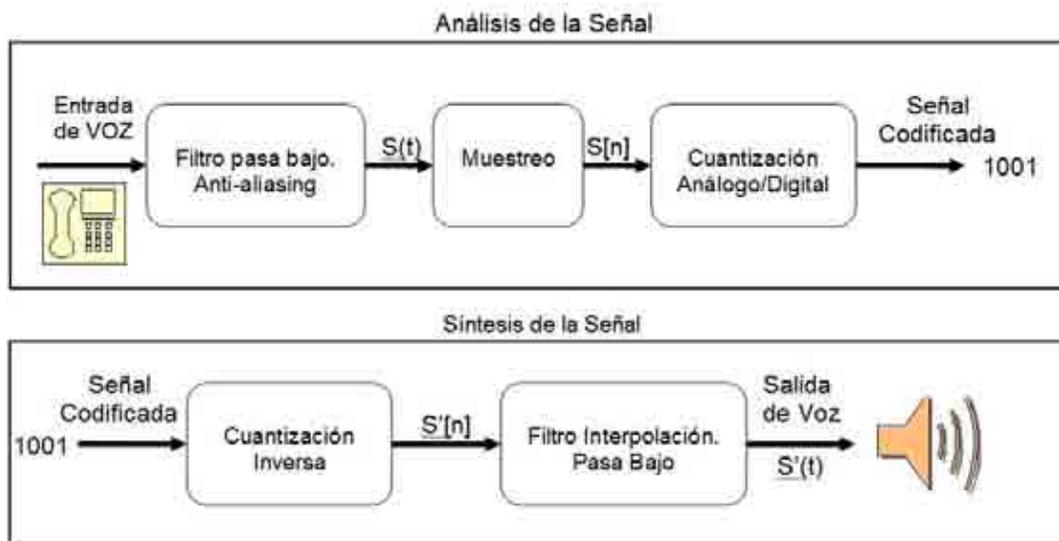
Existe un compromiso entre la degradación producida por la codificación y el número de bits utilizados en su representación que normalmente, aunque no lineal, es monótonamente decreciente. Esta degradación en la calidad de la señal de voz puede incorporar problemas muy complejos en ciertas aplicaciones como reconocimiento de voz sobre IP. Por su parte, el canal de transmisión también puede agregar distorsión en la señal por lo que los codificadores, en general, incorporan métodos de protección de bits (*bit protection*), que permite identificar secuencias de datos con errores. En medios de transmisión como Internet, el problema principal que los codificadores deben superar es la pérdida de paquetes, puesto que los protocolos de la capa de transporte descartan los paquetes dañados.

Los codificadores de voz más simples son los *waveform*, que codifican la señal muestra por muestra. Existen codificadores de este tipo que aprovechan características de la voz en el dominio del tiempo y en el dominio de la frecuencia. Un método más complejo de codificación son los sistemas vocoder, los cuales incorporan el modelo de producción del habla en las técnicas de compresión. En general, estos sistemas han logrado disminuir mucho las tasas de transmisión permitiendo el uso de aplicaciones como transmisión digital de voz, almacenamiento para mensajes de voz por correo electrónico, centros de llamados, reconocimiento de voz en VoIP, etc.

### 3.1. CODIFICADORES TIPO WAVEFORM

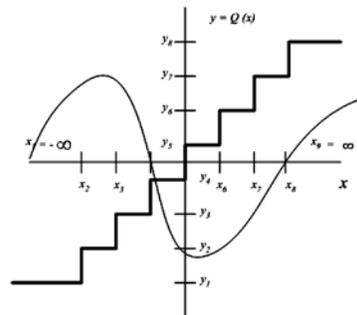
Los métodos para representar digitalmente características temporales o espectrales de la forma de onda de una señal de voz son denominados codificadores *waveform*. En el dominio del tiempo, la señal de voz presenta redundancia, como periodicidad, variación lenta de intensidad, etc., que son aprovechados por estos codificadores. Por otro lado, los métodos *waveform* basados en el dominio de la frecuencia, toman en consideración que la distribución de las componentes espectrales no es uniforme. Ejemplo de este tipo de codificadores son *Filter Bank Analysis*, *Sub-Band Coders* (SBC) y *adaptive Transform Coders* (ATC).

El teorema del muestreo es el principio fundamental detrás del procesamiento y de la transmisión de una señal análoga utilizando técnicas digitales. Este principio establece que una señal análoga con frecuencia de corte  $w_c$ , puede ser exactamente representada por una secuencia de muestras  $x(nT)$ , extraídas a una frecuencia mayor o igual al doble de  $w_c$ . Es importante notar que en este proceso no existe pérdida. La figura 6 muestra el procesamiento básico general para la codificación de voz. El primer bloque del proceso de codificación, conocido también como análisis, corresponde a un filtro pasa bajo que tiene por finalidad eliminar las componentes que superen la frecuencia  $w_c$ , para evitar error de *aliasing*. Luego la señal es muestreada y posteriormente cuantizada. El proceso de síntesis, o sea de decodificación, es realizado utilizando el proceso inverso.



**Figura 6.** Procesamiento básico de codificación de voz.

La cuantización es el proceso fundamental en la codificación de voz, pues a diferencia del muestreo, este proceso sí incorpora error en la señal decodificada. La idea fundamental de esta etapa es representar las muestras de la señal en niveles discretos, como se muestra en la figura 7. El número de niveles que se utilicen aquí, determinará el error de cuantización introducido.



**Figura 7.** Cuantización uniforme de una señal

### 3.1.1. Codificación PCM

La metodología más simple para codificar una señal de voz es utilizar solamente cuantización uniforme. Generalmente, se utiliza 16 bits por muestra por lo que si la señal es muestreada a 8000 Hz, uno obtendrá una tasa de codificación de 128 kbps. Sin embargo, se ha demostrado que dada la dinámica de una señal de voz, sólo es necesario utilizar 12 o 13 bits disminuyendo la tasa a 96 y 104 kbps respectivamente, sin pérdida considerable de calidad.

Una de las primeras características apreciables en una señal de voz es que la distribución en el tiempo no es uniforme, esto principalmente debido a silencios y sonidos con poca energía. Este tipo de cuantización no aprovecha bien el número de bits utilizados para representar la señal por lo que es posible utilizar otro tipo de metodología no uniforme que sea más efectiva. Además, el SNR aumenta en segmentos de la voz con mayor amplitud, por lo que la cuantización uniforme tiende a reproducir una señal cuya calidad varía en función del tiempo, debido a la dinámica de la voz. Para solucionar este problema, es posible utilizar cuantización logarítmica, proporcionando un error de cuantización más uniforme.

En 1972, el comité de consulta internacional telefónico y telegráfico (*Comité Consultatif International Téléphonique et Télégraphique*, CCITT) publicó la recomendación G.711, basado en la tecnología PCM. Rápidamente se transformó en el referente principal de los sistemas de transmisión de voz de la época. La vigencia del estándar se mantiene hasta nuestros días en las redes de telefonía fija. El codificador utiliza 8 bits por muestras, las cuales son muestreadas a 8 kHz codificando a una tasa de 64 kbps.

El estándar G.711, establece dos tipos de algoritmos de compresión: la ley  $\mu$ , la cual es utilizada como estándar PCM en Estados Unidos, y la ley A, la cual es utilizada como estándar PCM en Europa.

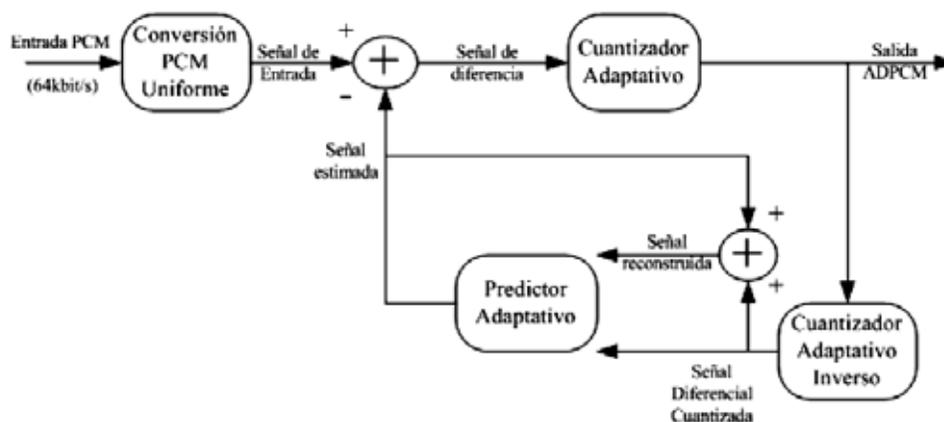
La idea fundamental detrás de este estándar es comprimir la señal de voz en forma logarítmica, para luego realizar cuantización uniforme. En ambas leyes, la compresión es casi lineal para señales de amplitud pequeña y logarítmica para señales de amplitud grande.

### **3.1.2. Codificación ADPCM**

En 1982, el CCITT formó un nuevo grupo de investigación para estudiar una nueva metodología de codificación de voz que redujese la tasa de codificación y que mantuviese la calidad perceptiva de la voz. Luego de importantes contribuciones de diferentes organizaciones, se optó por un método adoptivo de un sistema diferencial PCM (*Differential Pulse Code Modulation*, DPCM).

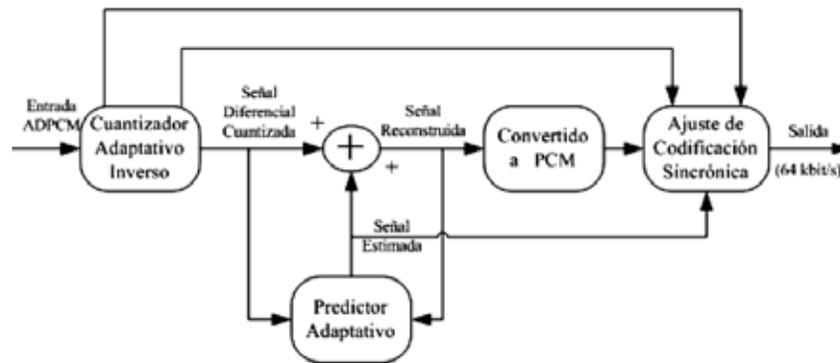
La idea básica detrás del sistema DPCM es cuantizar y transmitir el error de predicción en vez de la señal misma. Este sistema explota el comportamiento de la voz en el dominio del tiempo. En vez de cuantizar cada muestra de voz separadamente, DPCM cuantiza la diferencia entre la señal original y una estimación predictiva de la misma, basado en un promedio ponderado de muestras anteriores, usando la propiedad de correlación muestra a muestra que presentan las señales de voz. Además, el error de predicción, presenta menor varianza que la señal original, por lo que la cuantización del error de predicción introduce menos ruido de cuantización. Dado que la señal de voz no es estacionaria, los coeficientes del predictor deben ser recalculados cada cierto tiempo en forma adaptiva obteniendo un sistema ADPCM.

Luego de 18 meses de desarrollo y de pruebas de calidad subjetivas y objetivas, el grupo creado por CCITT publicó el estándar G.721 el cual codifica a una tasa de 32 kbps usando 4 bits por muestra. Entre 1985 y 1988, se desarrolló el estándar G.723 el cual extendía la tasa de codificación a 24 y 40 kbps usando 3 y 5 bits por muestras respectivamente. Entre 1989 y 1992, ambos estándares fueron unidos formando el estándar G.726, el cual mantuvo compatibilidad completa con los sistemas anteriores y además agregó una nueva tasa a 16 kbps usando tan sólo 2 bits por muestra. La figura 8 entrega un diagrama en bloque simplificado del codificador.



**Figura 8.** Diagrama en bloques simplificado del codificador ITU-T G.726

El codificador está diseñado para recibir como entrada una señal de voz previamente codificada con el estándar ITU-T G.711. El primer bloque tiene por función transformar esta entrada a una señal uniforme representada por 14 bits. Luego la señal estimada es sustraída a la señal de entrada. El residuo es cuantizado adaptativamente usando 31, 15, 7 o 4 niveles de cuantización dependiendo del modo de operación (40, 32, 24 y 16 kbps respectivamente). Luego el cuantizador adaptativo inverso produce una señal diferencial cuantizada usando la misma información que será transmitida. Esta señal es sumada a la señal estimada para producir la versión reconstruida de la señal de entrada y que será igual a la señal reproducida en el decodificador. Tanto la señal reconstruida como la señal diferencial cuantizada son utilizadas por el predictor adaptativo el cual produce una estimación de la señal de entrada, completando la retroalimentación.



**Figura 9.** Diagrama en bloques simplificado del decodificador ITU-T G.726

El decodificador, cuyo diagrama en bloque simplificado es mostrado en la figura 9, reconstruye la señal en forma análoga a la parte de retroalimentación del codificador. Incorpora además un sistema de ajuste de codificación síncrona. Finalmente, la salida uniforme es transformada al formato del estándar G.711 correspondiente a la ley  $\mu$  o A.

Muchos bloques adicionales han sido agregados para asegurar estabilidad independiente del estado inicial. Además, tanto el codificador como el decodificador son configurados de modo que ambos funcionen con valores iniciales comunes.

### 3.1.3. Otras tecnologías

Tanto PCM como ADPCM corresponden a codificadores *waveform* en el dominio del tiempo. Tecnologías similares a las anteriores son: DPCM, APCM (*Adaptive Pulse Code Modulation*), DM (*Delta Modulation*), ADM (*Adaptive Delta Modulation*) y APC (*Adaptive Predictive Coding*). Todas estas técnicas aprovechan características temporales de la voz.

Otra clase de tecnología *waveform* son las que aprovechan características de la voz en el dominio de la frecuencia. Estos métodos filtran la señal de voz usando varias bandas de frecuencias codificándolas en forma separada. La codificación de la forma de onda puede ser realizada tanto en el dominio del tiempo en cada banda como en el dominio de la frecuencia. Ejemplo de este tipo de codificadores es el ATC, el cual subdivide la señal en grupos de N muestras y los transforma al dominio espectral para

codificarlos. Para realizar eficientemente la codificación, se asignan mayor cantidad de bits a los coeficientes espectrales más importantes. Codificadores como *Filter Bank Analysis* y SBC son otros ejemplos de las tecnologías *waveform* que aprovechan características en frecuencia de la señal de voz.

### 3.2. CODIFICADORES TIPO VOICE CODER

Los codificadores de tipo *voice coder* o *vocoder*, se caracterizan principalmente por utilizar el modelo de producción y recepción del habla a través de señales de excitación y filtros. Estas tecnologías han logrado disminuir las tasas de transmisión considerablemente, logrando estándares de codificación por debajo de los 2.5 kbps.

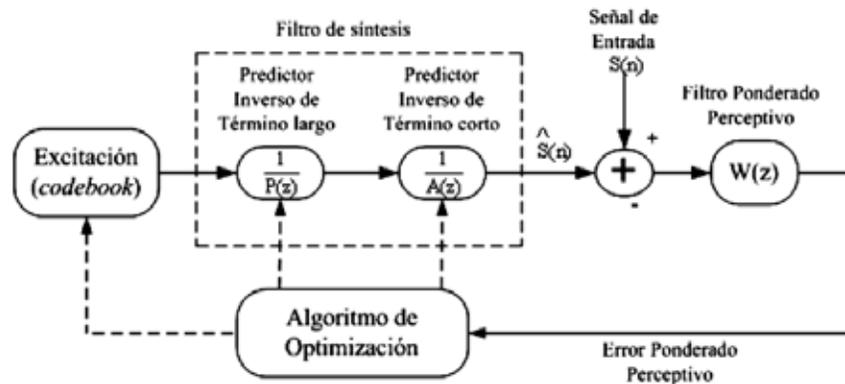
El oído humano tiene limitaciones que estas tecnologías aprovechan. Si dos señales se superponen en el tiempo, y una de ellas posee mayor energía que la otra, el sistema auditivo humano no es capaz de distinguir la señal de menor energía. A este fenómeno perceptivo se le denomina enmascaramiento (*masking*). Utilizando esta propiedad se llega a la conclusión que lo más importante es que la señal decodificada sea perceptiblemente similar a la original. Con esto se puede modificar el espectro del ruido producido en la cuantificación y hacerlo perceptiblemente inaudible logrando que estos codificadores no vean afectada su calidad perceptiva a pesar de utilizar tasas de transmisión bajas.

#### 3.2.1. Codificadores CELP

Las técnicas utilizadas por los *vocoders* revolucionaron los sistemas de transmisión de voz. Se lograron alcanzar tasas tan bajas como 2.4 y 4.8 kbps. Sin embargo, estos sistemas no lograban obtener calidades semejantes a los sistemas *waveform*. En 1985 se publicó un codificador denominado *code excited linear prediction* (CELP), en el cual se utilizó, por primera vez, la técnica de análisis por síntesis.

La figura 10 muestra un esquema básico de este sistema. El principio de este método es que el codificador de voz codifica la señal tomando en cuenta la señal decodificada. Utilizando los parámetros que posteriormente serán transmitidos, el codificador reconstruye la señal de modo de minimizar alguna medida de error. Más aun, un filtro perceptivo es agregado al esquema de análisis por síntesis para explotar la

propiedad de enmascaramiento, aprovechando las limitaciones del sistema auditivo humano.



**Figura 10.** Esquema básico de codificadores CELP.

Las tecnologías CELP descomponen la señal de voz, tomando en cuenta características de periodicidad, hasta dejar una señal plana semejante a un ruido aleatorio. Esta señal resultante o residuo es modelado por medio de sistemas de cuantización. El procesamiento es realizado sobre segmentos de voz denominados *frames*, y no muestra-a-muestra como en la mayoría de los codificadores *waveforms*. El número de muestras por *frame* varía generalmente entre 80 y 280, que muestreadas a 8 kHz corresponden a 10 y 35 milisegundos respectivamente.

El *pitch* incorpora una componente casi periódica en la señal de voz. En el dominio del tiempo, esta naturaleza periódica, que normalmente varía entre 2.5 y 20 milisegundos, es llamada correlación de término largo (*long term correlation*), la cual puede ser removida utilizando un filtro de predicción. Este filtro,  $1/P(z)$ , corresponde a un filtro todo polo y tiene por función predecir el *pitch*. La ecuación entrega la forma general de este filtro, donde  $b_i$  corresponde a los coeficientes del predictor, que son calculados utilizando técnicas de correlación, y  $D$  representa al período del *pitch*.

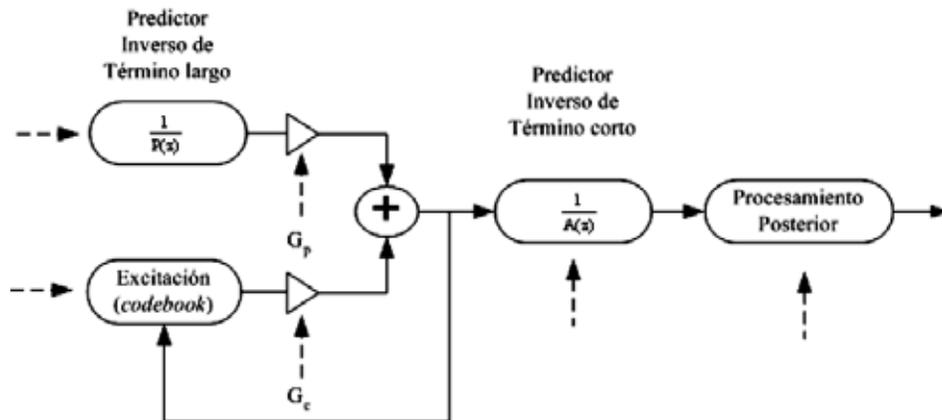
$$\frac{1}{P(z)} = \frac{1}{1 - z^{-D} \sum_{i=-q}^r b_i z^{-i}}$$

Adicionalmente, la señal de voz presenta correlación muestra-a-muestra tanto a nivel temporal como espectral. En el dominio del tiempo, este efecto es denominado correlación de término corto (*shortterm correlation*) el cual es posible reducir pasando la señal de salida del filtro  $1/P(z)$ , por un filtro,  $1/A(z)$ , semejante al utilizado por los *vocoders*. La ecuación de abajo, muestra la forma general de este filtro. Los coeficientes  $a_i$  son calculados utilizando predictores lineales de orden  $p$ , el cual usualmente toma el valor de 10 para voces muestreadas a 8 kHz.

$$\frac{1}{A(z)} = \frac{1}{1 - \sum_{k=1}^p a_k z^{-k}}$$

Luego que la señal de voz es procesada por los filtros de predicción de error de término corto y largo, la señal resultante carece de todo tipo de periodicidad. Dado que este residuo presenta características de ruido aleatorio, el codificador CELP lo modela usando segmentos o vectores de ruido blanco Gaussiano. Existe un número grande de estos vectores de representación, los cuales en conjunto forman un *codebook*, el cual está presente tanto en el codificador como en el decodificador. El residuo es comparado usando distancia euclidiana ponderada con cada uno de los vectores que forman el *codebook*, para luego seleccionar el más parecido al residuo, o sea el que tenga la distancia mínima. La ventaja es que en vez de enviar al decodificador el vector de residuo, sólo se envía el índice del *codebook* correspondiente al segmento seleccionado. Representar una secuencia de vectores en un *codebook* por su índice es denominado cuantización vectorial y es parte fundamental en la disminución de la tasa de transmisión de los codificadores tipo *vocoders*.

Lo que se manda al decodificador, corresponde al índice de la secuencia del *codebook* óptimo, los coeficientes de los predictores de término corto y largo, y las ganancias asociadas a los filtros. Con estos valores se reconstruye la señal de excitación para finalmente filtrarla por el filtro  $1/A(z)$ . La figura 11 muestra el diagrama del decodificador.



**Figura 11.** Esquema básico de decodificadores CELP

Se desarrollaron nuevos algoritmos para el cálculo y cuantización de los coeficientes del filtro  $1/A(z)$ ,  $a_i$ . Estas y muchas otras mejoras permitieron estandarizar codificadores de voz basados en tecnologías CELP: ITU-T G.728, ITU-T G.729, ITU-T G723.1, entre otros. Todos estos estándares fueron implementados orientados a diferentes escenarios, por lo que su utilización dependerá de los requerimientos del sistema tanto de costo, calidad y retardo. Entre ellos, el sistema de codificación más importante en transmisión de voz por Internet es el codificador ITU-T G.729. La tasa de transmisión que entrega este esquema es de 8 kbps, obteniendo una calidad perceptiva mayor a la mayoría de los otros sistemas *vocoders*. Modificando la forma de cuantizar los parámetros, se publicaron los anexos D, los cuales extienden las tasas de codificación a 6.4 y 11.8 kbps respectivamente.

### 3.3. CALIDAD DE CODIFICADORES

La necesidad de sistemas de medición, comenzó en los años 1950 con el desarrollo del sistema de comunicación análogo. Estas técnicas fueron esenciales en la optimización de diseños de sistemas de codificación. Actualmente, existen muchas formas de medir la calidad, los cuales, en general, pueden ser divididos en métodos subjetivos y objetivos.

La forma de comparar las distintas metodologías a nivel de calidad, dependerá del tipo de codificación que se trate. Dado que los codificadores *waveform* buscan reproducir

una señal lo más parecida, en forma, a la original, los parámetros objetivos de medición de calidad son perfectamente utilizables. Sin embargo, en codificadores tipo *vocoder* estos parámetros son poco efectivos, pues la codificación busca que la voz reproducida sea perceptivamente igual a la original, por lo que en forma no son necesariamente semejantes. Los métodos subjetivos de medición de calidad entregan una herramienta adecuada para este tipo de codificadores, pero producto del elevado costo de las evaluaciones, debido a los requerimientos necesarios para llevar a cabo este tipo de pruebas, hacen complicada su utilización.

### 3.3.1. Métodos Subjetivos

Los métodos subjetivos están basados en la opinión de grupos de personas sobre la calidad de frases codificadas. Las personas son entrenadas auditivamente, para posteriormente en laboratorios con equipos altamente especializados realizar las pruebas.

Algunos de los métodos subjetivos buscan medir la inteligibilidad de los codificadores. Para ello, utilizando palabras que se pronuncien parecidas se evalúa a los codificadores. Ejemplo de este tipo de sistemas de medida son MRT (*Modified Rhyme Test*) y DRT (*Diagnostic Rhyme Test*). Sin embargo, la inteligibilidad es sólo una parte en la calidad de voz, por lo que estos sistemas de medidas son incompletos, si el objetivo es medir la calidad.

Otra clase de sistema de evaluación que intenta medir no sólo la inteligibilidad de señal codificada es el *test* de calidad IAJ (*Isometric Absolute Judgment*). En esta prueba, las personas deben escuchar frases distorsionadas por los codificadores, para luego emitir un juicio usando la escala que va de excelente a insatisfactoria. Lo interesante de este método es que es isométrico en el sentido que a las personas no se les presentan frases de referencia antes de evaluar la calidad.

Otro tipo de metodología utilizada es entrenar a los evaluadores con señales de referencia, para evitar diferencias de criterios. Alguno de estos sistemas de medida son: MOS, PAR (*Paired Acceptability Rating*) y ACR. Una técnica distinta de sortear diferencias de preferencia individuales lo incorpora el método de medición PAJ (*Parametric Absolute Judgment*), el cual consiste en evaluar diferentes componentes de la calidad por

separado, en vez de evaluar sólo el promedio. Este método es efectivo, pues las personas comúnmente concuerdan en el grado de degradación de factores aislados. El proceso de ponderación de estos factores es el principal responsable de las diferencias en los criterios de los evaluadores. Para llegar a un valor promedio, el sistema de medida PAJ pondera los resultados usando un mismo criterio. Las pruebas QUART (Quality Acceptance rating test) y DAM (Diagnostic Acceptability Measure) son dos ejemplos más de este tipo de sistema de medida.

De los métodos subjetivos discutidos, sin lugar a duda el *test* MOS es el más importante y el más utilizado. En este sistema, un grupo de personas evalúan las frases codificadas usando una escala de cinco puntos según lo expuesto en la tabla 1. Una fase de entrenamiento es utilizada antes de la prueba para asociar criterios. En este período, varias frases estándares con calidades conocidas son reproducidas a los evaluadores para que les sirvan de referencia al momento de evaluar. Una de las características interesantes de destacar, es que este método es aplicable a un amplio rango de distorsiones. La desventaja que presenta el sistema de medida MOS es que los resultados pueden variar por factores como la selección de los evaluadores, las instrucciones dadas, el equipamiento utilizado, el orden en que se realiza la prueba, etc.

Nivel de Distorsión	Calidad de Voz	Nivel
Imperceptible	Excelente	5
Algo perceptible pero no molesto	Bueno	4
Perceptible y levemente molesto	Regular	3
Molesto pero no objetable	Pobre	2
Muy molesto y objetable	Insatisfactorio	1

**Tabla 1.** Escala de 5 puntos de MOS.

Algunos de estos métodos forman parte del estándar ITU-T P.800, el cual describe varios métodos para la medición subjetiva de la calidad de la voz transmitida.

### 3.3.2. Métodos Objetivos

Los métodos objetivos de medición de calidad para señales de voz se basan en comparaciones matemáticas entre la señal original y codificada. La mayoría de estos métodos utilizan medidas de distancia para cuantizar la diferencia entre la señal distorsionada y la original. Una de las ventajas de este tipo de prueba es que permite obtener un valor cuantitativo de la calidad que no depende de factores externos, como en el caso de las medidas de calidad subjetivas. Es decir, cada vez que se realice la medición se obtendrá el mismo valor. Esto permite medir variaciones pequeñas en la calidad producidos por modificaciones poco significativas en los codificadores. Si se quisiera medir estas variaciones con métodos objetivos, se tendría que utilizar un gran número de evaluadores para tener un resultado significativo estadísticamente.

Los métodos más elementales de este tipo de codificadores son el SNR y la razón señal a ruido segmentada (*Segmental Signal to Noise Ratio*, SEGSNR), los cuales miden la razón entre la potencia de la señal y la potencia del ruido. Estos métodos son ampliamente utilizados en codificadores tipo *waveform*. Sin embargo, estos métodos no son aplicables a los codificadores tipo *vocoder*. Para medir la calidad en este tipo de codificadores se han desarrollado métodos objetivos que permitan predecir los resultados que se obtendrían al utilizar pruebas de calidad subjetiva. Para ello se utilizan técnicas de análisis de voz, las cuales buscan imitar el comportamiento perceptivo del oído y en base a este modelo emitir un juicio sobre la calidad de las frases analizadas por los sistemas.

Producto del desarrollo experimentado por las técnicas de codificación, surgió la necesidad de estandarizar sistemas de medición objetivas. Una de las iniciativas presentadas fue el método de medida de calidad de voz perceptivo (*Perceptual Speech Quality Measure*, PSQM), publicado en el estándar ITUT P.861. Esta prueba, que fue diseñada para medir señales distorsionadas por redes telefónicas, ofrece una correlación muy baja con pruebas subjetivas en situaciones de recorte de la voz, ruido de fondo, *jitter* y pérdida de paquetes para VoIP. Otro método, que fue incluido como anexo de la recomendación ITU-T P.861, es el MNB (*Measure Normalized Block*), el cual fue desarrollado por el ITS (*Institute Telecommunication Sciences*). Este sistema incluye consideración de error de bits y tramas incompletas. A pesar que este método presenta buen índice de correlación con pruebas MOS, el rendimiento frente a retardos y efectos

de pérdida de paquete no es óptima. Un número de organizaciones instaron a la ITU a seleccionar un reemplazo para PSQM que se adaptara mejor a las redes existentes. Con este fin, el grupo de estudio 12 de la ITU organizó una competencia en la que participaron diversas compañías durante septiembre de 1998 y marzo del 2000. En la ocasión fueron seleccionados dos tecnologías: PAMS (*Perceptual Analysis Measurement System*) y PSQM99. El sistema PSQM99 brindaba un mejor rendimiento en condiciones de variación rápida de ganancia e intervalos intensos de cortes, mientras que el sistema PAMS era mejor en condiciones de pérdidas de paquete en redes IP y filtrado. Ambos sistemas fueron integrados logrando un modelo simple y mejorado, denominado PESQ, el cual fue publicado como estándar en la recomendación ITU-T P.862 en el año 2001.

