

ESCUELA POLITÉCNICA DEL EJÉRCITO

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA EN ELECTRÓNICA Y
TELECOMUNICACIONES

PROYECTO DE GRADO PARA LA OBTENCIÓN
DEL TÍTULO DE INGENIERÍA

“Diseño e implementación de un emulador de central telefónica IP
utilizando el software de código abierto ASTERISK para la red de
datos de la Facultad de Ingeniería Electrónica de la Escuela
Politécnica del Ejército”

José Luis Vaca Parra

SANGOLQUI – ECUADOR

2008

CERTIFICACIÓN

Certificamos que el presente proyecto titulado “Diseño e implementación de un emulador de central telefónica IP utilizando el software de código abierto ASTERISK para la red de datos de la Facultad de Ingeniería Electrónica de la Escuela Politécnica del Ejército” ha sido desarrollado en su totalidad por el Sr. José Luis Vaca Parra con cédula de ciudadanía No. 060218870-8, bajo nuestra dirección.

Sr. Ing. M.Sc. Gonzalo Olmedo
DIRECTOR

Sr. Ing. Julio Larco
CODIRECTOR

RESUMEN DEL PROYECTO

La importancia y desarrollo de las telecomunicaciones han permitido la aparición de nuevas tendencias tecnológicas en el transporte y administración de las conversaciones telefónicas. Se han adicionado muchos beneficios y servicios, brindando sistemas de comunicación integrales. Todos estos beneficios y nuevas tecnologías de transporte de información, están contenidos en la actualidad dentro de pequeñas centrales de telefonía, brindando iguales y mayores ventajas que las grandes y ya conocidas centrales telefónicas. En el presente proyecto se realizó un estudio e implementación de un sistema de telefonía IP basado en el software de libre distribución llamado Asterisk.

La investigación inicia con un estudio de la tecnología, identificando los elementos que conforman los sistemas de telefonía IP y analizando los factores que influyen en el desempeño de la red. A continuación se realizó un análisis del marco legal actual vigente en el Ecuador para este tipo de tecnologías y se analizó las perspectivas de regulación.

También se muestra la variedad de aplicaciones e innumerables ventajas de los sistemas de telefonía IP basados en Asterisk, a continuación se detalla la implementación del sistema y el análisis de la configuración. Además se realizó una evaluación al sistema instalado, realizando pruebas a los servicios y mostrando el detalle de llamadas establecidas por el sistema.

Finalmente se establecen las respectivas conclusiones y recomendaciones que se obtuvieron como experiencia de esta investigación.

DEDICATORIA

A directivos, profesores y alumnos que forman parte del Departamento de Eléctrica y Electrónica de la Escuela Politécnica del Ejército.

AGRADECIMIENTO

Mi agradecimiento a todas las personas que han aportado en mi formación personal y profesional, de manera infinita al amor y cariño de mis padres, mis hermanos y mi familia.

De igual modo, mi eterno agradecimiento a mis profesores, educadores y orientadores que a lo largo de los años han sabido formar mi carácter y actitud.

A mis compañeros y amigos, por ese convivir diario que aún se mantiene perenne en mis pensamientos.

PRÓLOGO

Voz sobre Protocolo de Internet, también llamado Voz sobre IP, VoIP, VoIP (por sus siglas en inglés), o Telefonía IP, es un grupo de recursos que hacen posible que la señal de voz viaje a través de Internet empleando un protocolo IP (*Internet Protocol*). Esto significa que se envía la señal de voz en forma digital en paquetes en lugar de enviarla a través de circuitos utilizables solo para telefonía.

La transmisión de voz en forma de datos es una tecnología que las empresas están dispuestas a adoptar debido al ahorro que representa. La VoIP es la transmisión de tráfico de voz de dos vías sobre una red IP, permitiendo combinar transmisión de voz con comunicación de datos en una plataforma IP única e integrada. De esta forma las llamadas telefónicas corren sobre la misma red de computadoras en lugar de utilizar líneas telefónicas tradicionales, lo cual ofrece muchos beneficios a los proveedores de servicios, así como a los usuarios finales.

Los servicios de telefonía IP suponen un gran avance en las comunicaciones al conseguir varios objetivos, tales como: reducción de costos de telefonía, nivel de gestión, incorporación de nuevos servicios como videoconferencia, correo de voz, identificador de llamadas, entre otros y fundamentalmente el aspecto más interesante: el gran ahorro económico que representa utilizar una sola red para transmisión de voz y datos.

Asterisk es una aplicación software libre de una central telefónica (PBX). Como cualquier PBX, se puede conectar un número determinado de teléfonos para hacer llamadas entre sí e incluso conectar a un proveedor de VoIP o bien a una red digital se servicios integrados (RDSI). La utilización de una herramienta como Asterisk de código abierto y de libre distribución permite la libertad de uso, estudio y adaptación. También brinda la posibilidad de manipular y mejorar el código fuente y adoptar a las necesidades específicas de seguridad y funcionalidad. Además existe una libertad de copia, sin costo de licencias.

El estudio y las experiencias obtenidas con un software de código abierto basados en plataformas Linux como lo es Asterisk, fomentarán futuras investigaciones, prácticas de laboratorio y proyectos similares en el departamento de ingeniería eléctrica y electrónica. La implementación del emulador de una central telefónica IP se convertirá en una opción de comunicación eficiente que resuelva las necesidades internas de telefonía además que permitirá la incorporación de nuevas funcionalidades a través de herramientas y tecnología de vanguardia.

ÍNDICE DE CONTENIDO

CAPÍTULO 1: INTRODUCCIÓN

1.1. Introducción a la telefonía IP	1
1.1.1. Características del protocolo IP	1
1.1.2. Operación del protocolo IP	2
1.1.2.1. Datagramas IP	3
1.1.2.2. Cabecera	3
1.1.2.3. Direccionamiento IP	8
1.1.3. Ventajas de migrar a telefonía IP	10
1.1.3.1. Ventajas competitivas	10
1.1.3.2. Retorno de la Inversión	10
1.1.4. Servicios adicionales	11
1.2. Características del sistema de telefonía IP	11
1.2.1. Componentes del sistema	12
1.2.2. Escenario de redes de telefonía IP	14
1.2.2.1. Telefonía IP – Telefonía IP	14
1.2.2.2. Telefonía IP – Telefonía Convencional	14
1.2.2.3. Telefonía Convencional - Telefonía Convencional	15
1.2.3. Protocolos VoIP	16
1.2.3.1. Protocolo H323	17
1.2.3.2. Protocolo SIP	18
1.2.3.3. Protocolo IAX	19
1.2.3.4. Protocolo MGCP	19
1.2.3.5. Protocolos Propietarios	19
1.2.4. Codificadores Decodificadores de voz (Codecs)	20
1.2.4.1. Codec G.711	21
1.2.4.2. Codec G.726	22
1.2.4.3. Codec G.723.1	22
1.2.4.4. Codec G.729.A	22
1.2.4.5. Codec GSM	23

1.2.4.6. Codec iLBC	23
1.2.4.7. Codec Speex	23
1.2.4.8. Codec MP3	24
1.2.5. Calidad de Servicio QoS	24
1.2.5.1. Latencia	25
1.2.5.2. Jitter	26
1.2.5.3. Pérdida de paquetes	27
1.2.5.4. Eco	25

CAPÍTULO 2: MARCO REGULATORIO DE LA TELEFONÍA IP

2.1. Introducción	29
2.2. Marco regulatorio para la telefonía IP en Ecuador	30
2.3. Perspectivas de regulación de la Telefonía IP y VoIP.	31
2.3.1. Desregulación en Servicios	35

CAPÍTULO 3: CARACTERÍSTICAS DEL SOFTWARE ASTERISK

3.1. Introducción	37
3.2. Funcionalidades generales	37
3.3. Funcionalidades de central básica	39
3.4. Funcionalidades de central avanzada	44
3.4.1. Buzón de Voz	44
3.4.2. Conferencia de Voz	44
3.4.3. Música en espera	45
3.4.4. Salas de conferencias de llamadas	45
3.4.5. Interoperabilidad con bases de datos	45

CAPÍTULO 4: DISEÑO DE LA RED DE TELEFONÍA IP

4.1. Configuración del Servidor	46
4.1.1. Características del hardware.....	46
4.1.2. Instalación del sistema operativo	47
4.1.3. Instalación de Asterisk	48
4.1.3.1. Compilando Zaptel.....	50
4.1.3.2. Compilando Libpri.....	51
4.1.3.3. Compilando Asterisk.....	51
4.1.3.4. Compilando mpg123.....	52
4.1.3.5. Sonidos en español	52
4.1.3.6. Carga del modulo zaptel.....	53
4.1.3.7. Carga del modulo ztdummy	53
4.1.4. Configuración básica de Asterisk	54
4.1.4.1. Configuración de Hardware Zaptel (zaptel.conf).....	57
4.1.4.2. Configuración de Hardware Zapata (zapata.conf)	58
4.1.4.3. Configuración SIP (sip.conf)	58
4.1.4.4. Configuración IAX (iax.conf)	60
4.1.4.5. Plan de Mercado - Dialplan (extension.conf)	61
4.1.5. Funcionalidades de central avanzada	65
4.1.5.1. Configuración del buzón de voz.....	65
4.1.5.2. Configuración de música en espera	67
4.1.5.3. Configuración de salas de conferencias	69
4.1.5.4. Interoperabilidad con bases de datos	70
4.1.5.5. Tarifación de llamadas.....	71
4.2. Configuración de terminales.....	73
4.2.1. Tipos de terminales	74
4.2.1.1. Equipos dedicados	74
4.2.1.2. Softphones	77
4.2.2. Configuración de Softphone X-Lite.....	78
4.2.3. Configuración de Softphone SJ Phone	81

CAPÍTULO 5: EVALUACIÓN DEL SISTEMA

5.1. Introducción.....	85
5.2. Análisis del desempeño de la red.....	85
5.3. Evaluación del sistema desarrollado	88

CAPÍTULO 6: CONCLUSIONES Y RECOMENDACIONES

6.1. Conclusiones.....	93
6.2. Recomendaciones.....	96

BIBLIOGRAFÍA	97
---------------------------	-----------

ANEXOS:

Anexo 1: Plan de Mercado	98
Anexo 2: Comandos Consola de Asterisk	100
Anexo 3: Referencia de Aplicaciones.....	106

ÍNDICE DE TABLAS

CAPÍTULO 1: INTRODUCCIÓN

Tabla 1.1: Características de Codecs	41
--	----

CAPÍTULO 4: DISEÑO DE LA RED DE TELEFONÍA IP

Tabla 4.1: Requisitos generales del sistema	47
Tabla 4.2: Archivos de configuración de Asterisk.....	54
Tabla 4.2: Características Softphones	76

CAPÍTULO 5: EVALUACIÓN DEL SISTEMA

Tabla 5.1: Cuadro comparativo de codecs soportados por Asterisk	86
Tabla 5.2: Características del servidor Asterisk	88
Tabla 5.3: Medición de tráfico generado por el sistema	90
Tabla 5.3: Log de Softphone X-Lite al probar el sistema.....	92

.

ÍNDICE DE FIGURAS

CAPÍTULO 1: INTRODUCCIÓN

Figura 1.1: Modelo de red TCP/IP y protocolos principales	2
Figura 1.2: Estructura de un datagrama IP	4
Figura 1.3: Direccionamiento IP	9
Figura 1.4: Componentes del sistema.....	13
Figura 1.5: Escenario de Telefonía IP – Telefonía IP.....	14
Figura 1.6: Escenario de Telefonía IP – Telefonía Convencional.....	15
Figura 1.7: Telefonía Convencional – Telefonía Convencional	15
Figura 1.8: Pila de protocolos RTP.....	16
Figura 1.9: Modelo de red TCP/IP & OSI y protocolos principales	17

CAPÍTULO 3: CARACTERÍSTICAS DEL SOFTWARE ASTERISK

Figura 3.1: Logotipo de Asterisk PBX.....	38
Figura 3.1: Características de una central telefónica Asterisk.....	39

CAPÍTULO 4: DISEÑO DE LA RED DE TELEFONÍA IP

Figura 4.1: Sistema de telefonía IP	73
Figura 4.2: Terminal IP Cisco 7971G-GE	74
Figura 4.3. Terminal IP Avaya 4620	75
Figura 4.4. Terminal 3Com 3102 Business Phone	75
Figura 4.5. Terminal IP Siemens optiPoint 400 Advanced	76
Figura 4.6. Apariencia del Softphone X-Lite 3.0.....	77
Figura 4.7. Pantalla para acceder a configuración de cuentas SIP en X-Lite.....	78
Figura 4.8. Pantalla de configuración de cuentas SIP en X-Lite.....	78

Figura 4.9. Configuración de una cuenta SIP en X-Lite	79
Figura 4.10. Aspecto de X-Lite con una cuenta SIP registrada y activa	79
Figura 4.11. Aspecto del Softphone SJphone desarrollado por SoftJoys Labs ...	80
Figura 4.12. Configuración de cuentas SIP en SJphone	81
Figura 4.13. Configuración del proveedor de servicio en SJphone	81
Figura 4.14. Configuración de cuentas SIP en SJphone	82
Figura 4.15. Aspecto del Softphone SJphone con un registro satisfactorio	82

CAPÍTULO 5: EVALUACIÓN DEL SISTEMA

Figura 5.1: Gráfico comparativo de codecs soportados por Asterisk	87
Figura 5.2: Consola de Asterisk	88
Figura 5.3: Aspecto de FreePBX instalado en Asterisk	89
Figura 5.4: Tráfico generado por una llamada establecida en el sistema	90
Figura 5.5: Reporte de detalle de registro de llamadas (CDR)	91

GLOSARIO

AAA (*Authentication, Authorization, and Accounting*)

Son los tres pasos fundamentales en la seguridad de datos en informática: Autenticación, Autorización y Auditoría.

ADPCM (*Adaptive Differential Pulse Code Modulation*)

Código de modulación diferencial adaptativo de pulsos o pulsaciones, una forma de código de modulación de pulsaciones que produce una señal digital con un bit rate más bajo que el estándar

ADSL (*Asymmetric Digital Subscriber Line*)

Método de transmisión de datos a través de la línea telefónica de cobre tradicional a velocidad alta. Los datos pueden ser descargados a distinta velocidad que son cargados, esta es la razón por la cual se le denomina asimétrico.

AGI (*Asterisk Gateway Interface*)

Interfaz en el cual interactúa la central Asterisk con scripts creados por un usuario en lenguajes de programación como: Perl, PHP, C, Pascal, Bourne Shell; los cuales agregan funcionalidades al sistema Asterisk.

AH (*Authentication Header*)

Proporciona servicios de autenticación, aunque no de cifrado. Su función es asegurar la compatibilidad con homólogos IPSec que no admiten ESP, que suministra tanto autenticación como cifrado.

ANI (*Automatic Number Identification*)

Es una característica de los servicios de red de telefonía inteligente que permite a los abonados capturar o mostrar el número de teléfono de un grupo de llamadas.

ARP (*Address Resolve Protocol*)

Protocolo de resolución de direcciones, es un protocolo de nivel de red responsable de encontrar la dirección hardware (Ethernet MAC) que corresponde a una determinada dirección IP.

AS (*Autonomous System*)

Conjunto de redes sobre la cual existe una autoridad administrativa.

ATM (*Asynchronous Transfer Mode*)

Es una tecnología de telecomunicación desarrollada para hacer frente a la gran demanda de capacidad de transmisión para servicios y aplicaciones.

BGP (*Border Gateway Protocol*)

Es un protocolo mediante el cual se intercambian prefijos los ISPs registrados en Internet, en la actualidad la mayoría de los ISPs intercambian sus tablas de rutas con este protocolo.

BRI (*Basic Rate Interface*)

Interfaz de tasa básica (BRI, 2B+D, 2B+1D) de una Red Digital de Servicios Integrados (RDSI). Esta configuración se compone de dos canales de 64 kbps al portador (canales B) y uno de 16 kbps para el canal de datos, (canal D). Los canales B se utilizan para la transmisión de voz o datos, y el canal D para cualquier combinación de: datos, control/señalización y paquetes X.25.

CE (*Customer Edge*)

Router frontera del cliente que solicita el servicio VPN con MPLS.

CEF (*Cisco Express Forwarding*)

Es una técnica propietaria de Cisco para realizar switcheo a altas velocidades en WANs con garantía de velocidad y sin retraso.

CHAP (*Challenge Handshake Authentication Protocol*)

Es un método de autenticación remota o inalámbrica. Diversos proveedores de servidores y clientes de acceso a la red emplean CHAP.

CIDR (*Classless Inter-Domain Routing*)

Método de asignar y especificar direcciones Internet utilizados en enrutadores interdominios con mayor flexibilidad que el sistema original de clases de direcciones del protocolo Internet. Como resultado se ha ampliado en gran medida el número de direcciones Internet disponibles.

CPE (*Customer Premise Equipment*)

Es el conjunto de equipos instalado en los predios de los clientes y que permiten su conexión con las estaciones base de la red.

CR-LDP (*Constraint-Based Routing Label Distribution Protocol*)

Protocolo de distribución de etiquetas que contiene extensiones del LDP para extender sus capacidades. Puede trabajar con explicit route constraints, QoS constraints, entre otras.

CsC (*Carrier's Carrier*)

Servicio backbone a proveedores de servicio.

CTI (*Computer Telephony Integration*)

Tecnología que permite la interacción entre un teléfono y un ordenador para ser integrados y coordinados.

DD (*Database Description*)

Descripción de base de datos.

DID (*Direct Inward Dialing Number*)

También Conocido como DDI, la marcación directa permite que una llamada entrante acceda a la extensión deseada sin necesidad de intervención de operadora.

DiffServ (*Differentiated Services*)

Se utiliza para establecer diferentes tipos de servicio a diferentes usuarios.

DISA (*Direct Inward System Access*)

Características de centrales telefónicas que permite a un usuario externo de la central, obtener tono de llamada interno y actuar con los privilegios de la central telefónica.

DNS (*Domain Name Server*)

Es una base de datos distribuida y jerárquica que almacena información asociada a nombres de dominio en redes como Internet.

DoS (*Denial of Service*)

Es un incidente en el cual un usuario o una organización se ven privados de un recurso que normalmente podrían usar. Habitualmente, la pérdida del servicio supone la indisponibilidad de un determinado servicio de red, como el correo electrónico, o la pérdida temporal de toda la conectividad y todos los servicios de red.

DSL (*Digital Subscriber Line*)

Tecnología de transmisión que permite que los hilos telefónicos de cobre convencionales transporten hasta 16 Mbps (megabits por segundo) mediante técnicas de compresión.

DUNDi (*Distributed Universal Number Discovery*)

Servicio de Descubrimiento de Numero Universal Distribuido, una completa red P2P (Peer to Peer) sin ningún tipo de servidor en medio, diseñada para eliminar la dependencia de una central y sus equipos de comunicaciones, logrando una red de comunicaciones virtualmente gratuita.

ECN (*Explicit Congestion Notification*)

Regula dinámicamente las sesiones de tiempo real admitidas en una red y trata por tanto de reestablecerlas en el caso de que sea necesaria.

ENUM

(RFC 3761) Es un protocolo que utiliza el sistema DNS de Internet para traducir números de teléfono E.164 a esquemas de dirección de IP (como las de SIP, H323 o email).

ENUM (*Electronic Numbering*)

Es un protocolo que utiliza el sistema DNS de Internet para traducir números de teléfono E.164 a esquemas de dirección de IP. ENUM esta soportado en Proxys SIP como SER o SNOM 4S, gateways como Asterisk, Swyx y teléfonos SIP.

ESP (*Encapsulating Security Payload*)

Es un encabezado de red diseñado para proveer los servicios de seguridad de IPv4.

FEC (*Forwarding Equivalence Classes*)

Nombre que se le da al tráfico que se reenvía bajo una etiqueta. Subconjunto de paquetes tratados del mismo modo por el conmutador.

FTP (*File Transfer Protocol*)

Protocolo de transferencia de archivos que permite transmitir ficheros sobre Internet entre una máquina local y otra remota.

FXO (*Foreign eXchange Office*)

Oficina de intercambio remota, es un interfaz que conecta a la línea telefónica. Proporcionan a una centralita acceso a la red de telefonía pública.

FXS (*Foreign eXchange Station*)

Estación de intercambio remota, es un interfaz que controla al teléfonos. A los interfaces FXS se conectan los teléfonos, proporcionan carga y tono de timbre.

GRE (*Generic Routing Encapsulation*)

Protocolo de conexión virtual mediante túneles.

H.323

Es un conjunto de normas ITU para comunicaciones multimedia que hacen referencia a los terminales, equipos y servicios estableciendo una señalización en redes IP. Es utilizado comúnmente para Voz sobre IP (VoIP, Telefonía IP) y para videoconferencia basada en IP.

HTTP (*Hypertext Transfer Protocol*)

Protocolo base de la Web.y que ofrece un conjunto de instrucciones para que los servidores y navegadores funcionen. Es el lenguaje usado para escribir documentos para servidores World Wide Web.

IAX (*Inter-Asterisk Exchange Protocol*)

Protocolo de código abierto para el transporte de voz en tiempo real utilizando UDP.

ICMP (*Internet Control Message Protocol*)

Es usado principalmente por los sistemas operativos de las computadoras en una red para enviar mensajes de error, indicando por ejemplo que un servicio determinado no está disponible o que un router o un host no puede ser localizado.

IETF (*Internet Engineering Task Force*)

Es una organización internacional abierta de normalización, que tiene como objetivos el contribuir a la Ingeniería de Internet. Se divide en áreas como: transporte, encaminamiento, seguridad, etc.

IGP (*Internet Gateway Protocol*)

Hace referencia a los protocolos usados dentro de un sistema autónomo. Los protocolos IGP más utilizados son RIP y OSPF.

IKE (*Internet Key Exchange*)

Se utiliza para establecer una SA en el protocolo IPSec.

IntServ (*Integrated Services*)

Modelo de servicios diferenciados, realiza una reserva previa de recursos antes de establecer la comunicación.

IP (*Internet Protocol*)

Base del conjunto de protocolos que forman Internet y que permite que los paquetes de información sean direccionados y enrutados.

IPSec (*IP Security*)

Es una extensión al protocolo IP que añade cifrado fuerte para permitir servicios de autenticación y cifrado, y de esta manera asegurar las comunicaciones a través de dicho protocolo.

ISP (*Internet Service Provider*)

Organización que provee la conexión de ordenadores a Internet, ya sea por líneas dedicadas o por líneas conmutadas.

ITU (*Internacional Telecommunication Union*)

Unión Internacional de las Telecomunicaciones, es el organismo especializado de las Naciones Unidas encargado de regular las telecomunicaciones, a nivel internacional, entre las distintas administraciones y empresas operadoras.

IVR (*Interactive Voice Response*)

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

L2TP (*Layer 2 Tunneling Protocol*)

Fue diseñado por el IETF para corregir las diferencias de PPTP y establecerse como un estándar aprobado por el IETF. Utiliza PPP para proporcionar acceso telefónico que puede ser dirigido a través de un túnel por Internet hasta un punto determinado.

LAC (*L2TP Access Control*)

Dispositivo que termina las llamadas a sistemas remotos y sesiones de túnel PPP entre sistemas remotos y LNS.

LAN (*Local Area Network*)

Red de datos para dar servicio a un área geográfica máxima de unos pocos kilómetros cuadrados, por lo cual pueden optimizarse los protocolos de señal de la red para llegar a velocidades de transmisión de hasta 100 Mbps.

LCP (*Link Control Protocol*)

Protocolo de Control de Enlace.

LDP (*Label Distribution Protocol*)

Protocolo de distribución de etiquetas, utilizado principalmente en MPLS.

LER (*Label Edge Router*)

Elemento que inicia o termina el túnel (pone y quita encabezados), es decir el elemento de entrada/salida a la red MPLS. Un router de entrada se conoce como Ingress Router y uno de salida como Egress Router.

LFT (*Label Forwarding Table*)

Tabla de Reenvío de Etiquetas, la cual usa el componente de datos para reenviar los paquetes etiquetados a través de la red MPLS.

LNS (*L2TP Network Server*)

Dispositivo que puede terminar túneles L2TP desde un LAC y sesiones PPP a sistemas remotos mediante sesiones de datos L2TP.

LSA (*Link State Advertisement*)

Paquetes OSPF que contienen información del estado, métrica y otros de las interfaces del router.

LSP (*Label Switched Path*)

Nombre genérico de un camino MPLS (para cierto FEC), es decir, del túnel MPLS establecido en los extremos.

LSR (*Label Switch Router*)

Elemento que conmuta etiquetas.

MAC (*Media Access Control*)

Número exclusivo asignado por el fabricante a cada computadora u otro dispositivo de una red.

MGCP (*Media Gateway Control Protocol*)

Protocolo diseñado para transportar los datos de voz en tiempo real.

MPLS (*Multi-Protocol Label Switching*)

Es un mecanismo de transporte de datos estándar creado por la IETF. Opera entre la capa de enlace de datos y la capa de red del modelo OSI. Fue diseñado para unificar el servicio de transporte de datos para las redes basadas en circuitos y basadas en paquetes.

NAT (*Network Address Translation*)

Estándar creado por la IETF el cual utiliza una o más direcciones IP para conectar varios computadores a otra red, por lo general Internet.

NCP (*Network Control Protocol*)

Protocolo de Control de Red. Es un protocolo del Network Layer.

NetBEUI (*NetBios Extended User Interface*)

Es un protocolo de nivel de red sencillo utilizado en las primeras redes Microsoft. La comunicación entre equipos se consigue gracias al intercambio de sus nombres en una red de área local, es un protocolo sin encaminamiento por lo que no dispone de mecanismos para conectar equipos que estén en redes separadas.

NOC (*Network Operations Center*)

Representa al centro de operaciones en el cual se realiza el monitoreo y resolución de fallas en una red de servicios.

NSP (*Network Service Provider*)

Proveedores de servicio IP/Internet.

OSI (*Open System Interconnection*)

Modelo definido por la Organización Internacional de Normalización, es uno de los más importantes y el más utilizado. Se divide en 7 capas sucesivas: Física, Enlace de Datos, Red, Transporte, Sesión, Presentación y Aplicación.

OSP (*Open Settlement Protocol*)

Protocolo estándar que se ha diseñado para la transferencia de datos para sistemas facturación entre operadores.

OSPF (*Open Shortest Path First*)

Propone el uso de rutas más cortas y accesibles mediante la construcción de un mapa de la red con información sobre sistemas locales y vecinos, esto con el fin de conocer la ruta más corta.

OSS (*Open Sound Systems*)

Interfaz estándar para capturar sonidos en sistemas operativos Unix.

PAP (*Password Authentication Protocol*)

Método básico de autenticación, en el cual el nombre de usuario y la contraseña (clave) se transmiten a través de una red y se compara con una tabla de parejas nombre-clave, la no coincidencia provocará la desconexión. Típicamente, las contraseñas almacenadas en la tabla se encuentran encriptadas.

PCM (*Pulse Code Modulation*)

Mecanismo de modulación digital de la voz.

PDU (*Protocol Data Unit*)

Es la unidad de datos de protocolo de la capa de red. Las PDU tienen encapsuladas en su área de datos otras PDU.

PE (*Provider Edge*)

Router frontera del proveedor del servicio VPN con MPLS.

PERL (*Practical Extraction and Report Language*)

Lenguaje de programación diseñado por Larry Wall creado en 1987. Perl toma características del C, del lenguaje interpretado shell (sh), AWK, sed, Lisp y, en un grado inferior, muchos otros lenguajes de programación.

PoP (*Point Of Presence*)

Punto geográfico, especialmente una ciudad, desde donde un Proveedor de Servicios Internet ofrece acceso a la red Internet.

PPP (*Point to Point Protocol*)

El protocolo PPP permite establecer una comunicación a nivel de enlace entre dos computadoras. Generalmente se utiliza para establecer la conexión a Internet de un particular con su proveedor de acceso a través de un modem telefónico.

PPTP (*Point-to-Point Tunneling Protocol*)

Es un protocolo para implementar redes privadas virtuales desarrollado por Microsoft, Ascend Communications.

PRI (*Primary Rate Interface*)

La interfaz de velocidad primaria (PRI) es un estándar de telecomunicaciones para el transporte de múltiples DS0 (*Data Signal 0*) de voz y transmisiones de datos entre dos ubicaciones físicas.

PSTN (*Public Switched Telephone Network*)

La Red Telefónica Conmutada (también llamada Red Telefónica Básica o RTB) es una red de comunicación diseñada primordialmente para la transmisión de voz, aunque pueda también transportar datos, por ejemplo en el caso del fax o de la conexión a Internet a través de un módem acústico.

PTO (*Public telephony Operator*)

Operador de telefonía pública. Empresas u organismos que administran las empresas de telefonía.

PVC (*Permanent Virtual Circuit*)

Conexión que reemplaza las líneas privadas por un solo enlace en la red.

QoS (*Quality Of Service*)

Es la capacidad de las tecnologías de conmutación de paquetes de proveer todos los recursos necesarios a cada aplicación en un momento determinado dentro de la red.

RARP (*Reverse Address Resolve Protocol*)

Protocolo de resolución de direcciones inverso, es un protocolo utilizado para resolver la dirección IP de una dirección hardware dada.

RAS (*Remote Access Server*)

Servidor dedicado a la gestión de usuarios que no están en una red pero necesitan acceder remotamente a ésta. Permite a los usuarios, una vez autenticados, obtener acceso a los archivos y servicios de impresora de una LAN desde una localización remota.

RD (*Route Distinguisher*)

Es un clasificador de direcciones utilizado por MPLS para distinguir rutas VPN de diferentes clientes que se conectan al proveedor de servicio.

ISDN (*Integrated Services Data Network*)

La Red Digital de Servicios Integrados es una red que procede de la Red Digital Integrada (RDI) y que facilita las conexiones digitales extremo a extremo con el fin de proporcionar una amplia gama de servicios, tanto de voz como de otros tipos.

RFC (*Request for Comments*)

Conjunto de notas técnicas y organizativas donde se describen los estándares de Internet.

RSVP (*Resource Reservation Protocol*)

Protocolo que se encarga de reservar recursos para que estén disponibles cuando las aplicaciones lo necesiten.

RT (*Route Target*)

Identifica las VFR en que se instalan las rutas. Permite el control y manejo de prefijos VPNv4.

SA (*Security Associations*)

Describe un flujo unidireccional seguro de datos a través de dos puertas de enlace. Se utiliza en IPsec con IKE.

SAD (*Security Association Database*)

Base de datos donde se almacena todos los parámetros concernientes a las SAs, cada una de ellas tiene una entrada donde se especifican todos los parámetros necesarios para que IPsec realice el procesamiento de paquetes IP.

SCCP (*Cisco Skinny Client Control Protocol*)

Protocolo propietario de equipos VoIP de Cisco, este es un protocolo predeterminado para equipos terminales de sistemas de PBX de Cisco.

SDP (*Session Description Protocol*)

Protocolo diseñado para la negociación de las capacidades de los participantes en una comunicación de dispositivos multimedia como tipo de codificación entre otros aspectos.

SIP (*Session Initiation Protocol*)

Protocolo diseñado para transportar los datos de voz en tiempo real.

SMS (*Short Message Service*)

Servicio de mensajes cortos, servicio disponible en los sistemas telefónicos que permite el envío de mensajes cortos (también conocidos como mensajes de texto) entre teléfonos móviles, teléfonos fijos y otros dispositivos de mano.

SMTP (*Simple Mail Transfer Protocol*)

Protocolo de envío de correo electrónico.

SP (*Service Provider*)

Proveedor de servicio, empresa que proporciona el servicio de acceso a la red.

SPD (*Security Policy Database*)

Base de datos donde existe una lista ordenada de políticas de seguridad a ser aplicadas a los paquetes IP. Dichas políticas son en general reglas que especifican como los paquetes IP deben ser procesados.

SSL (*Secure Sockets Layer*)

Es un protocolo criptográfico que proporciona comunicaciones seguras en Internet. Proporciona autenticación y privacidad de la información entre extremos sobre Internet mediante el uso de criptografía.

SVC (*Switched Virtual Circuit*)

Conexión virtual entre dos extremos variables de la red. El conmutador establece la conexión al comienzo de la llamada y la interrumpe al finalizar esta.

TCP (*Transmission Control Protocol*)

Protocolo en los que se basa buena parte de Internet. Se encarga de dividir la información en paquetes en origen, para luego recomponerla en destino.

TFTP (*Trivial File Transfer Protocol*)

Se trata de un sencillo protocolo utilizado para transferir archivos. Se ejecuta en UDP.

ToS (*Type of Service*)

Campo dentro de un datagrama IP que indica el modo en que se debe manejar el datagrama.

TTS (*Text To Speech*)

Es la producción artificial de habla humana a través de un sistema informático utilizado para este propósito (sintetizador de voz), puede ser implementado en software o hardware.

UDP (*User Datagram Protocol*)

Es un protocolo que opera en el nivel de transporte del modelo OSI que se basa en intercambio de datagramas. Permite enviar los datagramas sin establecer previamente una conexión ya que su encabezado contiene suficiente información de direccionamiento.

VC (*Virtual Circuit*)

Conexiones virtuales que utiliza ATM para reemplazar las líneas privadas por un solo enlace de red, puede ser permanentes o conmutadas.

VLL (*Virtual Leased Lines*)

Enlaces punto a punto orientados a conexión entre los sitios de los clientes.

VoIP (*Voice Over IP*)

Voz sobre IP, transmisión de la voz utilizando técnicas de conmutación por paquetes basadas en el protocolo IP.

VPDN (*Virtual Private Dial Network*)

Enlaces que permiten a los clientes que el SP le aprovisiona y gestiona los accesos conmutados a su red.

VPLS (*Virtual Private LAN Segments*)

Es una forma de proveer comunicación multipunto a multipunto basado en Ethernet para redes IP/MPLS.

VPN (*Virtual Private Network*)

Red en la que al menos alguno de sus componentes utiliza la red Internet pero que funciona como una red privada, empleando para ello técnicas de cifrado.

VPRN (*Virtual Private Routed Network*)

Simulan redes dedicadas de enrutadores IP entre los sitios de los clientes, aunque transporte tráfico IP, esta debe ser tratada como un dominio de enrutamiento separado.

VRF (*VPN Routing and Forwarding*)

Es una tecnología utilizada en redes de computadoras que permite la coexistencia de múltiples instancias las tablas de ruteo de un mismo router. Esta tecnología es utilizada comúnmente en MPLS VPNs de capa 3.

WAN (*Wide Area Network*)

Es una red muy extensa que abarca computadoras separadas físicamente. Opera en la capa física y de enlace del modelo de referencia.

CAPITULO I

INTRODUCCION

1.1. INTRODUCCION A LA TELEFONIA IP

El desarrollo en la industria de las telecomunicaciones ha cambiando el uso de las redes actuales. Los usuarios de las redes telefónicas convencionales cada día incrementan el uso de estas redes para la transmisión de información y datos en vez de utilizarlas para llamadas de voz únicamente.

Los sistemas de comunicaciones en vez de ser elementos independientes y aislados para atender un determinado tipo de comunicación, ahora tienden a transmitir cualquier tipo de información sobre los medios existentes. Así, sobre la redes de área local (LAN) e Internet, mediante la adopción de ciertos estándares y la incorporación de algunos elementos, es posible enviar voz y datos, con la gran ventaja y ahorro que supone el utilizar una sola infraestructura. Además, el crecimiento y fuerte incursión de las redes IP, el desarrollo de técnicas avanzadas de digitalización de voz, mecanismos de control, priorización de tráfico, protocolos de transmisión en tiempo real, así como nuevos estándares que permiten la calidad de servicio en redes IP, han creado un entorno donde es posible transmitir la voz sobre IP de forma fiable y efectiva.

1.1.1. Características del protocolo IP

El Protocolo IP (*Internet Protocol*) está diseñado para su uso en sistemas interconectados de redes de comunicación de ordenadores por intercambio de paquetes. El protocolo Internet proporciona los medios necesarios para la transmisión de bloques de datos llamados datagramas desde el origen al destino,

donde origen y destino son hosts identificados por direcciones de longitud fija. El protocolo internet también se encarga, si es necesario, de la fragmentación y el reensamblaje de grandes datagramas para su transmisión a través de redes de trama pequeña [1].

El Protocolo Internet está específicamente limitado a proporcionar las funciones necesarias para enviar un paquete de bits (un datagrama internet) desde un origen a un destino a través de un sistema de redes interconectadas. En la figura 1.1 se puede observar el modelo TCP/IP y sus protocolos principales divididos de acuerdo a sus respectivas capas.

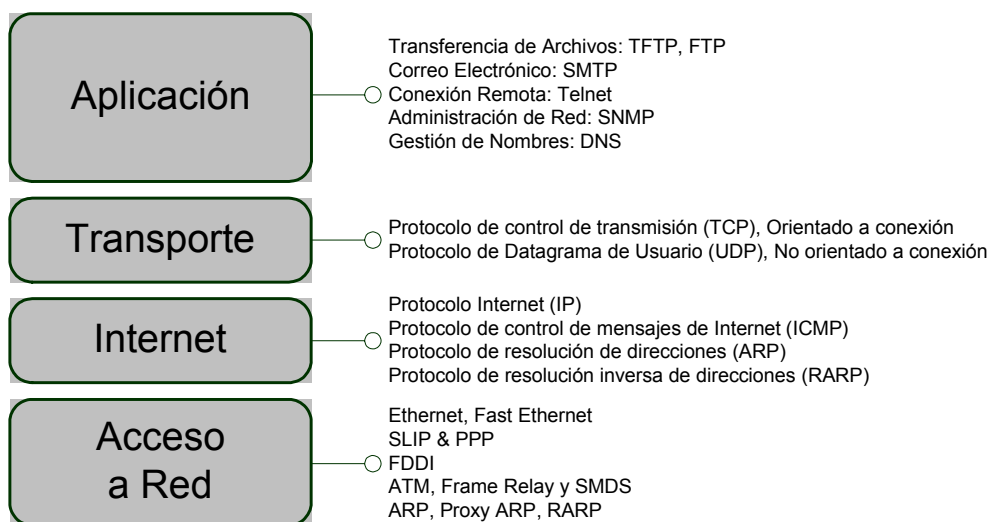


Figura 1. 1. Modelo de red TCP/IP y protocolos principales

1.1.2. Operación del Protocolo IP

El protocolo IP implementa dos funciones básicas: direccionamiento y fragmentación.

Los módulos de internet usan las direcciones que se encuentran en la cabecera para transmitir los datagramas hacia sus destinos. La selección de un camino para la transmisión se llama encaminamiento.

Los módulos internet usan campos en la cabecera para fragmentar y reensamblar los datagramas cuando sea necesario para su transmisión a través de redes de "trama pequeña".

El modelo de operación es que un módulo internet reside en cada host involucrado en la comunicación internet y en cada pasarela que interconecta redes. Estos módulos comparten reglas comunes para interpretar los campos de dirección y para fragmentar y ensamblar datagramas. Además, estos módulos (especialmente en las pasarelas) tienen procedimientos para tomar decisiones de encaminamiento y otras funciones.

El protocolo IP trata cada datagrama como una entidad independiente no relacionada con ningún otro datagrama. No existen conexiones o circuitos lógicos (virtuales o de cualquier otro tipo).

El protocolo no proporciona ningún mecanismo de comunicación fiable. No existen acuses de recibo ni entre extremos ni entre saltos. No hay control de errores para los datos, sólo una suma de control de cabecera. No hay retransmisiones. No existe control de flujo.

1.1.2.1. *Datagramas IP*

La estructura de un datagrama IP está dividido en bloques de 32 bits (4 bytes), cuyo tamaño máximo es de 65535 bytes (64K). El formato de los datagramas IP consiste en una parte de cabecera y en una parte de datos cuyo tamaño es variable.

1.1.2.2. *Cabecera IP*

En la cabecera hay una parte fija de 20 bytes y una parte opcional de longitud variable. En la figura 1.2. se puede ver el formato de la cabecera IP, donde:

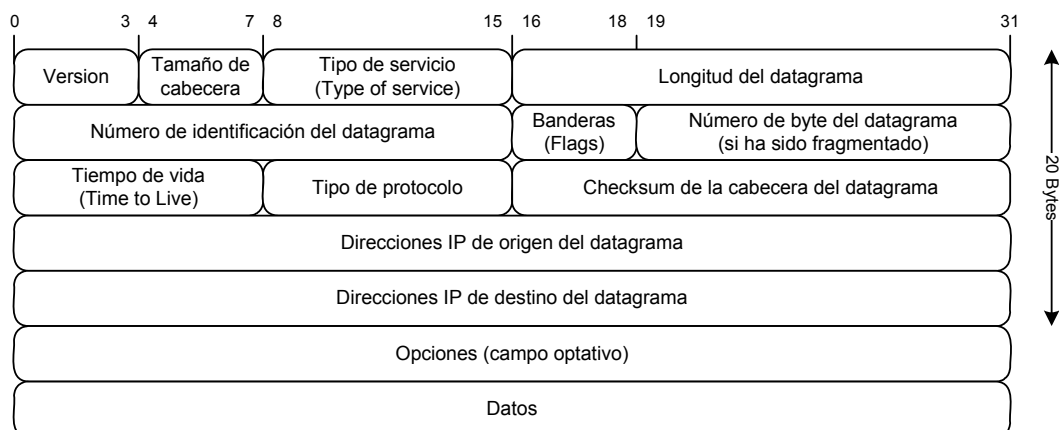


Figura 1. 2. Estructura de un datagrama IP

Versión (4 bits): Indica el número de versión del protocolo al que pertenece el datagrama, lo que permitirá la evolución futura del protocolo y que la transición entre las versiones se pueda hacer ejecutándose en unas máquinas la versión vieja y en otras la versión nueva.

Tamaño de la cabecera (4 bits): Indica el número de palabras de 32 bits que ocupa la cabecera. Estos 4 bits de tamaño máximo nos limitan a un tamaño de cabecera máximo de 60 bytes.

Tipo de servicio (8 bits): Permite que el host especifique que clase de servicio quiere, pudiéndose combinar confiabilidad y velocidad. Para la voz digitalizada es más importante realizar la entrega de forma rápida que precisa, mientras que para la transferencia de ficheros no importa a que velocidad se realiza la transferencia pero sí que esté libre de errores. De los 8 bits, 3 son para el campo de precedencia que en realidad es una prioridad de 0 (normal) a 7 (para los paquetes de control de red). A continuación aparecen los bits de seguridad (alta o baja), retardo (alto o bajo cuando se intenta minimizar el retardo) y rendimiento (normal o alto cuando se intenta maximizar el rendimiento durante la transmisión del datagrama).

Longitud del datagrama (16 bits) en bytes que tendrá todo el datagrama, considerando tanto la cabecera como los datos. Hay que tener en cuenta que el

tamaño máximo de un datagrama es de 65535 bytes lo que puede ser insuficiente en las redes de alta velocidad.

Identificador del datagrama (16 bits): es un número de secuencia que junto a la dirección origen, la dirección destino y el protocolo de usuario, sirven para que la máquina destino determine a que datagrama pertenece el fragmento que ha recibido. Todos los fragmentos de un datagrama contienen el mismo valor en el campo identificador y este número debe ser único para la dirección origen, la dirección destino y el protocolo de usuario durante el tiempo en el que el datagrama permanece en el conjunto de redes.

Banderas (3 bits): El primer bit no se utiliza actualmente. El indicador de mas fragmentos (MF) cuando vale 1 indica que este datagrama tiene mas fragmentos y toma el valor 0 en el último fragmento. El indicador de no fragmentar (DF) prohíbe la fragmentación cuando vale 1. Es una orden que se le da a los encaminadores de que no fragmenten el datagrama cuando el destino es incapaz de reensamblarlo. Si este bit vale 1, el datagrama se descartará si se excede el tamaño máximo en una subred de la ruta. Por lo tanto, cuando este bit vale 1, es aconsejable usar encaminamiento por la fuente para evitar subredes cuyo tamaño máximo de paquete sea menor que el tamaño del datagrama.

El número de byte en el datagrama (13 bits): Indica en que posición del datagrama original, medido en unidades de 8 bytes (64 bits), va el fragmento actual. Debido a esto, todos los fragmentos excepto el último contienen un campo de datos con una longitud múltiplo de 8 bytes. Como se proporcionan 13 bits, puede haber un máximo de 8912 (2^{13}) fragmentos por datagrama, y por lo tanto el tamaño máximo de un datagrama es de 65536 bytes, uno mas que el campo de longitud total.

Tiempo de vida (8 bits): Es un contador que sirve para limitar la vida de un paquete. Aunque lo lógico sería pensar que cuenta el tiempo en segundos, en realidad lo que cuenta es el número de saltos de dispositivo de encaminamiento que realiza. Cuando el contador llega a cero, el paquete se descarta y se envía de una paquete al computador origen avisándole. Con este mecanismo se consigue

que los datagramas no permanezcan indefinidamente en la red si, por ejemplo, se dañan las tablas de encaminamiento.

Tipo de protocolo (8 bits): Se utiliza por la capa de red para saber a que protocolo de la capa de transporte le tiene que enviar el datagrama una vez lo ha reensamblado. Existen diferentes protocolos de transporte, entre ellos TCP y UDP. En el RFC 1700 se definen todos estos protocolos.

Suma de comprobación (16 bits): Sirve para verificar el contenido de la cabecera y es útil para la detección de errores generados durante la transmisión del datagrama. Como algunos de los campos de la cabecera pueden cambiar en alguno de los dispositivos de encaminamiento (por ejemplo, el tiempo de vida y algunos campos relacionados con la segmentación), este valor es verificado y recalculado en cada uno de los dispositivos de encaminamiento. El algoritmo empleado consiste en sumar todas las medias palabras de 16 bits a medida que van llegando, usando la aritmética de complemento a 1, y luego obtener el complemento a 1 del resultado. Se supone que la suma de comprobación de la cabecera es cero cuando llega. Este algoritmo es algo mas robusto que una suma normal. Existen algunas técnicas para acelerar el cálculo.

Dirección origen (32 bits): Indica el número de red y el número del ordenador que envía el datagrama.

Dirección destino (32 bits): Indica el número de red y el número del ordenador al que se envía el datagrama.

Opciones (variable): Contiene las opciones solicitadas por el usuario que envía los datos y se diseñó para que las versiones posteriores del protocolo pudieran incluir información no considerada originalmente, para que los investigadores pudieran probar cosas nuevas y para que aquellas aquella información que es utilizada pocas veces no tuviera asignada unos bits determinados en la cabecera. Cada una de las opciones empieza en 1 byte que identifica la opción. Algunas de las opciones vienen seguidas de un campo de 1 byte para indicar la longitud de la opción y a continuación uno o mas bytes de

datos. Hay seis opciones definidas actualmente pero no todas son reconocidas por todos los dispositivos de encaminamiento, estas son:

- **Seguridad:** Permite añadir una etiqueta para indicar lo secreta que es la información que contiene el datagrama. Por ejemplo, se podría utilizar para que los dispositivos de encaminamiento no consideren redes en concreto.
- **Encaminamiento estricto desde el origen:** Es una secuencia de direcciones IP que sirve para indicar la trayectoria completa que debe seguir el datagrama desde el origen hasta el destino. Esta opción es usada sobre todo cuando los administradores de sistemas envían paquetes de emergencia porque las tablas de encaminamiento se han corrompido o para hacer mediciones de tiempo.
- **Encaminamiento libre desde el origen:** Es una secuencia de direcciones IP que sirve para indicar que el datagrama debe pasar obligatoriamente por esos dispositivos de encaminamiento y en ese orden, pero también puede pasar por otros dispositivos de encaminamiento. Esta opción es útil cuando por diversas consideraciones se deben pasar por algunos dispositivos de encaminamiento en concreto.
- **Registrar la ruta:** Sirve para indicar que los dispositivos de encaminamiento agreguen su dirección IP al campo de opción y de esta manera tener conocimiento de la ruta seguida por el datagrama. Se utiliza, por ejemplo, para poder determinar si los algoritmos de encaminamiento están funcionando correctamente. Los 40 bytes de tamaño máximo que puede tener el campo de opciones sólo permite registrar 9 saltos, lo que puede ser en las redes actuales en muchos casos insuficiente.