El método PESQ presenta mayor exactitud que cualquier otro modelo en promedio, es altamente robusto y da predicciones exactas de calidad para un amplio rango de condiciones. Es ideal para medir efecto de pérdida de paquete, *jitter*, ruido ambiental y errores en transmisión de canal en codificadores como G.711, G.726, G.728, G.729 y G.723.1. El resultado entregado por el estándar PESQ es normalizado a una escala similar al sistema MOS en el rango 0.5 y 4.5, sin embargo en la mayoría de los casos el rango de salida varía entre 1.0 y 4.5, rango normal para valores MOS encontrados en los experimentos de calidad subjetivos.

En la tabla 2, se resumen las características para los codificadores más utilizados en el servicio de Telefonía y Voz sobre IP

<b>Codificador</b>	<b>Norma / Recomendación</b>	<b>Velocidad Binaria</b>	<b>Calidad Vocal (MOS)</b>	<b>Retardo de codificador / decodificador [ms]</b>
PCM Temporal	G.711	64 kbit/s	4,1 a 4,4	0,1
ADPCM Temporal	G.726	32 kbit/s	3,8 a 4,2	12,0
LD-CELP	G.728	16 kbit/s	3,9 a 4,2	33,0
CS-ACELP	G.729	8 kbit/s	3,9 a 4,2	20,0
MP-MLQ-ACELP	G.723.1	6,3 y 5,3 kbit/s	3,9 a 3,7	16,0

**Tabla 2.** Codificadores más utilizados en el servicio de Telefonía y Voz sobre IP

### 3.4. CODIFICACIÓN DE VOZ EN REDES DE PAQUETES

Tanto en redes de telefonía fija como en redes IP, una de las consideraciones más importantes que deben ser analizadas es la calidad de la voz percibida por el usuario final. Para el estándar de telefonía fija PSTN (*Public Switched Telephone Network*) la calidad medida en escala MOS usualmente toma valores en torno a 4. Sin embargo, para niveles superiores a 3 permitirán una comunicación aceptable. La calidad perceptiva de una señal de voz codificada depende de numerosos factores de diversa naturaleza, por lo que ha sido necesario realizar numerosos estudios para identificar aquellos parámetros más importantes. Los factores que afectan la calidad perceptiva de la voz cuando son transmitidos sobre cualquier red de paquetes pueden ser clasificados de la siguiente forma:

#### 3.4.1. Parámetros de codificación

Corresponden a aquellas distorsiones de la calidad de la voz producidas por el proceso de codificación, el cual incorpora error en la señal decodificada. El más importante de ellos es sin lugar a duda el tipo de codificador que se utilice. Asociado a esta elección, se establecen parámetros como el número de bits que se utilizarán por muestra, la tasa con que será muestreada la señal, los intervalos de tiempo entre paquetes (*packetization interval*), etc. Además, algunos codificadores están implementados con técnicas de ocultamiento de error, por lo que en caso de falla en la transmisión de los datos los *frames* extraviados pueden ser recuperados. Estas técnicas pueden afectar negativamente la calidad de la voz al incorporar ruido, silencios, repeticiones y sustitución de forma de onda en la señal decodificada.

#### Parámetros de red

Corresponden a aquellos factores propios de la red que introducen distorsión en la calidad de la señal de voz transmitida. En particular, la utilización de un medio de difusión como la Internet, puede fuertemente distorsionar la inteligibilidad de una transmisión de voz. Los parámetros más importantes son las pérdidas de paquete, la distribución de las pérdidas, el tamaño de paquetes, el retardo en la transmisión y el  *jitter*.

### **Parámetros del locutor**

Tanto el sexo del locutor como el idioma en que se expresa son relevantes en la calidad perceptiva de la voz. En general, estudios han demostrado que los codificadores responden levemente mejor en presencia de un locutor del género masculino debido a que el *pitch* no es tan agudo como el producido por un locutor del género femenino. Debido a que los codificadores han sido probados y perfeccionados utilizando bases de datos de locutores de habla inglesa (en particular todos los *codebook*), la calidad decae cuando el locutor habla en un idioma diferente. Más aun, la variación en el acento de los locutores es también un factor que afecta la calidad.

### **Parámetros estructurales**

En esta categoría se incluyen todos aquellos factores que dependen de la infraestructura con que se realicen las pruebas. El diseño y el funcionamiento de los equipos terminales, los *softwares* utilizados y las condiciones ambientales son factores importantes. Otro foco de pérdida de calidad es el fenómeno denominado *eco* que corresponde al ruido introducido cuando en uno de los terminales no están aislados el micrófono y el audífono por lo que la parte de la señal recibida es retransmitida. Esto debe ser evitado mediante técnicas de cancelación de eco.

La calidad perceptiva de la voz depende de cada uno de estos parámetros en forma altamente no lineal. Más aun, la variación de estos parámetros no es predecible, por lo que anticipar un valor para la calidad basado en estos índices es un proceso complejo el cual no puede ser resuelto usando métodos matemáticos que consideren cada una de estas variables.

De todos los posibles parámetros que pueden influenciar la calidad de la voz mencionados anteriormente, sólo un pequeño conjunto de ellos tiene una directa relación con los parámetros de red. A continuación se explican las distorsiones que introducen el retardo, el *jitter*, las pérdidas de paquete y el tamaño de paquete.

### 3.4.2. Influencia del Retardo

El retardo total en la transmisión es el resultado final de los retardos individuales de cada uno de los nodos y procesos que componen la red. Algunos de los retardos son fijos, como el tiempo de codificación y decodificación, mientras que otros dependen de las condiciones y de la estructura de la red. Cada enrutador debe procesar y encaminar cada paquete que llega a él. El tiempo que demora este proceso dependerá de la capacidad del enlace y del enrutador. Por otro lado, la propagación física de los datos en la red depende del medio de transmisión que en general es cercano a la velocidad de la luz, por lo que si las distancias entre los terminales no son grandes, este tiempo será pequeño. Además cada uno de los *buffers* que se implementan aumentará el retardo total.

El retardo tiene un impacto en la calidad de la transmisión, pues a medida que aumenta el tiempo de envío disminuye el nivel de interacción en la conversación. Si la conexión es unidireccional un retardo en torno a 150 ms no produce mayor distorsión, pero si éste supera los 400 ms la calidad comienza a verse afectada. Por otro lado, en comunicaciones bidireccionales aparece el problema de eco, por lo que para mantener la calidad en niveles aceptables, si el retardo supera los 30 ms, es necesario implementar mecanismos de cancelación de eco.

A continuación se muestran en la tabla 3, los valores que indican las clases de calidad e interactividad de acuerdo con el retardo de transmisión en una conversación telefónica.

Clase N°	Retardo	Observaciones
1	De 0 a 150 ms	Aceptable para la mayoría de las conversaciones; sólo algunas funciones altamente interactivas pueden experimentar degradación.
2	De 150 a 300 ms	Aceptable para las llamadas de baja interactividad (satélite con 250 ms por salto).
3	De 300 a 700 ms	Prácticamente una llamada semidúplex.
4	Mas de 700 ms	Inútil, a menos que los llamantes estén habituados a conversar en semidúplex (como los radioaficionados).

**Tabla 3.** Calidad e interactividad de una llamada, según el retardo de transmisión.

### **Retardo en el transmisor**

En el extremo transmisión, la voz es codificada y comprimida antes de encapsularla en paquetes IP. El tamaño del paquete representa una solución intermedia entre la necesidad de reducir el retardo de transmisión y la optimización de la anchura de banda. Los componentes del retardo en el transmisor son:

- La digitalización y codificación, es decir, el tiempo necesario para que una tarjeta de sonido o una pasarela digitalice y codifique una señal analógica.

La compresión, que se divide a su vez en tres partes:

- Retardo de trama: a diferencia de la digitalización de señal, que se lleva a cabo de una manera continua, la compresión se relaciona con determinada longitud de datos, cuya espera puede hacer necesario un tiempo de procesamiento especial.
- Retardo de codificación: este retardo, que también involucra la compresión por síntesis basada en la predicción, es solicitado por el codificador para saber, mientras está funcionando, cómo evoluciona la señal.
- Retardo de procesamiento: tiempo que toma el algoritmo para comprimir una trama. Depende del procesador y del tipo de algoritmo.
- Empaquetado: periodo de tiempo durante el cual la aplicación arma un paquete (creación del encabezamiento e inserción de los datos).
- Transmisión: este periodo de tiempo dependerá de la configuración utilizada, es decir, si la conexión se hace mediante un módem o un acceso directo en una LAN/WAN.

### **Retardo de red**

- Propagación: en una red alámbrica la velocidad de propagación es de 200 000 km/s, lo que resulta en un tiempo de propagación largo.
- Encaminamiento y colas: según el tipo de red, se pueden indexar varios tiempos diferentes.

Cuando se trate de una red IP bien controlada, tal como una intranet o equivalente, la transmisión de paquetes toma entre 50 y 100 ms (propagación y compensación de fluctuación de fase), y los encaminadores introducen un retardo de unos 50 ms, con un retardo total resultante de entre 200 y 250 ms de un extremo a otro, para una red de este tipo. En el caso de Internet, estos retardos son mucho mayores e incluso indeterminados.

#### **3.4.3. Influencia del Jitter**

Se puede describir el Jitter ó también conocido como la fluctuación de fase a la variación del retardo de transmisión. El protocolo utilizado para transportar paquetes vocales por Internet (en una red IP) es el protocolo de datagrama de usuario (UDP). El lado de señalización utiliza la capa de protocolo de control de transmisión (TCP).

El UDP funciona en un modo sin conexión, en el que los paquetes no siguen necesariamente la misma ruta y, por tanto, hay variaciones en el tiempo de tránsito. Otra posible causa de variación del tiempo de tránsito puede ser el número de encaminadores que se encuentran y la carga de cada uno de ellos. Para reconstituir un flujo síncrono en el extremo receptor, se instalan buffers de compensación de fluctuación de fase. No obstante, este procedimiento incrementa aún más el retardo de transmisión. Para mantener un nivel de calidad aceptable, la fluctuación de fase debe ser menor que 100 ms.

Buffer de fluctuación de fase: esta memoria permite la resincronización de los paquetes que llegan con retardos variables, compensando así los desfases temporales y restableciendo el orden correcto de los paquetes.

- Empaquetado.
- Descompresión.

- Decodificación y conversión digital-analógica.

El resultado de esto, en las actuales condiciones tecnológicas usadas en Internet y su dimensionamiento, es que la voz sobre IP, sería posible solamente en una red IP controlada del tipo intranet, mientras que en Internet será muchísimo más impredecible.

#### **3.4.4. Influencia de las Pérdidas de Paquetes**

La pérdida de un paquete hace que falte información cuando se recibe la señal de audio. Dependiendo del número de paquetes perdidos, la calidad sonora en el extremo receptor puede ser deficiente. En el IP la pérdida de paquetes es parte integrante del sistema y los encaminadores tienen que (utilizando el algoritmo de detección prematura aleatoria) destruir paquetes para evitar una posible congestión.

Existen cuatro causas posibles para la pérdida de paquetes:

- Duración de vida expirada (TTL = 0).
- Retardo en el extremo receptor superior a la fluctuación de fase de la memoria tampón.
- Destrucción por un módulo congestionado.
- Paquete no válido debido a fallos de transmisión.

El protocolo UDP se utiliza para transmitir voz mediante el IP, puesto que tiene la ventaja de utilizar menos tara y depende menos de protocolos de capa superior (como RTPC/RTP) para proporcionar control de errores o de flujo, o cuando las necesidades de tiempo real, hacen que la transmisión, tal como la utiliza el protocolo TCP, sea inadecuada.

La tasa de pérdida de paquetes dependerá de la calidad de las líneas utilizadas y del dimensionamiento de la red. Para que la calidad vocal sea aceptable, dicha tasa de pérdida de paquetes ha de ser menor que el 20%. Una solución posible para reducir la pérdida de paquetes consiste en utilizar sistemas de corrección de errores que tengan codificación redundante y adaptable, es decir, variable de acuerdo con las pérdidas de paquete estadísticamente observadas en la red en determinado momento. Es posible

obtener, cuando se utilizan dichos sistemas, unos niveles muy altos de calidad sonora, incluso por Internet. No obstante, esta solución genera otras dificultades relacionadas con el retardo total de transmisión, que, como ya se ha indicado, ha de controlarse si se ha de usar la red para telefonía.

#### **3.4.5. Influencia del Eco**

El eco es el tiempo que transcurre entre la transmisión de una señal y su regreso al transmisor. Por lo general, este problema aparece en el contexto de las comunicaciones de PC a teléfono, de teléfono a PC o de teléfono a teléfono, y es causado por los componentes electrónicos de las partes analógicas del sistema que reflejan una parte de la señal procesada.

Un eco menor que 50 ms es imperceptible. Por encima de este valor, el hablante oirá su propia voz después de haber hablado. Si se desea ofrecer un servicio de telefonía IP, las pasarelas (gateway) tendrán que procesar el eco generado por la transferencia de dos a cuatro hilos, de lo contrario, no será posible utilizar el servicio con equipos analógicos clásicos. Como solución, se están instalando compensadores de eco de alta calidad en la pasarela de la red.

## **4. CAPITULO IV. VOZ SOBRE IP (VOIP)**

### **4.1. ESTADO DEL ARTE**

#### **4.1.1. Evolución Histórica hacia la VoIP.**

La telefonía ha tenido grandes avances a través del tiempo, desde su inicio con los experimentos de la telegrafía de Marconi (1874 – 1937) hasta la actualidad, con los avances que ha tenido la informática y las telecomunicaciones, las cuales hoy hacen posible la comunicación a través de Internet y el envío de paquetes de voz a través de redes de datos que es lo que se denomina Voz sobre IP (VoIP).

Estos avances han traído consigo grandes ventajas a las empresas y usuarios de servicios de telefonía, debido al ahorro en costos de infraestructura y tarifas, puesto que al aprovechar el cableado de la red de datos para el envío de voz se minimizan los gastos de llamadas e incluso se pueden llegar a reducir a cero.

#### **4.1.2. Antecedentes de desarrollo de VoIP**

En lo que respecta al tema específico de VoIP, este comenzó como el resultado del trabajo de un grupo de jóvenes en Israel durante 1995. En aquella época, la única comunicación posible era de PC a PC. Poco más tarde Vocaltec, Inc. anunció el lanzamiento del primer Softphone que llamaron “Internet Phone Software”. Este Softphone estaba hecho para ser usado en un PC que tenía tarjeta de sonido, micrófono, parlantes y modem. El software funcionaba comprimiendo la señal de voz, convirtiéndola en paquetes de voz que eran enviados por Internet (exactamente igual que hoy). El software sólo funcionaba si los dos PC tenían el mismo software y el mismo hardware. Este fue comercialmente un fracaso principalmente porque las comunicaciones de banda ancha todavía no estaban disponibles.

En 1997 Jeff Pulver decide juntar por primera vez a los pocos usuarios, fabricantes, e interesados en esta tecnología en VON, la primer feria/congreso que actualmente sigue siendo el mayor evento de VoIP. Ahora Pulver organiza VON, 2 veces por año en EEUU, y ahora también una vez por año en varios países de Europa. También formó una compañía prestadora de servicio VoIP llamada FreeWorldDialup comúnmente

llamada FWD y es co-fundador de Vonage, el proveedor de VoIP más grande de EEUU. Pulver tiene varias empresas relacionadas con VoIP, entre ellas PulverMedia, su empresa encargada de organizar VON y publicar medios en todo el mundo.

En 1998 VoIP dio otro gran salto. Un grupo de emprendedores comenzó a fabricar los primeros ATA/Gateways para permitir las primeras comunicaciones PC a teléfono y, finalmente, las primeras comunicaciones teléfono a teléfono (con ATAs en cada extremo). Algunos de estos emprendedores inicialmente daban el servicio sin cargo a sus clientes para que pudieran probar la calidad y la tecnología. Estas llamadas contenían publicidad en el inicio y al final de cada comunicación. Estos servicios sólo se prestaban en EEUU y funcionaban gracias a esta publicidad. A menudo debía comenzarse la comunicación a través de una PC para luego pasar a un teléfono convencional. En este punto VoIP sumaba el 1% del total del tráfico de voz. Durante 1998 tres fabricantes comenzaron a fabricar switches de Capa 3 con QoS.

En 1999, Cisco vende sus primeras plataformas corporativas para VoIP. Se utilizaba principalmente el protocolo H.323 de señalización.

En el año 2000 VoIP representaba más del 3% del tráfico de voz. El mismo año Mark Spencer un estudiante de la Universidad de Auburn crea Asterisk, la primer central telefónica basada en Linux con un PC personal y la utilización de código fuente abierto. Asterisk hoy ofrece una solución freeware para hogares, pequeñas empresas y soluciones IP-PBX corporativas.

En 2002 el protocolo SIP comienza a desplazar al H.323.

En 2003 dos jóvenes universitarios - Jan Friis y Niklas Zennstrom - crean un softphone gratuito fácilmente instalable en cualquier PC que puede atravesar todos los firewalls y routers inclusive los corporativos. Ese producto es Skype, que se propaga con una velocidad increíble y llega en julio de 2007 a contar con 196 millones de usuarios en todo el mundo.

Lo anterior puede resumirse de la siguiente forma:

- 1995 – Inicio de la Voz sobre IP: La VoIP empieza con pequeñas aplicaciones gratuitas y de código abierto a raíz de la posibilidad de enviar pequeños fragmentos de voz codificados con algoritmos de compresión y pérdida. Rápidamente se empiezan a desarrollar aplicaciones para transmitir video aunque con un gran costo de ancho de banda y muy mala calidad de imagen.
- 1996 – Aparecen los protocolos de comunicaciones con aplicaciones como NetMeeting o GnomeMeeting, ICQ y muchísimos más, además de terminales análogos a teléfonos que funcionan con este protocolo.
- 1997 – Aparecen los primeros PBX de software. El protocolo H.323 se hace el “dueño y señor” de la VoIP ofreciendo voz y video aunque con mala calidad debido al ancho de banda: limitado y poco económico. De esta manera, empiezan a desarrollar hardware y software que actúa como centrales de VoIP para empresas utilizando la red local como transmisor y módems para realizar llamadas convencionales.
- 1998 - 1999 – La revolución de la banda ancha. Las conexiones de banda ancha empiezan a proliferar y la VoIP se mantiene estable aunque empiezan a nacer empresas que ven la VoIP como el futuro para llamadas telefónicas de bajo costo.
- Netmeeting permite conexión con un servidor H.323, CU-SeeMe se afianza como una de las aplicaciones de voz y vídeo más utilizados hasta el momento. Aparece el protocolo SIP evolución del arcaico H.323. Comienzo de Asterisk de la mano de Mark Spencer.
- 2000 – La revolución llega a la Voz sobre IP. Asterisk comienza como un software abierto y con un gran número de seguidores y apoyo. Las empresas aún no se fían de este software ni de Linux y continúan utilizando software y hardware de grandes empresas que continúan utilizando H.323.

- 2001 – 2007, Asterisk se afianza como símbolo de VoIP. Asterisk gana más y más adeptos. La empresa “Linux-support” se convierte en Digium especializada en la venta de hardware especial para Asterisk. No tardan en aparecer otros fabricantes que crean hardware exclusivamente compatible con Asterisk: como Sangoma y Junghanns, entre otros.

La evolución y desarrollo que ha tenido VoIP, tiende a desplazar el sistema de teléfonos tradicionales por teléfonos IP. Lo anterior, si se tiene en cuenta que poco a poco todo el mundo cambiará sus teléfonos tradicionales por teléfonos IP apoyados por los propios operadores de telefonía y servicios IP.

Pero hay que destacar que el camino a seguir por parte de los investigadores de voz sobre IP no ha sido fácil, pues han encontrado inconvenientes a través de su proceso de investigación, el cual, de una u otra forma, han contribuido para que el proyecto no se haya llevado a cabalidad ni desarrollado a un punto más alto; algunos de estos factores se relacionan a continuación:

- El constante freno que han aplicado las empresas de telefonía existentes, valiéndose de herramientas de tipo jurídicas.
- El afán de las grandes compañías telefónicas por adueñarse de esta tecnología para sacar el mejor provecho.
- El alto costo de los DSP's (Procesador Digital de señal).

Las primeras investigaciones hacia la convergencia de las actuales redes de telecomunicaciones se basaban en la utilización de multiplexores que permitían utilizar las redes WAN de datos de las empresas (típicamente conexiones punto a punto y *Frame Relay*) para la transmisión del tráfico de voz. La falta de estándares, así como el largo plazo de amortización de este tipo de soluciones han imposibilitado su implementación.

La necesidad de un estándar era evidente y observando el incuestionable éxito a nivel mundial de Internet, el protocolo IP parecía ser el medio que simplificaba en cierta

manera esta integración. A esta nueva solución se le conoce como VoIP (*Voice over Internet Protocol*).

El estándar VoIP define la tecnología que permite encapsular la voz en paquetes para poder ser transportados sobre redes IP sin necesidad de disponer de circuitos conmutados como es el caso de la red de telefonía conmutada (PSTN). La red convencional se basa en la conmutación de circuitos, es decir, al iniciarse la comunicación se establece un circuito físico durante el tiempo que dura la conversación. Esto implica la reserva de los recursos hasta que finalice la comunicación no pudiendo ser utilizado por otra, incluso durante los silencios que se suceden dentro de una conversación típica. En cambio, la telefonía IP no utiliza circuitos físicos para la conversación, sino que envía múltiples conversaciones a través del mismo canal (circuito virtual) codificadas en paquetes y en flujos independientes. Cuando se produce un silencio en una conversación, los paquetes de datos de otras conversaciones pueden ser transmitidos por la red, lo que implica un uso más eficiente del ancho de banda.

Desde que en 1995 la empresa *VocalTec* iniciara las primeras aplicaciones de comunicación de voz entre dos PCs a través de la red IP, han aparecido distintos niveles de desarrollo hacia la convergencia de redes:

- **Voz en Internet:** servicios de telefonía prestados sobre la red pública global formada por la interconexión de redes de conmutación de paquetes basadas en IP.
- **Voz sobre IP (VoIP):** servicios de telefonía prestados sobre redes IP "privadas" sin interconexión a la red PSTN o a la Red Digital de Servicios Integrados (RDSI).
- **Telefonía IP:** servicios de telefonía prestados sobre Redes IP privadas en interconexión con la PSTN y/o RDSI.
- **Voz sobre Frame Relay (VoFR):** servicios de telefonía prestados sobre redes Frame Relay orientadas a la transmisión de datos.

- **Voz sobre ATM (VoATM):** servicios de telefonía prestados sobre redes ATM donde existe posibilidad de ofrecer calidad de servicio (QoS: *Quality of Service*).
- **Multimedia sobre IP (MoIP):** servicios multimedia (vídeo, audio, imagen, etc.) prestados sobre redes IP.
- **Fax sobre IP (FoIP):** servicios de transmisión de fax sobre redes IP.
- **XoIP:** integración global de todos los servicios actuales y futuros que se puedan ofrecer sobre una red IP. El término X puede referirse por ejemplo a:
  - F = fax
  - M = multimedia
  - V = voz
  - D = datos

En definitiva, las redes IP parecen ser a priori la solución más inmediata para alcanzar la convergencia de redes debido sobre todo a su ámbito de cobertura actual, su aceptación por parte del usuario y la próxima aparición del protocolo IPv6. Integrar la voz en las redes IP aporta múltiples ventajas:

- La reducción de costes debido, por ejemplo, al mantenimiento, gestión y administración de una única red; llamadas gratuitas entre las distintas sedes de una empresa, etc.
- Mejor utilización del ancho de banda
- La integración de servicios en una misma infraestructura permite una mayor estandarización.
- Permite el control del tráfico de la red por lo que se disminuyen las posibilidades de que se produzcan caídas importantes en el rendimiento.
- Es independiente del tipo de red física que lo soporta.
- Permite utilizar tanto terminales hardware como software.
- Permite la integración de vídeo.
- Ofrece nuevos servicios de valor añadido como el correo de voz (voicemail), centro de llamadas vía Web, etc.

Sin embargo, existe un importante inconveniente que ha hecho que la expansión de la VoIP no sea tan rápida como se esperaba: la dificultad en ofrecer QoS. En la transmisión de voz es necesario que todos los paquetes lleguen ordenados, que no haya pérdidas y garantizar una mínima tasa de transmisión lo que implica la necesidad de QoS. En otros servicios como el correo, ofrecer QoS no es crítico, ya que si un paquete no ha llegado al destino se solicita su retransmisión; pero esto no es posible en la VoIP, ya que se trata de un servicio en tiempo real. La solución radica en diferenciar los paquetes de voz de los paquetes de datos, priorizar la transmisión de los paquetes de voz y evitar que la transmisión de los paquetes no supere los 150 milisegundos, tal y como se especifica en la recomendación ITU-TG 114. La calidad de servicio se está logrando en base a los siguientes criterios:

- La supresión de silencios y VAD (*Voice Activity Detection*), otorga más eficiencia a la hora de realizar una transmisión de voz, ya que se aprovecha mejor el ancho de banda al transmitir menos información.
- Compresión de cabeceras aplicando los estándares RTP/RTCP (*Real Time Protocol*).
- Cancelador de eco
- Priorización de los paquetes que requieran menor latencia:

**Custom Queuing (CQ):** asigna un porcentaje del ancho de banda disponible.

**Priority Queuing (PQ):** establece prioridad en las colas.

**Weight Fair Queuing (WFQ):** prioriza el tráfico de menor carga.

**DiffServ:** establece decisiones de rutas por paquete.

**Random Early Discard (RED):** control de congestión basado en descarte de paquetes de forma aleatoria.

- La implantación de IPv6 que proporciona mayor espacio de direccionamiento y la posibilidad de tunneling.

## 4.2. Principales componentes de la VOIP

Para la transmisión de voz sobre una red IP, el estándar 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 VoIP nativo y puede conectarse directamente a una red IP.

**Software:** un softphone es una aplicación audio ejecutable desde PC que se comunica con las PABX a través de la LAN. Para interactuar con el usuario se basa en la utilización de un micrófono y altavoz o mediante un teléfono USB.

**Servidor:** provee el manejo y funciones administrativas para soportar el enrutamiento de llamadas a través de la red. Este servidor puede adoptar diferentes nombres dependiendo del protocolo de señalización utilizado. Así en un sistema basado en el protocolo H.323, el servidor es conocido como **Gatekeeper**; en un sistema SIP, **servidor SIP**; y en un sistema basado en MGCP o MEGACO, **Call Agent** (Agente de llamadas). El servidor es un elemento opcional, normalmente implementado en software, y en caso de existir, todas las comunicaciones pasarían por él.

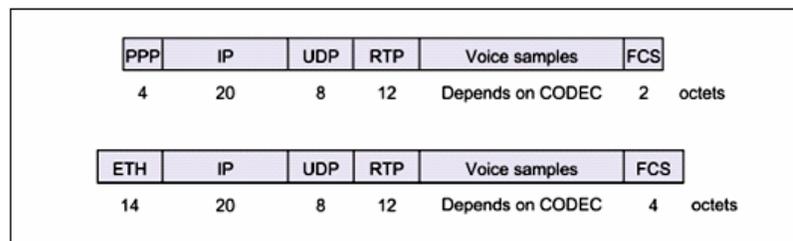
**Gateways:** enlace de la red VoIP con la red telefónica analógica o RDSI. Se encarga de adaptar las señales de estas redes a VoIP y viceversa, actuando de forma totalmente transparente para el usuario. El gateway posee, además de puertos LAN, interfaces de conexión a estas redes: FXO, FXS, E&M, BRI, PRI, G703/G.704

**Red IP:** provee conectividad entre todos los terminales. La red IP puede ser una red IP privada, una Intranet o Internet.

Los distintos elementos pueden residir en plataformas físicas separadas o bien pueden convivir varios elementos en la misma plataforma. De este modo es bastante habitual encontrar juntos servidor y gateway.

### 4.3. Encapsulamiento de una trama VoIP

Una vez que la llamada ha sido establecida, la voz será digitalizada y entonces transmitida a través de la red en tramas IP. Las muestras de voz son primero encapsuladas en RTP (protocolo de transporte en tiempo real) y luego en UDP (protocolo de datagrama de usuario) antes de ser transmitidas en una trama IP. La figura 12 muestra un ejemplo de una trama VoIP sobre una red LAN y WAN.



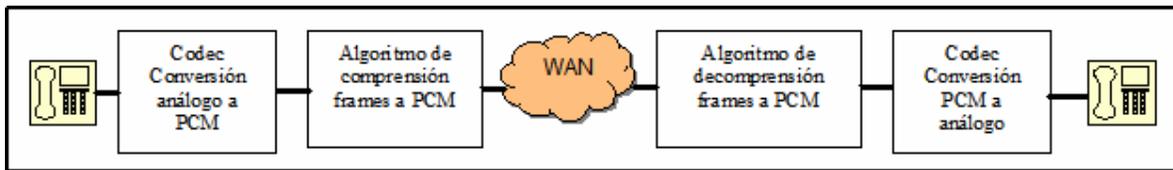
**Figura 12.** Ejemplo de una trama VoIP sobre una red LAN y WAN.

Por ejemplo, si el CODEC usado es G.711 y el periodo de paquetización es 20 ms, la carga útil será de 160 bytes. Esto resultara en una trama total de 206 bytes en una red WAN y en 218 bytes en una red LAN.

### 4.4. Funcionamiento de una red VoIP

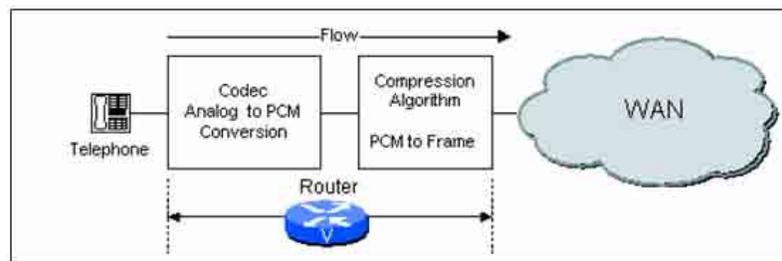
Años atrás, se descubrió que enviar una señal a un destino remoto también se podría enviar de manera digital es decir, antes de enviar la señal se debía digitalizar con un dispositivo ADC (conversor análogo a digital), transmitirla y en el extremo de destino transformarla de nuevo a formato análogo con un dispositivo DAC (conversor digital a análogo).

VoIP funciona de esa manera, digitalizando la voz en paquetes de datos, enviándola a través de la red y reconvirtiéndola a voz en el destino. Básicamente el proceso comienza con la señal análoga del teléfono que es digitalizada en señales PCM por medio del codificador/decodificador de voz (codec). Las muestras PCM son pasadas al algoritmo de compresión, el cual comprime la voz y la fracciona en paquetes (Encapsulamiento) que pueden ser transmitidos para este caso a través de una red privada WAN. En el otro extremo de la nube se realizan exactamente las mismas funciones en un orden inverso. El flujo de un circuito de voz comprimido es el mostrado en la figura 13.



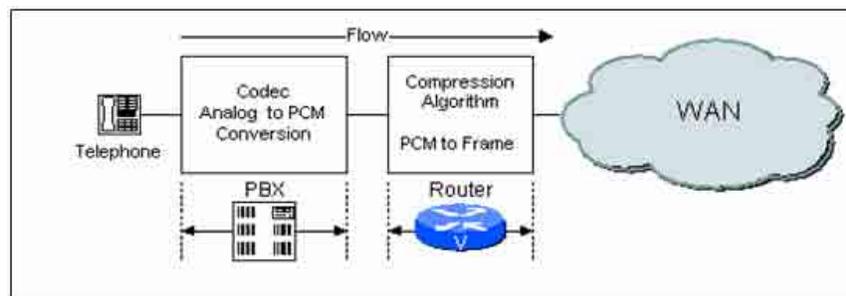
**Figura 13.** Flujo de una llamada VoIP

Dependiendo de la forma en la que la red este configurada, el Router o el gateway pueden realizar la labor de codificación, decodificación y/o compresión. Por ejemplo, si el sistema usado es un sistema análogo de voz, entonces el router o el gateway realizan todas las funciones mencionadas anteriormente como muestra la figura 14.



**Figura 14.** Router, Gateway de VoIP.

En cambio, como muestra la figura 15, si el dispositivo utilizado es un PBX digital, entonces es este el que realiza la función de codificación y decodificación, y el router solo se dedica a procesar y a encapsular las muestras PCM de los paquetes de voz que le ha enviado el PBX.



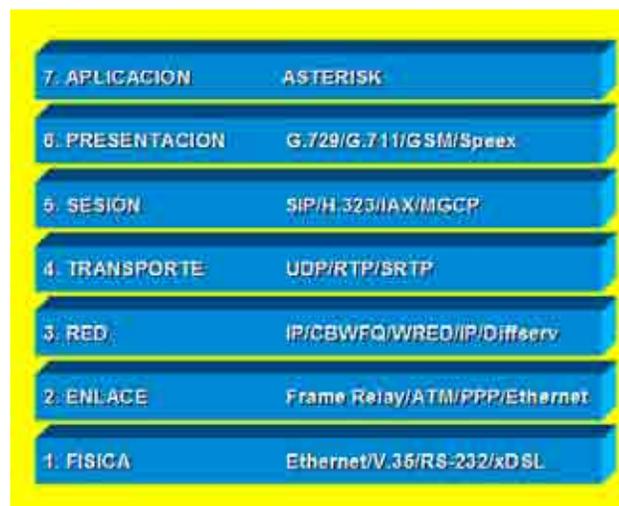
**Figura 15.** PABX y Router en VoIP

Para el caso de transportar voz sobre la red pública Internet, se necesita una interfaz entre la red telefónica y la red IP, el cual se denomina gateway y es el encargado en el lado del emisor de convertir la señal analógica de voz en paquetes comprimidos IP para ser transportados a través de la red. Del lado del receptor su labor es inversa, dado que descomprime los paquetes IP que recibe de la red de datos, y recompone el mensaje a su forma análoga original conduciéndolo de nuevo a la red telefónica convencional en el sector de la última milla para ser transportado al destinatario final y ser reproducido por el parlante del receptor.

Es importante tener en cuenta también que todas las redes deben tener de alguna forma las características de direccionamiento, enrutamiento y señalización.

El direccionamiento es requerido para identificar el origen y destino de las llamadas, también es usado para asociar las clases de servicio a cada una de las llamadas dependiendo de la prioridad. El enrutamiento por su parte encuentra el mejor camino a seguir por el paquete desde la fuente hasta el destino y transporta la información a través de la red de la manera más eficiente, la cual ha sido determinada por el diseñador. La señalización alerta a las estaciones terminales y a los elementos de la red su estado y la responsabilidad inmediata que tienen al establecer una conexión.

La siguiente imagen muestra en detalle la distribución de los protocolos VoIP dentro del modelo OSI:



**Figura 16.** Distribución de los protocolos VoIP dentro del modelo OSI

Como se puede ver en la figura de arriba, la voz sobre IP esta compuesta por diversos protocolos envolviendo varias capas del modelo OSI. De cualquier forma, VoIP es una aplicación que funciona sobre las redes IP actuales, tratando principalmente las capas de transporte, sesión, presentación y aplicación.

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

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

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

#### **4.5. 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. A continuación se describen algunos de los protocolos más importantes utilizados en VoIP.

##### **4.5.1. H.323**

H.323 es una familia de estándares desarrollado por la ITU en 1996 con el objetivo de ofrecer un mecanismo de transporte para servicios multimedia sobre redes que no garantizan QoS, 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 ido evolucionando para cubrir todas las necesidades de la VoIP. De hecho el protocolo VoIP generaliza los conceptos introducidos por H.323. Además especifica aspectos basados en el sistema de señalización número 7 (SS7) para la interconexión con la PSTN.

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. De hecho 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 y empresas actualmente ha disminuido debido principalmente a su complejidad y a ciertas ineficiencias detectadas en conferencias entre un número elevado de terminales.

#### **4.5.2. Protocolo SIP**

“Session Initiation Protocol” o SIP (Protocolo de Iniciación de Sesión), es un protocolo de señalización definido por el “Internet Engineering Task Force” o IETF que permite el establecimiento, la liberación y la modificación de sesiones multimedia (RFC3261). Este protocolo hereda de ciertas funcionalidades de los protocolos “Hyper Text Transport Protocol” o “http”, utilizados para navegar sobre el WEB y “Simple Mail Transport Protocol” o “SMTP”, utilizados para transmitir mensajes electrónicos (e-mails). SIP se apoya sobre un modelo transaccional cliente / servidor como http. El direccionamiento utiliza el concepto “Uniform Resource Locator” o “URL SIP” parecido a una dirección E-mail. Cada participante en una red SIP es entonces alcanzable vía una dirección, por medio de una URL SIP. Por otra parte, los requerimientos SIP son satisfechos por respuestas identificadas por un código digital. De hecho, la mayor parte de los códigos de respuesta SIP han sido tomados del protocolo http. Por ejemplo, cuando el destinatario no está ubicado, un código de respuesta «404 Not Found» está devuelto. Un requerimiento SIP está constituido de “headers” o encabezamientos, al igual que un mando SMTP. Por fin, SIP, al igual de SMPT es un protocolo textual.

SIP ha sido extendido con el fin de soportar numerosos servicios tales como la presencia, la mensajería instantánea (similar al servicio SMS en las redes móviles), la transferencia de llamada, la conferencia, los servicios complementarios de telefonía, etc.

SIP ha sido elegido por el 3GPP para la arquitectura “IP Multimedia Subsystem” o “IMS” como protocolo para el control de sesión y el control de servicio. El reemplazara en el futuro, los protocolos “ISUP”, utilizado para el control de llamada en la Red Telefónica Conmutada, y “INAP”, utilizado para el control de servicio en la arquitectura Red Inteligente.

El protocolo SIP es solo un protocolo de señalización. Una vez la sesión establecida, los participantes de la sesión intercambian directamente su trafico audio / video a través del protocolo “Real-Time Transport Protocol” o RTP. Por otra parte, SIP no es un protocolo de reservación de recursos, y en consecuencia, no puede asegurar la calidad de servicio. Se trata de un protocolo de control de llamada y no de control del medio.

SIP tampoco es un protocolo de transferencia de archivos tal como “http”, usado con el fin de transportar grandes volúmenes de datos. Ha sido concebido para transmitir mensajes de señalización cortos con el fin de establecer, mantener y liberar sesiones multimedia. Mensajes cortos, no relativos a una llamada pueden sin embargo ser transportados por SIP al estilo de SMS.

### **Entidades SIP**

SIP define dos tipos de entidades: los clientes y los servidores. De manera mas precisa, las entidades definidas por SIP son:

- **El Servidor Proxy** (Proxy Server): el recibe solicitudes de clientes que el mismo trata o encamina hacia otros servidores después de haber eventualmente, realizado ciertas modificaciones sobre estas solicitudes.
- **El Servidor de Redireccionamiento** (Redirect Server): se trata de un servidor quien acepta solicitudes SIP, traduce la dirección SIP de destino en una o varias direcciones de red y las devuelve al cliente. De manera contraria al Proxy Server, el Redirect Server no encamina las solicitudes SIP. En el caso de la devolución de una llamada, el Proxy Server tiene la capacidad de traducir el numero del destinatario en el mensaje SIP recibido, en un numero de reenvió de llamada y encaminar la llamada a este nuevo destino, y eso de manera transparente para el cliente de origen; para el mismo

servicio, el Redirect Server devuelve el nuevo numero (numero de reenvió) al cliente de origen quien se encarga de establecer una llamada hacia este nuevo destino.

- **El Agente Usuario** (User Agent) o “UA”: se trata de una aplicación sobre un equipo de usuario que emite y recibe solicitudes SIP. Se materializa por un software instalado sobre un « User Equipment » o UE : una PC, un teléfono IP o una estación móvil UMTS.

- **El Registrador** (Registrar): se trata de un servidor quien acepta las solicitudes SIP REGISTER. SIP dispone de la función de registro de los usuarios. El usuario indica por un mensaje REGISTER emitido al Registrar, la dirección donde es localizable (dirección IP).

El “Register” actualiza entonces una base de dato de localización. El registrador es una función asociada a un Proxy Server o a un Redirect Server. Un mismo usuario puede registrarse sobre distintas UAs SIP, en este caso, la llamada le será entregada sobre el conjunto de estas UAs.

## Métodos SIP

El RFC 3261 define seis solicitudes / requerimientos o métodos SIP.

El método “**INVITE**” es usado con el fin de establecer una sesión entre UAs. INVITE corresponde al mensaje ISUP IAM o al mensaje Q.931 SET UP y contiene las informaciones sobre el que genera la llamada y el destinatario así como sobre el tipo de flujos que serán intercambiados (voz, video,...).

Cuando un UA que emitió el método SIP INVITE recibe una respuesta final a la invitación (ejemplo: 200 OK), el confirma la recepción de esta respuesta por medio de un método “**ACK**”. Una respuesta del tipo “busy” o “answer” es considerada como final mientras una respuesta tipo “ringing” significando que el destinatario ha sido avisado es una respuesta provisoria.

El método “**BYE**”permite la liberación de una sesión anteriormente establecida. Corresponde al mensaje RELEASE de los protocolos ISUP y Q.931. Un mensaje BYE puede ser emitido por el que genera la llamada o el que la recibe.

El método “**REGISTER**” es usado por una UA con el fin de indicar al Registrar la correspondencia entre su Dirección SIP y su dirección de contacto (ejemplo: dirección IP).

El método “**CANCEL**” es utilizado para pedir el abandono de la llamada en curso pero no tiene ningún efecto sobre una llamada ya aceptada. De hecho, solo el método “**BYE**” puede terminar una llamada establecida.

El método “**OPTIONS**” es utilizado para interrogar las capacidades y el estado de un User Agent o de un servidor. La respuesta contiene sus capacidades (ejemplo: tipo de media siendo soportado, idioma soportado) o el hecho de que el UA sea indisponible.

## **Respuestas SIP**

Después de haber recibido y interpretado un requerimiento SIP, el destinatario de este requerimiento devuelve una respuesta SIP. Existen seis clases de respuestas:

- Clase 1xx : Información, el requerimiento ha sido recibido y esta en curso de tratamiento
- Clase 2xx: Éxito, el requerimiento ha sido recibido, entendido y aceptado.
- Clase 3xx: Reenrutamiento, la llamada requiere otros procesamientos antes de poder determinar si puede ser realizada.
- Clase 4xx: Error requerimiento cliente, el requerimiento no puede ser interpretado o servido por el servidor. El requerimiento tiene que ser modificado antes de ser reenviado.
- Clase 5xx: Error servidor, el servidor fracasa en el procesamiento de un requerimiento aparentemente válido.
- Clase 6xx: Fracaso global, el requerimiento no puede ser procesado por ningún servidor.

## **Funcionamiento del protocolo SIP**

### **Inscripción a la red SIP**

El método “**REGISTER**” es utilizado por un “**USER AGENT**” con el fin de indicar a la función Registrar (físicamente implantada en un Proxy Server o un Redirect Server) la correspondencia entre su dirección SIP (ejemplo: sip :mary.Taylor@orange.com) y su

dirección IP (ejemplo: sip:mary.Taylor@192.190.132.20). La dirección IP puede ser estática o obtenida de modo dinámico por DHCP. La función Registrar actualiza entonces una base de datos de localización. Desde este momento, el User Agent puede recibir llamadas ya que se encuentra ubicado. Si un usuario SIP desea reenviar sus llamadas de su dominio corriente hacia otro dominio, (ejemplo: del dominio orange.com al dominio francetelecom.com), solo tendrá que indicar a la función Registrar de orange.com su dirección SIP en el dominio francetelecom.com. Cuando un mensaje INVITE debe ser entregado por el Proxy Server del dominio orange.com a sip: mary.Taylor@orange.com, la base de datos actualizada por la función Registrar indica al Proxy Server que el mensaje tiene que ser relevado a sip:mary.Taylor@francetelecom.com. Entonces, el Proxy Server efectúa una búsqueda por el DNS de la dirección IP del Proxy Server del dominio francetelecom.com con el fin de relevar el mensaje SIP a encaminar al destino apropiado (sip:mary.Taylor@francetelecom.com).

En una red IP Multimedia Subsystem o IMS, el Proxy Server corresponde a una entidad CSCF (Call State Control Function), mientras la base de datos de localización es representada por la entidad Home Subscriber Server o HSS. El HSS en el IMS por los móviles es un HLR conteniendo por otra parte el perfil del usuario para los servicios IMS suscritos.

### **Establecimiento y liberación de sesión SIP**

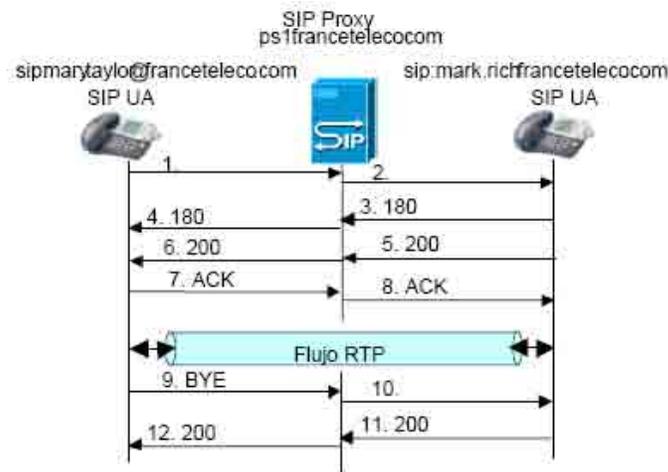
En el ejemplo siguiente, el que llama tiene como URL SIP sip: mary.taylor@francetelecom.com, mientras la URL SIP del destinatario de la llamada es sip: mark.rich@francetelecom.com.

Un mensaje de establecimiento de llamada SIP INVITE esta emitido por parte de la UA SIP del que llama al Proxy Server. Este ultimo interroga la base de datos de localización para identificar la localización del que esta llamado (dirección IP) y encamina la llamada a su destino. El mensaje INVITE contiene distintos "headers" o encabezamientos obligatorios, entre los cuales la dirección SIP de la persona que llama "From", la dirección SIP de la persona que recibe la llamada "To", una identificación de la llamada "Call-ID", un numero de secuencia "Cseq", un numero máximo de saltos "max-forwards". El encabezamiento "Via" esta actualizado por todas las entidades que

participaron al enrutamiento del requerimiento INVITE. Eso asegura que la respuesta seguirá el mismo camino que el requerimiento.

Por otra parte, el requerimiento SIP INVITE contiene una sintaxis "Session Description Protocol" o SDP. Esta estructura consiste en varias líneas que describen las características del media que el que llama "Mary" necesita para la llamada.

Mary Taylor indica que la descripción SDP utiliza la versión 0 del protocolo, que se trata de una sesión telefónica (m = audio), que la voz constituida en paquetes le debe ser entregada a la dirección de transporte (puerto UDP = 45450, dirección IP =192.23.34.45) con el protocolo RTP y utilizando un formato de codificación definido en el RFC "Audio Video Profile" o AVP y pudiendo ser G. 711  $\mu$ -law o G.728.



**Figura 17.** Establecimiento y liberación de sesión SIP

### Detalle del flujo de la llamada

**INVITE** sip:mark.rich@francetelecom.com SIP/2.0  
Via : SIP/2.0/UDP station1.francetelecom.com:5060

Max-Forwards : 20

To : Mark Rich <sip:mark.rich@francetelecom.com>

From : Mary Taylor <sip:mary.taylor@francetelecom.com>

Call-Id: 23456789@station1.francetelecom.com

CSeq: 1 INVITE  
Contact: mary.taylor@192.190.132.20  
Content-Type: application/sdp  
Content-Length:162

v = 0  
c = IN IP4 192.190.132.20  
m = audio 45450 RTP/AVP 0 15

La respuesta **180 RINGING** esta devuelta por el destinatario a la UA del que genera la llamada.

Cuando el destinatario acepta la sesión, la respuesta **200 OK** esta emitida por su UA y encaminada hacia la UA del que genera la llamada.

SIP/2.0 **200 OK**  
Via : SIP/2.0/UDP ps1.francetelecom.com:5060  
Via : SIP/2.0/UDP station1.francetelecom.com:5060  
Max-Forwards : 20  
To : Mark Rich <sip:mark.rich@francetelecom.com>  
From : Mary Taylor <sip:mary.taylor@francetelecom.com>  
Call-Id: 23456789@station1.francetelecom.com  
CSeq: 1 INVITE  
Contact: mark.rich@192.190.132.27  
Content-Type: application/sdp  
Content-Length:162

v = 0  
c = IN IP4 192.190.132.27  
m = audio 22220 RTP/AVP 0

La UA del que genera la llamada devuelve un método **ACK** al destinatario, relevada por la entidad Proxy Server.

La entidad Proxy Server participa al encaminamiento de la señalización entre UAs mientras que las UAs establecen directamente canales **RTP** para el transporte de la voz o de video en forma de paquetes sin implicación del Proxy Server en este transporte.

Cuando Mary cuelga, su UA envía un requerimiento **BYE** para terminar la sesión. Este requerimiento esta entregado al Proxy Server quien lo encamina a la UA de Mark. Este último, devuelve la respuesta **200 OK**.

**BYE** sip:mark.rich@francetelecom.com SIP/2.0  
 Via : SIP/2.0/UDP station1.francetelecom.com:5060  
 Max-Forwards : 20  
 To : Mark Rich <sip:mark.rich@francetelecom.com>  
 From : Mary Taylor <sip:mary.taylor@francetelecom.com>  
 Call-Id: 23456789@station1.francetelecom.com

CSeq: 2 **BYE**  
 SIP/2.0 **200 OK**  
 Via : SIP/2.0/UDP ps1.francetelecom.com:5060  
 Via : SIP/2.0/UDP station1.francetelecom.com:5060  
 Max-Forwards : 20  
 To : Mark Rich <sip:mark.rich@francetelecom.com>  
 From : Mary Taylor <sip:mary.taylor@francetelecom.com>  
 Call-Id: 23456789@station1.francetelecom.com  
 CSeq: 2 BYE

### **Extensiones del protocolo SIP**

Una entidad SIP puede suscribir a un evento con el fin de ser notificada de su ocurrencia. El requerimiento **SUBSCRIBE** permite la suscripción mientras el requerimiento **NOTIFY** es utilizado con el fin de notificar (RFC 3265). El método **PUBLISH** permite publicar su estado.

El método **REFER** (RFC3515) reenvía el receptor hacia un recurso identificado en el método. REFER permite emular distintos servicios o aplicaciones incluyendo la transferencia de llamada. Contemplamos T1, la entidad que origino la transferencia, T2 la

entidad transferida y T3, el destinatario de la transferencia. La transferencia de llamada permite a T1 transformar una llamada en curso entre T1 y T2 en una nueva llamada entre T2 y T3, elegida por T1. Si la transferencia de llamada se lleva a cabo, T2 y T3 podrán comunicar mientras que T1 no podrá seguir dialogando con T2 o T3.

El método **MESSAGE** (RFC 3428) ha sido propuesto como extensión al protocolo SIP con el fin de permitir la transferencia de mensajes instantáneos. La mensajería instantánea o “Instant Messaging” o “IM” consiste en el intercambio de mensajes entre usuarios en pseudo tiempo real. Este nuevo método hereda de todas las funciones ofrecidas por el protocolo SIP tales que el enrutamiento y la seguridad. El requerimiento MESSAGE puede transportar varios tipos de contenidos basándose sobre la codificación MIME.

El método **INFO** (RFC2976) permite transferir informaciones de señalización durante la llamada. Entre los ejemplos de información se encuentran los dígitos DTMF, las informaciones relativas a la tasación de una llamada, las imágenes etc.

Las respuestas finales 2xx, 3xx, 4xx, 5xx y 6xx a un requerimiento INVITE son satisfechas por el requerimiento **ACK** mientras las respuestas provisorias de tipo 1XX no son satisfechas. Ciertas respuestas temporarias tales como el 180 Ringing son criticas y su recepción es esencial para la determinación del estado de la llamada, entre otros durante el proceso de interconexión con la RTCP. El método **PRACK** (RFC3262) ha sido definido con el fin de satisfacer la recepción de respuestas temporarias de tipo 1XX.

El método **UPDATE** (RFC3311) permite a un terminal SIP actualizar los parámetros de una sesión multimedia (ejemplo : flujo media y sus codecs). El método UPDATE puede ser enviado antes de que la sesión sea establecida. UPDATE es entonces particularmente útil cuando se trata de poner al día los parámetros de sesión antes de su establecimiento, por ejemplo en puesta en espera del destinatario.

### **Arquitectura de servicios SIP**

La arquitectura de servicios SIP de base esta constituida de servidores de aplicación, de servidores de media y de S-CSCF.

El **servidor de aplicación** SIP ejecuta servicios (ejemplo: Push To Talk, Presence, Prepaid, Instant messaging etc...) y pueden influenciar el desempeño de la sesión a pedido del servicio. El servidor de aplicación corresponde al SCP de la Red Inteligente.

El **servidor de media** SIP (llamado en las recomendaciones el Multimedia Resource Function o MRF) establece conferencias multimedia, toca anuncios vocales o multimedia y colecta informaciones de usuario . Se trata de la evolución de la entidad Specialized Resource Point o SRP en el mundo multimedia.

El **servidor de llamada SIP (Proxy Server)** tiene el papel de punto desde el cual un servicio puede ser requerido. El dispone del perfil de servicio del abonado que le indica los servicios suscritos por el abonado y bajo cuales condiciones invocar estos servicios. Corresponde al SSP de la arquitectura Red Inteligente.

#### **4.5.3 MGCP-MEGACO**

*Media Gateway Control Protocol* (MGCP) 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, simplificando la integración con el protocolo SS7.

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. En definitiva, SIP y H.323 se utiliza para la señalización en los extremos, mientras que MEGACO es óptimo para los grandes operadores de telefonía.

#### **4.5.4. IAX**

*Inter-Asterisk eXchange protocol* (IAX) fue desarrollado por *Digium* para la comunicación entre centralitas basadas en *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 (*Network Address Translation*). 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. Se caracteriza por ser robusto y simple en comparación con otros protocolos. Permite manejar una gran cantidad de *códecs* y transportar cualquier tipo de datos.

#### **4.6. Tipos de arquitecturas**

En el pasado, todas las redes de voz fueron construidas usando una arquitectura centralizada en la cual los teléfonos fueron controlados por los conmutadores centralizados. Sin embargo este modelo trabajo bien para los servicios de telefonía básica.

Uno de los beneficios de la tecnología VoIP, es que permite a las redes ser construidas usando una arquitectura centralizada o bien distribuida. Esta flexibilidad permite a las compañías construir redes caracterizadas por una administración simplificada e innovación de Endpoints (teléfonos), dependiendo del protocolo usado.

##### **4.6.1. Arquitectura centralizada**

En general, la arquitectura centralizada esta asociada con los protocolos MGCP y MEGACO. Estos protocolos fueron diseñados para un dispositivo centralizado llamado Controlador Media Gateway o Call Agent, que maneja la lógica de conmutación y control de llamadas. El dispositivo centralizado comunica al Media Gateways, el cual enruta y transmite la porción audio/media de las llamadas (la información de voz actual).

En la arquitectura centralizada, la inteligencia de la red es centralizada y los dispositivos finales de usuario (endpoints) son relativamente mudos (con características limitadas). Sin embargo, muchas arquitecturas VoIP centralizadas usan protocolos MGCP o MEGACO.

Los defensores de la arquitectura VoIP centralizada, apoyan este modelo porque centraliza la administración, el provisionamiento y el control de llamadas. Simplifica el flujo de llamadas repitiendo las características de voz. Es fácil para los ingenieros de voz entenderlo. Los críticos de la arquitectura VoIP centralizada demandan que se suprimen las innovaciones de las características de los teléfonos (endpoints) y que llegara a ser un problema cuando se construyan servicios VoIP que muevan mas allá de características de voz.

#### **4.6.2. Arquitectura distribuida**

La arquitectura distribuida esta asociada con los protocolos H.323 y SIP. Estos protocolos permiten que la inteligencia de la red se distribuida entre dispositivos de control de llamadas y endpoints. La inteligencia en esta instancia se refiere a establecer las llamadas, características de llamadas, enrutamiento de llamadas, provisionamiento, facturación o cualquier otro aspecto de manejo de llamadas. Los Endpoints pueden ser Gateways VoIP, teléfonos IP, servidores media, o cualquier dispositivo que pueda iniciar y terminar una llamada VoIP. Los dispositivos de control de llamadas son llamados Gatekeepers en una red H.323, y servidores Proxy o servidores Redirect en una red SIP.

Los defensores de la arquitectura VoIP distribuida apoyan este modelo por su flexibilidad. Permite que las aplicaciones VoIP sean tratadas como cualquier otra aplicación IP distribuida, y permite la flexibilidad para añadir inteligencia a cualquier dispositivo de control de llamadas o Endpoints, dependiendo de los requerimientos tecnológicos y comerciales de la red VoIP. La arquitectura distribuida son usualmente bien entendida por los ingenieros que manejan redes de datos IP. Los críticos de la arquitectura distribuida comúnmente apuntan a la existencia de la Infraestructura PSTN como el único modelo de referencia que debiera ser usado cuando intentamos repetir los servicios de voz. Ellos también notan que las redes distribuidas tienden a ser más complejas.

#### 4.7. CENTRALES TELEFONICAS

Una central de telefonía privada (PBX) es un dispositivo que permite a las empresas conectar sus terminales telefónicos de forma independiente al proveedor de telefonía. De esta forma se consigue que todas las llamadas internas de una misma empresa sean conmutadas directamente sin necesidad de salir al exterior por la red pública de telefonía (PSTN o RDSI) disminuyendo notablemente los costos mensuales.

Las primeras PBX requerían la intervención de una persona encargada de conectar distintos cables para establecer la comunicación entre los distintos terminales de una empresa. Estas centrales eran conocidas como PBMX (*Manual PBX*).

El avance tecnológico rápidamente permitió prescindir de estos operadores para dar paso a un nuevo sistema electromecánico de conmutación totalmente automático llamados PABX (*Automatic PBX*).



PBMX



Nortel BCM 400 PABX

A todos los dispositivos conectados a la PBX se les conoce como extensiones y tanto pueden ser teléfonos, como faxes o módems, aunque estos últimos pueden degradar la calidad de la línea. Además, también es posible conectar a la central un determinado número de líneas troncales para poder realizar y recibir llamadas del exterior e incluso conectar varias PBX entre si para realizar llamadas entre las distintas sedes de una compañía. Normalmente para establecer la comunicación con el exterior, la PBX requiere que se marque el dígito 9 ó 0 seguido del número destino. De esta forma, la central es capaz de identificar que se trata de una llamada hacia el exterior y así poder seleccionar la utilización de una de las líneas troncales disponibles.

#### 4.7.1. Funcionalidades

Tal y como se ha definido anteriormente, el objetivo principal de una central PBX es establecer y mantener la comunicación entre dos puntos finales durante todo el tiempo requerido por los usuarios.