- **Identificación de secuencia.** Se utiliza cuando hay recursos reservados para un servicio, por ejemplo voz.
- **Marca de tiempo:** En este caso, además de registrar las direcciones de los dispositivos de encaminamiento como se hacia en la opción registrar la ruta, se utilizan 32 bits para guardar una marca de tiempo expresada en milisegundos. Esta marca es usada principalmente para buscar fallos en los algoritmos de encaminamiento.

Relleno (variable): El campo de opciones se rellena para que su tamaño sea múltiplo de 32 bits (4 bytes).

1.1.2.3. *Direccionamiento IP*

Cada computador y cada dispositivo de encaminamiento tendrá una dirección única cuya longitud será de 32 bits, que será utilizada en los campos dirección origen y dirección destino de la cabecera. Esta dirección consta de un identificador de red y de un identificador de computador. La dirección, como puede verse en la figura 1.3, está codificada para permitir una asignación variable de los bits utilizados al especificar la red y el computador. Este formato de direcciones permite mezclar las tres clases de direcciones en el mismo conjunto de redes. La dirección IP más pequeña es la 0.0.0.0 y la mayor es 255.255.255.255. [2]

Existen tres clases de redes que se pueden clasificar teniendo en cuenta la longitud del campo de red y del campo ordenador. La clase a la que pertenece una dirección puede ser determinada por la posición del primer 0 en los cuatro primeros bits. Las direcciones están codificadas para permitir una asignación variable de bits para especificar la red y el ordenador.

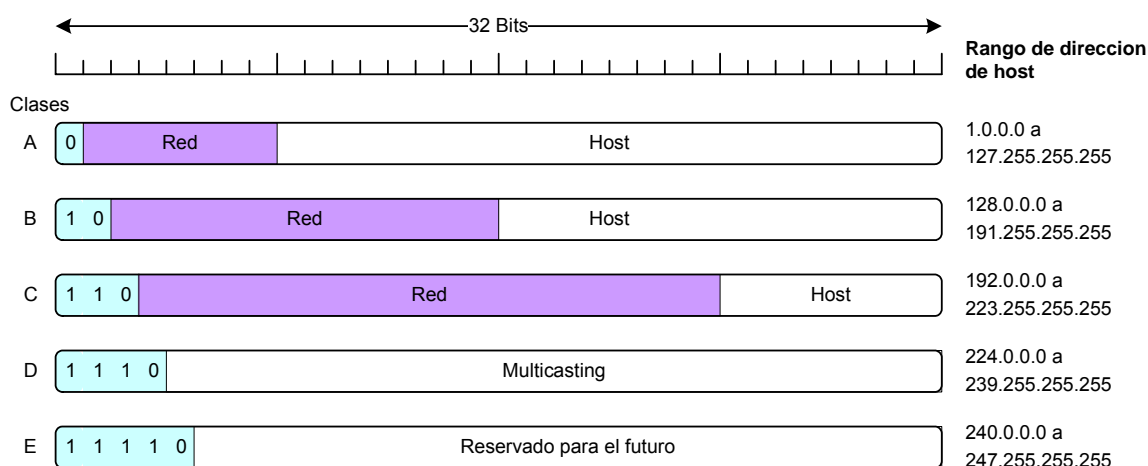


Figura 1. 3. Direccionamiento IP

- **Clase A:** Son aquellas que tienen un 0 en el bit más significativo del primer octeto (números menos de 127 en decimal). En estas direcciones, el primer octeto es el Identificador de red, y los tres siguientes identifican al equipo.
- **Clase B:** Son aquellas cuyos dos primeros bits más significativos son: 10 (primer número entre 128 y 191). En ellas, los dos primeros octetos son el identificador de red, los dos siguientes el del equipo.
- **Clase C:** Los tres bits más significativos del primer octeto son: 110 (primer número entre 192 y 223). En ellas, los tres primeros octetos identifican a la red, y el último al equipo.
- **Clase D:** Los cuatro bits más significativos del primer octeto son: 1110 (primer número entre 224 y 239). Son utilizadas para el direccionamiento punto a multipunto (multicast).
- **Clase E:** Los cuatro bits más significativos del primer octeto son: 1111 (primer número entre 240 y 255). Se encuentran reservados para usos futuros.

1.1.3. Ventajas de migrar a telefonía IP

La Telefonía IP utiliza la red de datos para proporcionar comunicaciones de voz, a través de una sola red de voz y datos. Esta convergencia de servicios de voz, datos y vídeo en una sola red implica ventajas como un menor coste de capital, procedimientos simplificados de soporte y configuración, y una mayor integración de las ubicaciones remotas y oficinas sucursales de una red corporativa.

Las soluciones de telefonía IP, proporcionan una amplia gama de ventajas que pueden resumirse en dos áreas [3];

1.1.3.1. *Ventajas Competitivas*

- **Escalabilidad y Administración.** Añadir un nuevo teléfono IP resulta muy sencillo ya que existen sistemas de gestión, administración y monitoreo.
- **Rápida instalación de nuevos servicios.** Al utilizar una infraestructura IP común y una interfaz de estándares abierta es posible desarrollar aplicaciones innovadoras e implantarlas rápidamente.
- **Accesibilidad.** Los usuarios pueden acceder a todos los servicios de la red, siempre que tengan un punto de acceso.

1.1.3.2. *Retorno de la inversión*

- **Menores costes globales en la infraestructura de la red.** Al implantar una sola red de voz, datos y video en todas las ubicaciones se consigue reducir los costes de capital. Además, como el teléfono y el PC comparten el mismo cable ethernet, los costes de cableado disminuyen.

- **Reducción de los costes de administración de la red.** Ahorros a corto y largo plazo en la administración de una red, un solo proveedor, un solo contrato de mantenimiento, cableado común, reducción en la transacción de llamadas, menor complejidad de integración de una aplicación.

1.1.4. Servicios adicionales

La telefonía IP es mucho más que VoIP; es integrar servicios que tradicionalmente se ofrecían en PBX con la ubicuidad de Internet (o redes IP) y la cantidad de servicios posibles, convergentes y en cierta manera, revolucionarios.

Entre algunos aspectos que incorpora la telefonía IP están:

- Transferencia de llamadas
- Desvío de llamadas
- Conferencias
- Identificador de llamadas
- Llamada en espera
- Colas de llamadas con prioridades
- Buzón de mensajes de voz
- Interoperabilidad con bases de datos
- Gestión remota del sistema

1.2. CARACTERÍSTICAS DEL SISTEMA DE TELEFONÍA IP

Las redes desarrolladas a lo largo de los años para transmitir las conversaciones vocales, se basaban en el concepto de conmutación de circuitos [4], o sea, la realización de una comunicación requiere el establecimiento de un circuito físico durante el tiempo que dura ésta, lo que significa que los recursos que intervienen en la realización de una llamada no pueden ser utilizados en otra

hasta que la primera no finalice, incluso durante los silencios que se suceden dentro de una conversación típica.

En contraposición a esto tenemos las redes de datos, basadas en el concepto de conmutación de paquetes [4], o sea, una misma comunicación sigue diferentes caminos entre origen y destino durante el tiempo que dura, lo que significa que los recursos que intervienen en una conexión pueden ser utilizados por otras conexiones que se efectúen al mismo tiempo.

1.2.1. Componentes del sistema

La arquitectura de una red de telefonía IP consiste en cuatro tipos de elementos: terminales, gatekeepers, gateways y MCU (*multipoint control units*)[5]. La configuración básica consiste de un mínimo de dos terminales conectados a una red de área local (LAN). Sin embargo, en la práctica es necesario añadir otros elementos con el fin de crear un eficiente sistema de comunicaciones con conexiones a redes públicas externas. El propósito y funcionalidad de cada elemento de red se detalla a continuación y se muestran en la figura 1.4, estos son:

Terminales, son clientes los cuales receptan e inician las llamadas. En ellos se genera y se recibe un flujo bi-direccional de información en tiempo real. Un terminal puede ser un software instalado en un computador o un equipo dedicado. Todos los terminales soportan voz, mientras que los datos y video son aplicaciones opcionales. Existe una variedad de software que simula terminales de telefonía IP a los cuales se les conoce como softphones.

Gateway, este dispositivo es el responsable de conectar la red de telefonía IP hacia otros tipos de redes; por ejemplo el gateway puede conectar redes H.323 con redes basadas en protocolo SIP, o a la red de telefonía convencional (PSTN) o redes digitales de servicios integrados (ISDN). El gateway es encargado de la conversión entre los diferentes formatos de transmisión y procedimientos de

comunicación. Los terminales se comunican con los gateways utilizando los protocolos H.245 y Q.931.

Gatekeeper, este dispositivo actúa en conjunción con varios Gateways, y se encarga de realizar tareas de autenticación de usuarios, control de ancho de banda, encaminamiento IP. Es el cerebro de la red de telefonía IP, realiza el control para el procesamiento de la llamada en protocolo H.323.

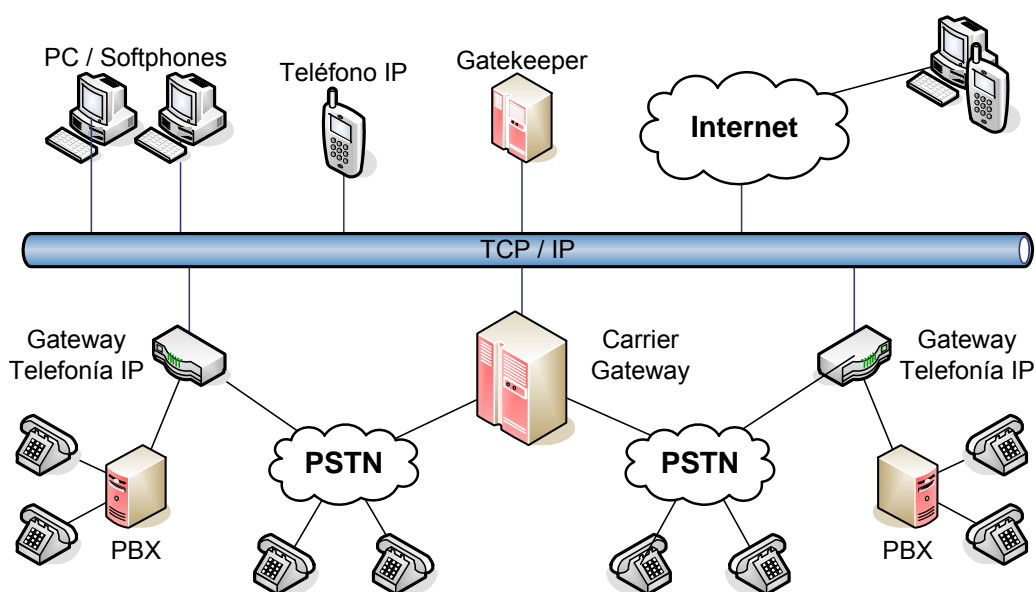


Figura 1. 4. Componentes del sistema

MCU (Multipoint Control Unit), es necesario para la distribución del flujo de datos utilizado en una conferencia multipunto centralizada e híbrida. Consiste de un controlador multipunto (MC) y un número de procesadores multipunto (MP). Los controladores multipunto manejan el control de la información y los procesadores multipunto manejan el flujo. Los terminales envían el flujo de información hacia los MCU, los cuales combinan u redistribuyen información de vuelta hacia los terminales.

A menudo es posible combinar diferentes tipos de elementos de la red en una misma unidad física. Por ejemplo, la funcionalidad de un gatekeeper puede venir incorporado dentro de un gateway y un MCU puede ser implementado dentro de las terminales con el fin de permitir conferencias multipunto.

1.2.2. Escenarios de Redes de Telefonía IP

Existen tres escenarios básicos de redes de telefonía IP, que son descritos a continuación [6]:

1.2.2.1. Telefonía IP - Telefonía IP

El primer escenario se presenta en la figura 1.5, es el más simple de las aplicaciones VoIP. Los terminales IP (Softphones, aplicaciones de PC o teléfonos IP) son conectados entre si en una red IP. La voz se comprime y descomprime a través del software instalado en los terminales.



Figura 1. 5. Escenario de Telefonía IP – Telefonía IP

1.2.2.2. Telefonía IP – Telefonía Convencional

Este segundo escenario se presenta en la figura 1.6, incorpora la posibilidad de conexión del terminal IP a una red de telefonía pública convencional (PSTN, ISDN, TDM, etc), para lo cual es necesario colocar un gateway en el extremo de la red IP, el cual es responsable de establecer la conexión entre la red de telefonía IP y otro tipo de telefonía convencional.

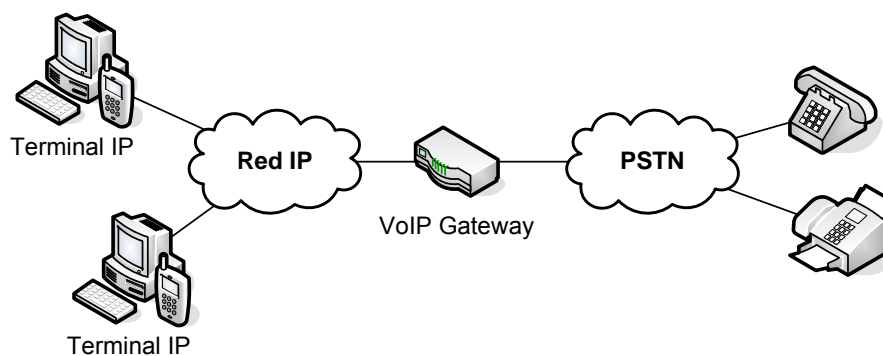


Figura 1. 6. Escenario de Telefonía IP – Telefonía Convencional

1.2.2.3. Telefonía Convencional – Telefonía Convencional

En este tercer escenario se presenta en la figura 1.7, donde ambos usuarios utilizan telefonía convencional, sin embargo la red IP se utiliza para una conexión de larga distancia. En este caso los gateways que se ubican en los extremos de la red IP son los encargados de establecer la conexión y la señalización entre las distintas redes.

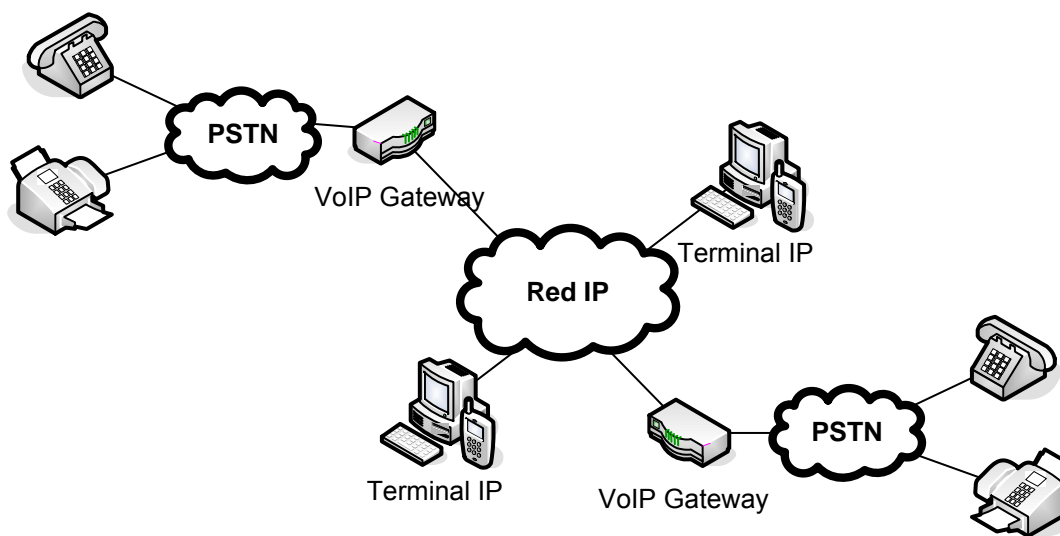


Figura 1. 7. Telefonía Convencional – Telefonía Convencional

1.2.3. Protocolos VoIP

El conjunto de protocolos de Voz sobre IP (VoIP) se descompone en dos categorías [7]:

- Protocolos del plano de datos
- Protocolos del plano de control

El **plano de datos (Voz)** es el protocolo necesario para llevar el tráfico de un usuario a otro.

Los protocolos de tiempo real RTP (*real time protocol*) y cRTP (*compressed real time protocol*) son protocolos de plano de datos y están disponibles en cualquiera de las arquitecturas de VoIP. El tráfico propio de VoIP a veces va por caminos diferentes a la señalización, esto significa que pueden viajar de forma independiente. RTP es el protocolo que soporta la voz del usuario. Cada paquete RTP contiene una muestra pequeña de la conversación de voz. El tamaño del paquete y el tamaño de la muestra de voz, dentro de dicho paquete, dependerán del CODECs utilizados. En la figura 1.8, se muestra la pila de protocolos RTP.



Figura 1. 8. Pila de protocolos RTP

La parte del **plano de control de VoIP** es el tráfico necesario para conectar y mantener el tráfico actual de usuario. Es también responsable de mantener toda la operación de toda la red (comunicaciones router-router).

Hay muchos tipos de protocolos de señalización diferentes, IAX, SIP, H.323, MGCP, Skinny/SCCP, UNISTIM. Los más ampliamente utilizados son H.323 y SIP. La figura 1.9, muestra una comparación entre el modelo TCP/IP y el modelo OSI, así como el conjunto de protocolos utilizados en cada capa de red.



Figura 1. 9. Modelo de red TCP/IP & OSI y protocolos principales

1.2.3.1. Protocolo H.323

El protocolo H.323 fue diseñado por la ITU (*International Telecommunication Union*) para proveer a los usuarios mecanismos para tele-conferencias que tienen capacidades de voz, video y datos sobre redes de conmutación de paquetes [8].

Un punto fuerte de H.323 era la relativa y temprana disponibilidad de un grupo de estándares, no solo definiendo el modelo básico de llamada, sino que además definía servicios suplementarios, necesarios para dirigir las expectativas de comunicaciones comerciales. H.323 fue el primer estándar de VoIP en adoptar el estándar de IETF de RTP para transportar audio y vídeo sobre redes IP.

H.323 está basado en el protocolo RDSI, Q.931 y está adaptado para situaciones en las que se combina el trabajo entre IP y RDSI, y respectivamente entre IP y QSIG (Protocolo de señalización en una Central PBX en una red PSTN). Un modelo de llamada, similar al modelo de RDSI, facilita la introducción de la Telefonía IP en las redes existentes de RDSI basadas en sistemas PBX. Por

esto es posible el proyecto de una migración sin problemas hacia el IP basado en sistemas PBX.

1.2.3.2. Protocolo SIP

El protocolo SIP (*Session Initiation Protocol*) fue desarrollado por el grupo MMUSIC (*Multimedia Session Control*) del IETF (*Internet Engineering Task Force*), definiendo una arquitectura de señalización y control para VoIP [9]. Inicialmente fue publicado en febrero del 1996 en la RFC 2543, ahora obsoleta con la publicación de la nueva versión RFC 3261 que se publicó en junio del 2002.

El propósito de SIP es la comunicación entre dispositivos multimedia. SIP hace posible esta comunicación gracias a dos protocolos que son RTP/RTCP y SDP.

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

SIP fue diseñado de acuerdo al modelo de Internet. Es un protocolo de señalización extremo a extremo que implica que toda la lógica es almacenada en los dispositivos finales. El estado de la conexión es también almacenado en los dispositivos finales. El precio a pagar por esta capacidad de distribución y su gran escalabilidad es una sobrecarga en la cabecera de los mensajes producto de tener que mandar toda la información entre los dispositivos finales.

SIP es un protocolo de señalización a nivel de aplicación para establecimiento y gestión de sesiones con múltiples participantes. Se basa en mensajes de petición y respuesta y reutiliza muchos conceptos de estándares anteriores como HTTP y SMTP.

1.2.3.3. Protocolo IAX

IAX (*Inter-Asterisk Exchange protocol*) es un protocolo abierto, lo cual significa que cualquier usuario puede a partir del código fuente seguir desarrollándolo, pero este no llega a ser aun un estándar.

El protocolo IAX, fue desarrollado por Digium con el propósito de establecer una comunicación entre servidores Asterisk. IAX es un protocolo de transporte que utiliza el puerto UDP (4569) para ambos canales de señalización y cadenas de datos del protocolo de transporte en tiempo real (RTP) [8].

IAX soporta Trunking, donde un simple enlace permite enviar datos y señalización por múltiples canales. Cuando se realiza Trunking, los datos de múltiples llamadas son manejados en un único conjunto de paquetes, lo que significa que un datagrama IP puede entregar información para más llamadas sin crear latencia adicional. Esto es una gran ventaja para los usuarios de VoIP, donde las cabeceras IP son un gran porcentaje del ancho de banda utilizado.

1.2.3.4. Protocolo MGCP

MGCP (*Media Gateway Control Protocol*), también fue desarrollado por el grupo IETF. Si bien MGCP aún se encuentra en desarrollo, éste protocolo es más difundido de lo que se imagina y está dejando atrás a otros protocolo como SIP y IAX. MGCP está definido en la RFC 3534 y fue diseñado para que los equipos terminales sean los más simple posible y que todo el procesamiento de la llamada se ejecute en los gateways y en otros agentes de control. A diferencia de SIP, MGCP utiliza un modelo centralizado. Los teléfonos MGCP no pueden establecer una llamada directamente [8].

MGCP separa conceptualmente estas funciones en los tres elementos: un MGC (*Media Gateway Controller*), uno o más MG (*Media Gateway*), y uno o más SG, (*Signaling Gateway*). Un gateway tradicional, cumple con la función de ofrecer conectividad y traducción entre dos redes diferentes e incompatibles como

lo son las de Conmutación de Paquetes y las de Conmutación de Circuitos. En esta función, el gateway realiza la conversión del flujo de datos, y además realiza también la conversión de la señalización, bidireccionalmente. Así, la conversión del contenido multimedia es realizada por el MG, el control de la señalización del lado IP es realizada por el MGC, y el control de la señalización del lado de la red de Conmutación de Circuitos es realizada por el SG.

MGCP introduce esta división en los roles con la intención de aliviar a la entidad encargada de transformar el audio para ambos lados terminales, de las tareas de señalización, concentrando en el MGC el procesamiento de la señalización.

1.2.3.5. Protocolos Proprietarios

Existen también varios protocolos VoIP propietarios, desarrollados por compañías que se dedican a la comercialización de Telefonía IP, entre ellos: Skinny/SCCP y Unistim [8].

- **Skinny/SCCP (Skinny Client Control Protocol);** es un protocolo propietario de equipos VoIP de Cisco, este es un protocolo predeterminado para equipos terminales de sistemas de PBX de Cisco.
- **UniStim;** es un protocolo VoIP propietario de Nortel Networks, el cual es utilizado en los equipos terminales y dispositivos de control de los productos desarrollados por Nortel.

1.2.4. Codificadores Decodificadores de voz (CoDecs)

Los codecs son el medio por el cual una señal analógica puede ser convertida en una señal digital y ser transportada a través del Internet. El ancho de banda y el número de conversaciones simultáneas que se pueden transportar en una conexión en particular están directamente relacionadas con el tipo de codec a ser

implementado. Entre cada tipo de codec existen diferencias en requerimientos de ancho de banda, calidad y características de codificar el flujo de datos de la señal, algunos codecs son los siguientes [8]:

- G.711
- G.726
- G.723.1
- G.729A
- GSM
- iLBC
- Speex
- MP3

Las características de cada uno de los codecs mencionados anteriormente se presentan en la tabla 1.1:

Tabla 1.1. Características codecs

Codec	Compresión de Datos (Kbps)	Requerimientos de Licencia
G.711	64 Kbps	No
G.726	16, 24 o 32 Kbps	No
G.723.1	5.3 o 6.3 Kbps	Si*
G.729A	8 Kbps	Si*
GSM	13 Kbps	No
iLBC	13.3 Kbps o 15.2 Kbps	No
Speex	Variable entre 2.15 y 22.4 Kbps	No

* Existen versiones de prueba gratuitas.