Existe un gran número de empresas, como *Nortel*, *Alcatel*, *Cisco*, *Ericsson*, *Fujitsu*, *NEC*, *Nortel*, *Panasonic*, *Samsung*, *Siemens* o *Toshiba* entre otros, que ofrecen una gran variedad de PBXs. Cada uno de estos fabricantes hacen esfuerzos para diferenciar sus productos sobre el de sus competidores, por eso añaden a sus centrales nuevos servicios de valor agregado. A continuación se enumeran alguno de los servicios más extendidos en las centralitas telefónicas:

**Operadora automática/virtual:** permite al llamante transferir la llamada a la extensión deseada mediante menús interactivos sin la intervención física de una operadora. Es un sistema basado en el reconocimiento de voz y/o de tonos DTMF (Dual Tone MultiFrequency), generados al marcar el teclado del teléfono. De esta forma se consigue sustituir la labor efectuada por una persona que sólo podrá atender una llamada al tiempo, por un servicio de atención automatizado capaz de atender múltiples llamadas simultáneamente.

**Marcación rápida** a números de servicio público como emergencias, policía o bomberos.

**Buzón de voz:** servicio de almacenamiento de mensajes de voz (contestador automático). El mensaje de bienvenida puede personalizarse.

**Transferencia de una llamada** a otra extensión para que sea atendida, por ejemplo, por otro departamento.

**Desvío de llamada** a otra terminal en caso de que la extensión no conteste o esté ocupada.

**Follow-me:** listado de números a los que redireccionar la llamada en caso de que la extensión no conteste. Los usuarios pueden configurar esta lista, por ejemplo, para desviar la llamada a su móvil en caso de no encontrarse en su puesto de trabajo.

**Llamada en espera, parking de llamadas (*call park*):** posibilidad de mantener conversaciones en espera para atender una nueva llamada entrante.

**Música en espera (MOH: *Music on Hold*):** servicio de reproducción de música para rellenar el silencio producido al mantener al llamante espera.

**Tarificación de llamadas:** sistema de cálculo del coste de una llamada.

**CallerID** o identificación de llamada.

**DDI (*Direct Dialling-In*):** enrutación de llamadas mediante la marcación directa a una extensión desde el exterior.

**Salas de conferencia:** conversación entre más de dos terminales.

**Listas negras:** restricción del acceso a determinados números.

**Registro y listado de llamadas** entrantes y salientes.

**Envío y recepción automática de faxes.**

**Grabación y escucha** de llamadas.

**Integración con bases de datos:** posibilidad de almacenar y recuperar información.

#### **4.7.2. Interfaces de acceso a la red pública**

Se entiende por interfaz al circuito físico que establece la conexión entre dos sistemas permitiendo su comunicación. Las interfaces no son universales, sino que existen diferentes estándares que establecen especificaciones técnicas concretas.

En la mayoría de los casos la comunicación se realiza desde una PBX a la red pública de telefonía (PSTN). Básicamente existen tres opciones de líneas telefónicas hoy en el mercado. Las líneas analógicas, que son las más comunes y son entregadas usando un par de cobre. Las líneas Digitales son usadas cuando son necesarias muchas líneas analógicas. La línea digital es normalmente entregada la través de un MODEM HDSL de 2 Mbps a través de fibra óptica con un MUX Digital como transporte. La tercera opción y más reciente, es la entrega de la línea telefónica usando voz sobre IP normalmente través del protocolo SIP.

#### **Interfaces FX (Foreign eXchange)**

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

#### **FXO (Foreign eXchange Office)**

Básicamente utilizadas para la comunicación entre una central de conmutación pública (PSTN) con una PBX.

Una puerta FXO en Asterisk se comunica directamente con la red pública, esta comunicación requiere tono de discado, indicación de campanilla y proveer indicadores de llamadas en progreso. Las interfaces FXO conectan el PBX a otro conmutador (PBX, Red Pública, gateway de voz sobre IP).

Es muy común conectar una interface FXO de una central telefónica a un gateway VoIP y transportar la voz empaquetada para otro gateway donde una interface FXS conecta un teléfono. Esta operación es conocida como OPX (Off-Promises Extension).

### **FXS (Foreign eXchange Station)**

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

### **Interfaces E & M**

Son también interfaces analógicas. El termino “E & M” viene de “Ear (receive) and Mouth (transmit)”. Usadas principalmente en las comunicaciones entre PBXs de una red a otra.

Por el momento no existe en el mercado hardware E&M para Asterisk. Estas tarjetas son conocidas en el mercado de telefonía como “tie-lines” analógicas. La mayoría de las centrales no vienen con este tipo de interface, mucha de las centrales de marcas conocidas poseían E&M como un opcional. Las placas E&M permiten una comunicación bi-direccional, pueden dar y recibir tono. Si se requiere usar una interface E&M con Asterisk la mejor opción es la integración de un gateway de voz.

### **Líneas Digitales E1/T1, Señalización CAS y CCS.**

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

- **ISDN** (Red Digital Integrada de Servicios): es una nueva forma de entroncamiento (desde 1990) Un simple par de cobre, puede transportar dos líneas, más un circuito de datos de 16kbps usados como señalización.

ISDN permite una forma bastante elegante de ofrecer el servicio de llamadas. Por ejemplo, servicios como: Caller-ID, llamada en espera, servicios de SMS, entre otros fueran originalmente desarrollados para ISDN.

- **MFC/R2** es una Señalización definida para la ITU (Q.421/Q.441), usada principalmente en América Latina y Asia. La Señalización usa CAS, ahora pasa las señalizaciones de cada canal por el canal 16. El R2 posee variaciones específicas para cada país.

**PRI:** acceso primario RDSI (30B+D)

**G703/G.704. (E&M digital)** conexión específica a PBX a 2 Mbps.

El número de interfaces dependerá del tamaño y las necesidades del escenario a implementar.

#### **4.7.3. IPBX**

La tendencia actual de los fabricantes de PBXs es incorporar a sus centrales la posibilidad de transmitir la voz sobre redes de datos. No es sólo la reducción de costos por la gestión de una única infraestructura lo que se le ofrece al cliente, sino que la integración simplifica y amplía las posibilidades de generar nuevos servicios de valor agregado.

El término IPBX (*Intranet* PBX) hace referencia a aquellas centrales capaces de transmitir la voz sobre redes IP basándose en el protocolo VoIP (*Voice over Internet Protocol*). Para la conexión a la red de Área Local (LAN) hace uso de tarjetas ethernet, y al igual que el resto de PBXs, también posee alguna de las interfaces anteriormente definidas para la conexión con otras redes de voz. Esto implica la necesidad de complejos mecanismos software que adapten la señal de voz durante la comunicación a cada uno de los diferentes estándares.

## 5. CAPÍTULO V. ASTERISK

Uno de los objetivos del presente proyecto es la implementación de una central para el servicio de VoIP.

*Asterisk* es la solución más económica y flexible que permite construir una IPBX en un computador convencional. Antes de iniciar la instalación y configuración del software es imprescindible describir las características que posee esta herramienta, así como el escenario de pruebas que definirá las especificaciones de dicha IPBX.

### 5.1. ASTERISK IPBX

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

Asterisk permite conectividad en tiempo real entre las redes PSTN y redes Voip. Asterisk es mucho mas que un PABX central. Con Asterisk en su red, se pueden crear cosas nuevas en telefonía como:

- Conectar empleados trabajando desde casa para un PABX de escritorio sobre conexiones de banda ancha.
- Conectar equipos en varias provincias sobre IP. Esto puede ser hecho por Internet o por una red IP privada.
- Dar a los funcionarios, correo de voz, integrándolo con una “web” y sus e-mail.
- Construir aplicaciones de respuesta automática por voz, que pueden conectase a un sistema de pedidos, o a otras aplicaciones internas.

- Dar acceso al PABX de la compañía para usuarios que viajan, conectando sobre la VPN de un aeropuerto o un hotel.

Asterisk incluye muchos recursos que solo eran encontrados en sistemas de mensajería unificada como:

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

## **5.2. El Rol de Digium**

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

Digium es hoy el principal patrocinador de Asterisk y uno de los líderes de la industria de PABX en código abierto, siendo Mark Spencer el creador y principal soporte de Asterisk.

### **5.3. El proyecto Zapata**

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

### **5.4. Reducción de Costos Utilizando Asterisk**

Al comparar un PABX tradicional con Asterisk talvez la diferencia sea pequeña, principalmente por los costos de hardware y los teléfonos IP. Entretanto, Asterisk solo puede ser comparado a un PABX digital. Comparar una central analógica de cuatro troncales y 16 líneas con Asterisk es injusto.

Cuando se agregan recursos avanzados como Voz sobre IP, la diferencia de costo es menor, en diversas oportunidades.

#### **Tener control sobre el sistema de telefonía**

Este es uno de los beneficios más citados, en vez de esperar que alguien configure su PABX propietario, puede ser configurado por uno mismo. Total libertad y interfaces estándar. En fin de cuentas es LINUX y es libre.

#### **Ambiente de desarrollo fácil y rápido**

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

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

#### **Posibilidad de proveer contenido dinámico.**

Como Asterisk es programado con C u otros lenguajes de dominio de la mayoría de los programadores, las posibilidades de proveer contenido dinámico, así como servicios de valor agregado, no tienen límites.

## Corre bajo Linux y es código abierto

Una de las cosas más destacables de Linux es la comunidad de software libre. Cuando se accede al Wiki, o los foros de software en código abierto se percibe que la adopción de nuevos usuarios es muy rápida, millones de preguntas y comentarios de problemas son enviados todos los días.

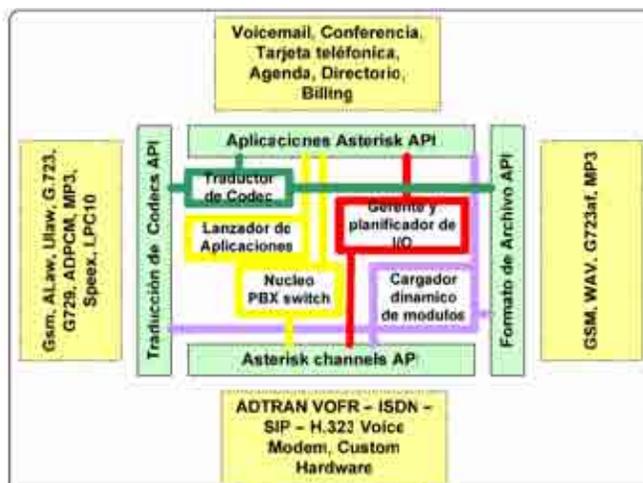
Asterisk es probablemente uno de los software que más personas tienen disponible para probar. Esto torna el código estable y permite una rápida resolución de problemas.

## 5.5. Limitaciones de la arquitectura de Asterisk

Asterisk usa una CPU de servidor para procesar los canales de voz, en vez de tener un DSP (procesador de señales digitales) dedicado a cada canal. En tanto que esto permitió que el costo fuese reducido para las tarjetas E1/T1, el sistema es muy dependiente de la performance de CPU.

Una recomendación es preservar al máximo la CPU de Asterisk, correrlo siempre en un servidor dedicado y probar el dimensionamiento antes de implementarlo.

## 5.6. Arquitectura de Asterisk



**Figura 18.** Arquitectura de ASTERISK

La figura muestra la arquitectura básica de Asterisk. A continuación se explicaran los conceptos relacionados con este esquema como los canales, los codecs y las aplicaciones.

### 5.6.1. Canales

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

Algunos de los hardwares que Asterisk soporta son:

- Zaptel – Wildcard T410P – Placa E1/T1 con cuatro puertas (PCI 3.3 volts)
- Zaptel – Wildcard T405P – Placa E1/T1 con cuatro puertas (PCI 5 volts apenas)
- Zaptel – TDM400P – Placa con cuatro puertas para tel. analógicos y ADSI
- Zaptel - TE110P – Placa con E1/T1 con una puerta, medio-comprimido.
- Quicknet, - las placas quicknet, tanto PhoneJack como Line Jack pueden ser usadas con Asterisk.
- ISDN4Linux – Es un driver antiguo para placas ISDN BRI, acceso básico. Placas de este standard pueden ser usadas en Asterisk.
- ISDN CAPI – Es la otra forma de soportar las placas ISDN BRI en Linux. Placas que soportan este standard pueden ser usadas con Asterisk.
- Voicetronix: poseen placas con mayor densidad de canales FXS y FXO que las placas de Digium.

#### **Canales que soporta Asterisk:**

- Agent: Un canal de agente DAC.
- Console: Cliente de consola de Linux, driver para placas de sonido (OSS o ALSA).

- H323: Uno de los protocolos mas antiguos de VoIP, usado en muchas implementaciones.
- IAX e IAX2: Inter-AsteriskExchange Protocol, el protocolo propio de Asterisk.
- MGCP: Media Gateway Control Protocol, otro protocolo de VOIP.
- Modem: Usado para líneas ISDN y en modems.
- NBS: Usado para broadcast de sonido.
- Phone: Canal de telefonía de Linux.
- SIP: Session Initiation Protocol, el protocolo de VoIP más común.
- Skinny: Un driver para el protocolo de los teléfonos IP de Cisco.
- VOFR: voz sobre frame-relay de Adtran.
- VPB: Líneas telefónicas para placas de Voicetronix.
- ZAP: Para conectar teléfonos y líneas con placas de Digium.
- Unicall: Usado para líneas digitales con señalización E1/R2.

### 5.6.2. Codecs y Conversores de CODEC

Obviamente es deseado poner tantas llamadas como sea posible en una red de datos. Esto puede ser hecho codificando en una forma que use menos banda ancha. Este es el papel de CODEC (COder/DECoder), algunos CODECs como el g.729 permite codificar a 8 kilo bits por segundo, una compresión de 8 para 1. Otros ejemplos son ulaw, alaw, gsm, ilbc y g729.

Asterisk soporta los siguientes CODECs:

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

### 5.6.3. Protocolos

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

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

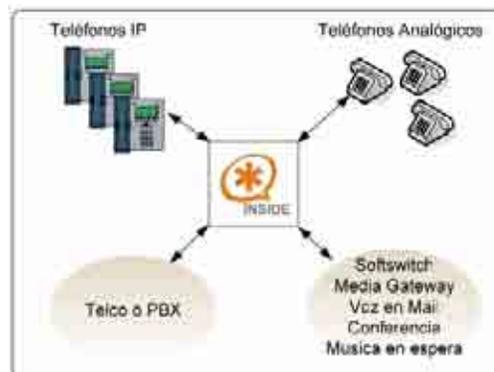
### 5.6.4. Aplicaciones

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

## 5.7. ESCENARIOS DE USO DE ASTERISK

A continuación se muestran algunos escenarios de uso de Asterisk y como ellos encajan en el modelo actual de telefonía.

### 5.7.1. Visión General



**Figura 19.** Esquema híbrido de comunicaciones IP y TDM

Dentro de una visión general, Asterisk es un PABX híbrido que integra tecnologías como TDM y telefonía IP con funcionalidad de unidades de respuesta automática y distribución automática de llamadas. En la figura de arriba se puede ver que Asterisk se puede conectar a una operadora de telecomunicaciones o un PABX usando interfaces analógicas o digitales. Los teléfonos pueden ser IP, analógicos o ADSI que es un teléfono analógico con display digital.

Descripción de los conceptos:

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

**Sistema de mensajería unificada:** Es un sistema donde todas los mensajes son diseccionados para un único lugar, por ejemplo, la casilla de correo electrónico de un usuario. En este caso los mensajes de e-mail, junto con los mensajes de correo de voz y

fax serian encaminados para buzón de voz del usuario. En Asterisk también se da la posibilidad de hacerlo.

**Servidor de música en espera:** Parece esto algo sin mucha importancia, sin embargo, en la mayoría de las centrales telefónicas es preciso colocar un aparatos reproductor de CD unido a uno

o varios redes, para que el usuario permanezca oyendo la música en espera. Si me permiten, En era digital esto ya quedo obsoleto!!. Asterisk, con MP3 es vanguardia hoy!!”.

**Discador automático:** Esto es muy útil en telemarketing, se puede programar el sistema para discado automático y distribuir en una lista. Pero esta es una tecnología que es vendida separadamente en otros PABX. En Asterisk se puede programar un discado y existen diversos ejemplos de discadores disponibles en Internet.

**Sala de Conferencia:** Permite que varios usuarios hablen en conjunto. Se selecciona un red para armar la sala de conferencia y todos los que disquen hacia ella estarán inmediatamente conectados.

Estas son algunas de las funcionalidades actuales de Asterisk, nuevas aplicaciones están surgiendo a cada día.

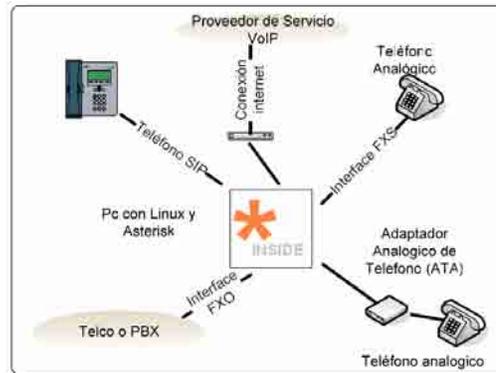
## **5.8. TELEFONÍA CON ASTERISK**

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

### **5.8.1. Sistema PABX 1x1**

En la figura 20, se ve un ejemplo de un PABX de una troncal y una línea analógica. Este es uno de los sistemas más simples que usted puede construir con Asterisk. A pesar de tener poca utilidad práctica este permite que se conceptualicen algunos puntos importantes. En primer lugar el PABX 1x1 posee una placa FXO (Foreign Exchange

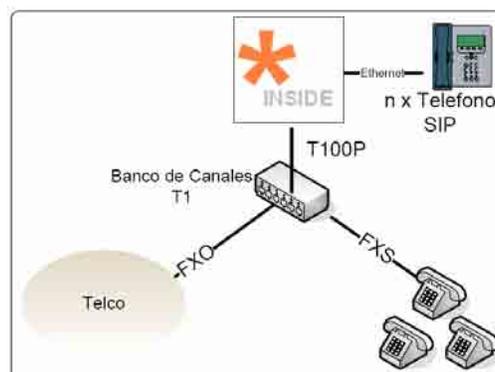
Office) para ser ligada a las operadoras o a una interfase de red. Se puede adquirir una de estas placas de Digium modelo TDM400P, para implementar el servicio. Otras de las posibilidades para una interfase FXO son un voice-modem con chipset Intel MD3200 y Motorola.



**Figura 20.** Ejemplo de un PABX de una troncal y una línea analógica.

### 5.8.2 Ampliación de PABX usando un banco de canales

Llega una hora que es difícil continuar insertando tarjetas en el PC. La mayoría de las tarjetas madres no permiten mucho más de 4 o 5 slots PCI. Si se quisiera atender ocho troncales y 16 redes, ya sería difícil de lograr. Por ejemplo, si se usa una TDM400P apenas cuatro canales por tarjeta serían posibles. En este caso se puede usar un banco de canales. Un banco de canales es un multiplexor donde entra un E1 (30 canales) o T1 (24 canales) y en el banco de canales estas señales son abiertas en diversas interfaces analógicas FXS, FXO.

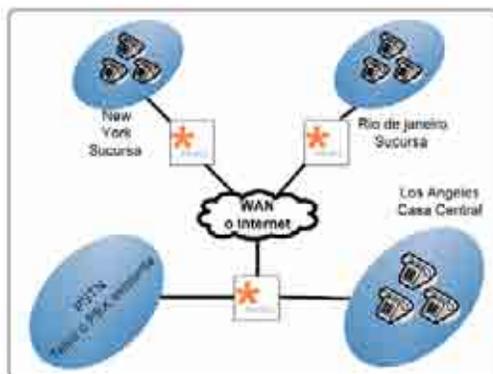


**Figura 21.** Ampliación de Canales VOIP.

Existen diversos fabricantes que producen bancos de canal GSM, lo que permite que se conecten hasta 30 líneas de celular en Asterisk. Recientemente Digium y otros fabricantes lanzaron tarjetas analógicas de alta densidad como a TDM2400 que puede atender hasta 24 puertos (FXS o FX). Otra solución interesante son los bancos de canal de Xorcon que usan un USB como interface.

### 5.8.3. Intercomunicación de sitios remotos con un sitio central.

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



**Figura 22.** Funcionalidad de un Gateway.

Media Gateway – Es un gateway permite que sus conexiones en telefonía analógica puedan ser convertidas en Voz sobre IP, por ejemplo, y transmitidas por la red de datos ante otro equipo sin pasar por la tarificación de la red pública. Este es el punto número uno de la implementación de voz sobre IP, reducir la cuenta. Si se tiene un servidor Asterisk en cada filial, usted puede interconectarlos usando IAX trunked, una de las mejores tecnologías de conexión de PABX por IP.

**Finalmente, la arquitectura de Asterisk se compone basicamente de:**

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

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

- Crear llamadas en espera y dejar a grupos de agentes que manejen las llamadas entrantes

Los comandos que se pueden utilizar en la consola de asterisk (CLI>), son:

- h: Help muestra las opciones de parametros de linea de comando.
- C **<configfile>**: Inicia Asterisk con archivo de configuración diferente del standard /etc/asterisk/asterisk.conf
- f: Foreground. Inicia Asterisk, pero no coloca un proceso en Background.
- c: Habilita el modo de consola. Inicia Asterisk en Foreground (adelante, implica la opción -f), con una consola con interface de linea de comando.
- r: Consola remota.
- n: Desabilita el color en consola.
- i: Pide los códigos criptográficos de inicialización.
- p: Corre como pseudo-realtime. Corre con prioridad de tiempo real.
- q: Modo silencioso suprime los mensajes.
- v: Incluye mensajes detalladas, (múltiplos v's = más verbose).
- d: Habilita debug extra en todos los módulos
- g: Hace que Asterisk descargue el núcleo en caso de segment violation.
- x**<cmd>**: Ejecuta el comando <cmd> (válido solo con r)

## **6. CAPITULO VI. IMPLEMENTACIÓN**

En el presente capítulo se explicara los requerimientos necesarios para la implementación y puesta en servicio de la plataforma Asterisk, así como los requerimientos mínimos y los componentes imprescindibles para su puesta en marcha.

### **6.1. Consideraciones sobre la instalación de Asterisk**

#### **6.1.1. Sistemas en producción**

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

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

#### **6.1.2. Consideraciones sobre la red**

Si se va a usar teléfonos IP, lo que es muy probable es importante que se preste atención a algunos recursos sobre la red. Los protocolos de voz sobre IP son muy buenos y resistentes a perdidas de paquetes, retardos y variaciones de tiempo. Si se abusa, la calidad de voz no será buena. Solo es posible garantizar la calidad de voz utilizando QoS extremo-a-extremo, lo que es inviable principalmente en Telefonía IP.

De esta forma se sugieren las siguientes recomendaciones:

- Implementar QoS extremo-a-extremo siempre que se pueda. Lo mismo en switches de 100Mbps donde es raro tener un congestionamiento, una condición de red inesperada puede echar a perder todo.

- Se recomienda utilizar una conexión de red exclusiva para softphones y teléfonos IP. En la mayoría de los casos, los backbones tienen bajo tráfico, pero una conexión puede ser congestionada por los propios usuarios con descargas, navegación, e-mail entre otras cosas.
- Se deben evitar hubs de 10 y 100 Mbps, las colisiones en estos equipamientos, causan Jitter.
- Cuando usa una red IP privada con equipamientos que soportan QoS extremo-a-extremo, si la calidad de voz estuviese baja, se debe verificar inmediatamente los enlaces, es probable que exista algún problema en la red. Con QoS bien implementado la calidad de voz es opera perfectamente.

### 6.1.3. Escenario y especificaciones

El diseño de una central depende directamente del tipo de servicio que se quiere ofrecer. Las dimensiones de una IPBX funcionando como *Call Center* para dar un servicio de atención al cliente serán muy diferentes que las necesitadas en una pequeña oficina con menor tráfico de voz. El objetivo principal que se quiere conseguir a la hora de montar la central es familiarizarse, analizar el potencial y entender el funcionamiento de *Asterisk*.

El PC destinado a realizar las funciones de central tiene un procesador Intel Centrino Duo a 1,6 GHz y 1 GB de RAM. Tal y como se puede observar en la tabla 4, un sistema de estas características soportará mas de 15 canales de voz simultáneos. Cabe destacar que el procedimiento correcto es realizar un análisis del tráfico esperado antes de diseñar una central, (tráfico total de datos más tráfico de los canales de voz).

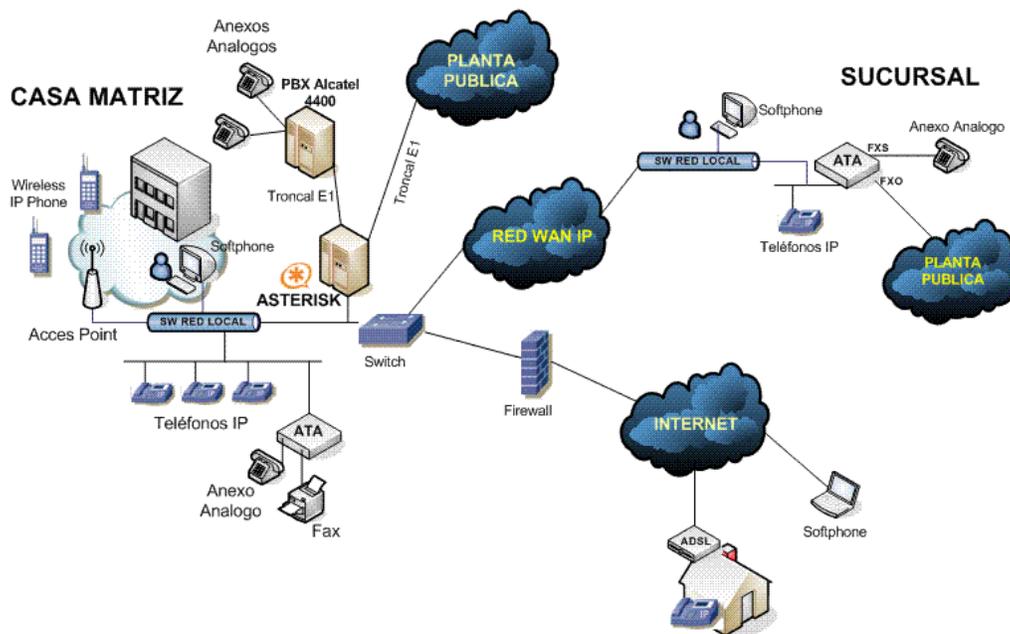
Proposito	Número de canales	Mínimo recomendado
Hobby	No mas de 5	400 MHz –x86, 256MB RAM
Pequeña oficina – hogar	5 a 10	1 GHz – x86, 512MB RAM
Pequeña empresa	Sobre 15	2 GHz – x86, 1GB RAM
Grandes empresas	Mayor a 15	CPU Doble núcleo, incluso posibilidad de múltiples servidores en arquitectura distribuida

**Tabla 4.** Guía de requisitos del sistema para Asterisk.

Se ha decidido trabajar con la plataforma *Linux*, y más concretamente sobre *Ubuntu* principalmente por ser una distribución muy estable, segura y fácil de administrar. Se ha instalado *Ubuntu 7.04* con el *kernel v.2.6.15*, última versión estable del núcleo en la fecha de instalación. Este servidor además dispone de una puerta ethernet de 10/100 Mbps para conectarse a la LAN a través de un switch para poder comunicarse con otros dispositivos VoIP e Internet.

Como extensiones se dispone de un softphone SIP "X-Lite", gratuito de la compañía *CounterPath* disponible para sistemas *Windows*, *Mac* y *Linux*.

La central a implementar no únicamente ha de establecer y mantener cada una de las posibles comunicaciones de voz en la que puedan intervenir alguna de sus extensiones, sino que ha de ser capaz de ofrecer servicios de valor agregado. Los servicios que se han implementado, entre los muchos que integran a Asterisk, son voicemail, operadora virtual y desvío de llamada, a continuación se muestra un escenario global de las soluciones que se pueden implementar con Asterisk y luego el esquema a implementar para las pruebas de este trabajo de tesis.



**Figura 23.** Escenario Global de VoIP y Telefonía IP

## **6.2. Metodología para el desarrollo de las experiencias de laboratorio**

En este capítulo se mostrarán los aspectos metodológicos considerados en la realización del trabajo:

- Hipótesis de trabajo: Define el estado inicial y el alcance del trabajo realizado en el trabajo de tesis desde el punto de vista docente.
- Estructura de las guías del laboratorio: Muestra la estructura empleada en la confección de las guías del laboratorio.
- Planificación de las experiencias: Consiste en dividir el trabajo realizado en este trabajo de tesis en segmentos, los que serán cubiertos por cada una de las experiencias del laboratorio.

### **6.2.1. Hipótesis de Trabajo**

A continuación se mostrarán los requisitos mínimos que necesitan satisfacer los alumnos que realizarán las experiencias del laboratorio de la IPBX. Además se mostrará cual es la infraestructura necesaria para la realización de los experimentos tanto en lo que se refiere al software como al hardware.

### **6.2.2. Perfil de los Alumnos**

El laboratorio de IPBX está ideado para alumnos de Ingeniería orientados al área de telecomunicaciones o para profesionales con una mediana preparación técnica y teórica en el tema de la VoIP. En particular, se dirige el trabajo a los alumnos de Ingeniería Civil Electrónica de la Universidad Austral de Chile, quienes de acuerdo a su malla curricular, poseen los conocimientos y habilidades necesarias para el desarrollo del laboratorio.

Es importante destacar que un requisito fundamental para los alumnos es que estén altamente motivados con el tema. Para un mejor entendimiento y aprovechamiento de los contenidos abordados en las actividades del laboratorio, es necesario que los alumnos posean conocimientos básicos en tecnologías de redes TCP/IP y Linux. Por lo cual es altamente recomendable para los candidatos al laboratorio que realicen algún estudio formal o informal sobre redes TCP/IP como de Linux, previo a la realización de los laboratorios.

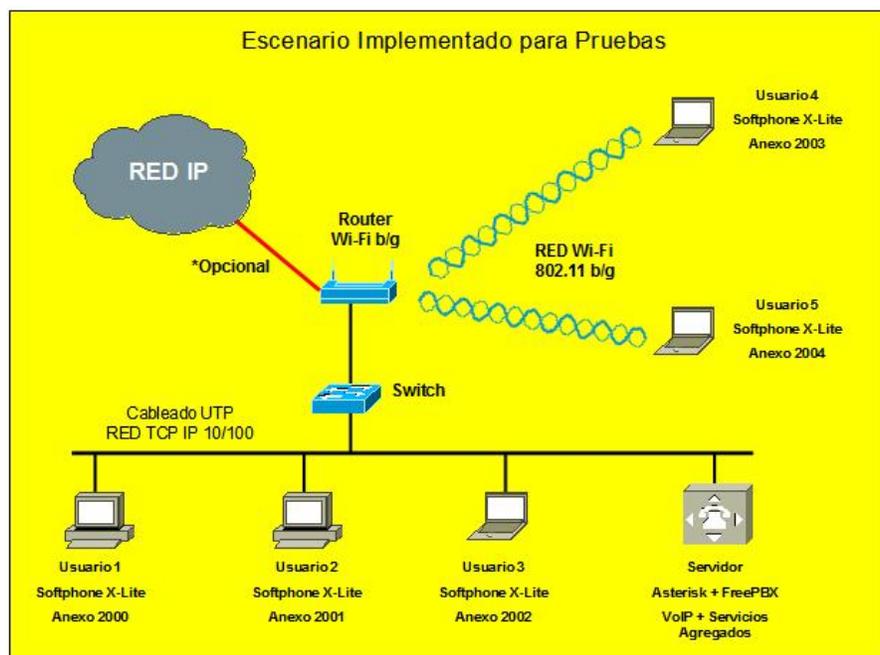
Estas experiencias de laboratorio están consideradas principalmente para alumnos que estén cursando o hayan cursado la asignatura de “Comunicaciones Modernas, ELEI 249”, ya que en esta se estudia en detalle las topologías de redes existentes y todo el los conceptos asociados al estudio e implementación de redes IP.

### 6.2.3. Infraestructura Necesaria

Para la realización del laboratorio es necesario contar con un conjunto de computadores conectados entre sí formando una pequeña red TCP/IP (capa 2, modelo OSI). El detalle de la implementación de la red propiamente tal no es objetivo del trabajo, por lo que no se ahondará el análisis en tópicos de equipamiento de redes, sino que el foco del trabajo apunta a los servicios que puede suministrar la IPBX mediante LINUX.

El equipamiento de Hardware necesario es relativamente sencillo, el cual consiste en una serie de computadores conectados en una red cableada y alternativamente inalámbrica, ambas bajo un mismo segmento de red.

El escenario recomendado de pruebas es el siguiente:



**Figura 24.** Esquema de red utilizado para trabajo de tesis.

Computador	Sistema Operativo	Memoria RAM	Velocidad Procesador	Softphone
Usuario 1	Windows XP SP2	512 Kb	Intel 1.8 Gb	X-lite 3.0
Usuario 2	Windows XP SP2	512 Kb	Intel 1.8 Gb	X-lite 3.0
Usuario 3	Windows XP SP2	512 Kb	Intel 1.66 Gb	X-lite 3.0
Usuario 4	Windows XP SP2	1 Gb	Intel 1.8 Gb	X-lite 3.0
Servidor	Ubuntu Feisty 7.04	1 Gb	Intel 1.66 Gb	X-lite 3.0

Router Utilizado	Marca, modelo	Descripción
Router	Belking wireless 54G	4 puertos mini switch

**Tabla 5.** Detalle de Equipos utilizados:

Resulta más interesante poder utilizar un router inalámbrico para poder efectuar pruebas en un escenario cableado como wi-fi, de manera que el alumno pueda establecer comparaciones en cuanto a la calidad de servicio involucrada así como el ruteo de paquetes.

Tanto para los computadores clientes como para el servidor no se exigen grandes requerimientos, pero se recomienda la siguiente configuración mínima:

- Procesador Intel Pentium de 1.2 Ghz, para equipos cliente.
- Procesador Intel Pentium de 1.6 Ghz, para equipo servidor.
- 256 MB memoria RAM para equipos cliente.
- 512 MB memoria RAM para equipo servidor
- 4 GB disco duro.
- 1 Equipo con tarjeta compatible para Wi-Fi 802.11 b/g
- 1 Tarjetas de red 10BaseT por cada equipo.
- Lector de CD-ROM 12x ATAPI IDE.
- Monitor de 14 pulgadas.
- 1 head-set por cada equipo para poder escuchar y hablar al momento de efectuar las llamadas.

#### 6.2.4. Estructura de las Guías de Laboratorio

La estructura de las guías está compuesta por los siguientes puntos:

1. Título de la experiencia: Es una frase que identifica fácilmente a la experiencia y su contenido.
2. Resumen: En el resumen se muestran cuales son los objetivos generales de la experiencia de modo de entregar al alumno una visión global y una motivación de la sesión.
3. Objetivos: Se detallan los objetivos específicos de la sesión.
4. Desarrollo de la actividad: Es la agenda de trabajo de la sesión.

#### 6.2.5. Planificación de las Experiencias

Una vez definida la infraestructura mínima, se realizó una segmentación de las etapas de su implementación, de modo de definir los temas a abordar en cada experimento. Con esta segmentación se definen 4 grandes bloques, que constituyen las experiencias 1 a la 4.

En la experiencia 1 se muestra el el procedimiento de instalación y operación de Linux, de modo de motivar al alumno y comenzar el proceso de familiarización en el tema de configuración y manipulación del software. Esta experiencia se ve complementada con la experiencias 2, el la cual se procede con la implementación y puesta en marcha del servidor de VoIP. Por último en las experiencias 3 y 4 se habilitan los clientes de VoIP y posteriormente se realiza la medición de calidad de servicio subjetiva en VoIP, de manera de poder familiarizar al alumno con los factores que influyen directamente sobre esta tecnología.

La planificación de las experiencias se ve en la siguiente tabla:

Contenidos	LAB. 1	LAB. 2	LAB. 3	LAB.4
Instalación de Linux				
Instalación de Asterisk y FreePBX				
Configuración de Anexos en FreePBX y Sopftphone				
Medición de QoS				

**Tabla 6.** Planificación de experiencias de laboratorio.

Los experimentos fueron diseñados considerando aspectos tales como dificultad creciente, y la secuencia necesaria para poner en marcha los servicios. Se debe considerar que existen servicios que suponen que otros ya estén operando.

Con la segmentación propuesta se consigue poner en marcha el IPBX mínimo, ya que logra habilitar los servicios básicos necesarios que debe prestar un IPBX.

Respecto al tiempo que dura cada sesión experimental se espera que no supere las 3 horas.

## 7. CAPITULO VII. CONCLUSIONES

- Se estructuraron y segmentaron 4 experiencias de laboratorio, las cuales apuntan a la implementación de un servidor de voz sobre IP, mas conocido como IPBX basado en el protocolo SIP. Todas las extensiones conectadas a este servidor, pueden comunicarse entre sí de forma totalmente gratuita mediante la utilización de softphones, siempre y cuando esten dentro de una red privada. Es posible conectarse hacia sitios remotos mediante un estudio previo de las condiciones de red que se tengan, asi como el equipo se se requiera habilitar como servidor.
- El servicio de VoIP para el cual se crearon las experiencias de laboratorio, fue implementado y probado en las oficinas de Telefonica Chile, San Martín N° 50, Santiago – Centro. Se crearon 4 anexos y se realizaron llamadas en ambiente cableado e inalambrico dentro de una red LAN, además se probo el servicio de buzón de voz, se midio la calidad de servicio de forma subjetiva, en este caso el valor promedio alcanzado resulto ser de 4, es decir, bueno. Finalmente todas las experiencias fueron simuladas, obteniendo resultados satisfactorios en todos los casos, lo que permitio la validación de las experiencias para ser implementadas en cualquier dependencia que cumpla con las condiciones y recomendaciones descritas en esta tesis.
- Con la segmentación de las actividades utilizada se consiguió plantear una metodología sistemática para la implementación y puesta en marcha del servicio VoIP mínimo, a un bajo costo, además de permitir a los alumnos conocer y comprender en forma practica, temas como codificación y señalización de voz sobre redes IP, ademas de poder determinar los parámetros de calidad de servicio, entre otros.
- La segmentación de las actividades se materializó en las guías experimentales, en donde a través de una serie de pasos el alumno será capaz de poner en marcha los servicios que presta la VoIP.

- Es importante destacar el valor docente de un laboratorio, en donde se permite un alto grado de entendimiento de las actividades e interacción de los alumnos.
- Las experiencias están diseñadas para alumnos con mediana preparación en el tema de redes, pero altamente motivados. Esto es importante destacarlo, puesto que será usual en el desarrollo de las sesiones la corrección de errores o malas maniobras por parte de los alumnos.
- Este laboratorio ofrece a sus alumnos, sean ellos profesionales o estudiantes, la posibilidad de desarrollarse y mejorar sus conocimientos relacionados a las comunicaciones de VoIP e inclusive adentrarse a la Telefonía IP. Esto ofrece una excelente posibilidad de distinguirse positivamente del resto de sus colegas.
- El diseño de la IPBX se ha basado en la utilización de herramientas GNU con la consecuente reducción de costos. Gracias a las licencias GPL de Linux y Asterisk es posible entregar una solución con las mismas características que las actuales PABX de las grandes marcas comerciales a un precio más económico.
- El software de aplicación utilizado para implementar la IPBX fue Asterisk, esto ya que es una solución software para centrales IP robusta, flexible y ante todo económica y potente. Serían necesarios varios proyectos dedicados exclusivamente al estudio de Asterisk para alcanzar todo este potencial, por lo que una de las futuras líneas de investigación consistiría en una mayor profundización en este software. Algunas de estas posibles vías podrían basarse en una mayor inversión para conseguir un incremento del tamaño del escenario de manera que se pudiera simular las características de un Call Center. También es posible implementar servicios de Telefonía IP, mediante la integración de Asterisk con troncales analógicas, en este caso se puede considerar un proyecto de tesis que permita determinar los requerimientos necesarios para su correcta implementación y utilización.

- Para hacer posible la comunicación de voz sobre IP a nivel de LAN, se requiere de un PC destinado a ser usado como servidor de Asterisk debidamente configurado, disponer de un cableado estructurado correctamente instalado, se recomienda la utilización de cable UTP categoría 6. Los usuarios que requieran utilizar la aplicación de VoIP, deberán contar con un PC, la instalación y configuración de un softphone (se recomienda el X-Lite de Counterpath), los anexos deben estar correctamente configurados tanto en el servidor como en los softphone y disponer de head set con micrófono o al menos parlantes y un micrófono.
- La conectividad de la aplicación de Voz sobre IP se implementa a través del softphone, obteniendo buena calidad de voz y conectividad, además del fácil manejo y configuración del mismo.
- Se debe tener en cuenta el realizar un estudio previo de la red y la capacidad de los computadores a utilizar, en especial el que se destinara a ser utilizado como servidor, ya que Asterisk en este caso, no opera con DSP (Procesador Digital de Señales), sino que utiliza directamente los recursos de la CPU que posea el computador para poder entregar el servicio.

## **PROYECCIONES FUTURAS SUGERIDAS**

- Hacer extensible el proyecto a través de Internet ó una red VPN IP con sitios remotos.
- Permitir la comunicación con telefonía tradicional y móvil, es decir, implementar Telefonía IP, usando las tarjetas respectivas.
- Diseñar e implementar experiencias de laboratorio asociados a los servicios de Telefonía IP, que considere servicios de valor agregado, tales como buzón de voz, tarificador, fax, etc.
- Realizar una solución de VoIP ó Telefonía IP, basada en Asterisk para la Universidad Austral.

## BIBLIOGRAFÍA

*Asterisk The Open Source PBX.*

Sitio web: <http://www.asterisk.org/>

*Asterisk Architecture. Digium – The Asterisk Telephony Company:*

Sitio web: <http://www.digium.com/en/asteriskbusinesses/applications/>

Artículo voz IP, Ingeniería en Redes y Comunicaciones.

Sitio web: <http://comunicaciones.unitronics.es/tecnologia.htm>

*Aspectos básicos para la utilización de SIP como protocolo de señalización en la red de acceso de sistemas UMTS.*

Sitio web: <http://www.dit.upm.es/info/memodit02.pdf>

Andrew S. Tanenbaum, "Redes de Computadoras", Tercera Edición, Prentice-Hall, Hispanoamericana S.A., México, 1997.

Ballén Jiménez, J y Muñoz Serna, D. (2006). *Diseño e implementación de un sistema de comunicación y de mensajería de voz sobre IP a través de las redes LAN conectadas conectadas por protocolo TCP/IP*. Trabajo de grado para optar al título de Ingeniero de Sistemas. Universidad El Bosque. Bogotá, Colombia.

Bartomeu Islas, H. (2005). *Evaluación de los parámetros de QoS en entornos de movilidad IP*. Trabajo de fin de carrera para optar al título de Ingeniero de Sistemas Computacionales. Universidad Politécnica de Cataluña. Barcelona, España.

Sitio web: <http://sam.ccaba.upc.es/pdf/EvaluacionMIP.pdf>

Certain Yance, A. (2006). Gecko Networks. *TrixBos al descubierto*.

Sitio web: [www.alfredcertain.com/?dl=pdf/GN\\_eBook\\_TRIXBOXALDESCUBIERTO.pdf](http://www.alfredcertain.com/?dl=pdf/GN_eBook_TRIXBOXALDESCUBIERTO.pdf)

Ebook, "Trixbos al Descubierto", Alfredo Certain Yance, 2006 GECKO EU, GECKO NETWORKS.

Estudio y Desarrollo de Centrales Telefonicas PBX Basado en Tecnologia VoIP, Orlando Pinto Soto, Universidad Técnica Federico Santa María Departamento de Electrónica, diciembre de 2005.

Gotxi García, I. (2005). *Diseño e implementación de un agente de usuario SIP*. Tesis doctoral, Instituto superior de ingenieros de Bilbao. Bilbao, España.

Sitio web: <http://sua.sourceforge.net/Presupuesto.doc>

*IPCorp Server.*

Sitio web: [http://www.ipcorp.cl/servicios\\_callcenters.htm](http://www.ipcorp.cl/servicios_callcenters.htm)

ITU, Unión Internacional de telecomunicaciones.

Sitio web: <http://www.itu.int/home/index-es.html>

ITU-T G.711 (1988). *Recommendation G.711, Pulse code modulation (PCM) of voice frequencies*, Nov. 1988. Sitio web: <http://www.itu.int/rec/T-REC-G.711/e>

ITU-T G.723.1 (1996-2). *Recommendation G.723.1 Dual Rate Speech Coder for Multimedia Communications Transmitting at 5.3 and 6.3 kbps.* . Marzo 1996.

Sitio web: <http://www.itu.int/md/T05-SG16-060403-TD-PLN-0258/es>

ITU-T G.726 (1990). *Recommendation G.726, "40-,32-,24-, and 16-Kb/s adaptive differential pulse code modulation."*. Dec. 1990.

Sitio web: <http://www.itu.int/md/T01-SG13-030721-TD-WP3-0006/es>

ITU-T G.726 (1994). *Recommendation G.726 Appendix III: Comparison of ADPCM Algorithms*. Geneva 1994.

Sitio web: <http://www.itu.int/md/T01-SG13-030721-TD-WP3-0006/es>

ITU-T G.728 (1992). *Recommendation G.728 Coding of Speech at 16 kbps Using Low Delay Code Excited Linear Prediction*. Septiembre 1996.

Sitio web: <http://www.itu.int/itudoc/itu-t/com16/implgd/g728ig-es.html>

ITU-T G.729 (1998). *Recommendation G.729 Annex D: 6.4 kbit/s CS-ACELP speech coding algorithm*. Sep 1998. Sitio web: <http://www.itu.int/rec/T-REC-G.729/e>

ITU-T P.800 (1996-3). *Recommendation P.800- Methods for subjective determination of transmission quality*. Agosto 1996. Sitio web: <http://www.itu.int/rec/T-REC-p>

“Informe esencial sobre telefonía IP”, Unión Internacional de Telecomunicaciones, GRUPO DE EXPERTOS SOBRE TELEFONÍA IP DEL UIT- D. Unidad de Ciberestrategias de la UIT, año 2003.

Joskowickz, J. (2005). *Redes Unificadas*.

Sitio web de la Universidad de la República de Uruguay:

<http://ie.fing.edu.uy/ense/asign/redcorp/material/2004/Redes%20Unificadas.pdf>

Kaschel, H y San Juan, E. (2006). *Consideraciones técnicas para elaborar un estándar definitivo VoIP*. Sitio web de la Universidad de Santiago de Chile:

<http://www.senacitel.cl/downloads/senacitel2002/ID002.pdf>

Management Solutions, Madrid, Departamento Marketing y Comunicación. (2006, noviembre). *VoIP: convergencia y comoditizacion*.

Sitio web: <http://www.msspain.com/esp/files/newsletter-06.pdf>

Montalbán Alvarado, R. (2005). *Telefonía IP con señalización por canal común*. Trabajo de titulación para optar al título de Ingeniero Electrónico. Universidad Austral de Chile. Valdivia, Chile.

Pinto Soto, O. (2005). *Estudio y desarrollo de centrales telefónicas PBX basado en tecnología VoIP*. Memoria presentada como requisito parcial para optar al título de Ingeniero Civil Electrónico con mención en computadores y sistemas digitales.

Quintana Cruz, D. (2007). *Diseño e implementación de una red de telefonía IP sobre con software libre en la RAAP*. Tesis para optar al título de Ingeniero de las Telecomunicaciones. Pontificia Universidad Católica del Perú. Lima, Perú.

Sitio web: <http://gst.telecom.pucp.edu.pe/~dqintana/tesis.pdf>

Ramírez Andrade, T. (2006). *Desarrollo de aplicaciones en la central Asterisk de Telefonía IP en el Departamento de Electrónica de la UTFSM*. Memoria presentada como requisito parcial para optar al título de Ingeniería Civil Electrónica con mención en control automático. Universidad Técnica Federico Santa María, Valparaíso, Chile.

Sitio web: [http://alumnos.elo.utfsm.cl/~tramirez/Documentos/memoria\\_word.doc](http://alumnos.elo.utfsm.cl/~tramirez/Documentos/memoria_word.doc)

Rodera Rodera, S. (2005). *Diseño e implementación de un punto neutro para VoIP*. Trabajo de fin de carrera presentado para optar al título de Ingeniero en Telecomunicaciones. Universidad Politécnica de Cataluña. Barcelona, España.

Ramal Morcillo, F. (2005). *Diseño de un sistema de telefonía IP y plan de empresas para su comercialización*. Proyecto de fin de carrera presentado como requisito para optar al título de Ingeniero en Telecomunicaciones. Universidad Politécnica de Cataluña. Barcelona, España.

Ramírez, Tamara. *Desarrollo de Aplicaciones en la Central Asterisk de Telefonía IP en el Departamento de Electrónica de la UTFSM*. Proyecto de título.

Sitio web: <http://alumnos.elo.utfsm.cl/~tramirez/>

Rodríguez Salas, C. (2007, 07 de marzo). COR Technologies S.R.L. *NEX-Revista de Networking y programación*. Número 15.

Sitio web: [http://www.nexweb.com.ar/free/pdf\\_13\\_15\\_24\\_27/NEX%20IT%2015.pdf](http://www.nexweb.com.ar/free/pdf_13_15_24_27/NEX%20IT%2015.pdf)

Ruíz Muzquiz, P. (2005). *Redes y sistemas de comunicación versión 0.6.0*.

Sitio web: [http://alqua.org/alquadocs/RSC-0\\_6\\_0.pdf](http://alqua.org/alquadocs/RSC-0_6_0.pdf)

Roper Smith, J. (2005). *Introducción a Asterisk Home*.

Sitio web del Instituto Internacional para la comunicación y el desarrollo: [http://www.imaginar.org/iicd/tus\\_archivos/TUS9/3\\_manual\\_asterisk.pdf](http://www.imaginar.org/iicd/tus_archivos/TUS9/3_manual_asterisk.pdf)

Sáenz Rivera, T. (2005). *Videoconferencia como apoyo a la educación a distancia y el trabajo colaborativo*. Tesis para obtener el grado de Maestro en ciencias área Telemática. Universidad de Colima. Colima, Perú.

Sitio web: [http://digeset.ucol.mx/tesis\\_posgrado/Pdf/Tania%20Saenz%20Rivera.pdf](http://digeset.ucol.mx/tesis_posgrado/Pdf/Tania%20Saenz%20Rivera.pdf)

Session Initiation Protocol (sip) in IETF.

Sitio web: <http://www.ietf.org/html.charters/sip-charter.html>

Telefonía IP: sus características y protocolos utilizados. Pablo Naveas Farías, Universidad Técnica Federico Santa María.

Universidad Técnica Federico Santa María. Valparaíso, Chile.

Sitio web:

<http://docencia.mat.utfsm.cl/~opinto/download/presentaciones/presentaciones.html>

Unión Internacional de Telecomunicaciones, Ginebra, Oficina de desarrollo de las telecomunicaciones. (2005, octubre). *Informe esencial sobre telefonía IP*.

Sitio web:

[http://www.itu.int/osg/spu/wtpf/wtpf2001/sgreport/revisedversion9march/SecGen9march\\_s.pdf](http://www.itu.int/osg/spu/wtpf/wtpf2001/sgreport/revisedversion9march/SecGen9march_s.pdf)

Valero Henríquez, J y Jiménez Opazo, X. (2006). Capa tres soluciones tecnológicas SL. *Cómo instalar Asterisk 1.2 estable desde fuente de código*.

Sitio web: <http://www.realidadfutura.com/docu/asterisk/instalacion-asterisk-1.2.pdf>

Viegas Sandoval, E y Correa Stuardo F. (2006, 17 de octubre). *Asterisk desconsolidado*.

Sitio web: [http://asterio.com.ar/resources/downloads/Asterisk\\_desconsolidado.pdf](http://asterio.com.ar/resources/downloads/Asterisk_desconsolidado.pdf)

Van Meggelen J., Smith J., Madsen L.; "Asterisk. *The Future of Telephony*". Ed. O'Reilly (2005).

## REFERENCIAS ELECTRÓNICAS

[www.voip-news.org](http://www.voip-news.org)

*The VoIP Wiki.* <http://www.voip-info.org/wiki/>

*Speech Encoding. Wikipedia The Free Encyclopedia:*

[http://en.wikipedia.org/wiki/Speech\\_encoding](http://en.wikipedia.org/wiki/Speech_encoding)

*Softphone SIP X-lite. CounterPath:* <http://www.xten.com/>

[http://asterisktutorials.com/videos/Getting\\_Started\\_with\\_TrixBot/Getting%20Started%20with%20TrixBot.html](http://asterisktutorials.com/videos/Getting_Started_with_TrixBot/Getting%20Started%20with%20TrixBot.html)

<http://www.asterisk-peru.org/>

<http://asterisktutorials.com/>

[http://asterisktutorials.com/videos/Getting\\_Started\\_with\\_TrixBot/Getting%20Started%20with%20TrixBot.html](http://asterisktutorials.com/videos/Getting_Started_with_TrixBot/Getting%20Started%20with%20TrixBot.html)

[http://asterisktutorials.com/videos/Advanced\\_FreePBX/Advanced\\_FreePBX.html](http://asterisktutorials.com/videos/Advanced_FreePBX/Advanced_FreePBX.html)

<http://www.trixbox.org/>

[http://about.skype.com/2007/06/make\\_international\\_calls\\_from.html](http://about.skype.com/2007/06/make_international_calls_from.html)

<http://descargarlinux.com.ar/mambo/moodle/course/view.php?id=5>

<http://www.linux-cd.com.ar/manuales/rh9.0/rhl-gsg-es-9/s1-managing-working-with-files.html>

[http://structio.sourceforge.net/guias/AA\\_Linux\\_colegio/AA\\_Linux\\_colegio.html](http://structio.sourceforge.net/guias/AA_Linux_colegio/AA_Linux_colegio.html)

## ANEXO I GUIAS DE LABORATORIO

### Experiencia N°1



Universidad Austral de Chile  
Instituto de Electricidad y Electrónica

#### **Nombre Experiencia:** Instalación de Linux Ubuntu 7.04

#### **Resumen**

*La presente práctica de laboratorio tiene como fin, familiarizar al alumno con el proceso de instalación de Linux en un entorno gráfico, esto a su vez le permitirá para conocer los tipos de particiones existentes en Linux, así como de modo general, la estructura de archivos de este sistema operativo.*

#### **Objetivo General**

- **Instalación de Linux Ubuntu 7.04 Feisty**

#### **Objetivos Específicos:**

- Permitir al alumno familiarizarse con el proceso de instalación del sistema operativo Linux.
- Apreciar las diferencias existentes con los sistemas operativos de Microsoft.
- Permitir al alumno conocer la estructura de archivos de Linux.

#### **MARCO TEORICO**

#### **LINUX**

Linux es una de las tantas variantes de Unix. Se trata de un sistema operativo de 32 bits de libre distribución, desarrollado originalmente por Linus Torvalds, un estudiante de la universidad finlandesa de Helsinki, quien, en 1991, se abocó a la tarea de reemplazar a Minix, un clon de Unix de pequeñas proporciones y finalidad académica desarrollado años antes por Andrew Tannenbaun.

A medida que avanzaba en su desarrollo, Linux fue dejando el código fuente de las sucesivas versiones del kernel y utilidades de Linux a disponibilidad de los usuarios de Internet. Este fue sin duda un gran acierto, ya que hizo posible que una multitud de desarrolladores de todo el mundo se familiarizaran con el código, lo cual en primera instancia significó un gran aporte de sugerencias, evolucionado luego hacia un espectacular ejemplo de desarrollo distribuido de software: centenares de desarrolladores independientes, desde diferentes puntos del planeta tomaron a su cargo la producción de software para Linux, ya sea escribiéndolo desde cero o portándolo desde otras plataformas Unix. Esta modalidad de desarrollo continúa aún hoy y ha permitido a Linux alcanzar un alto nivel de desarrollo y madurez, así también como un amplio grado de aceptación.

Actualmente, Linux posee todas las características que pueden encontrarse en cualquier sistema Unix moderno, incluyendo direccionamiento lineal de 32 bits, memoria virtual, multitarea real, shared libraries, módulos de kernel cargables on-demand, soporte TCP/IP (incluyendo SLIP, PPP, NFS, etc.), y entorno gráfico X-Windows.

Linux es distribuido bajo la Licencia General Pública de GNU, lo cual significa que puede ser distribuido, copiado y modificado gratuitamente, a condición de no imponer ninguna restricción en sucesivas distribuciones. En pocas palabras: Linux es un sistema operativo gratuito.

### **Componentes de linux**

Linux se puede dividir generalmente en cuatro componentes principales: el núcleo (kernel), el shell (consola), el sistema de archivos y las utilidades.

**El núcleo:** es el programa medular que ejecuta programas y gestiona dispositivos de hardware tales como los discos y las impresoras.

**El shell:** proporciona una interfaz para el usuario. Recibe órdenes del usuario y las envía al núcleo para ser ejecutadas.

**El sistema de archivos:** organiza la forma en que se almacenan los archivos en dispositivos de almacenamiento tales como los discos. Los archivos están organizados en

directorios. Cada directorio puede contener un número cualquiera de subdirectorios, cada uno de los cuales puede a su vez, contener otros archivos.

El núcleo, el shell y el sistema de archivos forman en conjunto la estructura básica del sistema operativo. Con estos tres elementos puede ejecutar programas, gestionar archivos e interactuar con el sistema. Además, Linux cuenta con unos programas de software llamados utilidades que han pasado a ser considerados como características estándar del sistema.

**Las utilidades:** son programas especializados, tales como editores, compiladores y programas de comunicaciones, que realizan operaciones de computación estándar. Incluso uno mismo puede crear sus propias utilidades.

### ***Archivos y Directorios***

Un archivo es un conjunto de información al que se le ha asignado un nombre (llamado nombre del archivo). Ejemplo de archivo son un mensaje de correo, o un programa que puede ser ejecutado, esencialmente, cualquier cosa guardada como un archivo individual.

Los archivos son identificados por sus nombres. Estos nombres usualmente identifican el archivo y su contenido de alguna forma significativa para el usuario.

No hay un formato estándar para los nombres de los archivos como lo hay en MS-DOS y en otros sistemas operativos; en general, estos nombres pueden contener cualquier carácter (excepto /), y están limitados a 256 caracteres de longitud.

Un directorio es simplemente una colección de archivos. Puede ser considerado como una carpeta que contiene muchos archivos diferentes.

Los directorios también tienen nombre con el que se pueden identificar. Además, los directorios mantienen una estructura de árbol; es decir, los directorios pueden contener otros subdirectorios.

## **El Árbol de directorios**

Los sistemas Linux tienen una distribución de archivos estándar, de forma que recursos y archivos puedan ser fácilmente localizados. Esta distribución forma el árbol de directorios, el cual comienza en el directorio /, también conocido como directorio raíz.

Directamente por debajo de / hay algunos subdirectorios importantes: /bin, /etc, /dev y /usr, entre otros. Éstos a su vez contienen otros directorios con archivos de configuración del sistema, programas, etc.

En particular, cada usuario tiene un directorio home. Éste es el directorio en el que el usuario guardará sus archivos. Usualmente, los directorios home de los usuarios se encuentran en /home y son nombrados con el nombre del usuario al que pertenecen. Por tanto, el directorio home de Usuario es /home/Usuario.

### ***Explorando el sistema de archivos***

El sistema de archivos es la colección de archivos y la jerarquía de directorios de su sistema. Entre los directorios principales destacan:

**/bin:** es la abreviación de binaries, o ejecutables. Es donde residen la mayoría de los programas esenciales del sistema. La mayoría de los archivos de /bin tienen un asterisco ( \* ) añadido al final de sus nombres. Esto indica que son archivos ejecutables.

**/dev:** Los archivos en /dev son conocidos como controladores de dispositivo (device drivers) y se utilizan para acceder a los dispositivos del sistema y recursos, como discos duros, modems, memoria, etc.