1.2.4.1. Codec G.711

G.711 es un codec fundamental para las redes PSTN [8]. En efecto si se hace referencia a la modulación por impulsos codificados PCM (*Pulso Code Modulation*) con respecto a las redes telefónicas, siempre se relaciona con G.711. En esta técnica existen dos métodos de compresión de la señal: la ley μ usada en

Norte América y la ley A usada en el resto del mundo, cada uno de ellos transmite una palabra de 8 bits 8000 veces por segundo, lo cual da como resultado una transmisión de 64,000 bits por segundo (64 Kbps).

1.2.4.2. Codec G.726

El codec G.726 se encuentra actualmente en vigencia y reemplaza al G.721 que se encuentra obsoleto [8], utiliza una forma original de compresión, esta técnica es conocida como un modulación diferencial adaptativa de impulsos codificados ADPCM (*Adaptative Differential Pulse-Code Modulation*) y puede ejecutarse en varias tasas de transmisión de bits, entre ellas: 16 kbps, 24 kbps y 32 kbps.

G.726 ofrece una calidad muy similar al G.711, pero utiliza la mitad del ancho de banda para su transmisión. Esto se debe a que se envía el resultado de la cuantización únicamente describiendo la diferencia existente entre el muestreo actual y uno previo a este. G.726 decayó en la década de los 90's por su incapacidad de transmitir señales de modem y fax, pero debido a sus características de ancho de banda y procesamiento de información vuelve a ser atractivo sobre todo por que no requiere un gran trabajo computacional.

1.2.4.3. Codec G.723.1

Este codec esta diseñado para una tener bajas tasas de compresión de voz [8]. Existen dos tipos de tasas de transmisión 5.3 kbps y 6.3 kbps. G.723.1 es uno de los codec requeridos en el protocolo H.323. Este codec esta actualmente protegido por patentes y se requiere una licencia para su aplicación comercial.

1.2.4.4. Codec G.729A

Considerando el limitado ancho de banda que G.729 utiliza, este codec ofrece una impresionante calidad. Utiliza una técnica conocida como CS-ACELP

(*Conjugate-Structure Algebraic-Code-Excited Linear Prediction*). Debido a las patentes que protegen a G.729A no se puede ser usado sin pagarlo, sin embargo este codec es extremadamente popular y es usado en diferentes teléfonos y sistemas de VoIP [8].

Para lograr esa impresionante tasa de compresión, este codec requiere de igual forma un impresionante esfuerzo en el procesamiento de información. G.729A requiere un ancho de banda de 8 kbps.

1.2.4.5. Codec GSM

Este codec no requiere de una licencia para su aplicación como G.723.1 y G.729A, sin embargo ofrece un excepcional rendimiento con respecto a la demanda de procesamiento de información y la calidad de sonido. GSM requiere un ancho de banda de 13 kbps [8].

1.2.4.6. iLBC (Internet Low Bitrate Codec)

Este codec ofrece una atractiva combinación entre calidad y ancho de banda por lo cual hace muy atractiva su aplicación, pero esta técnica no es muy popular entre los codecs ITU por lo que puede no ser compatible con los teléfonos IP comunes y los sistemas de VoIP comerciales. Los RFC 3951 y 3952 publicados por la IETF hacen referencia a iLBC [8].

Para lograr los altos niveles de compresión que iLBC ofrece se utiliza complejos algoritmos, lo cual demanda una alta capacidad en el procesamiento de información. Para la utilización iLBC es necesario registrar la aplicación a GIPS (*Global IP Sound*). iLBC opera con una tasa de bits de 13.3 kbps y 15.2 kbps.

1.2.4.7. Speex

Speex es un codec de tasa de bits variable VBR (*Variable Bitrate*), esto significa que es posible modificar la tasa de transmisión de bits dinámicamente en función al cambio de las condiciones de la red. Existen dos versiones disponibles: banda ancha y banda angosta, la aplicación depende de la calidad de voz requerida en los terminales [8].

Speex es un codec totalmente libre, auspiciado por Xiph.org; para su aplicación es necesario registrarse en su sitio oficial en Internet (<http://www.speex.org>). Este codec opera en un ancho de banda que oscila entre 2.15 kbps a 22.4 kbps.

1.2.4.8. MP3

MP3 es típicamente usado en la compresión de música, es la abreviación de Moving Picture Experts Group Audio Layer 3 Encoding Standard. MP3 no es un codec de telefonía, ya que se encuentra optimizado para la compresión de música y no para voz. Sin embargo es muy popular en los sistemas de telefonía VoIP en las aplicaciones de espera con música MoH (*Music on Hold*) [8].

1.2.5. Calidad de Servicio QoS

La calidad de servicio QoS (*Quality of Service*) es el término que se refiere a los desafíos de transmitir una cadena de información en un tiempo determinado a través de redes diseñadas para entregar datos en términos del mejor esfuerzo.

Los principales problemas QoS de una red de VoIP, son [9]:

- Latencia
- Jitter
- Pérdida de paquetes

- Eco

Los problemas de la calidad del servicio en VoIP vienen derivados de dos factores principalmente:

- Internet es un sistema basado en conmutación de paquetes y por tanto la información no viaja siempre por el mismo camino. Esto produce efectos como la pérdida de paquetes o el jitter.
- Las comunicaciones VoIP son en tiempo real lo que produce que efectos como el eco, la pérdida de paquetes y el retardo o latencia sean muy molestos y perjudiciales y deban ser evitados.

1.2.5.1. Latencia

A la latencia también se la llama retardo. No es un problema específico de las redes no orientadas a conexión. Es un problema general de las redes de telecomunicación. Por ejemplo, la latencia en los enlaces vía satélite es muy elevada por las distancias que debe recorrer la información.

La latencia se define técnicamente en VoIP como el tiempo que tarda un paquete en llegar desde la fuente al destino. Las comunicaciones en tiempo real (como VoIP) y full-duplex son sensibles a este efecto. Al igual que el jitter, es un problema frecuente en enlaces lentos o congestionados.

La latencia o retardo entre el punto inicial y final de la comunicación debiera ser inferior a 150 ms. El oído humano es capaz de detectar latencias de unos 250 ms, 200 ms en el caso de personas bastante sensibles. Si se supera ese umbral la comunicación se vuelve molesta.

No hay una solución que se pueda implementar de manera sencilla. Muchas veces depende de los equipos por los que pasan los paquetes, es decir, de la red misma. Se puede intentar reservar un ancho de banda de origen a destino o

señalizar los paquetes con valores de TOS para intentar que los equipos sepan que se trata de tráfico en tiempo real y lo traten con mayor prioridad pero actualmente no suelen ser medidas muy eficaces ya que no disponemos del control de la red.

Si el problema de la latencia está en nuestra propia red interna podemos aumentar el ancho de banda o velocidad del enlace o priorizar esos paquetes dentro de nuestra red.

1.2.5.2. Jitter

El jitter es un efecto de las redes de datos no orientadas a conexión y basadas en conmutación de paquetes. Como la información se discretiza en paquetes cada uno de los paquetes puede seguir una ruta distinta para llegar al destino.

El jitter se define técnicamente como la variación en el tiempo en la llegada de los paquetes, causada por congestión de red, pérdida de sincronización o por las diferentes rutas seguidas por los paquetes para llegar al destino.

Las comunicaciones en tiempo real (como VoIP) son especialmente sensibles a este efecto. En general, es un problema frecuente en enlaces lentos o congestionados. Se espera que el aumento de mecanismos de QoS (calidad del servicio) como prioridad en las colas, reserva de ancho de banda o enlaces de mayor velocidad (100Mb Ethernet, E3/T3, SDH) puedan reducir los problemas del jitter en el futuro aunque seguirá siendo un problema por bastante tiempo.

El jitter entre el punto inicial y final de la comunicación debiera ser inferior a 100 ms. Si el valor es menor a 100 ms el jitter puede ser compensado de manera apropiada. En caso contrario debiera ser minimizado.

La solución más ampliamente adoptada es la utilización del jitter buffer. El jitter buffer consiste básicamente en asignar una pequeña cola o almacén para ir

recibiendo los paquetes y sirviéndolos con un pequeño retraso. Si alguno paquete no está en el buffer (se perdió o no ha llegado todavía) cuando sea necesario se descarta. Normalmente en los teléfonos IP (hardware y software) se pueden modificar los buffers. Un aumento del buffer implica menos pérdida de paquetes pero más retraso. Una disminución implica menos retardo pero más pérdida de paquetes.

1.2.5.3. Pérdida de paquetes

Las comunicaciones en tiempo real están basadas en el protocolo UDP. Este protocolo no está orientado a conexión y si se produce una pérdida de paquetes no se reenvían. Además la pérdida de paquetes también se produce por descartes de paquetes que no llegan a tiempo al receptor.

Sin embargo la voz es bastante predictiva y si se pierden paquetes aislados se puede recomponer la voz de una manera bastante óptima. El problema es mayor cuando se producen pérdidas de paquetes en ráfagas.

La pérdida de paquetes máxima admitida para que no se degrade la comunicación deber ser inferior al 1%. Pero es bastante dependiente del codec que se utiliza. Cuanto mayor sea la compresión del codec más pernicioso es el efecto de la pérdida de paquetes. Una pérdida del 1% degrada más la comunicación si se usa el codec G.729 en vez del G.711.

Para evitar la pérdida de paquetes una técnica muy eficaz en redes con congestión o de baja velocidad es no transmitir los silencios. Gran parte de las conversaciones están llenas de momentos de silencio. Si solo transmitimos cuando haya información audible liberamos bastante los enlaces y evitamos fenómenos de congestión.

De todos modos este fenómeno puede estar también bastante relacionado con el jitter y el jitter buffer.

1.2.5.4. **Eco**

El eco se produce por un fenómeno técnico que es la conversión de 2 a 4 hilos de los sistemas telefónicos o por un retorno de la señal que se escucha por los altavoces y e ingresa de nuevo por el micrófono. El eco también se suele conocer como reverberación. El eco se define como una reflexión retardada de la señal acústica original y es especialmente molesto cuanto mayor es el retardo y cuanto mayor es su intensidad con lo cual se convierte en un problema. En los retardos suelen ser mayores que en la red de telefonía tradicional.

El oído humano es capaz de detectar el eco cuando su retardo con la señal original es igual o superior a 10 ms. Otro factor importante es la intensidad del eco ya que normalmente la señal de vuelta tiene menor potencia que la original. Es tolerable que llegue a 65 ms y una atenuación de 25 a 30 dB. Hay dos posibles soluciones para evitar este efecto tan molesto:

- **Supresores de eco**, consiste en evitar que la señal emitida sea devuelta convirtiendo por momentos la línea full-duplex en una línea half-duplex de tal manera que si se detecta comunicación en un sentido se impide la comunicación en sentido contrario. El tiempo de conmutación de los supresores de eco es muy pequeño y estos impiden una comunicación full-duplex plena.
- **Canceladores de eco**, es el sistema por el cual el dispositivo emisor guarda la información que envía en memoria y es capaz de detectar en la señal de vuelta la misma información (tal vez atenuada y con ruido). El dispositivo filtra esa información y cancela esas componentes de la voz. Requiere mayor tiempo de procesamiento.

CAPITULO II

MARCO REGULATORIO DE LA TELEFONIA IP

2.1. INTRODUCCION

La evolución en los sistemas de comunicación a través de Voz sobre IP (VoIP) ha permitido que en los últimos años se empiece a hablar de esta tecnología como un posible sustituto de la telefonía tradicional.

Sin embargo, los servicios de VoIP no han sido correctamente entendidos ni adecuadamente dimensionados. Hasta hace poco, se han visto como un simple equivalente de la telefonía local, desconociendo el potencial introducido gracias a la utilización del protocolo Internet (IP), lo cual se ha reflejado en las aproximaciones legales que se han efectuado hasta la entrada en vigor de nuevos marcos normativos de las comunicaciones electrónicas, los cuales reconocen la existencia de nuevos servicios y la convergencia de varios ya existentes.

Hay que precisar dos aspectos de importancia en la definición de voz sobre IP como tal y la telefonía IP.

La VoIP es la transmisión de voz a través de redes que utilicen el protocolo IP, sin considerar si la red portadora es Internet, u otro tipo de red pública o privada. El rasgo distintivo de la VoIP es la utilización de conmutación de paquetes en modo datagrama mediante la utilización del protocolo IP, por oposición a la conmutación de circuitos propia de la telefonía básica.

Por su parte, la telefonía IP consiste en tener centrales IP que se comunican entre sí y entregan información de alta calidad a los usuarios. En otras palabras consiste en tener teléfonos con direcciones IP que permiten efectuar aplicaciones

de datos, no sólo de voz. A través de un teléfono IP se puede navegar en Internet e incorporar otras aplicaciones que se tienen en la red de datos, por ejemplo, se puede tener acceso al directorio empresarial del área de recursos humanos mediante un teléfono IP, o a las páginas de agencias de viajes o estado del tiempo más usadas.

En el ambiente tecnológico y comercial actual, sobresale la exigencia de usuarios a empresas a ser eficientes en la prestación de los servicios de telecomunicaciones dentro de los linderos del concepto de costos eficientes y complementado por la regulación con un factor de ajuste por calidad y otro por productividad, lo que impone el uso de tecnologías para aumentar la productividad de los sectores público y privado como condición de competitividad cuando sea técnica y económicamente posible de acuerdo con las leyes vigentes, que constituye un incentivo al uso de la herramienta VoIP para ofrecer precios competitivos.

2.2. MARCO REGULATORIO PARA LA TELEFONIA IP EN ECUADOR

Si bien la concepción del Internet y el desarrollo de los protocolos como el TCP/IP constituyó una fuente de beneficios en términos de acceso a la información, hubo problemas correlativos a la obtención de sus beneficios, entre ellos los jurídicos ya que el derecho de ningún estado puede estar habilitado para asimilar oportunamente estos cambios. De hecho, la única manera de anticiparlos de forma consciente ha sido mediante la toma de decisiones enfocadas a no legislar sobre el particular como es el caso específico de Ecuador.

El panorama es cambiante, VoIP ha tenido un papel más que preponderante en el devenir actual de las comunicaciones, su importancia económica aumenta a niveles insospechados por el estado ecuatoriano, y amenaza con la desaparición de la Telefonía Pública Básica Conmutada Local Extendida (TPBCLE), Larga Distancia Nacional (TPBCLDN), y Larga Distancia Internacional (TPBCLDI) como las conocemos hoy en día, ya que sus costos de operación la convierten en una opción al rompe competitiva por lo menos en cuanto a precios.

En el Ecuador el marco legal en vigencia para el sector de las telecomunicaciones viene dado por la “Ley Especial de Telecomunicaciones” publicada en el registro oficial el 10 de agosto del 1992 y por el “Reglamento General a la Ley Especial de Telecomunicaciones” publicado en el registro oficial el 4 de septiembre del 2001. [10]

En este contexto legal no se definen las técnicas utilizadas para ofrecer servicios finales de telecomunicaciones, servicios portadores ni servicios de valor agregado. Sin embargo el artículo 5 del “Reglamento General a la Ley Especial de Telecomunicaciones” señala la necesidad de otorgar un título habilitante para la prestación de servicios de telecomunicaciones, de la misma manera que el artículo 36 señala la obligatoriedad de los prestadores de servicios a la interconexión de las redes públicas de comunicaciones. Con esto el Ecuador se alinea en el grupo de países en los cuales no existe una política concreta en materia de telefonía IP y VoIP.

2.3. PRESPECTIVAS DE REGULACION DE LA TELEFONIA IP Y VoIP

Los diversos Estados Miembros de la UIT (*United International Telecommunications*) consideran de diferente manera a la telefonía IP. Algunos la autorizan o no la someten a reglamentación, otros la prohíben, y algunos otros aplican toda una serie de controles y restricciones, ya sea a través de las licencias o de otros instrumentos reglamentarios. Cabe señalar asimismo que esta cuestión se plantea en el contexto de un periodo en el cual muchos estados miembros están flexibilizando sus regímenes de reglamentación de las telecomunicaciones y adoptando una postura que les da un margen más amplio a la política de competencia para asegurar un campo de juego equitativo en los mercados de telecomunicaciones, en contraposición a la reglamentación específica del sector. [11]

Dentro de esos marcos de política general, la telefonía IP plantea algunas cuestiones concretas a los encargados de elaborar políticas y los organismos

reguladores, cuestiones que exigen un equilibrio cuidadoso y fundamentado de intereses diferentes y en ocasiones antagónicos. ¿Dónde se "enmarca" la telefonía IP en los regímenes de reglamentación de las telecomunicaciones, en caso de hacerlo? ¿Qué relación hay entre los derechos y obligaciones de los IPTSP y los de los proveedores de servicios de telefonía tradicional, muchos de los cuales están sujetos a reglas de transmisión comunes y a compromisos de servicio universal? ¿Se ha de tratar igual o de manera diferente la telefonía por Internet, VoIP y el tráfico de voz RTPC? ¿Se les debe exigir a los IPTSP ser titulares de una licencia como la mayoría de las empresas tradicionales de telefonía de voz? ¿O se debería considerar a la telefonía IP como una tecnología incipiente que ofrece nuevos servicios y aplicaciones que se podrían desarrollar mejor con una reglamentación gubernamental mínima o nula?

Ante todo conviene fijar posibles objetivos de política gubernamental para la telefonía IP que puedan servir de base para cualquier enfoque reglamentario que se adopte y, en particular, para determinar si se han de aplicar o no marcos de telecomunicaciones tradicionales. Esos objetivos, que también podrían ser los parámetros para un análisis de los costos/beneficios de cualquier política, incluirían:

- Servicio universal/acceso universal;
- Servicios de telecomunicaciones asequibles;
- Reequilibrio de las tarifas;
- Establecimiento de un campo de juego equitativo para los competidores y los nuevos operadores que ingresan en el mercado;
- Promoción de nuevas tecnologías y servicios;
- Fomento de las inversiones en despliegue de redes y nuevos servicios;
- Efecto en las corrientes de ingresos de los operadores establecidos;
- Transferencia de tecnologías;
- Desarrollo de recursos humanos;
- Crecimiento económico en conjunto y, en particular, en el sector de las comunicaciones.

Actualmente se observan varios planteamientos generales de política nacional:

- En primer lugar, los países que incluyen alguna o todas las formas de telefonía IP dentro de su sistema de reglamentación.
- En segundo lugar, los países que prohíben la telefonía IP.
- En tercer lugar, los países que no someten a reglamentación la telefonía IP.
- Por último, los países en los cuales la situación es incierta o todavía se ha de abordar oficialmente el tema.

Este último grupo de países, en los cuales no existe una política concreta en materia de telefonía IP, representa la mayoría de los Estados Miembros de la UIT, los países han adoptado planteamientos reglamentarios muy diferentes, que pueden guardar relación con condiciones comerciales o grados de liberalización diferentes. Es importante observar que la mayoría de las veces la componente de servicio, es decir, el servicio de telefonía de voz proporcionado por medio de Internet o de redes basadas en el IP, y no la utilización de la propia tecnología IP, la que es objeto de una reglamentación.

Las prohibiciones a la telefonía IP se imponen principalmente en países en desarrollo y pueden estar relacionadas con la preocupación de que ese servicio o aplicación pueda absorber ingresos que hasta entonces correspondían a los operadores establecidos. En ciertos casos, se les ha pedido a las ISP (*Internet Service Provide*) que bloqueen el acceso a determinados sitios web, basados en otros países, que ofrecen la posibilidad de realizar gratuitamente llamadas de telefonía IP. No obstante, los PTO (*Public Telephony Operator*) de algunos países en desarrollo están asimilando la telefonía IP y haciendo frente a las consecuencias de una reducción de los ingresos por minuto generados por los servicios internacionales y de larga distancia, para no arriesgarse a perder la oportunidad de generar ingresos en futuros ámbitos de crecimiento relacionados con el Protocolo Internet. Muchos países que han conservado el monopolio de las telecomunicaciones no prohíben específicamente la telefonía IP. Sin embargo, es

probable que no autoricen a ningún operador que no sea el PTO titular a proporcionarla. Es probable, en cambio, que por motivos prácticos la telefonía IP (o por lo menos los servicios de ordenador personal a teléfono) se autorice en esos países porque no se considera en absoluto como telefonía de voz y, por consiguiente, no es un servicio competitivo. Además, para obtener un servicio de llamadas salientes de ordenador a teléfono aceptable se precisa un acceso a Internet relativamente rápido y fiable, lo cual a menudo no es el caso en los países en desarrollo. Por consiguiente, la terminación de las llamadas internacionales entrantes es el aspecto más importante de la telefonía IP para muchos países en desarrollo.

Hay diferentes razones que justifican las políticas de aquellos países que no someten a reglamentación la telefonía IP o han decidido incluirla de una manera positiva en su marco de reglamentación de las telecomunicaciones. En primer lugar, ello puede responder al deseo de fomentar y estimular las tecnologías incipientes, asociado a la inquietud que suscita la imposición de reglamentaciones a tecnologías que aún no se encuentran en una fase totalmente madura. Podría considerarse que la telefonía IP ejerce presiones competitivas a la baja en las tarifas telefónicas y que por consiguiente favorece el bienestar del consumidor. En segundo lugar, podría considerarse que las limitaciones de la telefonía IP son incompatibles con el enfoque concebido para estimular el despliegue de redes basadas en IP y la migración hacia estas redes. Por último, los organismos reguladores de esos países pueden dudar acerca de la conveniencia de intervenir en los nuevos mercados a menos que cuenten con evidencias de que esos mercados no funcionan correctamente.

La importancia que están tomando hoy en día los sistemas de telecomunicaciones y el manejo básico de negocios, obliga a pensar en la implantación de tecnologías que se adapten a las necesidades de una estructura básica existente y su respectiva regulación por parte del Estado. VoIP generó en el año 1999, negocios por el orden de 263 millones de dólares en Europa, para el año 2006 esta alcanzará los 91.480 millones de dólares, en el año 2000 VoIP absorbió el 3% de las comunicaciones de larga distancia en Estados Unidos y el 5% en Europa y la facturación conjunta llega a los 3.000 millones de dólares, se

marcaron 12.500 millones de minutos a través de VoIP y la progresión es que el 25% de las llamadas se haga a través de IP y que para este año haya igual número de líneas telefónicas que usuarios de la red, y que en apenas 2 años el 90% del tráfico sea de datos, lo que no equivale a decir que habrá menos tráfico de voz, solo que el tráfico de datos prosperará más rápidamente, al paso que seguirán implementándose dispositivos más competitivos que utilizarán los protocolos de Internet. Por todas estas estadísticas se hace necesaria la discusión de este tipo de tecnología por parte de los organismos de control de telecomunicaciones de los gobiernos.

2.3.1. Desregulación en Servicios

Como se ha visto, la convergencia tiene una base tecnológica constituida por la posibilidad de acceso, en cualquier momento y lugar, a información digitalizada, multimedia e interactiva, con capacidad de personalización y a través de una plataforma adaptada a las necesidades del usuario. Además desde el punto de vista de negocio y de mercado, la convergencia promueve la oferta de servicios sin importar los medios técnicos que se utilicen en cada caso.

Desde el punto de vista de la oferta de servicios se puede ver el tema justamente al contrario, es decir, el proveedor de servicios puede ser completamente ignorante de las características concretas de las infraestructuras que soportan sus servicios, ya que se encuentra aislado de ellas. En este sentido, y hasta cierto punto, no importa que la red que ofrece los servicios sea inalámbrica, bucle de abonado convencional, cable de televisión, red eléctrica o fibra óptica. Es decir, los servicios son independientes de las características físicas de las infraestructuras. Otra punto importante y que merece ser considerado es que las características esenciales de la Telefonía IP han descubierto a los proveedores de servicios una nueva forma de ofrecer servicios, que no requiere título habilitante alguno o un título simplificado (Autorización) en el caso de los Proveedores de Acceso a Internet, en contraste con los proveedores tradicionales sometidos a un régimen de otorgamiento de títulos y sobre todo a unas obligaciones (contenidos, servicio universal, calidad de

servicio). Hay que tener en cuenta que la Telefonía IP, puede llevarnos a un concepto de comunicaciones basado en Internet, red en la que se ofrecen ya servicios (voz, contenidos audiovisuales) que en el resto de redes exigirían un título habilitante, y que en la medida en que dichos servicios superen el carácter experimental que hoy en día se les podría conceder, la convivencia de agentes que realizan idénticas actividades con diferentes exigencias administrativas y obligaciones, no parece evidente.

Además, habría que imaginar la sobrecarga que podría enfrentar un regulador con decenas de miles de servidores a los que vigilar. Una opción, que requeriría un consenso internacional entre todos los Gobiernos, sería la de exigir títulos habilitantes para la realización de cierto tipo de actividades, relacionadas con servicios de telecomunicación o audiovisuales. Aunque este tipo de acuerdos no parecen viables hoy en día.

Por todo lo apuntado, es perfectamente factible que el final del proceso de la convergencia en cuanto a la regulación, sea el de que la provisión de servicios de comunicaciones y audiovisuales quede completamente desregulada, al ser incompatible la situación actual con una situación regulada como la actual. Además, si se contempla una sinergia entre lo planteado como Telefonía IP y las futuras aplicaciones desarrollados en Internet, se podría hacer notorio que, al igual que en Internet, son los propietarios de las redes quienes controlan el acceso a los servicios e información al margen de Gobiernos o reguladores, salvo en los elementos que pertenecen más al sector de las telecomunicaciones como circuitos alquilados, red de acceso de los usuarios y bandas de frecuencias.

CAPITULO III

CARACTERISTICAS DEL SOFTWARE ASTERISK

3.1. INTRODUCCION

Las telecomunicaciones son probablemente la última mayor industria electrónica que ha quedado intocable por la revolución del “*open source*” o código abierto. Los grandes fabricantes aún construyen ridículamente caros e incompatibles sistemas que corren bajos complicados y antiguos códigos de impresionante ingeniería sobre hardware obsoleto. Por ejemplo, *Nortel's Business Communications* maneja sistemas de telefonía en servidores Windows NT y sistemas administración que pueden llegar a costar entre 5 a 15 mil dólares sin incluir los dispositivos terminales. Si se requiere aplicaciones especiales o personalizaciones del sistema se tiene que pagar extra por el uso de licencias y funcionalidades limitadas. La mayoría de fabricantes de productos de telecomunicaciones ofrecen líneas similares de productos, donde no es posible tener una flexibilidad y adaptabilidad a necesidades individuales.

Asterisk cambia el enfoque del sector. Con Asterisk no solo un proveedor conoce como funciona el sistema telefónico, o que tecnología utiliza. Con Asterisk se acoge sutilmente el concepto de estandarización, pero también se disfruta la libertad de desarrollar innovaciones propias [8].

3.2. FUNCIONALIDADES GENERALES

Asterisk es una pequeña central controlada por software (PBX) de código abierto. Como cualquier central PBX permite interconectar teléfonos y conectar

dichos teléfonos a la red telefónica convencional. Su nombre viene del símbolo asterisco (*) en inglés.

El creador original de esta pequeña central es Mark Spencer de la compañía *Digium* que sigue siendo el principal desarrollador de las versiones estables. Pero al ser de código libre existen multitud de desarrolladores que han aportado funciones y nuevas aplicaciones. Originalmente fue creada para sistemas Linux pero hoy en día funciona también en varios sistemas operativos como: OpenBSD, FreeBSD, Mac OS X, Solaris Sun y Windows. Pero Linux sigue siendo la que más soporte presenta.



Figura 3. 1. Logotipo de Asterisk PBX.

El paquete básico de Asterisk incluye muchas características que antes sólo estaban disponibles en caros sistemas propietarios como creación de extensiones, envío de mensajes de voz a e-mail, llamadas en conferencia, menús de voz interactivos y distribución automática de llamadas. Además se pueden crear nuevas funcionalidades mediante el propio lenguaje de Asterisk o módulos escritos en C o mediante scripts AGI¹ escritos en Perl o en otros lenguajes.

Para poder utilizar teléfonos convencionales en un servidor Linux corriendo Asterisk o para conectar a una línea de teléfono analógica se suele necesitar hardware especial. *Digium* y otras compañías venden tarjetas para este fin.

Pero quizás lo más interesante es que Asterisk soporta numerosos protocolos de VoIP como SIP y H.323, además que puede operar con muchos teléfonos SIP, actuando como servidor de registro, como *gateway* o entre

¹ AGI es la abreviatura de (*Asterisk Gateway Interface*) el cual es un interface para agregar funcionalidades al sistema Asterisk mediante scripts de programación en lenguajes como Perl, PHP, C.

teléfonos IP y la red telefónica convencional. Los desarrolladores de Asterisk han diseñado un nuevo protocolo llamado IAX para una correcta optimización de las conexiones entre centrales Asterisk.

Al soportar una mezcla de la telefonía tradicional y los servicios de VoIP, Asterisk permite a los desarrolladores construir nuevos sistemas telefónicos de forma eficiente o migrar de forma gradual los sistemas existentes a las nuevas tecnologías. Algunos sitios usan Asterisk para reemplazar a antiguas centralitas propietarias, otros para proveer funcionalidades adicionales y algunas otras para reducir costos en llamadas a larga distancia utilizando la red de Internet.

3.3. FUNCIONALIDADES DE CENTRAL BÁSICA

La solución de telefonía basada en Asterisk dispone de un vasto conjunto de funciones. Asterisk ofrece las funciones propias de las centrales clásicas y además características avanzadas, pudiendo trabajar tanto con sistemas de telefonía estándar tradicionales como con sistemas de Voz sobre IP. [12]

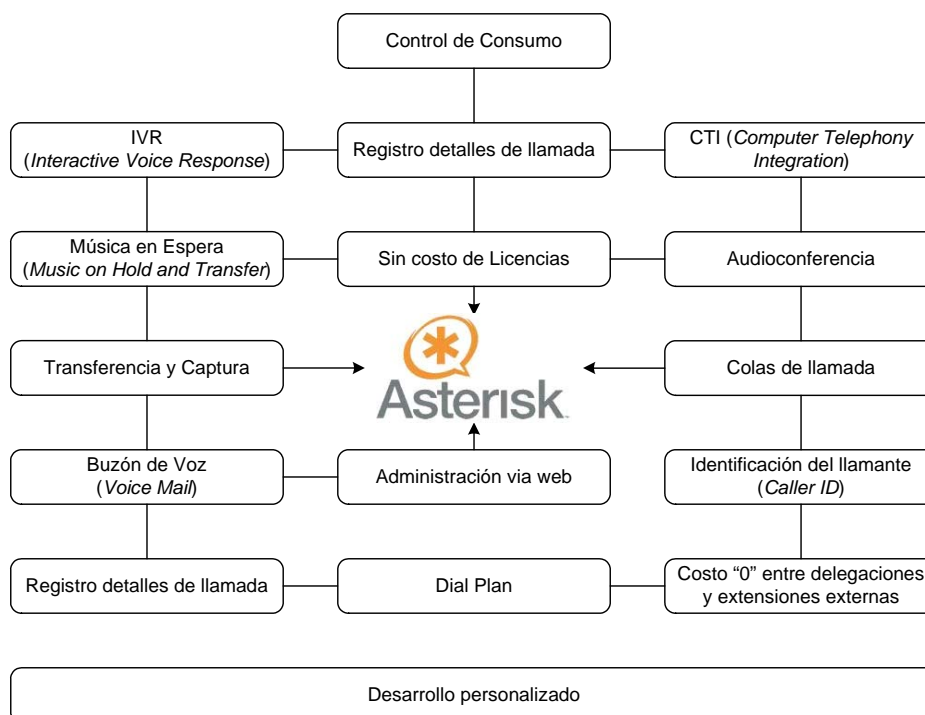


Figura 3. 2. Características de una central telefónica Asterisk.

Las características de llamada que presenta la central Asterisk son:

- Sistema de menú en Pantalla ADSI (Interfaz Analógico para presentación de Servicios)
- Receptor de Alarmas
- Agregar mensaje (*Append Message*)
- Autenticación/Autenticación
- Respuesta automatizada
- Listas negras
- Transferencia no supervisada
- Registros de llamada detallados
- Desvío de llamada si la extensión está ocupada
- Desvío de llamada si la extensión no responde
- Desvío de llamada variable
- Monitorización de llamadas
- Aparcamiento/estacionamiento de llamadas
- Encolado de llamadas
- Grabación de llamadas
- Recuperación de llamadas
- Encaminamiento de llamadas (DID y ANI)²
- Escucha de llamadas
- Transferencia de llamadas
- Llamada en espera
- Identificación del llamante
- Bloqueo de llamante
- Identificación del llamante durante la llamada en espera
- Tarjetas de llamadas
- Conferencia de voz
- Almacenamiento y recuperación en Base de Datos
- Integración con Base de Datos
- Marcación por nombre

² DID (*Direct Inward Dialing Number*) y ANI (*Automatic Number Identification*), características de encaminamiento de llamadas, ver glosario de términos.

- Acceso directo al sistema interno (DISA)
- Tonos de llamada distintivos
- Descubrimiento de Número Universal Distribuido (DUNDi)
- Función No Molestar
- Llamadas de emergencia
- ENUM
- Recepción y transmisión de Fax
- Lógica de extensiones flexible
- Presentación interactiva del directorio
- Respuesta vocal interactiva (IVR)
- Agentes locales y remotos
- Macros
- Música durante la espera (*Music on Hold*)
- Música en transferencia (*Music on Transfer*)
 - Sistema flexible basado en Mp3
 - Reproducción aleatoria o Lineal
 - Control de Volumen
- Marcación predictiva
- Privacidad
- Protocolo de establecimiento abierto (OSP)
- Conversión de protocolos
- Captura de llamada remota
- Soporte de oficina remota
- Extensiones itinerantes (*Roaming*)
- Encaminamiento en función de la identificación del llamante
- Mensajería (SMS)
- Deletreo / Habla
- Transferencia supervisada
- Detección de habla
- Texto a Voz (TTS)
- Llamada a tres
- Fecha y hora
- Transcodificación
- Pasarelas VoIP

- Buzón de Voz
 - Indicador visual de mensaje en espera
 - Tono de marcado entrecortado para mensaje en espera
 - Envío de mensajes del buzón al correo electrónico
 - Grupos de buzones
 - Interfaz web para acceder a los Buzones

Integración ordenador-telefonía (CTI)

- AGI (*Asterisk Gateway Interface*): Interfaz a la Pasarela Asterisk
- Gestor gráfico de llamadas
- Gestión de colas para llamadas salientes
- Marcación predictiva
- Interfaz de gestión

Escalabilidad

- TDMoE (*Time Division Multiplex over Ethernet*): Multiplexación por División en el Tiempo sobre tramas Ethernet
 - Permite la conexión directa de la PBX Asterisk
 - Latencia nula
 - Utiliza hardware Ethernet normal
- Voz sobre IP
 - Permite la integración de instalaciones físicamente separadas.
 - Emplea las conexiones de datos más comúnmente desplegadas.
 - Permite un plan de marcación unificado en todas las oficinas.

Codecs soportados por Asterisk

- Código de Modulación Diferencial Adaptativo de Pulsos (ADPCM)
- G.711 (A-Law y U-Law)
- G.723.1
- G.726
- G.729
- GSM
- iLBC

- Lineal
- LPC-10
- Speex

Protocolos Soportados

- IAX
- H.323
- SIP
- MGCP
- SCCP (Cisco® Skinny®)

Interacción con la Telefonía Tradicional

- E&M, E&M Wink
- FXS
- FXO
- GR-303
- Señalización por bucle (*Loopstart*)
- Señalización por tierra (*Groundstart*)
- *Kewlstart*
- Soporte para tonos MF(multifrecuencia) y DTMF (multifrecuencia con doble tono)
- *Robbed-bit Signaling* (RBS)

Protocolos PRI Soportados

- 4ESS
- BRI (ISDN4Linux)
- DMS100
- EuroISDN (RDSI europea)
- Lucent 5E
- National ISDN2
- NFAS

3.4. FUNCIONALIDADES DE CENTRAL AVANZADA

Asterisk está dotado con características que sólo ofrecen los grandes sistemas PBX propietarios como: buzón de voz, conferencia de voz, colas de llamadas, música en espera, registros de llamada detallados, entre otras funcionalidades.

3.4.1. Buzón de Voz

Una de las más populares características de un sistema moderno de telefonía es el buzón de voz. Asterisk naturalmente cuenta con un flexible sistema de buzón de voz cuyas características incluyen:

- La creación de cuentas ilimitadas protegidas por contraseñas y organizadas en carpetas.
- Saludos de bienvenida predeterminados y personalizados.
- Diferentes saludos de acuerdo al estado del usuario: ocupado o no disponible
- La posibilidad de asociar varios teléfonos a un solo buzón o asociar varios buzones a un solo teléfono.
- Notificación por correo electrónico de un buzón de voz con la posibilidad de adjuntar el archivo de sonido generado.
- Reenvío de un buzón de voz o envíos masivos de mensajes de voz.
- Directorio de la compañía, basado en buzones de voz.

3.4.2. Conferencia de Voz

El servicio de conferencia de voz de Asterisk, permite conversar simultáneamente con dos o más teléfonos, estableciendo conversaciones simultáneas entre varios usuarios a través de la central telefónica. Esta facilidad permite efectuar otra llamada independiente de si fue el suscriptor el que llamo o fue llamado.

3.4.3. Música en espera

Reproduce un archivo de sonido de formato WAV o MP3, mientras se espera una transferencia de llamada o cuando el canal este ocupado.

3.4.4. Salas de Conferencia de llamadas

Esta característica permite que múltiples personas que realicen una llamada puedan conversar juntas como si estuviesen dentro de un mismo lugar. Algunas de las principales características de esta aplicación son las siguientes:

- Salas de conferencias de llamadas con protección y contraseñas
- Administración de conferencias (bloquear, incluir y no admitir participantes)
- Llamadas de *broadcast*.
- Creación estática o dinámica de salas de conferencias.

3.4.5. Interoperabilidad con bases de datos

Asterisk provee un poderoso mecanismo para el almacenamiento de información de las llamadas entrantes y salientes. Este mecanismo se denomina AstBD (*Asterisk Database*) y almacena la información dentro del plan de marcado. A través de este mecanismo se puede lograr un registro detallado de llamadas.

CAPITULO IV

DISEÑO DE LA RED DE TELEFONÍA IP

4.1. CONFIGURACION DEL SERVIDOR

Para el desarrollo de sistemas dedicados a telefonía IP, se deben considerar ciertos parámetros con los cuales Asterisk pueda ejecutarse sin problemas y permita una escalabilidad de acuerdo a las necesidades.

4.1.1. Características del Hardware

En requerimientos de hardware, Asterisk necesita un sistema similar al de aplicaciones en tiempo real. Esto debido a que es necesario tener una prioridad en el acceso al procesador y buses del sistema; por lo tanto en sistemas no relacionados con tareas de procesos dedicados, Asterisk puede ejecutarse mostrando un bajo rendimiento. En pequeños sistemas y sistemas de prueba, esto puede no ser una característica trascendental, pero en sistemas de alta capacidad, pueden ocurrir defectos en el rendimiento los cuales se manifiestan en problemas con la calidad del audio hacia los usuarios. También al incrementar la carga en el sistema, se incrementa la dificultad de mantener las conexiones, por lo tanto es necesario considerar los requerimientos de hardware necesarios durante el proceso de selección y montaje del sistema.

La tabla 4.1 muestra los criterios, que se deben considerar al planificar nuestro sistema.

Tabla 4.1. Requisitos generales del sistema.

PROPÓSITO	NÚMERO DE CANALES	HW MÍNIMO RECOMENDADO
Sistemas de Prueba	No mas de 5	400-MHz x86, 256 MB RAM
Oficinas pequeñas	De 5 a 10	1-GHz x86, 512 MB RAM
Pequeños Sistemas	Hasta 15	3-GHz x86, 1 GB RAM
Medianos y Grandes sistemas	Más de 15	Doble CPU, posiblemente con múltiples servidores en una arquitectura distribuida

En la instalación de grandes sistemas es común desarrollar varios servidores. Uno o más unidades centrales de procesamiento de datos complementadas con uno o más servidores auxiliares para el manejo de bases de datos, correo de voz, conferencias, gestión de red, interfaces web, *firewall*, entre otros.

Asterisk está diseñado para crecer de acuerdo a las necesidades: un sistema pequeño que utiliza un solo equipo para el proceso de datos y demás funciones puede ser distribuido en varios servidores de acuerdo al incremento de la demanda. La flexibilidad es una razón clave en el costo-beneficio de empresas de rápido crecimiento, ya que no existe un tamaño máximo o mínimo efectivo cuando se considere el presupuesto inicial para el desarrollo del sistema.

4.1.2. Instalación del Sistema operativo

Asterisk siendo un software muy flexible, puede ser instalado exitosamente en casi cualquier plataforma Linux y en varias otras también como plataformas Unix y ahora en sistemas Microsoft también.

Para nuestra aplicación en particular se instaló el sistema operativo CentOS 4.5 el cual está basado en la versión de kernel 2.6 de los sistemas Linux.

Para la correcta instalación, configuración y operación de un sistema de telefonía IP basado en Asterisk debe comprobar la instalación de los siguientes paquetes con los cuales interactúa nuestro sistema:

- gcc
- cvs
- libxml2-devel
- libtiff-devel
- mysql-server
- php-gd
- php-mysql
- kernel-devel
- bison
- ncurses-devel
- audiofile-devel
- subversion
- libogg-devel
- zlib-devel
- lame

Para verificar la correcta instalación de estos paquetes se utiliza el comando `rpm nombre_paquete -q` de la siguiente manera:

```
rpm nombre_paquete -q
```

Con este mismo comando, se instala también un paquete en el caso de que este no existiera. Así:

```
rpm -ivh nombre_paquete
```

4.1.3. Instalación de Asterisk

Asterisk, se compone de tres paquetes principales: el programa Asterisk (*Asterisk*), los controladores de telefonía Zapata (*Zaptel*), y las librerías PRI (*libpri*).

Si la red a implementarse es VoIP pura, se requiere únicamente el paquete Asterisk, mientras que los controladores Zaptel se requieren cuando se utiliza un hardware análogo o digital, o se utiliza el controlador ztdummy, y la librería libpri es opcional a menos de que se utilice interfaces ISDN PRI (*Integrated Service Data Network Primary – Primary Rate Interface*) de telefonía tradicional

Para compilar Asterisk, se debe instalar previamente el compilador GCC (Versión 3.x o posterior) y su archivos dependientes. También se requiere el paquete bison que es un programa generador para las funcionalidades en CLI (*Command Line Interface*)

Los paquetes de Asterisk se los puede obtener del servidor FTP de Digium, localizado en <ftp://ftp.digium.com>, en la pagina web de asterisk <http://www.asterisk.org> u obtener la ultima versión del código fuente, a través del comando wget, de la siguiente forma:

```
# cd /usr/src/  
# wget --passive-ftp ftp.digium.com/pub/asterisk/asterisk-1.*.tar.gz  
# wget --passive-ftp ftp.digium.com/pub/asterisk/asterisk-sounds-*.tar.gz  
# wget --passive-ftp ftp.digium.com/pub/zaptel/zaptel-*.tar.gz  
# wget --passive-ftp ftp.digium.com/pub/libpri/libpri-*.tar.gz
```

Una vez que se obtiene el código fuente, es necesario extraerlo y compilarlo, para lo cual es necesario ejecutar el comando *tar* y que los archivos de ubiquen dentro del carpeta /usr/src.

```
# cd /usr/src/  
# tar zxvf zaptel-*.tar.gz  
# tar zxvf libpri-*.tar.gz  
# tar zxvf asterisk-*.tar.gz  
# tar zxvf asterisk-sounds*.tar.gz
```

4.1.3.1. Compilando Zaptel

El interfase Zaptel es un modulo del kernel el cual presenta una capa de abstracción entre los controladores de hardware y el módulo zaptel en asterisk. En este concepto los controladores del dispositivo pueden ser modificados sin realizar cambios en el código fuente de asterisk. Los controladores son usados para comunicar el hardware directamente y enviar información entre zaptel y el hardware.

En asterisk, ciertas aplicaciones y características requieren un dispositivo de reloj, con el fin de operar normalmente (Asterisk no se compilara correctamente, si el dispositivo de reloj no es encontrado). Todos los dispositivos de hardware de Digium PCI (*Peripheral Component Interconnect*) proveen un interfase de reloj de 1-kHz, pero si se carece de este hardware, el controlador ztdummy actúa como un dispositivo de reloj. En distribuciones de linux basados en un kernel de 2.4, el controlador ztdummy toma la señal de reloj del controlador UHCI del puerto USB. En versiones basadas en un kernel 2.6; ztdummy ya no requiere del controlador USB, debido a que el kernel ya provee una señal de reloj de 1-kHz con el cual el controlador actúa. En versiones anteriores a la 2.4.5 el controlador ztdummy, es incompatible.

Para configurar ztdummy, se manipula el archivo makefile del paquete de zaptel, en el cual se retira el símbolo de comentario (#) que antecede al comando ztdummy y se compila normalmente.

```
MODULES=zaptel tor2 torisa wcusb wcfxo wctdm \  
ztdynamic ztd-eth wctlxxp wct4xxp wctellxp # ztdummy
```

Para compilar el programa se ejecuta el siguiente proceso:

```
# cd /usr/src/zaptel-version  
# make clean  
# make install
```

Para compilar el modulo `zaptel` en una versión de kernel 2.6 se requiere realizar el siguiente proceso previo para su carga sin errores:

```
[xyz src]# cd /usr/src/zaptel-1.2.5
[xyz zaptel-1.2.5] mv ztdummy.c ztdummy.c.orig
[xyz zaptel-1.2.5] sed "s/if 0/if 1/" < ztdummy.c.orig > ztdummy.c
[xyz zaptel-1.2.5]# make
[xyz zaptel-1.2.5]# cd /usr/src/kernels/2.6.9-34.EL-i686/include/linux/
[xyz linux]# mv spinlock.h spinlock.h.orig
[xyz linux]# sed "s/rw_lock_t/rwlock_t/" < spinlock.h.orig > spinlock.h
[xyz linux]# cd /usr/src/zaptel
[xyz zaptel-1.2.5]# make
[xyz zaptel-1.2.5]# make install
[xyz zaptel-1.2.5]# make config
```

4.1.3.2. Compilando Libpri

El modulo `libpri` es utilizado por varios fabricantes de hardware TDM para multiplexar una señal en un sistema de transmisión. La compilación del modulo `zaptel` sigue el mismo procedimiento descrito anteriormente.

```
# cd /usr/src/libpri-version
# make clean
# make
# make install
```

4.1.3.3. Compilando Asterisk

Una vez compilado e instalado los paquetes `zaptel` y `libpri`, la instalación estándar del asterisk es muy similar, el procedimiento es el descrito a continuación:


```
# cd /usr/src/asterisk-version
# make clean
# make
# make install
# make samples
```

El comando `make samples` instala una configuración inicial en los archivos, esta configuración cambia de acuerdo a los requerimientos particulares, pero contiene una descripción detallada de cada archivo a ser editado.