**/etc:** contiene una serie de archivos de configuración del sistema. Éstos incluyen /etc/passwd (la base de datos de usuarios), /etc/rc (guiones de inicialización del sistema), etc.

**/sbin:** se usa para almacenar programas esenciales del sistema, que usará el administrador del mismo.

**/home:** contiene los directorios home de los usuarios. Por ejemplo, /home/Usuario es el directorio del usuario. En un sistema recién instalado, no habrá ningún usuario en este directorio.

**/lib:** contiene las imágenes de las librerías compartidas. Estos archivos contienen código que compartirán muchos programas. En lugar de que cada programa contenga una copia propia de las rutinas compartidas, éstas son guardadas en un lugar común, en /lib. Esto hace que los programas ejecutables sean menores y reduce el espacio usado en disco.

**/proc:** es un sistema de archivos virtual. Los archivos que contiene realmente residen en memoria, no en disco. Hacen referencia a varios procesos que corren en el sistema, y le permiten obtener información acerca de qué programas y procesos están ejecutándose en un momento dado.

**/tmp:** Muchos programas tienen la necesidad de generar cierta información temporal y guardarla en un archivo temporal. El lugar habitual para esos archivos es /tmp.

**/usr:** es un directorio muy importante. Contiene una serie de subdirectorios que contienen a su vez algunos de los más importantes y útiles programas y archivos de configuración usados en el sistema.

Los directorios descritos anteriormente son esenciales para que el sistema esté operativo, pero la mayoría de las cosas que se encuentran en /usr son opcionales para el sistema. De cualquier forma, son estas cosas opcionales las que hacen que el sistema sea útil e interesante.

**/var:** contiene directorios que a menudo cambian su tamaño o tienden a crecer. Muchos de estos directorios solían residir en /usr, pero desde que estamos trabajando de dejarlo relativamente inalterable, los directorios que cambian a menudo han sido llevados a /var.

## **UBUNTU**

Ubuntu es una distribución Linux que ofrece un sistema operativo predominantemente enfocado a equipos personales, aunque también proporciona soporte para servidores.

Ubuntu esta basado en Debian GNU/Linux, Ubuntu concentra su objetivo en la facilidad de uso, la libertad en la restricción de uso, los lanzamientos regulares (cada 6 meses aproximadamente) y la facilidad en la instalación. Ubuntu es patrocinado por Canonical Ltd., una empresa privada fundada y financiada por el empresario sudafricano Mark Shuttleworth.

La versión más reciente a la fecha, es la versión 7.04 (*Feisty Fawn*) fue lanzada el 19 de abril de 2007.

### **Características**

Basada en la distribución Debian.

Disponible en 4 arquitecturas: Intel x86, AMD64, SPARC (para esta última sólo existe la versión servidor).

Los desarrolladores de Ubuntu se basan en gran medida en el trabajo de las comunidades de Debian y GNOME.

Las versiones estables se liberan cada 6 meses y se mantienen actualizadas en materia de seguridad hasta 18 meses después de su lanzamiento.

El entorno de escritorio oficial es Gnome y se sincronizan con sus liberaciones.

El navegador web oficial es Mozilla Firefox.

El sistema incluye funciones avanzadas de seguridad y entre sus políticas se encuentra el no activar, de forma predeterminada, procesos latentes al momento de instalarse. Por eso mismo, no hay un firewall predeterminado, ya que no existen servicios que puedan atentar a la seguridad del sistema.

Para labores/tareas administrativas en terminal incluye una herramienta llamada sudo, con la que se evita el uso del usuario root (administrador).

No sólo se relaciona con Debian por el uso del mismo formato de paquetes deb, también tiene uniones muy fuertes con esa comunidad, contribuyendo con cualquier

cambio directa e inmediatamente, y no solo anunciándolos. Esto sucede en los tiempos de lanzamiento. Muchos de los desarrolladores de Ubuntu son también responsables de los paquetes *importantes* dentro de la distribución de Debian.

La versión reciente es la 7.04, cuyo nombre en clave es "Feisty Fawn". Este lanzamiento incluyó mejoras en el uso de la memoria de softwares como Evolution y Nautilus. Además incluyó GNOME 2.18.1, Mozilla Firefox 2.0.0.3, OpenOffice.org 2.2, Xorg 7.2, GCC 4.1.1, y la versión 2.6.20 del núcleo Linux.

### **Requisitos del sistema**

A continuación se detallan los requisitos de hardware mínimos y recomendados para la instalación del sistema operativo:

#### **Requisitos Mínimos**

Procesador Intel™ o compatible a 200 Mhz

256 Mb de RAM

Tarjeta SVGA

3Gb de espacio libre en el disco duro

#### **Requisitos Recomendados**

Procesador Intel™ o compatible a 1 Ghz

512 Mb de RAM

Aceleradora gráfica 3D compatible con OpenGL

5 Gb de espacio libre en el disco duro

### **Desarrollo de la actividad**

En la presente actividad, se requiere realizar lo siguiente:

1. Formateo y partición de disco duro para la instalación de Linux, versión Ubuntu Feisty 7.04. Este procedimiento se debe realizar en el equipo que tenga más recursos de procesador y memoria RAM, ya que se usara como servidor del servicio VoIP.
2. Instalar Ubuntu Feisty 7.04 en modo grafico.

3. Preparar al menos 2 computadores adicionales para instalar los softphones que permitan utilizar el servicio. Se puede utilizar sistema operativo Windows ó Linux.
4. Verificar que las condiciones de cableado sean las necesarias para permitir la comunicación entre los equipos a nivel LAN.
5. Comprobar si existe la posibilidad de utilizar un router con tecnología Wi-Fi IEEE 802.11 a/b ó g, que permita comunicaciones inalámbricas a nivel LAN, así como comprobar que los computadores a utilizar, posean interfaces Wi-Fi, compatibles con la norma IEEE 802.11 a/b ó g (Se recomienda 802.11g).
6. Comprobar que los equipos posean head-set ó al menos un micrófono y audifonos.
7. Una vez instalado Ubuntu 7.04, comprobar la interoperabilidad entre los equipos, para esto, se deben realizar pruebas de ping, además de monitorear el tráfico de red mediante el software Ethereal.
8. Explorar la estructura de archivos de Linux, ejecutar comandos en la consola de Linux para crear archivos, modificar los permisos de estos y moverlos de carpeta.

#### **Información a considerar en el Informe:**

1. Detallar completa y claramente las características de los equipos utilizados:
  - Velocidad de la CPU
  - Memoria RAM
  - Disco Duro
2. Se debe detallar los equipos de comunicaciones utilizados y sus características, router, switch, etc.
3. Detallar el tipo de cable UTP y categoría utilizado en la Red LAN, en el caso de utilizar en forma conjunta conexiones inalámbricas, especificar la norma utilizada.
4. Detallar paso a paso el proceso de formateo, particionado e instalación de Ubuntu Linux.

**Nota:** En caso de tener problemas con el desarrollo de la actividad, citar la guía de ayuda para la implementación de experiencias de laboratorio suministrada.



## **Experiencia N° 2**

**Nombre Experiencia:** Instalación de Asterisk y FreePBX

### **Resumen**

*La presente práctica de laboratorio tiene como fin, permitir al alumno comprobar en forma práctica la correcta instalación de Asterisk y FreePBX, además de familiarizarlo con las instrucciones que permiten la habilitación y configuración de estas herramientas para una posterior puesta en marcha de una IPBX de baja escala.*

### **Objetivo General**

- Instalación de Asterisk y FreePBX

### **Objetivos Específicos:**

- Comprender como realizar una correcta instalación de Asterisk, FreePBX y sus componentes.
- Conocer los comandos básicos que permiten la operabilidad de la IPBX mediante Asterisk y FreePBX.
- Permitir al alumno, dimensionar las capacidades de la aplicación al ser utilizada como IPBX

### **Desarrollo de la Actividad**

1. Instalar Asterisk IPBX 1.2
2. Instalar FreePBX versión 2.2.2
3. Comprobar el correcto funcionamiento de FreePBX
4. Instalar y activar aplicaciones de valor agregado dentro de FreePBX, tales como:
  - Buzon de Voz
  - IVR
  - Musica en Espera, entre otros.

**Preguntas:**

1. Indicar en que directorios se guardan los archivos de configuración de Asterisk y cuales son.
2. Detalle cuales son los requisitos mínimos para poder habilitar Asterisk.
3. Indicar cual es el software necesario para implementar VoIP con Asterisk en una red LAN.
4. Nombre los 4 componentes básicos de la arquitectura de Asterisk y sus características.
5. Indique porque se utiliza FreePBX en conjunto con Asterisk e indique 3 características de esta aplicación.

**Nota:** El informe debe entregar un detalle de todo el procedimiento de instalación y habilitación de la IPBX, así como la activación de los servicios de valor agregado.

En caso de tener problemas con el desarrollo de la actividad, citar la guía de ayuda para la implementación de experiencias de laboratorio suministrada.



### **Experiencia N° 3**

#### **Nombre Experiencia:**

Configuración de Clientes SIP y realización de llamadas en una LAN cableada y Wi-Fi

#### **Resumen**

En la experiencia a desarrollar, el alumno aprenderá a configurar y habilitar anexos IP, establecer llamadas entre usuarios y además habilitar servicios de valor agregado, tal como buzón de voz. Además experimentará efectuando llamadas a través de una red LAN cableada e inalámbrica de manera de poder tener una experiencia más completa del servicio.

#### **Objetivo General**

- Permitir al alumno, conocer, comprender y utilizar la aplicación de VoIP mediante Asterisk IPBX.

#### **Objetivos Específicos:**

- Permitir al alumno conocer y comprender la configuración y utilización del softphone X-Lite de Counterpath en los equipos cliente.
- Permitir al alumno conocer y comprender la correcta configuración y habilitación de los anexos IP.
- El alumno debe ser capaz de habilitar al menos 1 servicio de valor agregado que ofrece Asterisk, mediante la utilización de FreePBX

#### **Desarrollo de la Actividad**

1. Configurar los anexos para al menos 2 usuarios mediante la aplicación de FreePBX, instalada en el servidor de VoIP.
2. Instalar y configurar al menos 2 softphones X-Lite, en los equipos de los usuarios que utilizarán el servicio de VoIP.
3. Realizar llamadas entre los usuarios y de ser posible repetir el procedimiento mediante una red LAN inalámbrica.

4. Habilitar el servicio de buzón de voz para los anexos y comprobar que este operativo cuando el anexo al que se le esta llamando, no contesta ó no esta disponible.
5. monitorear las llamadas realizadas y los mensajes de voz guardados por medio de la aplicación FreePBX, instalada en el equipo servidor (IPBX).

Preguntas:

1. Indique cuales son los parámetros que se deben configurar en Asterisk usando FreePBX, para poder crear una extensión que opere con protocolo SIP.
2. Configure 4 anexos en FreePBX que tengan las siguientes características:
  - Protocolo de señalización: SIP.
  - 2 usuarios con buzón de voz habilitado.
  - 2 usuarios sin buzón de voz habilitado.
  - Largo de la extensión: 4 dígitos.
  - 2 usuarios opereraran en la red LAN cableada
  - 2 usuarios operaran en la red LAN Wi-Fi (del mismo segmento de red).
  - a) Comprobar el funcionamiento del servicio de buzón de voz.
  - b) Comprobar el correcto funcionamiento de llamadas entre usuarios de la red cableada y la red Wi-Fi pertenecientes al mismo segmento de red.
  - c) Ingresar en FreePBX, al menú de registro de llamadas y de mensajes de voz, para comprobar que hayan sido almacenadas.
  - d) Probar recuperar los mensajes de voz desde los softphones.

**Nota:** En caso de tener problemas con el desarrollo de la actividad, citar la guía de ayuda para la implementación de experiencias de laboratorio suministrada.



## **Experiencia N° 4**

### **Nombre Experiencia:**

Medición subjetiva de Calidad de Servicio en VoIP mediante Ethereal

### **Resumen**

En la presente experiencia, el alumno aprenderá a determinar los parámetros de calidad de servicio que se pueden ver afectados para el servicio de VoIP. Además aprenderá a utilizar el software Ethereal para medir estos parámetros. El alumno deberá realizar mediciones de calidad de servicio para llamadas efectuadas entre redes cableadas e inalámbricas de manera de establecer las diferencias si existieran.

### **Objetivo General**

- Permitir al alumno conocer y comprender uno de los métodos de medición de calidad de servicio subjetiva en VoIP.

### **Objetivos Específicos:**

- Permitir al alumno familiarizarse con la captura de paquetes mediante la utilización del software Ethereal, para su posterior análisis.
- Permitir al alumno conocer, comprender y determinar los parámetros que afectan a la calidad de servicio en VoIP.
- El alumno deberá ser capaz de evaluar en forma estadística el resultado de la calidad de servicio obtenida y establecer comparaciones con la telefonía tradicional.

## **MARCO TEORICO**

Todos los administradores de red necesitan tarde o temprano una herramienta que pueda capturar paquetes de la red y analizarlos. En el pasado, estas herramientas eran muy caras, propietarias, o ambas cosas. Sin embargo, con la llegada de Ethereal, todo eso ha cambiado.

Ethereal es quizá uno de los mejores sniffers open source disponibles hoy en día. Algunas de sus características son:

- Disponible para UNIX y para Windows.
- Captura y muestra paquetes desde cualquier interface en un sistema UNIX.
- Muestra paquetes capturados por otros programas de captura
- Filtra paquetes por muchos criterios.
- Busca paquetes usando filtros.
- Colorea paquetes basándose en filtros

Ethereal es un proyecto de software open source, liberado bajo la Licencia Pública GNU (GPL). Todo el código fuente está gratuitamente disponible bajo la GPL. Se puede modificar Ethereal para adecuarlo a las propias necesidades.

El código fuente de Ethereal y los kits binarios para algunas plataformas están disponibles en el web de Ethereal: <http://www.ethereal.com>.

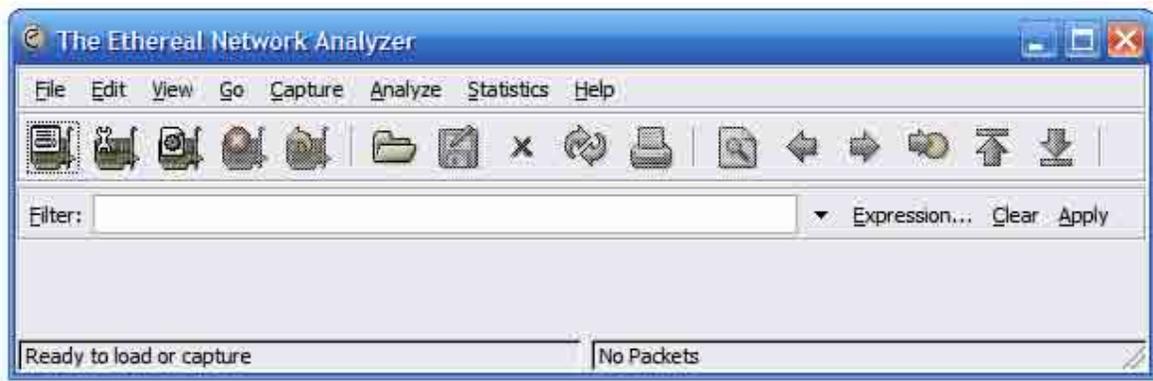
### ***Desarrollo de Ethereal***

Ethereal fue inicialmente desarrollado por Gerald Combs. El desarrollo y mantenimiento actual de Ethereal lo lleva el equipo Ethereal, un grupo que arregla bugs y proporciona nuevas funcionalidades.

Hay también un gran número de gente que ha contribuido con dissectores de protocolo para Ethereal, y se espera que esto continuará. Se puede encontrar una lista de la gente que ha contribuido al código de Ethereal en el enlace "Authors" en el web.

### ***Los menus de Ethereal***

El menú de Ethereal descansa a lo largo de la parte superior de la ventana de Ethereal, como se muestra en el ejemplo.



Contiene los siguientes elementos:

**File:** Este menú contiene elementos de menú para abrir y releer ficheros de captura, guardar ficheros de captura, exportar bytes, imprimir ficheros de captura, y salir de Ethereal.

**Edit:** Este menú contiene elementos de menú para encontrar una trama, marcar una o más tramas, establecer referencias de tiempo y establecer las preferencias (cortar, copiar y pegar no están actualmente implementados).

**View:** Este menú contiene elementos de menú para modificar opciones de visualización, ampliar/reducir la fuente de los paneles, colorear tramas, expandir todas las tramas, colapsar todas las tramas, mostrar un paquete en una ventana aparte, y recargar el fichero de captura actual.

**Go:** Este menú contiene elementos de menú para desplazarse entre las tramas.

**Capture:** This menu permite iniciar y detener capturas, y crear filtros de captura.

**Analyze:** Este menú contiene elementos de menú para seleccionar y filtrar paquetes coincidentes, seguir una sesión TCP, crear filtros de visualización, habilitar o deshabilitar la disección de protocolos, y configurar decodificadores especificados por el usuario.

**Statistics:** Este menú contiene elementos de menú para obtener un resumen de los paquetes que han sido capturados, mostrar estadísticas de jerarquía de protocolos,

gráficos de entrada/salida, listas de conversaciones y hosts, tiempos de respuestas y distintos análisis de protocolos particulares.

**Help:** Este menú permite mostrar los plugins cargados, contiene el elemento de menú About Ethereal... y acceso a alguna Ayuda básica.

### ***El menú Capture de Ethereal***

Elemento	Acelerador	Descripción
Start...	Ctrl-K	Este elemento muestra el cuadro de diálogo Preferencias de Captura (descrito más adelante en la sección Capturando paquetes con Ethereal) y permite empezar a capturar paquetes.
Stop	Ctrl-E	Este elemento detiene la captura actualmente en curso.
Interfaces		Muestra los interfaces de captura disponibles y su tráfico actual.
Capture Filters...		Este elemento muestra un cuadro de diálogo que permite crear y editar filtros de captura. Se puede dar nombre a los filtros, y se pueden guardar para uso futuro. Hay más detalles sobre este tema en la sección Definiendo y guardando filtros

### ***El menú Statistics de Ethereal***

Elemento	Descripción
Summary	Este elemento muestra una ventana de estadísticas con información sobre los paquetes capturados. Vea la Figura 3-10.
Protocol Hierarchy Statistics	Este elemento muestra un árbol jerárquico de estadísticas de paquetes Vea la Figura 3-11.
Conversations	Este elemento muestra un resumen de todas las conversaciones existentes. Los protocolos pueden ser Ethernet, FDDI, Fibre Channel, IPX, IPv4, TCP (IPv4 y IPv6), Token Ring o UDP (IPv4 y IPv6).
Endpoints	Este elemento muestra un resumen de todos los nodos (hosts) existentes. Los protocolos pueden ser Ethernet, FDDI, Fibre Channel, IPX, IPv4, TCP (IPv4 y IPv6), Token Ring o UDP (IPv4 y IPv6).
IO Graphs	Este elemento muestra una gráfica de las tramas capturadas en función del tiempo. Vea la Figura 3-12.

Elemento	Descripción
Conversation List	Este elemento muestra una lista de todas las conversaciones existentes y el tráfico en uno y otro sentido según un protocolo concreto. Los protocolos pueden ser Ethernet, FDDI, Fibre Channel, IPX, IPv4, TCP (IPv4 y IPv6), Token Ring o UDP(IPv4 y IPv6).
Endpoint List	Este elemento muestra una lista de todos los nodos (hosts) existentes y el tráfico entrante y saliente según un protocolo concreto. Los protocolos pueden ser Ethernet, FDDI, Fibre Channel, IPX, IPv4, TCP (IPv4 y IPv6), Token Ring o UDP(IPv4 y IPv6).
Service Response Time	Este elemento muestra estadísticas de tiempos de respuesta ante distintos servicios. Los servicios pueden ser DCE-RPC, Fibre Channel, ITU-T H.225 RAS, LDAP, ONE-RPC o SMB.
ANSI	Este elemento lleva la cuenta de las llamadas a las distintas funciones de distintos protocolos ANSI. Incluye A-Interface BSMAP, A-Interface DTAP y la operación MAP.
BOOTP-DHCP	Este elemento lleva la cuenta de los distintos tipos de mensajes DHCP.
GSM	Este elemento lleva la cuenta de las llamadas a las distintas funciones del protocolo GSM. Incluye A-Interface BSSMAP, A-Interface DTAP (incluyendo el control de llamada; la gestión de la movilidad GPRS, de la sesión GPRS, de la movilidad y de los recursos de radio; el servicio de mensajes cortos; y los servicios suplementarios) y la operación MAP.
H.225	Lleva la cuenta de los distintos tipos o razones de mensajes ITU-T H.225.
H.223 Conversations	Lleva la cuenta y analiza las conversaciones ITU-T H.223.
HTTP	Este elemento lleva la cuenta de las solicitudes HTTP y las distintas respuestas obtenidas (informativas, éxito, etc.).
ISUP Message Types	Este elemento lleva la cuenta de los distintos tipos de mensajes ISUP.
MTP3	Este elemento analiza el tráfico MTP3.
ONC-RPC Programs	Este elemento lleva la cuenta de las llamadas a los distintos programas ONE-RPC y sus tiempos de respuesta.
RTP	Este elemento muestra las estadísticas de los flujos RTP de la captura o analiza un flujo RTP concreto.
SIP	Este elemento lleva la cuenta de los distintos tipos de mensajes SIP.
TCP Stream Graph	Este Elemento Analiza Gráficamente Flujos TCP. Permite representar el RTT, el throughput o la secuencia de tiempos (estilo Stevens o tcptrace). Vea un ejemplo en la Figura 3-13.
WAP-WSP	Este elemento lleva la cuenta de los distintos tipos de PDU y códigos de estado.

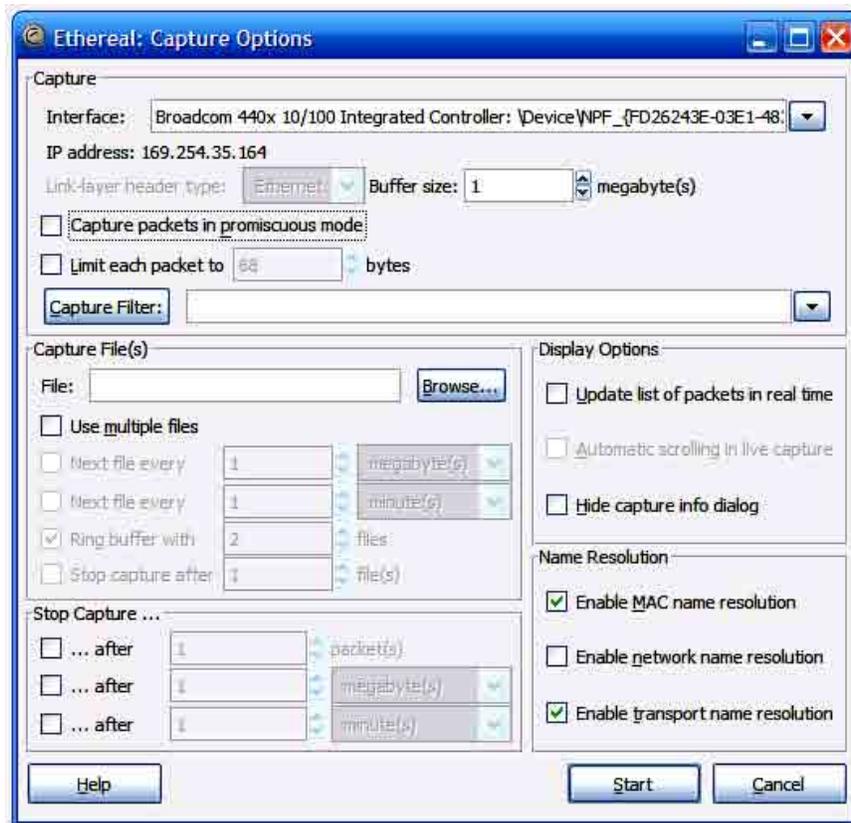
## Capturando paquetes con Ethereal

Iniciando Ethereal y seleccionando Start... desde el menú Capture. Esta muestra el cuadro de diálogo Preferencias de Captura y se tratará en más detalle en la sección Cuadro de diálogo Preferencias de Captura.

### Cuadro de diálogo Preferencias de Captura

Cuando se selecciona Start desde el menú Capture, Ethereal muestra cuadro de diálogo

Preferencias de Captura.



Se pueden establecer los siguientes campos en este cuadro de diálogo:

**Interface:** Este campo especifica el interface sobre el que se desea capturar. Sólo se puede capturar en un interface, y sólo se puede capturar en los interfaces que Ethereal ha encontrado en el sistema. Es una lista desplegable, por tanto sólo hay que pulsar el botón del lado derecho y seleccionar el interface que se desee. Por defecto es el primer interface no loopback que soporta capturas, y si no hay ninguno, el primer. En algunos

sistemas, los interfaces loopback no se pueden utilizar para capturas. Este campo realiza la misma función que la opción de línea de comandos -i <interface>.

**Capture limits:** Aquí se especifica el número de paquetes que se quieren capturar. Se puede poner el límite de la captura por número de paquetes, por volumen de datos o por tiempo transcurrido.

**Filter:** Este campo permite especificar un filtro de captura. Los filtros de captura se discuten en más detalle en la sección Filtrando mientras se Captura. Por defecto está vacío, o sin filtro. También se puede pulsar el botón/etiqueta Filter, y Ethereal mostrará el cuadro de diálogo Filtros y permitirá crear o seleccionar un filtro. Vea la sección Definiendo y guardando filtros

**File:** Este campo permite especificar el nombre de archivo que se usará para la captura cuando más tarde se escoja Save... o Save As... del menú File de Ethereal. No hay valor por defecto para este campo.

**Capture length:** Este campo permite especificar la cantidad máxima de datos que se capturarán para cada paquete, y es a lo que algunas veces se llama snaplen. Por defecto es 65535, que será suficiente para la mayoría de los protocolos. Debería ser al menos la MTU del interface en el que se está capturando.

**Capture packets in promiscuous mode:** Este botón de radio permite especificar que Ethereal debería poner el interface en modo promiscuo durante la captura. Si no se especifica, Ethereal sólo capturaré los paquetes que van desde y hacia su ordenador (no todos los paquetes que pasan por el interface).

**Nota:** Si algún otro proceso ha puesto el interface en modo promiscuo se puede estar capturando en modo promiscuo incluso si se desmarca esta opción.

**Update list of packets in real time:** This botón de radio permite especificar que Ethereal debería actualizar el panel de lista de paquetes en tiempo real. Si no se especifica, Ethereal no muestra ningún paquete hasta que se cancele la captura. Cuando se pulsa

este botón de radio, Ethereal captura en un proceso aparte y envía las capturas al proceso de visualización.

**Nota:** A veces esta opción no funciona correctamente en Windows

**Automatic scrolling in live capture:** This botón de radio permite especificar que Ethereal debería hacer scrolling del panel de lista de paquetes según llegan nuevos paquetes, de modo que siempre se vea el último paquete. Si no se especifica, Ethereal simplemente añade nuevos paquetes al final de la lista, pero no hace scrolling del panel de lista de paquetes.

**Enable MAC name resolution:** This botón de radio permite controlar si Ethereal traduce o no los primeros tres octetos de las direcciones MAC al nombre del fabricante al que le ha asignado ese prefijo el IETF.

**Enable network name resolution:** This botón de radio permite controlar si Ethereal traduce o no las direcciones IP a nombres de dominio DNS. Pulsando este botón de radio, el panel de lista de paquetes tendrá más información útil, pero también ocasionará que se produzcan solicitudes de búsqueda de nombres, lo que podría interferir con la captura.

**Enable transport name resolution:** This botón de radio permite controlar si Ethereal traduce o no los números de puerta a protocolos.

Una vez que se han establecido los valores deseados y seleccionado los botones de radio que se necesitan, pulse simplemente en OK para comenzar la captura, o Cancel para cancelarla. Si se inicia una captura, Ethereal abre un cuadro de diálogo que muestra el progreso de la captura y permite detener la captura cuando se tienen suficientes paquetes capturados.

## Desarrollo de la Actividad

1. Realizar captura de paquetes mediante Ethereal, mientras se realizan llamadas entre los anexos IP.
2. Analizar el detalle de los parámetros de red obtenidos en la captura del tráfico de las llamadas y tomar nota de los parámetros de Jitter, Latencia y Pérdida de Paquetes.
3. De ser posible, realizar cambios de Codec (ej. G.711 a G.723) y repetir los pasos 1 y 2.

## Preguntas:

1. Indicar cuales son los parámetros que afectan la QoS en VoIP.
2. Indicar una forma de medir la QoS en VoIP.
3. Indique al menos 3 codec utilizados en VoIP y sus principales características, ventajas y desventajas.
4. Detalle escrita y gráficamente el flujo de una llamada realizada con protocolo SIP. Utilice como referencia los resultados obtenidos con Ethereal en el laboratorio.
5. De la experiencia realizada utilizando el software Ethereal, se requiere el siguiente detalle:
  - a) Indicar los valores obtenidos de Jitter, Latencia y pérdida de paquetes por cada llamada realizada, llenando la siguiente tabla:

Tabla con resultados Obtenidos			
Llamadas realizadas	Jitter [ms]	Latencia [ms]	Perdida Paquetes [%]
Llamada a extensión 1			
Llamada a extensión 2			
Llamada a extensión 3			

- b) Indicar los resultados obtenidos en las mediciones de QoS según la escala MOS completando la siguiente tabla, posteriormente realice un gráfico detallando la información.

<b>Resultados de QoS Obtenidos de la Experiencia</b>	
<b>Llamadas realizadas</b>	<b>Valoración de Escala MOS</b>
Llamada a extensión 1	
Llamada a extensión 2	
Llamada a extensión 3	

- c) Repita los pasos a) y b) pero realizando llamadas entre equipos conectados en una red Wi-Fi y compare los resultados finales con los obtenidos en una red cableada.