4.1.3.4. Compilando mpg123

El paquete `mpg123` contiene un reproductor de archivos `mp3` desde consola, en otras palabras es un decodificador de `mp3` para los sistemas operativos basados en Unix. Para compilar es necesario obtener el paquete `mpg123` que se obtiene del sitio web <http://www.mpg123.com> y seguir el procedimiento de instalación en sistemas Linux:

```
# make clean
# make linux
# make install
# ln -s /usr/local/bin/mpg123 /usr/bin/mpg123
```

4.1.3.5. Sonidos en español

Los instaladores Asterisk desarrollados por Digium cuentan con el paquete de sonidos en Inglés, sin embargo se puede conseguir en la red Internet, los sonidos de la central telefónica Asterisk en español. Una vez descargado el archivo se lo descomprime en el directorio `/var/lib/asterisk/sounds/` con el comando `tar`, De la siguiente manera:

```
# cd /var/lib/asterisk/sounds
```

```
# tar xfvz /ubicacion origin/asterisk-sonidos-version.tar
# ln -s /var/lib/asterisk/sounds/es/digits /var/lib/asterisk/sounds/digits
```

4.1.3.6. Carga del modulo Zaptel

El módulo Zaptel se carga antes de cualquier otro módulo a ser usado. Si se usa un módulo Zaptel de hardware PCI, se debe configurar el archivo */etc/zaptel.conf* antes de la puesta en operación. Si se usa el modulo zaptel solo para acceder al ztdummy, se puede probar su puesta en operación con el comando *modprobe*.

```
# modprobe zaptel
```

Si el proceso de instalación es exitoso, no debe mostrarse una salida en la pantalla. Para verificar la carga del modulo zaptel se utiliza el comando *lsmod*, el cual arroja como resultado la memoria promedio en uso.

```
# lsmod | grep zaptel
zaptel 201988 0
```

4.1.3.7. Carga del modulo ztdummy

El módulo ztdummy es un interfaz hacia un dispositivo que provee una señal de reloj para varias aplicaciones de Asterisk. Para verificar la carga de este modulo se utiliza el comando *modprobe* luego de que el modulo zaptel este ya cargado.

```
# modprobe ztdummy
```

Si el proceso de instalación es exitoso, no debe mostrarse una salida en la pantalla. Para verificar la carga de `ztdummy` y que esté siendo usado por `zaptel` se utiliza el comando `lsmod`, el cual arroja como resultado la memoria promedio en uso.

```
# lsmod | grep ztdummy
Module Size Used by
ztdummy 3796 0
zaptel 201988 1 ztdummy
```

4.1.4. Configuración básica de Asterisk

Asterisk utiliza varios directorios dentro del sistema Linux para manejar varios aspectos del sistema como mensajes de voicemail, mensajes de bienvenida y configuraciones de usuario, registro de errores, entre otros aspectos, sin embargo los archivos de configuración más importantes de Asterisk se los resume en la tabla 4.2:

Tabla 4.2. Archivos de configuración de Asterisk.

Archivo de Configuración	Detalle
<code>zaptel.conf</code>	Realiza una configuración a bajo nivel del interfaz de hardware. Se configura los canales FXO y el canal FXS.
<code>zapata.conf</code>	Este archivo configura el interfaz de Asterisk con el hardware.
<code>sip.conf</code>	Archivo de configuración del protocolo SIP.
<code>iax.conf</code>	Archivo de configuración de los canales IAX
<code>extension.conf</code>	Este archivo configura el plan de marcado

A continuación se presenta un detalle de los archivos más relevantes de configuración y operación para el sistema Asterisk de acuerdo al directorio dentro del sistema operativo Linux:

/etc/asterisk

En este directorio se encuentran todos los archivos necesarios para configurar el sistema y los servicios que Asterisk provee.

- *asterisk.conf*
Configuraciones generales de la ubicación de directorios de configuraciones, módulos compilados, correos de voz, etc. En general contiene las configuraciones predeterminadas, por lo cual no es recomendable modificar este archivo salvo en casos especiales.
- *cdr.conf*
Configuraciones referentes al detalle de llamadas. Los CDR (*Call Detail Record*). Modificar datos en este archivo puede repercutir en la integridad de los CDR.
- *codecs.conf*
A menos que utilice codecs como SPEEX, o se requiera configuraciones especiales con la forma en la que los codecs se comportan, no es recomendable modificar este archivo.
- *extconfig.conf*
Archivo para mapear archivos de configuración hacia tablas en alguna base de datos, de forma que no es necesario guardar las configuraciones en archivos.
- *extensions.conf*
Tal vez el archivo más importante de Asterisk. En este archivo se toman las decisiones de ruteo de las llamadas.
- *features.conf*
Permite habilitar y configurar servicios genéricos de un PBX como la transferencia asistida y monitoreo de llamadas.
- *iax.conf*
Importante archivo para el funcionamiento del canal chan_iax que le

permite a Asterisk interactuar con otros dispositivos IAX, incluyendo otros PBX Asterisk.

- *indications.conf*

Configuraciones para los grupos de frecuencias a utilizar para la indicación del proceso de las llamadas.

- *logger.conf*

Que nivel de verbosidad deben tener los mensajes de registro (*log*) y a donde deben ser enviados.

- *manager.conf*

Configuración del importante servicio AMI (*Asterisk Manager Interface*) que permite conectarnos mediante TCP y administrar la central PBX.

- *modules.conf*

Este archivo determina los módulos que serán cargados por Asterisk al iniciar. Es frecuente que cuando se instala asterisk por primera vez, no arranque debido a que no puede cargar un módulo para el que no tenemos soporte. Esto se soluciona comentando la línea del módulo en este archivo.

- *sip.conf*

Importante archivo que le permite a Asterisk interactuar con otros dispositivos de protocolo SIP.

- *zapata.conf*

Configuración de los canales Zap. Las configuraciones de este archivo deben coincidir con el hardware instalado y la configuración de los contradores zaptel. Existe un archivo muy similar a este llamado unicall.conf, no relacionado directamente con Asterisk.

/var/log/asterisk

En esta carpeta se encuentran los archivos de registro de las operaciones de Asterisk.

- *cdr.db*

Este archivo se encuentra disponible si se cuenta con el controlador para CDR para la base de datos SQLite. El archivo contiene la base de datos de los registros de las llamadas.

- *event_log*
Registro de eventos sucedidos en el PBX.
- *full*
Creado con la intención de contener todos los mensajes de depuración (*debug*) del sistema.
- *messages*
Contiene un listado de los mensajes de alertas (*warnings*), depuración (*debug*) y demás niveles de registro.
- *queue_log*
Archivo utilizado principalmente por la aplicación *app_queue*, que distribuye llamadas entrantes en la orden de la llegada al primer agente disponible.

/var/lib/asterisk

Directorio con archivos de audio, scripts AGI (*Asterisk Gateway Interface*), base de datos AstDB (*Asterisk Data Base*) y archivos para el pequeño servidor HTTP para AJAM (*Asynchronous Javascript Asterisk Manager*). Aquí veremos una descripción de cada uno de los directorios incluidos dentro:

- *agi-bin/*
Aquí se contienen programas en C, PHP, Python o cualquier otro lenguaje con el que se pretenda interactuar desde Asterisk.
- *keys/*
Directorio que contiene llaves RSA para la autenticación de llamadas con el protocolo IAX2.
- *sounds/*
Directorio con todos los sonidos que serán utilizados por aplicaciones que emiten sonidos como: *Playback()* y *Background()*. Ver anexo 3.
- *mohmp3/*
Archivos MP3 para la música en espera (*MusicOnHold*)

4.1.4.1. Configuración de Hardware Zaptel (*zaptel.conf*)

El archivo *zaptel.conf*, localizado en */etc/asterisk/* se utiliza para configurar el hardware que define los canales externos entre la red PSTN y la red de VoIP. Esta configuración define un puerto FXO con uno de señalización FXS.

Un puerto FXO (*Foreign eXchange Office*), es una interfaz que se conecta a la línea telefónica, no genera un tono de llamada, solo acepta uno. Un ejemplo común es el tono de llamada el cual provee la compañía telefónica local. Mientras que un puerto FXS (*Foreign eXchange Station*), es una interfaz que controla al teléfono, provee el tono de llamada y genera un voltaje de timbre el cual alerta al usuario una llamada entrante, ambos interfaces proveen una comunicación bidireccional

4.1.4.2. Configuración de Hardware Zapata (*zapata.conf*)

Asterisk utiliza el archivo *zapata.conf* para determinar la configuración en el hardware de telefonía instalado en el sistema. El archivo *zapata.conf* también controla las características y funcionalidades de las variables asociadas con los canales de hardware como el identificador de llamada, llamada en espera, cancelación de eco, entre otras opciones.

Cuando se realiza la configuración de hardware a través del archivo *zaptel.conf* y se carga el modulo, Asterisk no reconoce ningún cambio de configuración realizado. Estos cambios de configuración son reconocidos a través de los módulos *zaptel*, que actúan como un interfaz entre software y hardware. Toda la configuración de hardware y las características asociadas a ellos se realiza a través del archivo *zapata.conf* ubicado en el directorio */etc/asterisk/*

4.1.4.3. Configuración SIP (*sip.conf*)

El protocolo SIP (*Session Initiation Protocol*) es a menudo usado por la mayoría de terminales telefónicas VoIP y Softphones. Este archivo de la configuración para los canales del SIP de Asterisk se ubica en */etc/asterisk* y aquí

se configura las llamadas de entrada y de salida bajo este protocolo. Este protocolo de señalización de la capa de aplicación utiliza el puerto 5060 para comunicarse.

El archivo sip.conf comienza con una sección [general], en la cual se configura los canales y las opciones predeterminadas para todos los usuarios definidos dentro del archivo.

Cada conexión se define como: *user*, *peer* o *friend*. El tipo de conexión *user* se utiliza para autenticar llamadas entrantes, el tipo *peer* define llamadas salientes y el tipo *friend* es utilizado para definir ambos tipos de llamadas. El nombre de la extensión se define dentro de corchetes ([]).

El campo *secret*, defina una clave de autenticación para ingresar al sistema. Se puede monitoriar la latencia entre el servidor Asterisk y el teléfono, configurando el parámetro *quality*, aquí se puede determinar si el dispositivo remoto es alcanzable, *quality=yes* se utiliza para monitoriar cualquier dispositivo incluyendo otro servidor Asterisk, por defecto se considera una extensión alcanzable si la latencia es menor a 2000 ms. (2 segundos)

Si una extensión es un dispositivo que realiza NAT (*Network Address Translation*) como un *router* o un *firewall*, se configura la opción *nat=yes* para forzar a Asterisk a ignorar la información de contacto para la extensión y utilizar la dirección por la cual los paquetes están siendo recibidos.

La configuración del archivo sip.conf desarrollada es la siguiente:

```
[general]
context=incoming      ;Contexto para llamadas entrantes
;recordhistoty=yes    ;Guardar historia de llamadas
real=mydomail.tld     ;Dominio para autenticación por defecto "asterisk"
port=5060              ;Puerto UDP donde Asterisk escucha SIP
bindaddr=0.0.0.0      ;Dirección IP
srvlookup=yes         ;Habilita llamadas salientes en llamadas salientes
                      buscando en DNS
```



```
;tos=184 ;Calidad de servicio IP: numero del compo TOS
;tos=lowdelay ;Calidad de servicio IP: bajo retardado
;videosupport=yes ;Habilita soporte para video SIP
;disallow=all ;Prohibir todos los codecs
allow=ulaw
;allow=allow ;Codecs por orden de preferencia
;allow=all ;Habilita todo tipo de codecs
allow=ilbc
;language=es ;Lenguaje por defecto configuración individual usuario
;rtptimeout=60 ;Termina una llamada después de 60 seg de inactividad
RTP si no esta en espera
;rtpholdtimeout=600 ;Termina una llamada después de 600 seg de inactividad
;nat=no ;Configuración NAT
;dtmfmode=inband ;Se permite inband, rfc2833 e info
videosupport=yes

[jose.vaca]
type=friend ;Perfil para realizar y recibir llamadas
username=jose.vaca ;Nombre usuario
secret=6001 ;Contraseña usuario
host=dynamic
quality=200 ;Configuración monitoreo latencia
nat=no ;Configuración NAT
canreinvite=no
context=DEE ;Contexto al que pertenece el usuario
callerid=<JoseVaca> ;Identificación usuario
disallow=all ;Codecs por orden de preferencia
allow=alaw
allow=ulaw

[fabian.saenz]
type=friend ;Perfil para realizar y recibir llamadas
username=fabian.saenz ;Nombre usuario
secret=1001 ;Contraseña usuario
host=dynamic
quality=200 ;Configuración monitoreo latencia
nat=no ;Configuración NAT
canreinvite=no
context=DEE ;Contexto al que pertenece el usuario
callerid=<FabianSaenz>;Identificación usuario
disallow=all ;Codecs por orden de preferencia
allow=alaw
allow=ulaw

[carlos.romero]
type=friend ;Perfil para realizar y recibir llamadas
```

```
username=carlos.romero      ;Nombre usuario
secret=1002                 ;Contraseña usuario
host=dynamic
quality=200                 ;Configuración monitoreo latencia
nat=no                      ;Configuración NAT
canreinvite=no
context=DEE                 ;Contexto al que pertenece el usuario
callerid=<CarlosRomero>     ;Identificación usuario
disallow=all                ;Codecs por orden de preferencia
allow=alaw
```

4.1.4.4. Configuración IAX (iax.conf)

En el archivo de configuración iax.conf las secciones están definidas por un nombre encerrados en corchetes ([]). Cada archivo de iax.conf necesita al menos una sección denominada [general]. En esta sección [general] se define la configuración que utiliza el protocolo IAX como los codecs predeterminados. De igual manera que en el archivo sip.conf a continuación se definen las extensiones o terminales que utilizan este tipo de protocolo.

La configuración del archivo iax.conf desarrollada es la siguiente:

```
[general]
bandwidth=low
disallow=lpc10
jitterbuffer=no
forcejitterbuffer=no
tos=lowdelay
autokill=yes

[jose.vaca_iax]
type=friend
username=jose.vaca_iax     ;Nombre usuario
secret=6001                ;Contraseña usuario
host=dynamic
context=DEE                ;Contexto al que pertenece el usuario
callerid=<JoseVaca_IAX>    ;Identificación usuario
disallow=all                ;Codecs por orden de preferencia
```

```
allow=ulaw
trunk=no
```

4.1.4.5. Plan de Mercado - Dialplan (*extension.conf*)

El plan de marcado es quizás la parte más importante del sistema, maneja las llamadas entrantes y salientes, en otras palabras es el conjunto de instrucciones que Asterisk ejecuta para su funcionamiento. A diferencia de los sistemas de telefonía tradicionales es completamente personalizable.

El plan de marcado de Asterisk se configura en el archivo `extension.conf`, este archivo se encuentra en el directorio `/etc/asterisk/` pero esta ubicación puede variar dependiendo de la instalación de Asterisk, otros directorios comunes son `/usr/local/asterisk/etc/` y `/opt/asterisk/etc/`.

El plan de marcado se encuentra conformado por 2 partes: Contexto y extensiones.

Se denomina **contexto** a un grupo de extensiones que se define al colocar un nombre dentro corchetes (`[]`). El nombre que adopta puede ser la combinación de letras de la A a la Z, mayúsculas o minúsculas, números entre el 0 y el 9, guión medio y guión bajo; por ejemplo, el contexto de las llamadas entrantes puede definirse de la siguiente manera:

```
[incoming]
```

Todas las instrucciones colocadas después de la definición de un contexto son parte de dicho contexto, a menos de que se defina uno nuevo. Antes de empezar el plan de marcado existen dos contextos denominados `[general]` y `[global]` los cuales son utilizados para cumplir políticas de seguridad, al utilizarlos correctamente, se puede dar ciertos privilegios de marcación a los usuarios, como salida internacionales o la restricción de estas.

Una **extensión** es una instrucción que Asterisk ejecutará al recibir una llamada entrante o por dígitos marcado en un canal. Dentro de cada contexto se puede definir una o más extensiones. Las extensiones especifican lo que sucede con las llamadas y como estas se procesan en el plan de marcado. Las extensiones también se utilizan para especificar extensiones telefónicas en el sentido tradicional.

La sintaxis que se utiliza viene dada por la palabra "exten" seguida del signo igual y del signo de mayor que. Luego se definen tres elementos que son:

1. El nombre o número de la extensión,
2. La prioridad, que ejecuta una aplicación específica y se numera secuencialmente empezando desde 1, y
3. La aplicación o comando, que ejecuta una acción específica en la llamada como contestar la llamada o colgar.

Estos tres elementos se encuentran separados por comas, de la siguiente manera:

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

Lo cual define una extensión de nombre 123, de prioridad 1, que ejecuta la aplicación Answer()

Las aplicaciones son las acciones que ejecuta el plan de marcado. Cada aplicación realiza una acción específica sobre un canal, los cuales pueden ser: tocar sonidos, aceptar un todo de llamada entrante o realizar una llamada. Algunas aplicaciones necesitan información adicional denominadas argumentos para llevar a cabo sus funciones, los argumentos en una aplicación se colocan dentro de los paréntesis separados por comas. Ver anexo 3.

El plan de marcado desarrollado es el siguiente:

```
[general]
language=es
static=yes          ;Configuración estática
writeprotect=no    ;No se pone protección contra escritura para poder
                   ;actualizar mediante el comando save Dialplan del CLI de
                   ;ASTERISK

;Nuestro DialPlan

[default]
Ignorepat => 9
exten => s,1,Answer()
exten => s,2,BackGround(welcome)
include => DEE

[DEE]
exten => 1001,1,Dial(SIP/fabian.saenz,10.rtT)
exten => 1001,2,Voicemail(u1001)
exten => 1001,3,Hangup

exten => 1002,1,Dial(SIP/carlos.romero,10.rtT)
exten => 1002,2,Voicemail(u1002)
exten => 1002,3,Hangup

exten => 6001,1,Dial(SIP/jose.vaca,10.rtT)
exten => 6001,2,Voicemail(u6001)
exten => 6001,3,Hangup

;Funciones Adicionales

exten => 9000,1,Record(espe.gsm,3,30)
exten => 9090,4,Hangup

exten => 9001,1,Playback(espe)
exten => 9001,2,Hangup

exten => 9090,1,SetLanguage(en)
exten => 9090,2,Meetme()

exten => 9000,1,SetLanguage(en)
exten => 9000,2,VoiceMailMain()
;exten => 9000,1,VoiceMailMain(mailbox@context)
```

```
include => DEE

[from-pstn-custom]
switch => Realtime/from-pstn-custom@extensions
```

4.1.5. Funcionalidad de central avanzada

Asterisk está dotado con características que sólo ofrecen los grandes sistemas PBX propietarios como: buzón de voz, conferencia de voz, colas de llamadas, música en espera, registros de llamada detallados, entre otras funcionalidades.

4.1.5.1. Configuración del buzón de voz

Una de las más populares características de un sistema moderno de telefonía es el buzón de voz. La configuración de esta funcionalidad en Asterisk esta definida en el archivo de configuración denominado voicemail.conf. Este archivo contiene una diversidad de componentes que pueden ser personalizados.

Al igual que en el plan de marcado, el concepto de contextos se mantiene con el fin de mantener varios buzones de voz con características diferentes dentro de un mismo servidor. Los contextos están definidos por un nombre encerrado entre corchetes, como por ejemplo el contexto [default].

Dentro de cada buzón de voz se definen diferentes cuentas de correo. La sintaxis para definir una cuenta de correo de voz es la siguiente:

```
mailbox => password,name[,mail[,pager_email[,options]]]
```

Donde:


```

sendvoicemail=yes           ;habilita el envío de mensajes de voicemail

[internas]                  ;Buzones de voicemail creados

;mailbox => password,name[,mail[,pager_email[,options]]]
;1001 => 1234,Nombre_Usuario,nombreusuario@dominio.com, ,attach=yes
1001 => 1001,1001,,         ;Configuración Mailbox para extensión 1001
1002 => 1002,1002,,         ;Configuración Mailbox para extensión 1002
6001 => 6001,6001,,         ;Configuración Mailbox para extensión 6001

```

4.1.5.2. Configuración de Música en espera

La música en espera es una de las tantas características que incorpora la central telefónica Asterisk. Esta opción permite escuchar un fondo musical mientras se espera una transferencia de llamada o cuando el canal esta ocupado. La configuración de esta funcionalidad en Asterisk esta definida en el archivo de configuración denominado `musiconhold.conf`.

El archivo `musiconhold.conf` configura los diferentes clases de música y sus ubicación para las aplicaciones de música en espera (*Music on Hold*). Asterisk utiliza la aplicación `mpg123` para tocar la música en los canales. Se puede especificar ciertos argumentos para utilizar una aplicación externa que permita tocar una lista de canciones ubicados localmente o alojados sobre una red.

Una configuración típica de este archivo de configuración es la siguiente:

```

[default]
mode=quietmp3
directory=/var/lib/asterisk/mohmp3

; valid mode options:
; quietmp3      -- default
; mp3           -- loud
; mp3nb        -- unbuffered
; quietmp3nb    -- quiet unbuffered
; custom       -- run a custom application
; files        -- read files from a directory in any Asterisk
supported format

;[manual]

```



```

;mode=custom
; Note that with mode=custom, a directory is not required, such as
when reading
; from a stream.
;directory=/var/lib/asterisk/mohmp3
;application=/usr/bin/mpg123 -q -r 8000 -f 8192 -b 2048 --mono -s

;[ulawstream]
;mode=custom
;application=/usr/bin/streamplayer 192.168.100.52 888
;format=ulaw

; mpg123 on Solaris does not always exit properly; madplay may be a
better
; choice
;[solaris]
;mode=custom
;directory=/var/lib/asterisk/mohmp3
;application=/site/sw/bin/madplay -Q -o raw:- --mono -R 8000 -a -12

;[native]
;mode=files
;directory=/var/lib/asterisk/moh-native
;
;[native-random]
;mode=files
;directory=/var/lib/asterisk/moh-native
;random=yes           ; Toca los archivos en orden randómico

```

El commando `SetMusicOnHold()` define la clase para la música en espera en el canal actual. Cuando la música en espera esta activada esta clase define el archivo de música a ser ejecutado. Las clases son definidas en el archivo de configuración `musiconhold.conf`. Por ejemplo:

```

exten=s,1,Answer( )
exten=s,2,SetMusicOnHold(default)
exten=s,3,WaitMusicOnHold( )

```

El commando `MusicOnHold()` toca una canción especificada en la clase indefinidamente. Si se omite la cancion, se toca la cancion predeterminada que se encuentra configurada en el archivo `musiconhold.conf` o la canción predeterminada que define el comando `SetMusicOnHold()`

```

exten => 123,1,Answer( )
exten => 123,2,Playback(tt-allbusy)
exten => 123,3,MusicOnHold(default)

```

El comando `WaitMusicOnHold()` espera un tiempo especificado en segundos antes de tocar una canción.

```
exten => 123,1,Answer( )
exten => 123,2,WaitMusicOnHold(300)
exten => 123,3,Hangup( )
```

4.1.5.3. Configuración de salas de conferencias

Esta característica permite que múltiples llamadas puedan conversar juntas como si estuviesen dentro de un mismo lugar.

La configuración de esta opción se lo hace en el archivo `meetme.conf` en el cual se definen las salas de conferencia y las claves de acceso a los mismos. La configuración de este archivo en nuestro sistema es la siguiente:

```
[general]
;audiobuffers=32           ; Tiempo usado en audio buffers.
[rooms]
;
; Sintaxis: conf => confno[,pin][,adminpin]
;
conf => 800                 ;Configurar extensión 800 sala común.
conf => 801,1111           ;Con clave/pin de ingreso.
conf => 802,1111,1111     ;Con clave/pin de ingreso y moderador.
```

Mediante el comando `MeetMe()` es posible incluir a un llamante en una sala de conferencia. La sintaxis de este comando es el siguiente:

```
MeetMe([confno],[options],[pin])
```

Donde el argumento *confno*, es el número de extensión asignada para salas de conferencia dentro del dial plan, el argumento *options* define las múltiples

opciones para el ingreso, y *pin* es la clave de acceso hacia la sala, un ejemplo del uso de esta aplicación se muestra a continuación:

```
exten => 123,1,Answer() ;Contestar.  
;Añadir al llamante a la conferencia 501 con el pin 1234  
exten => 123,2,Meetme(501,DpM,1234)
```

4.1.5.4. Interoperabilidad con bases de datos

Asterisk provee un poderoso mecanismo para el almacenamiento de información de las llamadas entrantes y salientes. Este mecanismo se denomina *AstBD* y almacena la información dentro del plan de marcado. A través de este mecanismo se puede lograr un registro detallado de llamadas.