**Nota:** En caso de tener problemas con el desarrollo de la actividad, citar la guía de ayuda para la implementación de experiencias de laboratorio suministrada.

## **ANEXO II. GUIA DE AYUDA PARA EXPERIENCIAS DE LABORATORIO**



**Universidad Austral de Chile**

Escuela Electricidad y Electrónica

UNIVERSIDAD AUSTRAL DE CHILE

FACULTAD DE CIENCIAS DE LA INGENIERÍA

ESCUELA DE ELECTRICIDAD Y ELECTRÓNICA

**GUÍA DE AYUDA PARA LA IMPLEMENTACIÓN DE LOS LABORATORIOS DE VOIP,  
MEDIANTE LA UTILIZACIÓN DE ASTERISK IPBX.**

AUTOR: ALEX HANS HUNTER MENA  
FECHA DE EMISIÓN: NOVIEMBRE DE 2007

## INDICE

	Página
1. Introducción.....	1
2. Procedimiento para realizar experiencia N° 1.....	2
2.1. Instalación de Linux Ubuntu Feisty 7.04.....	2
3. Procedimiento para realizar experiencia N° 2.....	8
3.1. Instalación de Asterisk 1.2.....	8
3.2. Instalación de FreePBX 2.2.2.....	10
4. Procedimiento para realizar experiencia N° 3.....	17
4.1. Configuración extensiones SIP.....	17
4.2. Configuración extensiones SIP en FreePBX.....	17
4.3. Configuración de los anexos utilizando el Softphone X-LITE.....	22
5. Procedimiento para realizar experiencia N° 4.....	25
5.1. Medición Subjetiva de Calidad de Servicio en Voz IP (QoS).....	25
5.2. Descripción del Software.....	25

## **1. INTRODUCCIÓN**

La presente guía de ayuda, tiene como fin poder ser un complemento en cada una de las experiencias a realizar en el laboratorio, la idea es que el alumno cada vez que necesite ayuda en alguna etapa de la actividad, pueda consultar paso a paso el como llevar a cabo el desarrollo de las experiencias y resolver el problema que se le presente, además de poder complementar cada informe una vez que estas sean terminadas satisfactoriamente.

Esta guía de ayuda, se entrega en forma separada a las guías de laboratorio, de modo que el alumno pueda descubrir por sus propios medios el procedimiento para llegar a cumplir lo solicitado, de esta forma, se busca que el alumno desarrolle la capacidad autodidacta y pueda familiarizarse aún más con la implementación.

## 2. PROCEDIMIENTO PARA REALIZAR EXPERIENCIA N° 1

### 2.1. INSTALACIÓN DE LINUX UBUNTU FEISTY 7.04

**Paso 1:** Lo primero que se debe hacer es descargar la versión de Linux, “Ubuntu 7.04 – Feisty” ó pedir el cd de instalación, el cual se ofrece en forma gratuita, por medio del link <http://www.ubuntu-es.org/>, clicar en el menú “conseguir”, para solicitar un cd gratuito, es necesario registrarse.

**Paso 2:** Encender el PC, este será el futuro servidor ó plataforma para el servicio de VoIP, luego insertar el CD que contiene la instalación de Ubuntu Feisty 7.04.

Al iniciar el PC, se observara lo siguiente:



Se deberá seleccionar el Idioma presionando “F2”, Language y seleccionar Español.

**Paso 3:** Comenzará a cargar el sistema operativo, que llegara directamente al escritorio del LiveCD. Es posible utilizarlo para instalar la versión completa, para testear un poco y probar. Una vez termine de iniciar, dentro del escritorio, se observara lo siguiente:



Se debe realizar doble click sobre el ícono “Instalar” para comenzar a instalar el sistema operativo en el disco duro.

**Paso 4:** Seleccionar la zona horaria, en este caso se debe seleccionar la zona horaria de Santiago de Chile.

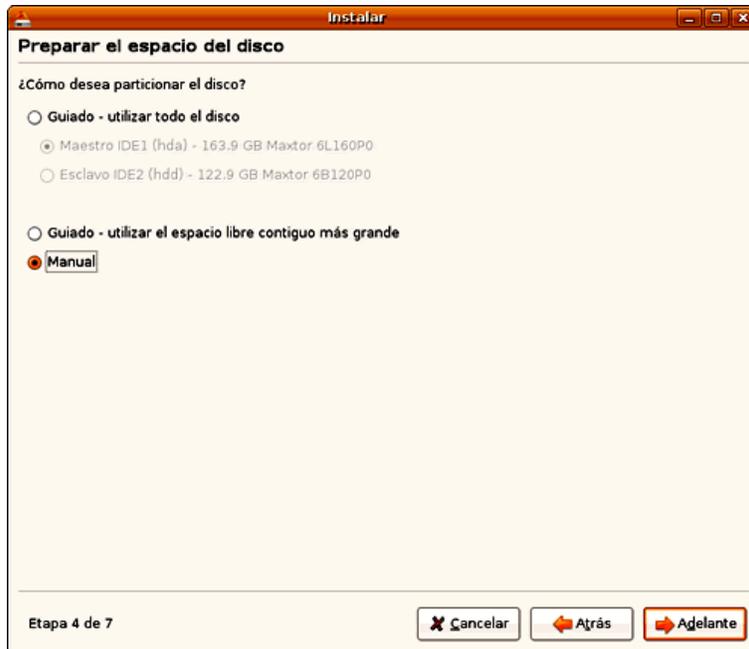


**Paso 5:** Seleccionar la distribución de caracteres del teclado, en este caso, se debe seleccionar “Spain”, tal como aparece en la imagen a continuación:



**Paso 6:** Selección de la partición en que se instalara el sistema operativo, en esta sección se ofrecen las siguientes alternativas:

- **Guiado - utilizar todo el disco:** Se coge todo un disco y se hacen dos partes de forma automática, una para Ubuntu y otra para la memoria SWAP (algo parecido al archivo de intercambio de Windows).
- **Guiado - utilizar el espacio libre contiguo más grande:** Versión de la anterior opción en la que se respetan las particiones previamente establecidas en el disco duro. Es el método automático para el caso en que se tenga instalado Windows u otro sistema operativo y se haya considerado espacio suficiente para la instalación de Ubuntu.
- **Manual:** El usuario configura la partición según las necesidades, este es el método mas recomendado.



**Paso 7:** Consiste en la creación de un usuario que tendrá los privilegios de Administrador del equipo (root) mediante la utilización del comando *sudo*.



**Paso 8:** Una vez finalizado el proceso, comenzaran a instalarse los componentes básicos de Ubuntu Feisty 7.04, luego el sistema solicitara que sea reiniciado, se deberá retirar el CD de instalación y ya se podra comenzar a utilizar Linux.



**Paso 9:** Instalación de repositorios adicionales para Ubuntu 7.04 (Feisty - Gnome).

Se recomienda realizar la instalación de repositorios adicionales en Ubuntu 7.04 (Feisty), de forma de poder encontrar software y paquetes adicionales que pueden ser utilizados durante la instalación de Asterisk y sus dependencias. A continuación se indican los pasos a seguir:

Se recomienda realizar un respaldo de la lista de repositorios que posee Ubuntu por defecto para luego agregar los nuevos repositorios, esto se debe hacer copiando y pegando en la consola de Ubuntu los comandos que se indican a continuación:

```
Sudo mv /etc/apt/sources.list /etc/apt/sources.list_backup  
gksudo gedit /etc/apt/sources.list
```

**Paso 10:** Estando en el archivo sources.list, se deben copiar y pegar las siguientes líneas al final del archivo, luego guardar y cerrar:

```
## Uncomment the following two lines to fetch updated software from the network  
deb http://archive.ubuntu.com/ubuntu feisty main restricted  
deb-src http://archive.ubuntu.com/ubuntu feisty main restricted  
## Uncomment the following two lines to fetch major bug fix updates produced  
## after the final release of the distribution.  
deb http://archive.ubuntu.com/ubuntu feisty-updates main restricted  
deb-src http://archive.ubuntu.com/ubuntu feisty-updates main restricted  
## Uncomment the following two lines to add software from the 'universe'  
## repository.
```

```
## N.B. software from this repository is ENTIRELY UNSUPPORTED by the Ubuntu
## team, and may not be under a free licence. Please satisfy yourself as to
## your rights to use the software. Also, please note that software in
## universe WILL NOT receive any review or updates from the Ubuntu security
## team.
deb http://archive.ubuntu.com/ubuntu feisty universe
deb-src http://archive.ubuntu.com/ubuntu feisty universe
deb http://security.ubuntu.com/ubuntu feisty-security main restricted
deb-src http://security.ubuntu.com/ubuntu feisty-security main restricted
deb http://security.ubuntu.com/ubuntu feisty-security universe
deb-src http://security.ubuntu.com/ubuntu feisty-security universe
deb http://archive.ubuntu.com/ubuntu feisty multiverse
deb-src http://archive.ubuntu.com/ubuntu feisty multiverse
deb http://archive.ubuntu.com/ubuntu feisty-backports main restricted universe multiverse
deb http://archive.canonical.com/ubuntu feisty-commercial main
deb http://medibuntu.sos-sts.com/repo/ feisty free non-free
deb-src http://medibuntu.sos-sts.com/repo/ feisty free non-free
```

**Paso 11:** Una vez que se haya terminado con el paso anterior, se debe ir nuevamente a la consola de comandos de Ubuntu para actualizar la lista de software. Esto se realizara escribiendo las siguientes líneas:

```
sudo apt-get update
```

### 3. Procedimiento para realizar experiencia N° 2

#### 3.1. INSTALACIÓN DE ASTERISK 1.2

A continuación se detalla la instalación de Asterisk 1.2, programa esencial para poder implementar nuestra IPBX que permitirá la comunicación VoIP, todas sus ventajas están listadas en la siguiente link <http://www.asterisk.org/features>. Se utilizara la versión 1.2, ya que es la versión mas estable hasta agosto de 2007, además se instalara FreePBX 2.2.2, este ultimo, nos permite realizar todas las configuraciones de Asterisk de una forma sencilla mediante interface web, toda la información necesaria puede ser encontrada en [www.freepbx.org](http://www.freepbx.org), en este caso se utilizo la versión 2.2.2 estable ya que la versión 2.2.3 beta esta aún en desarrollo (Agosto de 2007).

Debido a que se implementara VoIP pura en una red LAN, no sera necesario instalar los paquetes de Zaptel, los cuales permiten trabajar con tarjetas TDM, tampoco se instalara el paquete Libpri, ya que no nos conectaremos a la red PSTN, es decir, no utilizaremos servicios ISDN PRI.

Se requerirá descargar los siguientes paquetes al directorio **/usr/src**

Los archivos utilizados es este trabajo de tesis se detallan a continuación, recordando que no se utilizaron tarjetas TDM ni enlaces primarios, ya que el trabajo se acoto al servicio de VoIP:

- Asterisk 1.2.18
- Addons 1.2.6
- Sounds 1.2.1

Una vez descargados estos archivos, es recomendable guardarlos en las librerias **/usr/src** para mantener la estructura de archivos definida por *Ubuntu*. Antes de compilar estos paquetes, hay que comprobar que estén instaladas las dependencias de cada uno de estos módulos.

- *Compilador GCC* (versión 3.x o mayor) y sus dependencias. Versión instalada *gcc-3.3*

- El intérprete de comandos *bison* es necesario para la CLI de *Asterisk* Versión instalada: *bison-1.875*
- La librería criptográfica de *Asterisk* requiere *openssl* y su paquete de desarrollo.

Es importante destacar que resulta mas sencillo descargar e instalar *Asterisk* con *Sinaptic*, esta es una aplicación de *Ubuntu*, la cual permite instalar programas con sólo seleccionarlos, siempre y cuando estén disponibles ó si los repositorios se han actualizado.

Se debe ingresar al directorio ***/usr/src*** mediante consola tal como en el caso anterior y para la descarga de los paquetes se debe escribir:

```
/usr/src# wget http://ftp.digium.com/pub/asterisk/asterisk-addons-1.2.6.tar.gz  
/usr/src# wget http://ftp.digium.com/pub/asterisk/old-releases/asterisk-sounds-1.2.1.tar.gz  
/usr/src# wget http://ftp.digium.com/pub/asterisk/releases/asterisk-1.2.18.tar.gz
```

Se descomprimen los paquetes que se acaban de bajar tipeando en la consola:

```
/usr/src# tar xvfz asterisk-addons-1.2.6.tar.gz  
/usr/src# tar xvfz asterisk-sounds-1.2.1.tar.gz  
/usr/src# tar xvfz asterisk-1.2.18.tar.gz
```

Para compilar *Asterisk* se debe ingresar al directorio en el cual se descomprimió escribiendo:

```
cd /usr/src/asterisk-1.2.18
```

Luego se escriben los siguientes comandos que instalaran el software:

```
/usr/src/asterisk-1.2.18# make clean  
/usr/src/asterisk-1.2.18# make all  
/usr/src/asterisk-1.2.18# make install
```

Si no han ocurrido errores, debe salir algo así:

```
+---- Asterisk Installation Complete -----+
*****
ASTERISK installed.
Installation finished.
*****
```

### 3.2 INSTALACIÓN DE FREEPBX

Cuando ya se tiene todo funcionando se debe instalar Freepbx, esta herramienta no es solo un portal para realizar configuraciones, añade muchas herramientas a asterisk, su modo de operación es simple, todas las configuraciones que se hacen en sus paginas las guarda en una base de datos y cuando se aplican las configuraciones escribirá estas en los ficheros de configuración de Asterisk, por ello, cuando se instale Freepbx no se deben modificar los archivos de configuración, para esto se creara un nombre\_de\_archivo\_de\_configuración\_custom.conf sera ahí donde freepbx guardara los archivos originales.

FreePBX puede ser descargado desde su sitio oficial en:

[www.freepbx.org](http://www.freepbx.org) y luego en la opción *download* descargar la ultima versión estable.

La última versión estable de FreePBX hasta agosto de 2007 aún no esta disponible dentro de los repositorios de Sinaptic, por lo que debe ser instalado en forma manual.

Para poder instalar FreePBX, se requiere un “*Subversion Client Install*” este programa, permite instalar versiones antiguas de un software. En ubuntu, el paquete Subversión, incluye un paquete llamado Subversión Cliente (svn), herramienta que permite utilizar repositorios en internet usando un programa como el ssh, el cual permite conectarse a multiples servidores.

Para instalar el svn se debe abrir una consola de comandos y escribir:

```
sudo apt-get install svn
```

Ahora se debe ingresar al directorio `/usr/src` y luego `parar asterisk` tipeando:

```
kill -9 `pidof asterisk`
```

Para poder instalar freepbx se deben realizar muchas modificaciones en archivos de configuración, así como cambiar permisos sobre directorios. Se comenzará editando `php.ini` de apache para que acepte archivos de 40 mb (la música en espera que se subirá por interfase la web):

```
sudo gedit /etc/php5/apache2/php.ini
```

Se debe buscar la línea que dice:

```
; Maximum allowed size for uploaded files.  
upload_max_filesize = 2M
```

Debe cambiarse el valor 2 Megas por **40M**

Luego se debe modificar el archivo `php.ini` del paquete `php4-cgi` y `php5-cgi` que utilice la extension `mysql.so` (con esto se le permite a este paquete realizar consultas sql contra el motor de bases de datos mysql):

```
sudo gedit /etc/php4/cgi/php.ini  
sudo gedit /etc/php5/cgi/php.ini
```

Se debe buscar la línea en los dos archivos y se descomenta (quitar el punto y coma del comienzo), esta es la línea a buscar:

```
;extension=mysql.so
```

Ahora se debe agrgar un usuario que se llame `asterisk` y un grupo que se llame `asterisk`:

```
groupadd asterisk
```

```
useradd -c "asterisk PBX" -d /var/lib/asterisk -g asterisk asterisk
```

El home del usuario será donde estarán todos los archivos necesarios para que asterisk arranque.

Se debe crear el siguiente directorio para que freepbx no genere conflictos en la instalación, todos los procesos que se inicien son ingresados en el directorio /var/run y Freepbx requiere que se sitúen en un directorio dentro de ese:

```
mkdir /var/run/asterisk
```

Se edita el archivo /etc/asterisk/asterisk.conf para indicar que agregue el proceso en el nuevo directorio:

```
sudo gedit /etc/asterisk/asterisk.conf
```

La línea a cambiar es la que dice:

*astrundir => /var/run* . Esta debe quedar así:

*astrundir => /var/run/asterisk*

En esta momento, se deben compilar paquetes requeridos para la ejecución de Freepbx, estos son, asterisk-addons y el asterisk-sounds:

```
cd /usr/src/  
cd asterisk-addons-1.2.6
```

Se debe ejecutar un comando antes de compilar, este es:

```
/usr/src/asterisk-addons-1.2.6# perl -p -i.bak -e  
's/CFLAGS.*D_GNU_SOURCE/CFLAGS+=-D_GNU_SOURCE\nCFLAGS+=-  
DMYSQL_LOGUNIQUEID/' Makefile
```

Ahora se debe apreciar un salto de línea:

```
/usr/src/asterisk-addons-1.2.6# make clean  
....  
/usr/src/asterisk-addons-1.2.6# make  
....  
/usr/src/asterisk-addons-1.2.6# make install  
....
```

Ahora se compilan los sonidos:

```
/usr/src/asterisk-addons-1.2.6# cd ../asterisk-sounds-1.2.1  
/usr/src/asterisk-sounds-1.2.1# make install
```

Se procede a descomprimir el paquete de FreePBX descargado:

```
/usr/src# tar xvfz freepbx-2.2.2.tar.gz
```

Se aconseja configurar todo con el mismo password para no tener problemas futuros, se indica a continuación en las líneas donde dice *PASSWORD*, las contraseñas que se deben ingresar para los archivos que involucran a FreePBX

```
/usr/src# mysqladmin -u root password 'TU_PASSWORD_MYSQL'  
/usr/src# mysqladmin create asteriskcdrdb -p  
/usr/src# mysql --user=root --password=TU_PASSWORD_MYSQL asteriskcdrdb <  
/usr/src/freepbx-2.2.1/SQL/cdr_mysql_table.sql  
/usr/src# mysqladmin create asterisk -p  
/usr/src# mysql --user root -p asterisk < /usr/src/freepbx-2.2.1/SQL/newinstall.sql  
/usr/src# mysql --user root -p  
  
mysql> GRANT ALL PRIVILEGES on asteriskcdrdb.* TO asteriskuser@localhost  
IDENTIFIED BY 'TU_PASSWORD_ASTERISK';  
Query OK, 0 rows affected (0.00 sec)  
mysql> GRANT ALL PRIVILEGES on asterisk.* TO asteriskuser@localhost
```

```
IDENTIFIED BY 'TU_PASSWORD_ASTERISK';
```

```
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> FLUSH PRIVILEGES;
```

```
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> quit
```

Ahora se debe cambiar el usuario y el grupo con el que se ejecuta apache2, al cambiarlo se le dara permiso al portal para que interactúe con la consola de asterisk y los demás servicios. Para esto se edita el archivo apache2.conf:

```
sudo gedit /etc/apache2/apache2.conf
```

Se deben buscar estas dos líneas:

```
User www-data
```

```
Group www-data
```

Debe cambiarse www-data por asterisk para dejarlo así:

```
User asterisk
```

```
Group asterisk
```

Se edita también el archivo `/etc/php5/apache2/php.ini` para habilitar `register_globals`, por defecto esta en Off, se debe cambiar a On.

Una vez hechas las dos cosas se reinicia el servicio apache2:

```
# /etc/init.d/apache2 restart
```

Ahora ha llegado el momento de instalar Freepbx, al comienzo hará unas cuantas preguntas ya que no tiene el archivo `/etc/ampportal.conf`. Incluso la instalación indicara que el archivo `/var/lib/asterisk/bin/retrieve.conf` no tiene permisos para acceder al archivo que el mismo acaba de crear `/ampportal.conf`.

Cuando se salga sin haber instalado se daran los permisos y se vuelve a lanzar la instalación. Ojo con los usuarios y contraseñas poner atención cuales se piden y cuales son las nuevas que se van creando. El comando para lanzar la instalación es el siguiente:

Ahora, se procede con la instalación, el programa va a solicitar sustituir los archivos de configuración existentes, se debe responder si a todo (all):

```
cd /usr/src/freepbx-2.2.1  
/usr/src/freepbx-2.2.1# ./install_amp
```

Posteriormente se deben cambiar los permisos de lectura y escritura a los archivos, se deben escribir los siguientes comandos:

```
/# chmod 777 -R /var/lib/asterisk/bin  
/usr/src/freepbx-2.2.1# chmod 777 /etc/amportal.conf
```

Antes de lanzar de nuevo la instalación se debe cargar Asterisk, tipeando:

```
# asterisk ó amportal start
```

Se ejecuta la instalación:

```
/usr/src/freepbx-2.2.1# ./install_amp
```

No debería dar ningún error, a lo mejor alguno del op\_flashpanel, no puede ser cargado, pero no debe dar error ninguno de permiso denegado ni nada parecido.

En el archivo que crea de configuración preguntara en que directorio va a ejecutarse el portal, por defecto lo crea en /var/www/html, por conveniencia se a cambiado a /var/www/freepbx, este donde este, se le debe indicar que el administrador es el usuario y el grupo asterisk:

```
/# chown asterisk:asterisk -R /var/www/freepbx
```

A continuación se procede a ejecutar y detener *amportal*:

```
#!/amportal stop
```

```
#!/amportal start
```

Se comprueba que *amportal* ejecutará Asterisk, pone permisos a los archivos que necesita y ejecutara el Flash Operator Channel.

Una vez ejecutado *amportal*, se prueba FreePBX escribiendo en un navegador web la siguiente dirección:

<http://direccionip/directoriodeinstalacion>

En este caso se escribió:

<http://192.168.1.2/freepbx>

Cuando se ingrese al área de administración, pedirá usuario y contraseña, esta puede ser *admin* y *admin*, aunque debe ser la misma configurada en *amportal*, en este caso fue:

User: asterisk

Password: asterisk

Para cambiar el password se debe configurar el mismo en los siguientes tres sitios:

*/etc/amportal.conf* (línea *AMPMGRPASS=tupassword*)

*/etc/asterisk/manager.conf* (línea *secret = tupassword*)

En el portal *freepbx* en modo administrador, en la pestaña *setup*, en el link *administrators*, cambiar la contraseña para *admin*.

Se debe crear el directorio */var/lib/asterisk/sounds/custom* con permisos (*chmod*) *777* y del grupo *asterisk*.

```
#!/mkdir /var/lib/asterisk/sounds/custom
```

```
#!/chown asterisk:asterisk -R
```

```
/var/lib/asterisk/sounds/custom  
/# chmod 777 -R /var/lib/asterisk/sounds/custom
```

En el portal de Freepbx en la zona de administración y en tools se encuentra una opción "*Module Admin*", este contiene los complementos que se pueden instalar, en este caso se instalaron todos y posteriormente se actualizaron.

## 4. Procedimiento para realizar experiencia N° 3

### 4.1. Configuración extensiones SIP

En esta experiencia, se crearan clientes SIP configurados en FreePBX, en conjunto con el *softphone X-Lite*. Una vez descargado e instalado en los computadores que actuarán de clientes es necesario introducir en cada softphone la configuración del SIP Proxy, es decir, la dirección IP de la Central (servidor), así como el nombre de usuario y clave para poder registrarse en ella (configurados en Asterisk).

Se utilizará el puerto 5060, definido por las especificaciones del protocolo SIP.

### 4.2. Configuración de las extensiones para usuarios SIP en FreePBX.

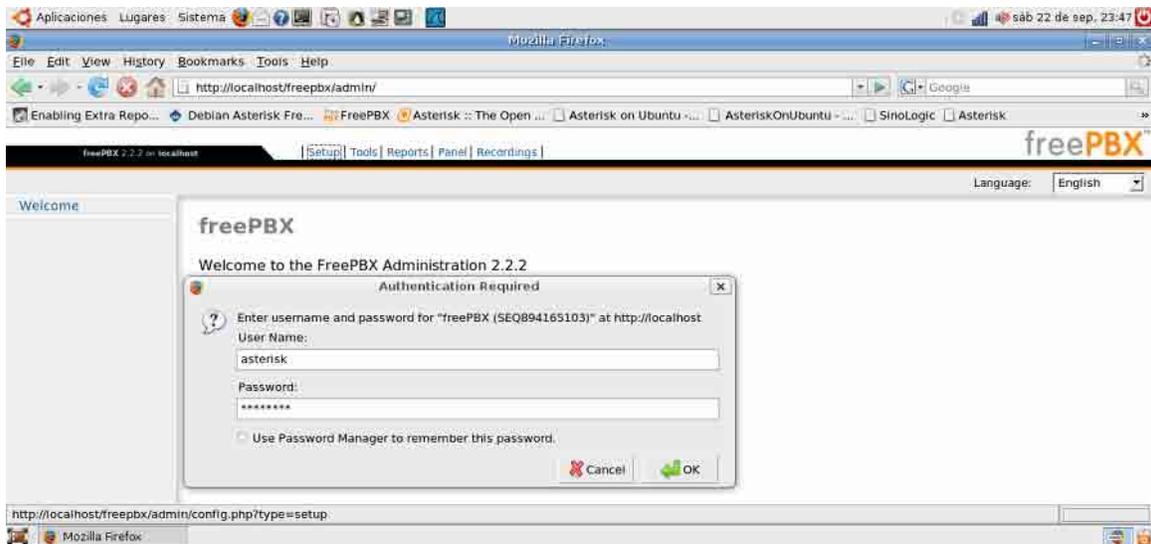
X-Lite es un softphone libre, puede ser descargado desde *www.counterpaht.com*, para habilitarlo sólo es necesario un número de anexo, password, la IP del servidor y la plataforma Asterisk.

Para la configuración de los anexos en FreePBX se requiere:

- Crear una extensión SIP
- Un password para que se registre el softphone "Secret"
- Configuración del NAT si las extensiones están en otra LAN (no aplica en este caso).

Descripción paso a paso para crear una extensión SIP en FreePBX:

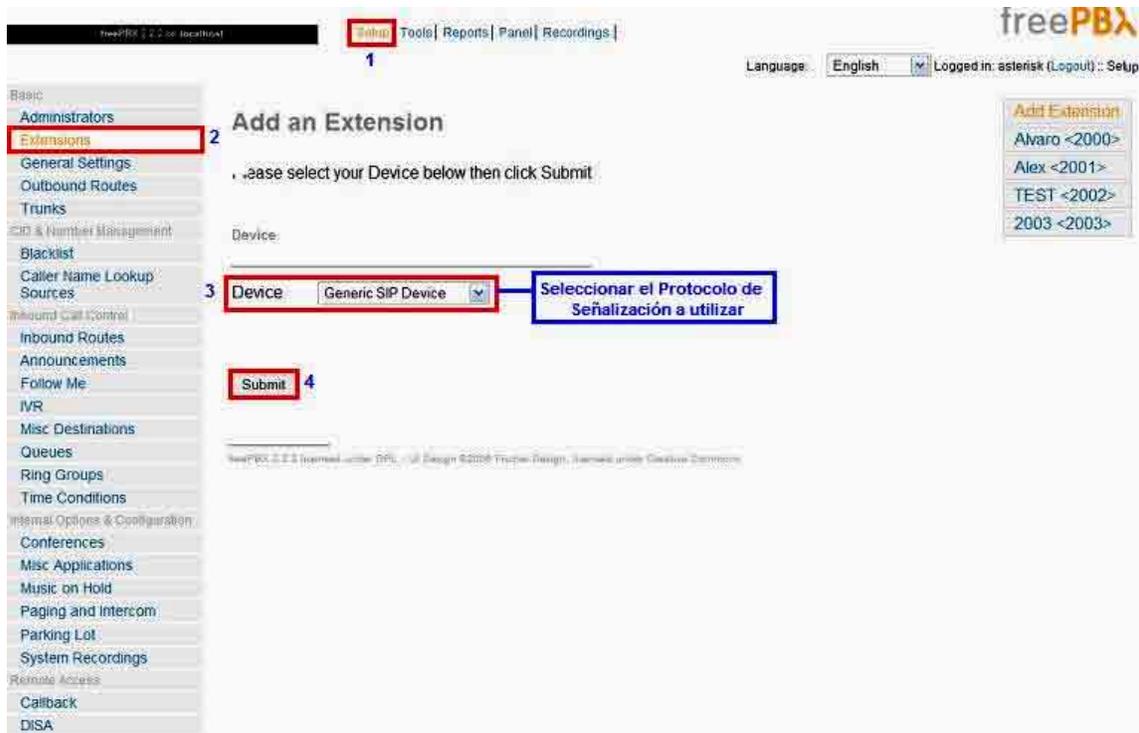
- **Paso 1:** Ingresar al interface web FreePBX escribiendo en el navegador del servidor: **<http://localhost/freepbxadmin/>**



**Paso 2:** Clickear en menú “Setup”, se solicitara *user* y *password*, en este caso se utiliza por conveniencia: *user = asterisk*, *password=asterisk* (creados en el momento de la instalación de FreePBX).

**Paso 3:** Seleccionar “Extensions” en el FreePBX para crear el anexo.

**Paso 4:** Seleccionar *Generic SIP Device* en el menú “Device” (dispositivo), ya que será el tipo de señalización a utilizar.



**Paso 5:** Una vez dentro se deben completar los siguientes campos:

**Extensión Number:** Número de la extensión (3 ó 4 dígitos). Ej.: 2001.

**Display Name:** Nombre que aparecerá en el softphone. Ej.: Alex.

**Paso 6:** Una vez creada la extensión, seleccionarla para realizar una configuración mas detallada, se debe clicar en el nombre de la extensión creada (menú superior derecho).

**Paso 7:** Ahora se procederán a completar algunos campos esenciales para poder habilitar la extensión, estos son:

- **Secret:** Es el password con el que se registrara el softphone en Asterisk (el mismo a ingresar en el softphone).