Para la aplicación de este mecanismo, debe estar instalada la aplicación *asterisk addons*, para la cual es necesario obtener el archivo *asterisk-addons-version.tar.gz*, descomprimirlo e instalarlo. Esto se lo realiza de la siguiente manera:

```
[xyz src]# cd /usr/src/asterisk-addons-version  
[xyz asterisk-addons-version]# make  
[xyz asterisk-addons-version]# make install
```

Debido a que asterisk realiza una escritura en una base de datos, es necesario comprobar la instalación del servidor de base de datos en nuestro sistema. Caso contrario es necesario instalar el servidor MySQL e inicializarlo. Los componentes del servidor MySQL para Red Hat Fedora Core 4.0 son los siguientes:

- MySQL-shared-compat-5.0.24-0.i386.rpm
- MySQL-client-5.0.22-0.i386.rpm
- MySQL-server-5.0.22-0.i386.rpm

- MySQL-devel-5.0.22-0.i386.rpm
- MySQL-bench-5.0.22-0.i386.rpm

Para arrancar nuestro servidor de base de datos es necesario ingresar en el *prompt* de Linux el siguiente comando:

```
$/sbin/service mysqld start
```

Para agregar el servidor MySQL al los niveles de corrida 3, 4 y 5 del sistema operativo linux, e inicializar el servidor MySQL por defecto, es necesario ingresar en el *prompt* el siguiente comando:

```
$/sbin/chkconfig --level 345 mysqld on
```

Para interactuar con las bases de datos y la central Asterisk es necesario definir en archivos los de configuración de asterisk: los usuarios, las claves de acceso, las bases de datos y los servidores que alojan las mismas.

4.1.5.5. Tarificación de llamadas

La aplicación de CDR (*Call Detail Record*) es una aplicación de Asterisk para el control de llamadas entrantes y salientes. Una de las maneras que cuenta el sistema Asterisk para almacenar CDR es interactuando con bases de datos. Para ello una vez instalado y configurado el servidor MySQL de base de datos, se configura asterisk para interactuar con él, para esto se modifica el archivo `cdr_mysql.conf` en la carpeta `/etc/asterisk` donde se define el *host* del servidor de base datos, el nombre de la base de datos, el usuario, la contraseña, el puerto, entre otros parámetros.

```
[global]
hostname=localhost      ;Dirección del host servidor base datos.
dbname=cdr             ;Nombre de la base datos.
password=cdruser       ;Contraseña ingreso base datos.
user=cdruser           ;Usuario base datos.
;port=3306              ;Puerto de conexión.
;sock=/tmp/mysql.sock
;userfield=1
```

En este archivo de configuración se indica a asterisk que se ha creado la base de datos *cdr* para registrar el ingreso y salida de llamadas. Y el usuario *cdruser* con contraseña *cdusr* para su edición.

La base de datos para registrar el detalle de llamadas (CDR) de asterisk requiere el siguiente esquema:

```
CREATE TABLE cdr (
  calldate datetime NOT NULL default '0000-00-00 00:00:00',
  clid varchar(80) NOT NULL default '',
  src varchar(80) NOT NULL default '',
  dst varchar(80) NOT NULL default '',
  dcontext varchar(80) NOT NULL default '',
  channel varchar(80) NOT NULL default '',
  dstchannel varchar(80) NOT NULL default '',
  lastapp varchar(80) NOT NULL default '',
  lastdata varchar(80) NOT NULL default '',
  duration int(11) NOT NULL default '0',
  billsec int(11) NOT NULL default '0',
  disposition varchar(45) NOT NULL default '',
  amaflags int(11) NOT NULL default '0',
  accountcode varchar(20) NOT NULL default '',
  uniqueid varchar(32) NOT NULL default '',
  userfield varchar(255) NOT NULL default ''
);
```

La tabla que se crea mediante la ejecución del comando anterior, tiene los siguientes campos:

Tabla 4. 3. Campos de la tabla SQL para registro de llamadas en Asterisk.

Variable	Tipo Variable	Registro	Descripción
calldate	Datetime	2008-01-12 01:38:15	Fecha realización llamada
clid	Carácter	Jose Vaca	Identificación usuario
src	Carácter	6001	Fuente llamada
dst	Carácter	1001	Destino llamada
dcontext	Carácter	DEE	Contexto
channel	Carácter	SIP/6001 – 099f0040	Canal
dstchannel	Carácter	SIP/6002 – 09a16f20	Canal destino
lastapp	Carácter		Aplicaciones perdidas
lastdata	Carácter		Datos perdidos
duration	Entero	11	Duración llamada (incluye marcación)
billsec	Entero	16	Duración llamada (excluye marcación)
disposition	Carácter	Answered	Disposición llamada
amaflags	Entero	1	Variable sistema
accountcode	Carácter		Código cuenta
uniqueid	Carácter		ID unico
userfield	Carácter		Campo usuario

4.2. CONFIGURACIÓN DE TERMINALES

Dentro de la arquitectura de una red de telefonía IP se encuentran los terminales, estos son los clientes, los cuales receptan e inician las llamadas. En ellos se genera y se recibe un flujo bi-direccional de información en tiempo real.

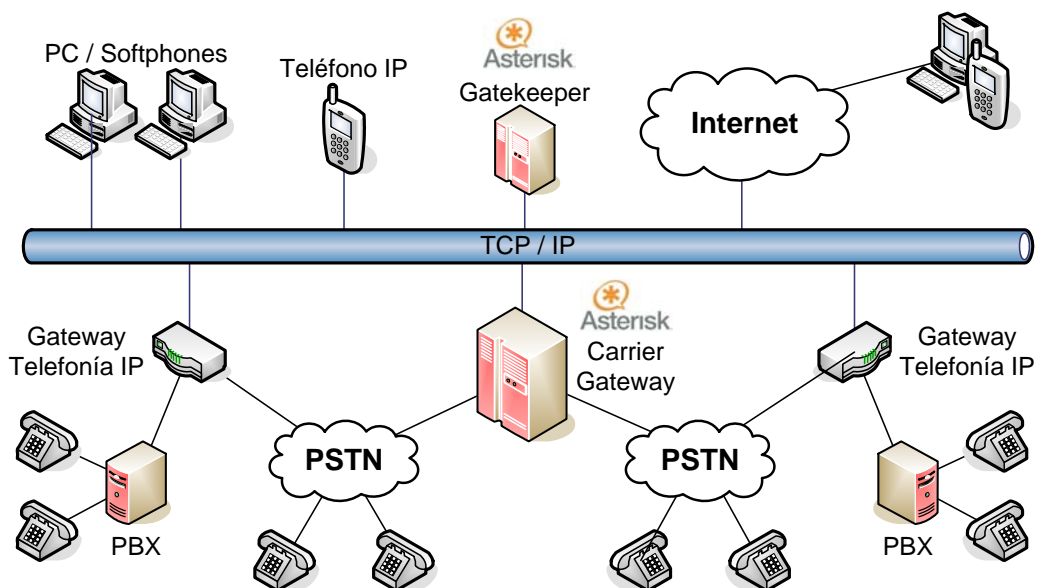


Figura 4. 1. Sistema de telefonía IP

Cualquiera que sea el tipo de terminal, es necesario que se registre dentro de un servidor de telefonía IP, para ello es necesario validar el terminal dentro del equipo que provee el servicio de telefonía IP. Para esto, se debe considerar los protocolos de comunicación (SIP, H323, IAX, etc), así como también definir los Codecs (G723, G711, GSM, etc) soportados por los equipos.

4.2.1. Tipos de Terminales

Un terminal puede ser un equipo dedicado (teléfono IP) o un software que puede ser instalado en un computador (*softphone*). Todos los terminales soportan voz y presentan infinidad de aplicaciones opcionales como datos y video.

4.2.2.1. Equipos dedicados

Existe una variedad de marcas que ofrecen equipos dedicados para telefonía IP, de acuerdo al fabricante ofrecen una variedad de protocolos y codecs de funcionamiento. La apariencia de estos dispositivos es similar a los terminales de telefonía convencional, pero ofrecen servicios adicionales como video, mensajes cortos de texto, envío de datos, entre otras características.

Cisco Systems

Cisco Systems es el líder mundial en soluciones de infraestructura de red y terminales de Telefonía IP dedicados, ofrecen soluciones de sistemas de telefonía IP incorporando sus grandes ventajas como: transmisión de datos y video. Los terminales Cisco operan en un protocolo propietario SCCP (*Skinny Client Control Protocol*), compatible con sus servidores y *gateways*. Ofrecen además terminales *IP cordless* y terminales *wieless* incorporando la tecnología 802.1x en sus series 792x G. La figura 4.2, muestra una Terminal IP cisco 7971G-GE.



Figura 4. 2. Terminal IP Cisco 7971G-GE.

Avaya

Los teléfonos IP de la serie 4600 de Avaya ofrecen las mismas características y funciones avanzadas de llamada que los teléfonos tradicionales de Avaya. Estos elegantes teléfonos incorporan una serie de características útiles como indicador de mensaje en espera, pantallas de fácil lectura, compatibilidad con audífonos, compatibilidad con *NetMeeting*, múltiples teclas de funciones programables y teclas de navegación en pantalla. Soportan protocolos de comunicación SIP y H323. La figura 4.3, muestra a un terminal IP Avaya 4620.



Figura 4. 3. Terminal IP Avaya 4620.

3-Com

3Com es considerada como líder en soluciones de telefonía IP. Ofrece una variedad de teléfonos IP incorporando todas las ventajas de la tecnología. Además ofrece soluciones con terminales *cordless* de alcances de hasta 300 metros y terminales con tecnología WiFi 802.11G. Sus productos soportan los protocolos de comunicación SIP y 3Com NBX H3. La figura 4.4, muestra un terminal IP 3Com 3102.



Figura 4. 4. Terminal 3Com 3102 Business Phone.

Siemens

La multinacional alemana también presenta en el mercado varios terminales IP, de la familia optiPoint 400, incorporan las ventajas de la telefonía IP, además que ofrecen una integración con las soluciones PBX tradicionales HiPath 3000 y HiPath 4000 de Siemens. Soportan protocolos: H323 y SIP. En la figura 4.5, se puede apreciar a un terminal IP Siemens OptiPoint 400 Advanced.



Figura 4. 5. Terminal IP Siemens optiPoint 400 Advanced.

4.2.2.2. Softphones

Los softphones, es un software que simula terminales de telefonía IP en el computador personal. Este tipo de software convierte a un computador en un terminal de telefonía IP utilizando sus recursos como procesador, tarjetas de sonido, red y video. La tabla 4.2, muestra los *softwares* más populares con sus principales características:

Tabla 4. 2. Características Softphones.

Nombre	Equipo de desarrollo	Características
X-LITE	CounterPath Solutions, Inc.	<ul style="list-style-type: none"> ▪ Entrada de dos líneas ▪ Mensajes instantáneos de texto ▪ Cancelación de eco, control de ganancia automática, detección de voz. ▪ Codecs de audio soportados: Broadvoice-32, Broadvoice-32 FEC, G.711aLaw, G.711uLaw, GSM, iLBC, L16 PCM Wideband. ▪ Codecs de video soportados: H.263, H.263 1998. ▪ Protocolos compatible: RFC 3261 SIP standard.
SJPHONE	SoftJoys Labs	<ul style="list-style-type: none"> ▪ Disponible en Windows, Linux y Mac. ▪ Soporta tanto H.323 como SIP. ▪ Admite codecs g711a, g711u, gsm y iLBC. ▪ Admite múltiples configuraciones. Gratuito.

EXPRESS TALK	Ahe NCH Swift Sound ad Software AG	<ul style="list-style-type: none"> ▪ Sólo en Windows. ▪ Sólo válido para SIP. ▪ Soporta 4 canales SIP simultáneamente. ▪ Admite codecs g711a, g711u y gsm. ▪ Cancelación de eco y reducción de ruido. ▪ Gratuito.
EYEBEAM (X-TEN)	CounterPath Solutions, Inc.	<ul style="list-style-type: none"> ▪ Disponible en Windows, Mac. ▪ Soporta SIP. ▪ Admite codecs G.711, iLBC, Speex, GSM, G.723.1, G.726, G.729a.
SIPSS	Nero	<ul style="list-style-type: none"> ▪ Sólo en Windows ▪ Tiene dos versiones. Gratuita con menos funcionalidades y de pago.
ADORE SOFTPHONE	Adore Softphone	<ul style="list-style-type: none"> ▪ Disponible en Windows y pocket-pc. ▪ Soporta protocolo G729 aparte de los protocolos habituales.
KPHONE	Wirlab	<ul style="list-style-type: none"> ▪ Disponible en linux. ▪ Gratuito.

4.2.2. Configuración de Softphone X-Lite

El *X-Lite* es un *softphone* gratuito desarrollado por *CounterPath Solution, Inc* y es el software gratuito líder para comunicaciones basadas en protocolo SIP. *X-Lite* combina llamadas de voz, video y mensajería instantánea con un interfaz muy familiar e intuitivo. En la figura 4.6, se muestra la apariencia que tiene el *software* una vez instalado en el computador.



Figura 4. 6. Apariencia del Softphone X-Lite 3.0 desarrollado por CounterPath Solutions, Inc.

Para configurar el servidor o proveedor de telefonía IP, se hace un clic derecho y a continuación se selecciona la opción *SIP Account Setting*, tal como se muestra en la figura 4.7.



Figura 4. 7. Pantalla para acceder a configuración de cuentas SIP en X-Lite.

Una vez allí, se despliega el detalle de todas las cuentas SIP, registradas por *X-Lite*. Si se quiere registrar una nueva cuenta se hace clic en el botón **ADD**, de la misma manera si se quiere eliminar una cuenta en **REMOVE** o si lo que se quiere es editar la cuenta se hace un clic en el botón **PROPERTIES**. La figura 4.8, muestra la pantalla de configuración de cuentas SIP.

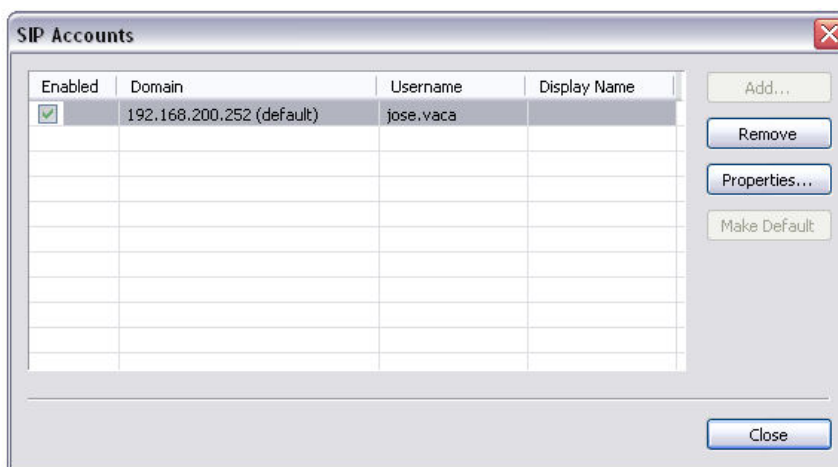


Figura 4. 8. Pantalla de configuración de cuentas SIP en X-Lite.

Al agregar o editar una cuenta, se despliega una pantalla con varias pestañas, en las cuales se establece los datos de la cuenta SIP del usuario de *X-Lite*, en esta pantalla se introducen los datos con los que ha sido creada la cuenta SIP en el servidor o proveedor del servicio.

La figura 4.9, muestra la pantalla de configuración de una cuenta SIP en *X-Lite*.

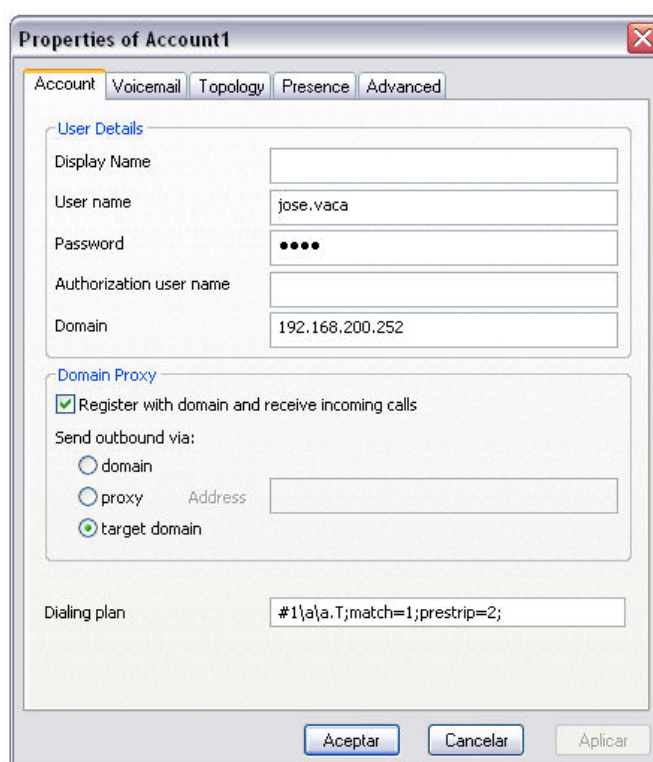


Figura 4. 9. Configuración de una cuenta SIP en X-Lite.

Una vez configurado correctamente una cuenta SIP activa dentro de *X-Lite*, en la pantalla principal del software aparece el nombre del usuario de la cuenta como registrado y listo para operar, tal como se muestra en la figura 4.10.



Figura 4. 10. Aspecto de X-Lite con una cuenta SIP registrada y activa.

4.2.3. Configuración de SJPhone

SJPhoneTM es un *softphone* desarrollado por *SoftJows Labs* que permite hablar en Internet con cualquier PC de escritorio, *notebooks*, PDAs, Teléfonos-IP de ordenadores independientes e incluso con líneas convencionales y teléfonos móviles. Soporta ambos estándares de industria SIP y H.323. Es compatible e interoperable con la mayoría de proveedores de servicio de telefonía de Internet (ITSP) y telefonía IP.

A continuación en la figura 4.11, se muestra el aspecto del *software* una vez instalado en el computador.



Figura 4. 11. Aspecto del Softphone SJphone desarrollado por SoftJoys Labs.

Para configurar una cuenta SIP en este *softphone* se ingresa a la pantalla de opciones, haciendo clic en el icono de opciones o haciendo clic derecho y seleccionando *OPTIONS*.

Una vez allí, aparece una pantalla con varias pestañas en la cual se configura todas las opciones relacionadas con la operación del *softphone* como Codecs, Interfaces, soporte, cuentas o perfiles de usuarios SIP.

En la figura 4.12, se muestra las ventanas de configuración para las cuentas SIP en *SJphone*.

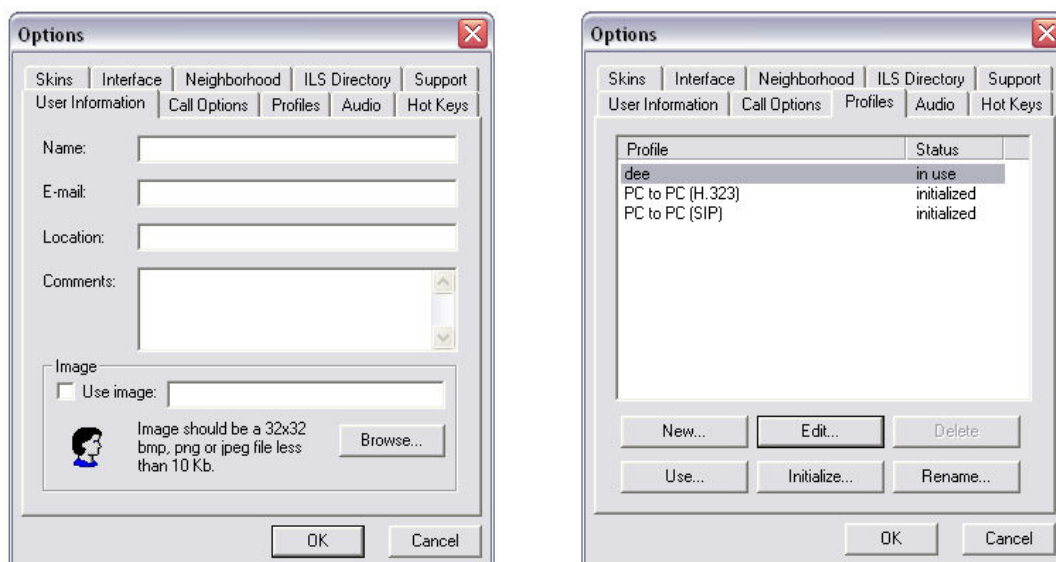


Figura 4. 12. Configuración de cuentas SIP en el Softphone SJphone.

Se ingresa a la pestaña de Perfiles (*Profiles*) y se agrega una nueva cuenta SIP, y allí se abra una nueva pantalla similar a la que se muestra en la figura 4.13. Allí se selecciona las opciones del servidor o proveedor del servicio como la dirección de dominio, el puerto en el que opera, número de correo de voz, usuario y contraseña.

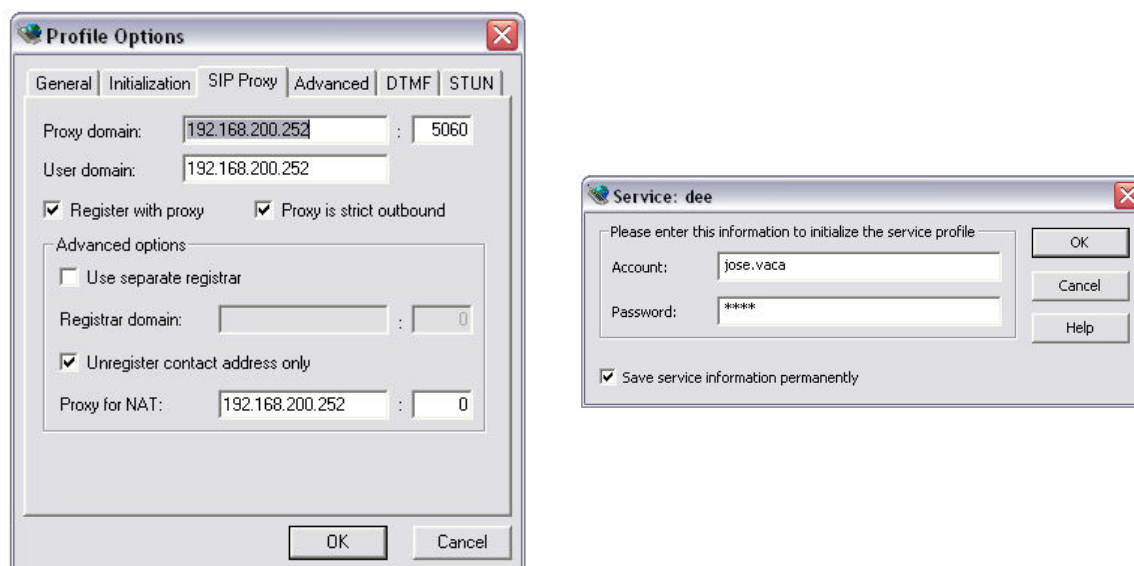


Figura 4. 13. Configuración de servidor o proveedor de servicio en el Softphone SJphone.

Si se requiere cerrar la sesión actual o volver a registrarse, se ingresa a la opción *SERVICE* haciendo clic derecho en el software, tal como lo muestra la figura 4.14 y se ingresa la información de la cuenta SIP.



Figura 4. 14. Configuración de cuentas SIP en el Softphone S_Jphone.