- **NAT:** se completara con “never” si la extensión se encuentra en un equipo cliente de la misma LAN ó con “yes” en caso contrario, como no aplica en este caso particular, completar con “never”.

- **Allow:** permite habilitar ó inhabilitar la extensión, completar con “yes”.

Los demás campos se mantienen por defecto.

**Paso 8:** Habilitación del buzón de voz. Dentro de las opciones de Asterisk, existe la posibilidad de crear un buzón de voz para cada extensión que se desee agregar, esto permitira recibir mensajes de los demas usuarios cuando no se este utilizando la aplicación. Dentro de la misma configuración de la extensión se deben completar los siguientes campos:

- **Voice mail:** permite habilitar (*Enable*) ó inhabilitar (*Disable*) el buzón de voz.

- **Voicemail password:** es la contraseña para poder rescatar los mensajes de voz desde el softphone.

- **Email address:** se puede configurar con una cuenta de email, para que cada vez que haya un mensaje, este sea enviado al e-mail especificado.

- **E-mail attachment:** se configura para que la grabación del buzón de voz llegue al e-mail indicado como archivo .wav adjunto.

- **Play CID (Caller ID):** al habilitarlo, indicara el número de extensión a la que pertenece el buzón de voz, cuando el usuario al que se esta llamando no esta disponible.

- **Play Envelope:** al habilitarlo, cuando se rescate un mensaje indicara la fecha y hora en que fue grabado el mensaje.

- **Delete vmail:** al habilitarlo indicara que se elimine el mensaje del servidor una vez que haya sido enviado al e-mail, en este caso se debe activar el campo “email attachment”.

Pantalla de configuración de la extensión:

freePBX 2.2.2 (ip:localhost) 1 **Submit** Tools | Reports | Panel | Recordings | freePBX

Language: English | Logged in: asterisk (Logout) | Setup

**Administrators:** 2 **Extension: 2001** Add Extension  
Alvaro <2000>  
Alex <2001>  
TEST <2002>  
2003 <2003>

Delete Extension 2001  
Add Gabcast Settings  
Add Follow Me Settings  
Edit Extension

Display Name: Alex 3

Extension Options

Direct DID  
DID Alert Info  
Outbound CID  
Emergency CID

Device Options

This device uses sip technology.

secret	2001 4 (Password)
dtmfmode	rfc2833
canreinvite	no
context	from-internal
host	dynamic
type	friend
nat	never 5 (No esta fuera de la red LAN "Never")
port	5060
qualify	yes
callgroup	
pickupgroup	
disallow	
allow	yes 6 (Habilita la extensión)
dial	SIP/2001
accountcode	
mailbox	2001@device

Fax Handling

Fax Extension: freePBX default  
Fax Email:  
Fax Detection Type: None  
Pause after answer: 0

Privacy

Privacy Manager: No

Dictation Services

Dictation Service: Disabled  
Dictation Format: Ogg Vorbis  
Email Address:

Recording Options

Record Incoming: On Demand  
Record Outgoing: On Demand

VoiceMail & Directory

Status: Enabled 7 (Habilita Buzón de voz)  
voicemail password: 2001 8  
email address:  
pager email address:  
email attachment:  yes  no  
Play CID: 9  yes  no  
Play Envelope:  yes  no  
Delete Vmail:  yes  no  
vm options:  
vm context: default

**Submit** 10 (Guarda los cambios)

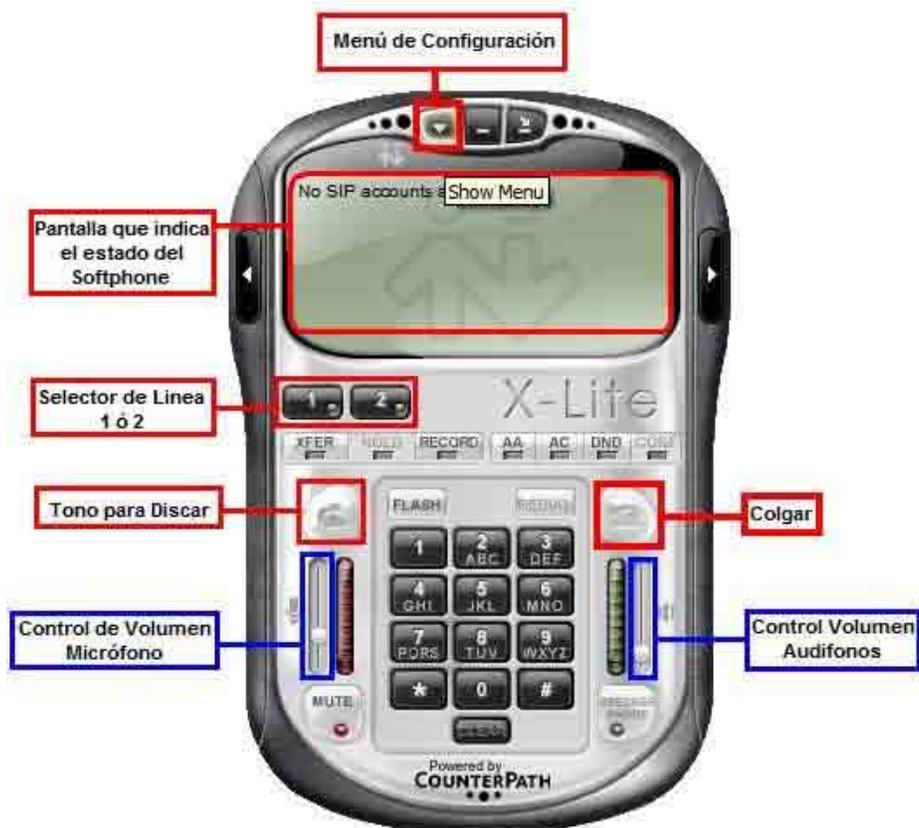
### 4.3. Configuración de los anexos utilizando el Softphone X-LITE:

**Paso 1:** Configuración del softphone X-Lite 3.0 (última versión encontrada en Agosto de 2007):

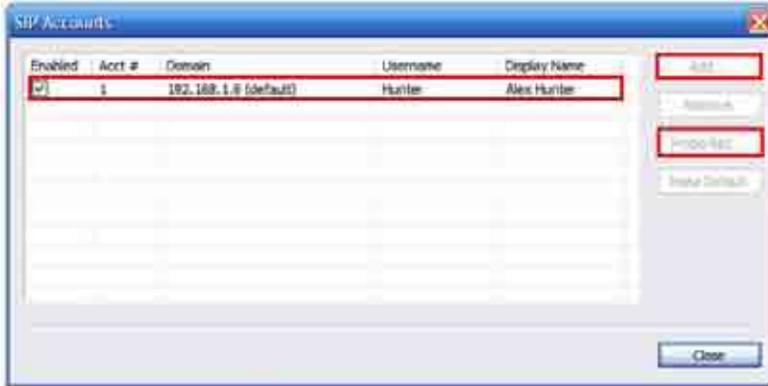
Programa (Software):	X-Lite
Versión:	3.0
Distribución para:	Windows XP
Tipo de Software:	Free
Sitio oficial:	<a href="http://www.counterpath.com/">http://www.counterpath.com/</a>



Para entrar al menú de configuración del softphone, se debe presionar en el icono *Show Menu*, en un principio dirá en pantalla “*No SIP accounts are enable*”, es decir, no hay cuentas SIP habilitadas. Al presionar en el icono, se desplegará el menú que aparece en la imagen derecha, en este menú se debe hacer click en *Add* para agregar los parámetros de configuración, los cuales serán los mismos que se configuraron en FreePBX.



Componentes Básicos del Softphone X-Lite



En el Menú de Configuración "SIP Account Settings..." se efectuara la configuración de la extensión

La opción Add... permite agregar una extensión

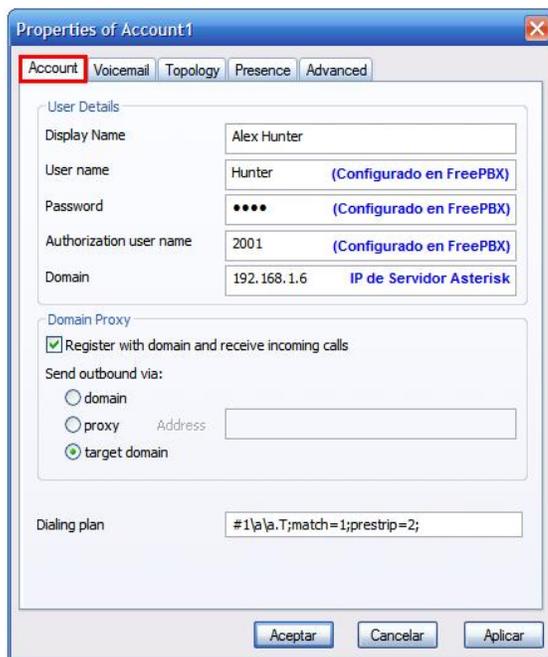
La opción Properties... permite modificar una cuenta ya creada y configurada



**Paso 2:** Propiedades de la cuenta "Properties", en este menú se procederá a configurar la siguiente información:

- **Display Name:** Nombre creado para la extensión en FreePBX, el cual identificara la extensión y que se mostrara en la pantalla del softphone.
- **User Name:** deberá tener la misma configuración que el campo "Display Name" de la extensión SIP creada en FreePBX, este es el usuario con el que se registrara el softphone en Asterisk.
- **Password:** es la contraseña que se creo en campo "Secret" de FreePBX.
- **Domain:** es la dirección IP que posee el computador que actua como servidor y que tiene instalado Asterisk.

Con estos campos ya es posible registrar el softphone para comenzar a utilizarlo, al finalizar y estar registrado, el softphone indicara en la pantalla "Ready" lo que significa que esta habilitado e indicando el número de extensión configurada, tal como aparece en la siguiente imagen:



Para llamar a otra extensión, simplemente se debe discar el número del otro usuario que este configurado en el servidor Asterisk, en caso de que no este disponible se puede dejar un buzón de voz, el cual el usuario al que se esta llamando puede rescatar una vez que se registre.

Se debe tener en cuenta que el administrador del servidor, esta capacitado para ver todo el registro de las llamadas, duración tiempo, número al que se llamo, etc., así como también escuchar y si es necesario, borrar los mensajes de voz de cada extensión, crear usuarios nuevos, eliminar los actuales, entre otros.

## 5. Procedimiento para realizar experiencia N° 4

### 5.1. Medición Subjetiva de Calidad de Servicio en Voz IP (QoS)

Para la medición de calidad de servicio, se recomienda la utilización del software Ethereal, también conocido como Wireshark, con este software es posible examinar (sniffear) los paquetes que se están transmitiendo en la red para su posterior análisis. En este caso en particular, se detallara como utilizar este software para poder determinar los parámetros como el Jitter, la Latencia y la pérdida de paquetes al realizar una llamada con señalización SIP en VoIP, con estos parámetros se logra establecer la calidad de servicio que se logra en la red de forma subjetiva, esto debido a que se utiliza la escala de medición MOS, ya descrita anteriormente. Además es posible visualizar el flujo de la señalización de la llamada SIP, para poder comprender aún mas el funcionamiento de esta tecnología.

### 5.2. Descripción del Software:

Ethereal es un analizador de protocolos de red para Unix y Windows, y es libre, es decir, no se paga una licencia para su utilización. Este software permite examinar datos de una red operativa o de un archivo de captura en algún disco. Se puede examinar interactivamente la información capturada, viendo información de detalles y sumarios por cada paquete. Ethereal tiene varias características poderosas, incluyendo un completo lenguaje para filtrar los paquetes que se deseen analizar y la habilidad de mostrar el flujo reconstruido de una sesión de TCP.

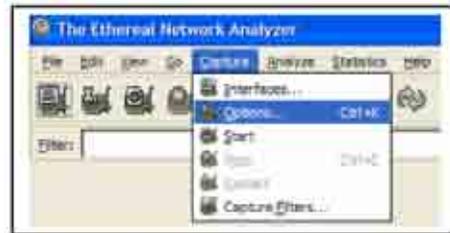
Para monitorear la calidad de servicio en voz sobre ip se utilizo:

Programa (Software):	Ethereal- Network Protocol Analyzer
Versión:	0.10.13
WinPcap version:	3.1 (packet.dll version 3, 1, 0, 27)
Basado en:	libpcap version 0.9[.x]
Distribución para:	Windows XP Service Pack 1, build 2600
Tipo de Software:	Open Source
Licencia:	GNU General Public License



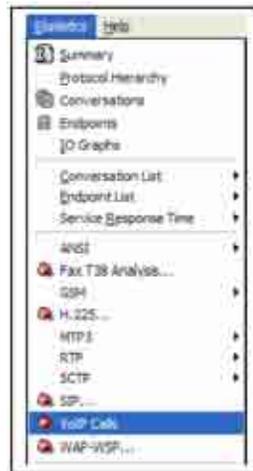
El primer paso que debe realizar para comenzar a utilizar Ethereal debe ser:

Dirigirse a **Capture**→**Options** y seleccione la interfaz correspondiente. Luego presione **Start** y en el momento que desee detener el monitoreo seleccione **Stop**.



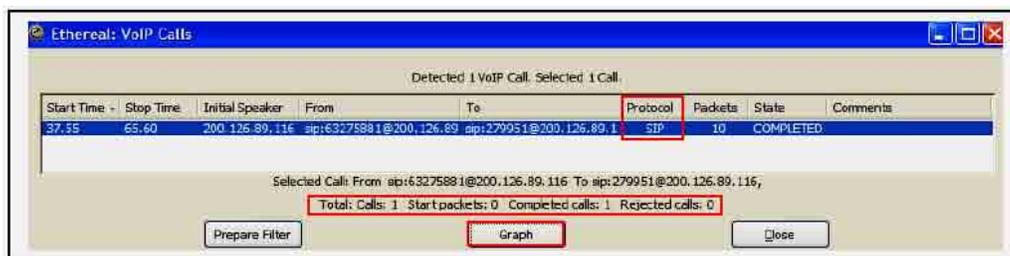
Pantalla Capture

Para acceder al análisis de llamadas VoIP debe seguir los siguientes pasos:  
**Statistics**→**VoIP Calls**



Pantalla Statistics

Donde se desplegará la siguiente pantalla:



Pantalla Voip Call

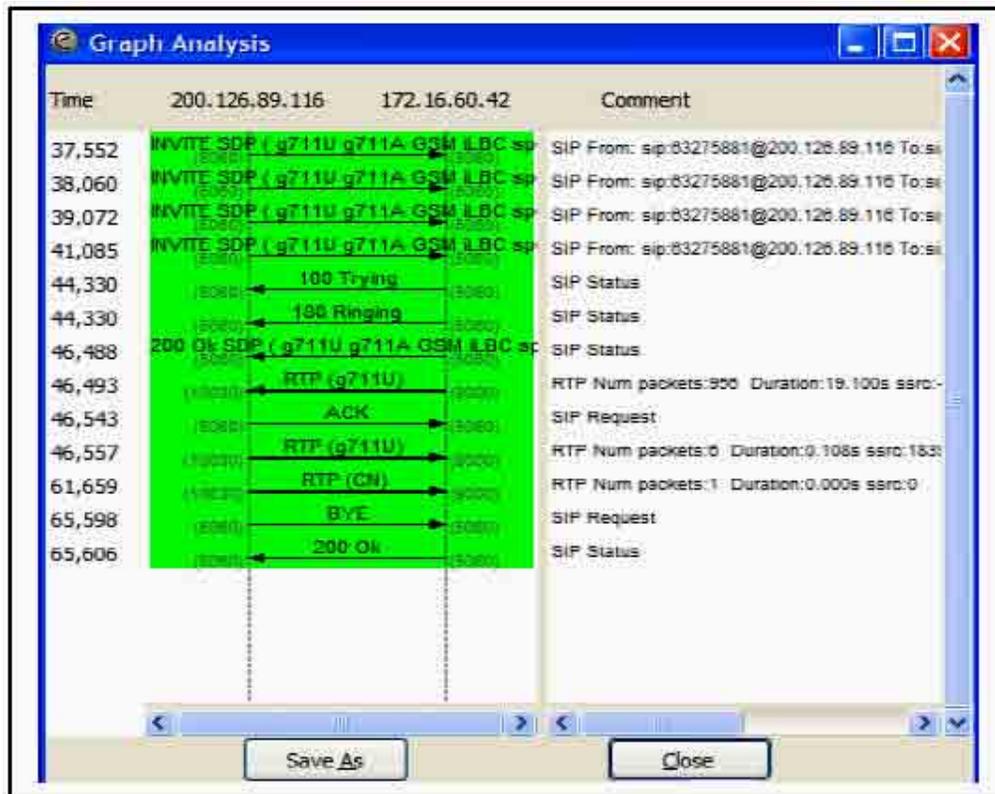
La lista de llamadas VoIP muestra la siguiente información por cada llamada:

- Start Time: Tiempo de inicio de la llamada.

- *Stop Time*: Tiempo de término de la llamada.
- *Initial Speaker*: La IP origen del paquete que inició la llamada.
- *From*: Para las llamadas H323 e ISUP corresponde al número que inició la llamada. Para llamadas SIP corresponde al campo "From" del INVITE. Para llamadas MGCP corresponde al EndpointID o numero llamante
- *To*: Para las llamadas H323 e ISUP corresponde al número que recibe la llamada. Para llamadas SIP corresponde al campo "To" del INVITE. Para llamadas MGCP corresponde al EndpointID o numero discado.
- *Protocol*: SIP, H323, ISUP o MGCP
- *Packets*: Número de paquetes involucrados en la llamada.
- *State*: El estado actual de la llamada. Los posibles valores son:
  - CALL SETUP: Llamada en estado setup (Setup, Proceeding, Progress o Alerting)
  - RINGING: llamada rengueando.
  - IN CALL: llamada está aun conectada.
  - CANCELLED: llamada fue liberada antes de conectarla.
  - COMPLETED: llamada fue completada y luego liberada.
  - REJECTED: llamada fue liberada antes de conectar por el destinatario.
  - UNKNOWN: se desconoce el estado de la llamada.

Para crear un filtro para una llamada en particular, sólo se debe seleccionar la llamada deseada (puede seleccionar más de una) y presionar en el botón **Prepare Filter**. Esto creará un filtro en la ventana principal de Ethereal para filtrar los paquetes relacionados con esta llamada.

Para poder acceder a un análisis más detallado acerca de la información de las llamadas, se debe seleccionar desde la lista la llamada que se desea analizar en detalle y luego presionar el botón **Graph**, deberá aparecer la siguiente imagen:



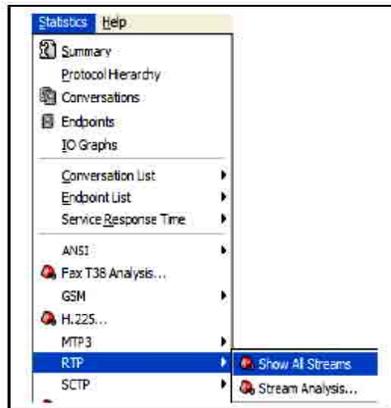
Pantalla Graph Analysis

El gráfico mostrará la siguiente información:

- Hasta diez columnas, cada una representando una dirección IP
- Todos los paquetes que pertenecen a la misma llamada se colorean con el mismo color.
- Una flecha mostrando la dirección de cada paquete en la llamada.
- Un label sobre cada flecha muestra el tipo de mensaje. Cuando esté disponible, también muestra el codec del medio.
- El tráfico RTP está resumido en una flecha más ancha con el correspondiente Codec.
- Muestra el puerto origen y destino UDP/TCP por paquete.

Para poder acceder a la información del RTP en detalle, se debe proceder de la siguiente manera:

Statistics→RTP→Show All Streams



Pantalla RTP

Se desplegará la siguiente pantalla, donde debe seleccionar de la lista la llamada a analizar y presionar el botón **Analyze**.

 A screenshot of the 'Ethereal: RTP Streams' dialog box. The title bar reads 'Ethereal: RTP Streams'. Below the title bar, it says 'Detected 3 RTP streams. Choose one for forward and reverse direction for analysis'. A table with 11 columns is displayed: Src IP addr, Src port, Dest IP addr, Dest port, SSRC, Payload, Packets, Lost, Max Delta (ms), Max Jitter (ms), and Mean. The table contains three rows of data. The 'Payload' column is highlighted in red in the original image.
 

Src IP addr	Src port	Dest IP addr	Dest port	SSRC	Payload	Packets	Lost	Max Delta (ms)	Max Jitter (ms)	Mean
172.16.60.42	8000	200.126.89.116	10030	2186531753	ITU-T G.711 PCMU	956	0 (0.0%)	35.44	3.60	
200.126.89.116	10030	172.16.60.42	8000	1839964803	ITU-T G.711 PCMU	6	0 (0.0%)	47.05	4.98	
200.126.89.116	10030	172.16.60.42	8000	0	Comfort noise	1	0 (0.0%)	0.00	0.00	

Pantalla RTP Streams

A través de ésta pantalla podrá realizar un mejor análisis del tráfico, visualizando el porcentaje de paquetes perdidos (*Lost %*), máx jitter [ms] y máx retardo [ms] (*Max Delta*). Luego de haber obtenido éstos parámetros se debe realizar una evaluación para especificar la calidad de Voz IP.

Se definen valores máximos de los siguientes parámetros Jitter, Perdida de paquetes y latencia, los cuales fueron clasificados como: Excelente, Bueno, Aceptable y Pobre.

Clasificación de QoS Subjetiva para VoIP (MOS)				
Parámetro	Excelente	Bueno	Regular	Pobre
Jitter [ms]	$t < 10$	$10 \leq t < 20$	$20 \leq t < 50$	$t \geq 5$
Latencia [ms]	$t < 50$	$50 \leq t < 150$	$150 \leq t < 300$	$t \geq 300$
% Pérdida paquetes	$p < 0.1$	$0.1 \leq p < 0.5$	$0.5 \leq p < 1.5$	$p \geq 1,5$

Calificación para metros Voip

*Umbral y Ponderaciones MOS:* Valores numéricos para cada categoría de Opinión de MOS y ponderaciones para el cálculo de MOS promedio.

Posteriormente para calificar los parámetros se debe evaluar mediante la recomendación E.800 del CCITT Calidad de Servicio, es el efecto conjunto del cumplimiento de un servicio, el cual determina el grado de satisfacción de un usuario de dicho servicio. La QoS se evalúa mediante el retardo (delay) y la disponibilidad de ancho de banda. Para tal efecto se sugiere realizar un reporte orientado a Medir Calidad de Voz IP, utilizando como unidad la escala MOS (Mean Opinion Score) la cual asigna un valor a la calidad de la llamada en toda la red. La medida tiene en cuenta tanto al codec como los efectos de la red.

Las marcaciones MOS tienen valores desde 1 (mala) a 5 (excelente). El valor de MOS real ha sido determinado en un ejercicio estadístico, un gran número de personas escuchando la misma llamada y valorándola de 1 a 5. Pueden darse datos MOS total o por llamadas y, normalmente, es la medida más utilizada, permitiendo visualizar rápida y fácilmente, para cualquier usuario no experto, los parámetros de Calidad de Voz IP.

Las Categorías de opinión MOS es subjetivo según la Calidad Voz IP individual de Jitter, Latencia y Pérdida de Paquetes, clasificándose en:

MOS	Jitter	Latencia	Perdida Paquetes
Excelente	Excelente	Excelente	Excelente
Bueno	Excelente	Bueno	Bueno
Bueno	Bueno	Excelente	Bueno
Bueno	Bueno	Bueno	Excelente
Aceptable	Bueno	Bueno	Bueno
Aceptable	Bueno	Bueno	Aceptable
Aceptable	Bueno	Aceptable	Bueno
Aceptable	Aceptable	Bueno	Bueno
Pobre	Aceptable	Bueno	Aceptable
Pobre	Aceptable	Aceptable	Bueno
Pobre	Bueno	Aceptable	Aceptable
Pobre	Aceptable	Aceptable	Aceptable
Pobre	Pobre	Aceptable	Aceptable
Pobre	Aceptable	Pobre	Aceptable
Pobre	Aceptable	Aceptable	Pobre
Malo	Pobre	Pobre	Aceptable
Malo	Pobre	Aceptable	Pobre
Malo	Aceptable	Pobre	Pobre
Inaceptable	Pobre	Pobre	Pobre

Categorías de opinión MOS.

## ANEXO III. COMANDOS DE LINUX

### Listado de Comandos Basicos en LINUX

A continuación se muestra un listado con los comandos básicos que se deben conocer al usar Linux y que se pueden ejecutar en cualquier shell (consola).

**cd directorio:** Permite el cambio entre directorios.

**cd .. :** Sube un directorio.

**cd :** Se posiciona en el directorio /home del usuario.

**cd - :** Se devuelve al directorio anterior.

**finger login:** Muestra información acerca del usuario por el cual se consulta.

**kill [Opción] PID:** Permite matar procesos.

**-9 :** Mata completamente el proceso deseado.

**find [DIR -iname] nombre archivo:** Busca desde el directorio DIR el nombre del archivo que se pide.

**ls [Opción]:** Muestra el contenido del directorio donde te encuentras.

**-a:** Se muestran todos los archivos del directorio, incluso los ocultos.

**-k:** Se muestra el tamaño de los archivos en KiloBytes.

**-l:** Se muestra toda la información sobre archivos y/o directorios.

**-x:** Se listan los ficheros en columnas, ordenados horizontalmente.

**-R:** Se listan los contenidos de todos los subdirectorios recursivamente.

**-X:** Se ordena el contenido del directorio alfabéticamente.

**-t:** Se ordena el contenido del directorio por fecha de modificación.

**mkdir [Opción] directorio:** Crea directorios.

**-m=codigo :** Establece los permisos (como en chmod).

**-v:** Muestra un mensaje por cada directorio creado.

**cp [Opción] archivo origen /destino:** Copia archivos.

**-p:** Preserva los atributos del archivo si es posible.

**-b:** Hace copias de seguridad de los archivos, en el caso que llegasen a hacer sobreescritos.

**-R:** Copia directorios recursivamente.

**mv [Opción] archivo origen /destino:** Mueve o renombra el archivo o directorio especificado.

**-b:** Hace copias de seguridad de los archivos que van a ser eliminados.

**-f:** Elimina archivos de destino ya existentes sin preguntar nada nunca al usuario.

**-v:** Muestra el nombre de cada archivo antes de moverlo.

**rm [Opción] archivo:** Borra o desenzala archivos

**-f :** No protesta de archivos que no existan y nunca pregunta nada al usuario.

**-i :** Pregunta antes de borrar un archivo o directorio.

**-r :** Borra recursivamente los contenidos de directorios.

**-v :** Muestra el nombre de cada archivo antes de borrarlo.

**rmdir directorio:** Borra directorio(s) vacío(s).

**cat [Opción] archivo:** Muestra el archivo archivo.

**-b :** Enumera las líneas que no están vacías.

**-n :** Enumera todas las líneas.

**man comando o programa:** Muestra el manual del comando, programa o instrucción especificado.

**pwd:** Muestra la ruta completa del directorio donde se estás trabajando.

**ps [Opción]:** Muestra los procesos que se estan llevando a cabo en el computador en es mismo instante.

**-e :** Muestra todos los procesos de los distintos usuarios.

**-f :** Muestra todos los detalles de los procesos que se estan ejecutando.

**-t :** Muestra los procesos por cada tty (ps -t tty1).

**-u :** Muestra los procesos por cada usuario (ps -u root).

**chmod [Opción] [Código] archivo o directorio:** Cambia los permisos del archivo o directorio.

**-c** : Informa los cambios que se hicieron.

**-v** : Muestra un mensaje por cada archivo procesado.

**-R** : Cambia los archivos o directorios recursivamente.

**Código:** Se compone de un número de 3 dígitos, que representan respectivamente los permisos para el usuario, el grupo y el resto. Cada uno puede ser la combinación (suma), de estos:

**1 : Read (r)**

**2 : Write (w)**

**4 : Execute (x)**

El comando **chmod** se usa para designar los permisos de un archivo. Sólo el *dueño del archivo* y el *root* pueden cambiar los permisos. La sintaxis de chmod es:

**chmod {a, u, g, o} {+, -} {r, w, x} nombre del archivo**

donde:

**u:** corresponde al dueño del archivo

**g:** corresponde al grupo

**o** o **a:** corresponde al resto de los usuarios, **a** para todos (all) y **o** para otros (others)

Para autorizar o desautorizar el permiso:

**+:** autoriza

**-:** desautoriza

**=:** resetea los permisos

Los tipos de permisos son:

**r:** lectura

**w**: escritura

**x**: ejecución

El comando **chmod** también acepta otros valores para cambiar los permisos. Es posible que se vea algo como:

```
sudo chmod 751 [nombre del archivo]
```

Es otro modo de gestionar los permisos; de forma binaria. El sistema es muy simple y cómodo: se considera un bit para lectura (**r**) otro para escritura (**w**) y otro para ejecución (**x**). Las combinaciones posibles son ocho y se muestran en la tabla siguiente:

Decimal	r w x
0	0 0 0
1	0 0 1
2	0 1 0
3	0 1 1
4	1 0 0
5	1 0 1
6	1 1 0
7	1 1 1

Un 1 equivale a activar y un 0 a desactivar los permisos. El equivalente decimal de los permisos se aplica en orden: *dueño, grupo y cualquiera* (**u,g,a**). Entonces en el ejemplo anterior el valor 751 actúa:

- 7, Cediendo todos los permisos al dueño.
- 5, Cediendo permiso de lectura y ejecución al grupo.
- 1, Cediendo permiso de ejecución a cualquiera.

**passwd [Opción] usuario:** Cambia tu password de acceso. Para cambiar la password de acceso solo basta colocar o ejecutar el comando "passwd".

**Otros Comandos útiles:**

**tar zxvf archivo:** Descomprime archivos .tgz o tar.gz

**tar cvzf archivo:** Crea un archivo .tar.gz

**tar tvzf archivo:** Permite listar los contenidos del tar sin extraerlos necesariamente.