Una vez configurado y registrado correctamente la cuenta SIP desde el *softphone S_Jphone*, en la pantalla principal del software aparece el estado del registro y el nombre del usuario registrado, tal como lo muestra la figura 4.15.



Figura 4. 15. Aspecto del Softphone S_Jphone con un registro satisfactorio.

CAPITULO V

EVALUACIÓN DEL SISTEMA

5.1. INTRODUCCIÓN

El crecimiento a largo plazo de la telefonía IP se verá impulsado por la mayor flexibilidad en los servicios ofrecidos por las redes IP, comparadas a la telefonía tradicional o PSTN. Esta flexibilidad deriva de las capacidades de señalización de los sistemas IP, superiores en versatilidad a los de la PSTN [15], además de la posibilidad de utilizar múltiples medios en una llamada. Sin embargo, para convertirse en un sistema de comunicaciones de escala global, la calidad de servicio de la telefonía IP debe al menos igualar a la del PSTN. Desde la perspectiva del usuario, el desempeño de un sistema de telefonía y su calidad de servicio consiste primero en la confiabilidad y cantidad de tiempo invertido en el establecimiento de la llamada y luego en la calidad del audio o del video transmitido.

5.2. ANÁLISIS DEL DESEMPEÑO DE LA RED

El análisis del desempeño de la red se lo puede relacionar con la calidad de servicio (QoS), el cual es un término que se refiere a evaluar el desafío de enviar información en tiempo real dentro de redes diseñadas para enviar información bajo el criterio del mejor esfuerzo. Debido a esto no existe una regla que defina la calidad de servicio sin embargo, es aceptado que si se puede enviar una conversación producida con hasta 300 milisegundos de retardo, esta conversación puede ser fluida y normal. Cuando el retardo excede los 500 milisegundos comienza a existir dificultades para el entendimiento del mensaje. Si el retardo es mas de 1 segundo, la conversación es totalmente incomoda.

Adicionalmente al tiempo de respuesta, es esencial asegurar que la información llegue intacta y completa hacia el receptor. Cuando existe mucha pérdida de paquetes, la interpretación del mensaje puede llegar a ser muy incómoda. Estos parámetros junto con otros como latencia, eco y *jitter* que fueron definidos previamente pueden ser considerados como parámetros para evaluar la calidad de servicio en un sistema de telefonía IP.

El desempeño de la calidad de servicio dentro una red de telefonía IP depende de muchos campos entre ellos y un fundamental es el *codec* utilizado para la compresión de la voz. De acuerdo al protocolo utilizado se puede optimizar el ancho de banda y la calidad de voz dentro de un sistema. La tabla 5.1 hace referencia a algunos *codecs* y su tasa de bits asociados:

Tabla 5. 1. Cuadro comparativo de codecs soportados por Asterisk.

Codec	Frecuenc Muestreo [kHz]	BR [kbps]	NEB [kbps]	Tamaño Bloque [ms]	Licencia	Comentarios	Ventajas	Desventajas
G.711	8	64	87.2	20	Open Source	G.711 u/a comúnmente referidos como u-law/a-law, donde a-law es la versión europea y u-law es la versión de EUA/Japón	Diseñado para entregar máxima calidad de voz. Muy bajo consumo de procesador	Incluyendo sobrecarga, consume más de 64 kbps, por lo que al menos 128 kbps de ancho de bandas son necesarios en cada dirección. (subida y bajada)
G.722	16	48 56 64	- - -	30	Open Source	Códec estándar ITU		
G.723.1	8	5.3 6.3	20.8 21.9	30	Propietario	Comúnmente usado por proveedores de VoIP para dar una buena calidad.	Compresión muy alta manteniendo una excelente calidad de audio.	Demanda un gran procesamiento de la información. Se requiere licencia para su uso.
G.726	8	16 24 32 40	- 47.2 55.2 -	20	Open Source	Una versión mejorada de G.721 y G.723		
G.728	-	16	31.5		Open Source	Códec estándar ITU		
G.729	8	8	31.2	20	Propietario	Códec estándar ITU	Excelente uso de ancho de banda.	Se requiere licencia para su uso.
GSM	8	13	-		Propietario	Misma codificación usada en los teléfonos celulares GSM.		
iLBC	-	13.33 15	-	30 20	Open Source		Alta Robustez contra pérdida de paquetes	
Speex	8 16 32	-	-		Open Source		Usa un bit rate variable para minimizar el consumo del ancho de banda.	

BR = Bit rate

NEB = Nominal Ethernet Bandwidth

De la misma manera se debe considerar parámetros como la capacidad (ancho de banda) de medio físico utilizado para la transmisión de los paquetes, capacidad de procesamiento de la información por parte de servidores y terminales, entre otros parámetros.

A continuación la figura 5.1 muestra un gráfico comparativo entre el ancho de banda y la tasa de bits utilizada por cada *codec* soportado por Asterisk.

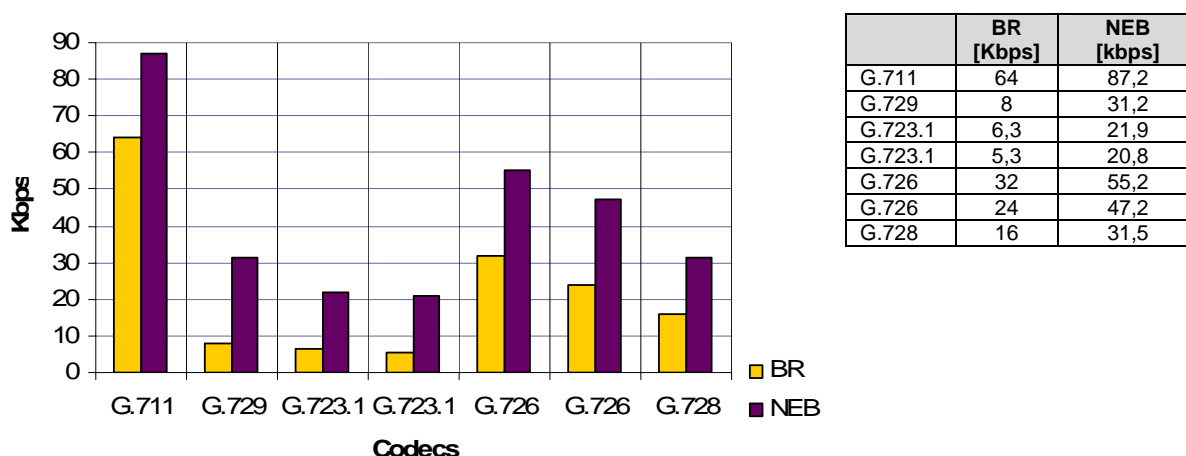


Figura 5. 1. Gráfico comparativo de codecs soportados por Asterisk.

En los que se refiere a la cantidad de tiempo invertido en el establecimiento de la llamada considerado en la telefonía tradicional como el *Call Setup Delay* y relacionado al tiempo transcurrido entre marcar el último dígito y la recepción del ringback, entran a discutirse varios factores que en telefonía IP actúan. Ya que la métrica de ese tiempo se puede simular basándose en el rastreo del retardo y las pérdidas de paquetes UDP factor que particularmente presenta diferencias al ocupar una señalización con el protocolo H.323, que utiliza TCP, y SIP, que usa principalmente UDP; además que entra en juego el desempeño de los servidores SIP, ya que en una llamada las tasas de transmisión pueden alcanzar, varios miles de requerimientos de proceso por segundo.

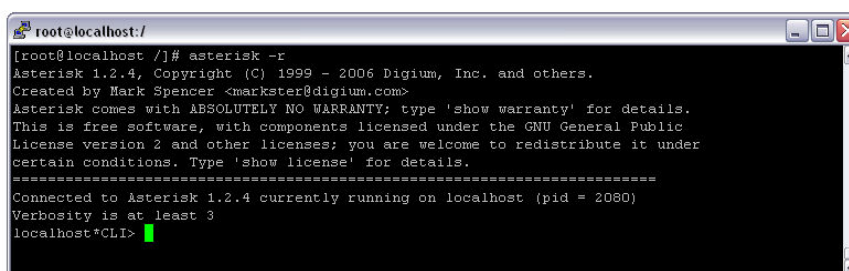
5.3. EVALUACION DEL SISTEMA INSTALADO

Las pruebas realizadas se encuentran ejecutadas sobre un sistema instalado en un servidor las características que muestra la tabla 5.2 y utilizando el *Codec* utilizado es el G.723 u-law.

Tabla 5. 2. Características del servidor Asterisk.

	Características
Procesador	Intel P4 2.0 Ghz.
Memoria RAM	512 Mb,
Disco Duro	4 Gb.
Sistema Operativo	Linux CentOS 4.5 final
Versión Kernel	2.6.9
Versión Asterisk	1.2.4

Una vez instalado el sistema, Asterisk presenta la consola de ejecución CLI (*Command Line Interface*) el cual es el interfaz de usuario para la administración y monitoreo del sistema, los comandos disponibles se encuentran detallados en el anexo 2. El aspecto del CLI se muestra en la figura 5.2.



```
root@localhost:/
[root@localhost /]# asterisk -r
Asterisk 1.2.4, Copyright (C) 1999 - 2006 Digium, Inc. and others.
Created by Mark Spencer <markster@digium.com>
Asterisk comes with ABSOLUTELY NO WARRANTY; type 'show warranty' for details.
This is free software, with components licensed under the GNU General Public
License version 2 and other licenses; you are welcome to redistribute it under
certain conditions. Type 'show license' for details.
=====
Connected to Asterisk 1.2.4 currently running on localhost (pid = 2080)
Verbosity is at least 3
localhost*CLI>
```

Figura 5. 2. Consola de Asterisk 1.2.4

La consola de Asterisk una vez ejecutada, muestra todas las acciones que el sistema realiza, tal como: registro de usuarios, realización de llamadas, errores registrados, etc.

Además se puede agregar programas de administración remota como *FreePBX*, que permiten al usuario final una administración y monitoreo del sistema amigable mediante páginas web, las mismas que modifican los archivos de configuración de Asterisk.

Free PBX es un proyecto de libre, desarrollado por *Atengo LLC* y sus aplicaciones se pueden obtener de forma gratuita desde su sitio web www.freepbx.com. El aspecto de *FreePBX* una vez instalado en el sistema se muestra en la figura 5.3:

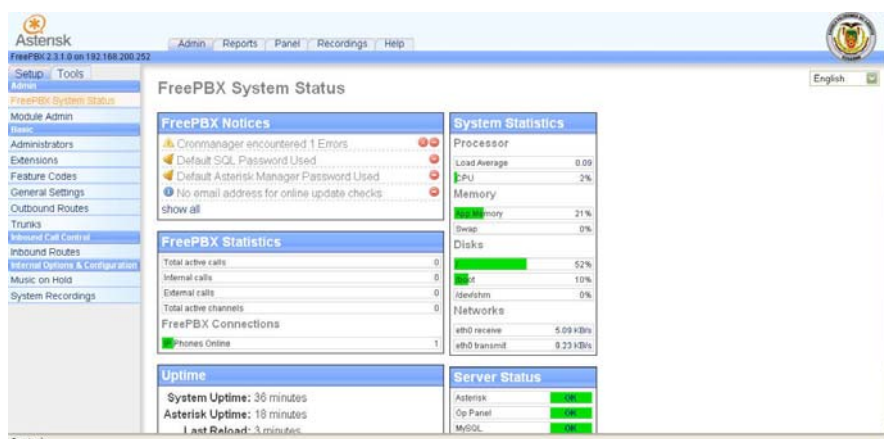


Figura 5. 3. Aspecto de *FreePBX* instalado en Asterisk.

Para medir el tráfico generado dentro de la red Lan en el transcurso de una llamada realizada por el sistema Asterisk, se utilizó el programa de medición *TrafMeter* versión 7.6.450 desarrollado por *MAREL IT Solutions* y el *Softphone X-Lite* versión 3.0 desarrollado por *Counter Path Corporation*, basado en el protocolo señalización *SIP*.

El tráfico generado se muestra en la figura 5.4, en la cual se puede observar que el ancho de banda utilizado por una llamada establecida en el sistema Asterisk es de 9,9 Kbps para el canal de entrada y 10 Kbps para el canal de salida.

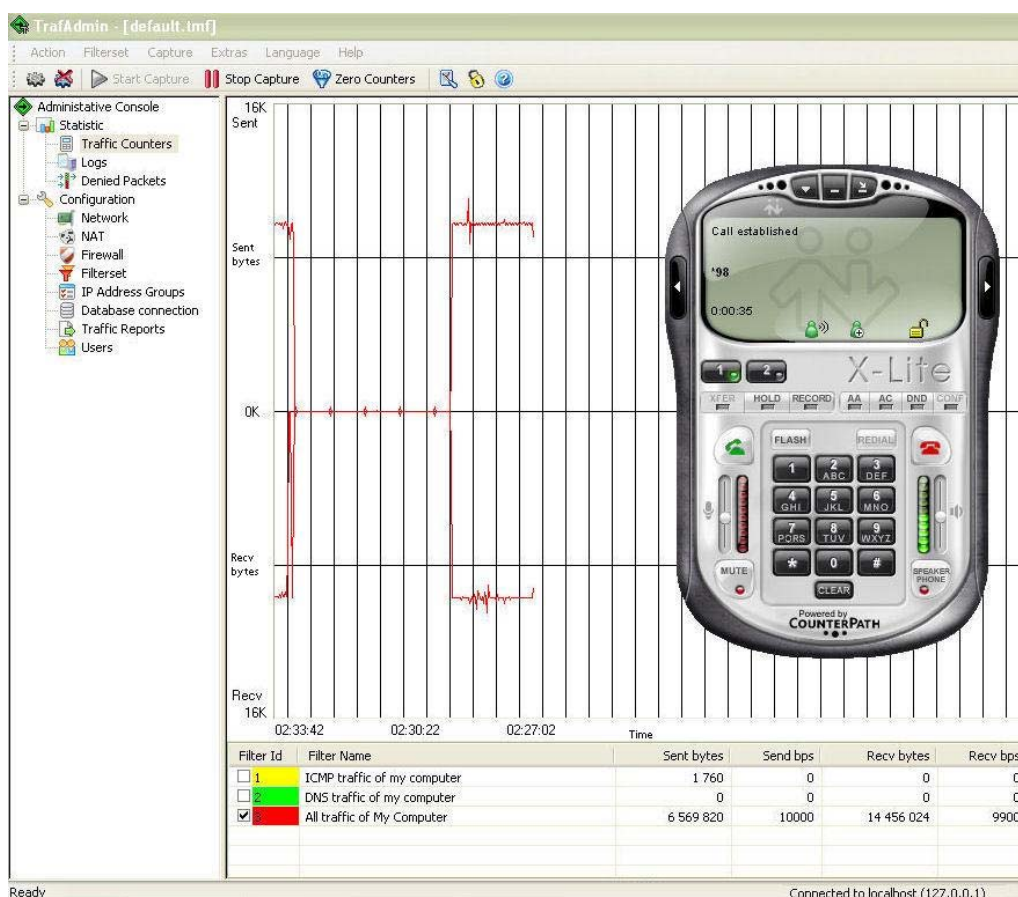


Figura 5. 4. Tráfico generado por una llamada establecida en Asterisk 1.2 Codec G.723 u-law.

En la tabla 5.2 se muestra las 3 distintas etapas que se aprecian en la figura 5.4, como resultado de la medición de tráfico. La primera etapa es una llamada establecida entre dos terminales donde se visualiza el tráfico generado por el establecimiento de una llamada; a continuación un espacio donde no se establece llamada alguna, por lo que el tráfico cursado es casi nulo y por último nuevamente una llamada establecida en donde se muestra de igual forma el tráfico generado por la transmisión de datos entre dos terminales de la misma red. Además se puede apreciar un tiempo de establecimiento de llamada (*Call Delay Setup*) el cual es el tiempo transcurrido en marcar el último dígito y la recepción del tono de timbre.

Tabla 5. 3. Medición de tráfico generado por el sistema.

	Duración [seg.]	Ancho Banda Canal de Entrada [Kbps.]	Ancho Banda Canal de Salida [Kbps.]
Llamada establecida	150	10	10
Espacio silencio	280	0	0
Llamada establecida	40	9,9	10

El detalle de registro de llamadas se almacena en la base de datos, y el reporte se lo obtiene mediante una página web que acceda a ella, los campos que se almacenan son varios, entre ellos están: la fecha de realización de la llamada, el origen, el destino, el canal de origen, el canal de destino, el estado que tuvo la llamada, la duración, la duración incluyendo el tiempo de marcación, entre otros valores. El interfaz implementado se muestra en la figura 5.5 y el reporte de detalle de llamadas muestra algunos de los campos descritos:

Date	Source	Destination	Src. Channel	Dst. Channel	Status	Duration	Recording
2008-01-07 01:14:15	6001	*98	SIP/6001-090ee9e0		ANSWERED	28	Not Recorded
2008-01-07 01:15:49	6001	*98	SIP/6001-09104350		ANSWERED	11	Not Recorded
2008-01-07 01:18:43	6002	*98	SIP/6002-09104350		ANSWERED	53	Not Recorded
2008-01-07 01:24:38	javaca	6002	SIP/6001-099f0040	SIP/6002-09a16f20	ANSWERED	12	Not Recorded
2008-01-07 01:29:20	javaca	6002	SIP/6001-099f0040	SIP/6002-09a16f20	NO ANSWER	0	Not Recorded
2008-01-07 01:34:29	6001	*98	SIP/6001-099f0040		ANSWERED	4	Not Recorded
2008-01-07 01:34:50	javaca	6002	SIP/6001-099f0040	SIP/6002-09a16f20	ANSWERED	16	Not Recorded
2008-01-07 01:37:08	javaca	6002	SIP/6001-099f0040	SIP/6002-09a16f20	ANSWERED	6	Not Recorded
2008-01-07 01:37:31	6001	*98	SIP/6001-099f0040		ANSWERED	3	Not Recorded
2008-01-07 01:38:33	6002	*98	SIP/6002-099f0040		ANSWERED	6	Not Recorded
2008-01-07 01:51:09	6001	*98	SIP/6001-09c079a0		ANSWERED	36	Not Recorded
2008-01-07 01:51:49	javaca	6002	SIP/6001-09c079a0	Local/s@from-pstn-2c9c,1	ANSWERED	61	Not Recorded
2008-01-07 01:52:43	javaca	s	Local/s@from-pstn-2c9c,2		ANSWERED	13	Not Recorded

Figura 5. 5. Reporte de detalle de registro de Llamadas (CDR).

Los *softwares softphone*, realizan también un registro de las acciones realizadas, estos registros se almacenan en el computador y muestran las principales acciones hechas por las terminales dentro del sistema. En este caso el registro del *Softphone X-Lite versión 3.0* desarrollado por *Counter Path Corporation*, se muestra en la tabla 5.4. Es este registro se detallan las fechas, mensajes, errores y otras acciones que realizó el simulador.

Tabla 5. 4. Log de Softphone X-Lite al probar el sistema.

Time	Type	Subsystem	Message	Function
[08-01-09]09:10:14.154	Info (min) GUI	"User Action: (keypad_star)"	eyeBeam::FunkyControl::DoAction	
[08-01-09]11:26:05.612	Info (min) AbstractPhone	"[6001@192.168.200.252]::DisableAccount"	AbPhone::CAccount::DisableAccount	
[08-01-09]11:26:05.652	Info (min) GUI	"Default account set to0"	eyeBeam::CdbAccountManager::OnClose	
[08-01-09]11:26:11.037	Info (min) AbstractPhone	"[6001@192.168.200.252]::EnableAccount"	AbPhone::CAccount::EnableAccount	
[08-01-09]11:26:11.240	Info (min) CCM	"Not using firewall SIP proxy. Reasons: [URI:6001@192.168.200.252]"	sua::CSIPAccess::CTransportCreator::UseFirewallSIP proxy	
[08-01-09]11:26:12.237	Info (min) CCM	"Response code to SIP request did not match any entry specified in retry-response-list. Response: 489[6001@192.168.200.252 -> 6001@192.168.200.252]"	sua::CSIPClientSubscriptionWatcher::OnSIPClientSubscriptionSessionDestroyed	
[08-01-09]11:26:15.398	Info (min) GUI	"User Action: (keypad_6)"	eyeBeam::FunkyControl::DoAction	
[08-01-09]11:26:15.400	Info (min) Utilities	"Detected CPU [AMD Turion(tm) 64 Mobile Technology ML-32]"	sua::CProcessorInformation::Detecti386CPU	
[08-01-09]11:26:15.401	Info (min) Utilities	"Detected CPU [AMD Turion(tm) 64 Mobile Technology ML-32]"	sua::CProcessorInformation::Detecti386CPU	
[08-01-09]11:26:15.441	Info (min) Utilities	"Detected CPU [AMD Turion(tm) 64 Mobile Technology ML-32]"	sua::CProcessorInformation::Detecti386CPU	
[08-01-09]11:26:15.509	Info (min) GUI	"User Action: (keypad_6)"	eyeBeam::FunkyControl::DoAction	
[08-01-09]11:26:15.879	Info (min) GUI	"User Action: (keypad_0)"	eyeBeam::FunkyControl::DoAction	
[08-01-09]11:26:15.879	Info (min) Utilities	"Detected CPU [AMD Turion(tm) 64 Mobile Technology ML-32]"	sua::CProcessorInformation::Detecti386CPU	
[08-01-09]11:26:15.959	Info (min) GUI	"User Action: (keypad_0)"	eyeBeam::FunkyControl::DoAction	
[08-01-09]11:26:16.311	Info (min) GUI	"User Action: (keypad_0)"	eyeBeam::FunkyControl::DoAction	
[08-01-09]11:26:16.311	Info (min) Utilities	"Detected CPU [AMD Turion(tm) 64 Mobile Technology ML-32]"	sua::CProcessorInformation::Detecti386CPU	
[08-01-09]11:26:16.383	Info (min) GUI	"User Action: (keypad_0)"	eyeBeam::FunkyControl::DoAction	
[08-01-09]11:26:16.775	Info (min) GUI	"User Action: (keypad_2)"	eyeBeam::FunkyControl::DoAction	
[08-01-09]11:26:16.775	Info (min) Utilities	"Detected CPU [AMD Turion(tm) 64 Mobile Technology ML-32]"	sua::CProcessorInformation::Detecti386CPU	
[08-01-09]11:26:16.828	Info (min) GUI	"User Action: (keypad_2)"	eyeBeam::FunkyControl::DoAction	
[08-01-09]11:26:17.519	Info (min) GUI	"User Action: Dial (cmd_dial)"	eyeBeam::FunkyControl::DoAction	
[08-01-09]11:26:17.532	Info (min) AbstractPhone	"[6001@192.168.200.252]::MakeCallRequest : sip:6002@192.168.200.252"	AbPhone::CAccount::MakeCallRequest	
[08-01-09]11:26:18.043	Info (min) Utilities	"Detected CPU [AMD Turion(tm) 64 Mobile Technology ML-32]"	sua::CProcessorInformation::Detecti386CPU	
[08-01-09]11:26:18.138	Info (min) Utilities	"Detected CPU [AMD Turion(tm) 64 Mobile Technology ML-32]"	sua::CProcessorInformation::Detecti386CPU	
[08-01-09]11:26:21.116	Info (min) RTP	"RTCP calculation 154 154 154 154 0"	sua::CRTPDestination::AppendAndNotifyAboutReceiver Report	
[08-01-09]11:26:22.140	Warning (min) Jitter	"Initiating jitter buffer increase"	sua::CJitterBuffer::HandleOneSecondTick	
[08-01-09]11:26:24.186	Info (min) RTP	"RTCP calculation 295 295 141 141 0"	sua::CRTPDestination::AppendAndNotifyAboutReceiver Report	
[08-01-09]11:26:25.217	Info (min) GUI	"User Action: Hang Up (cmd_hangup)"	eyeBeam::FunkyControl::DoAction	

CAPITULO VI

CONCLUSIONES Y RECOMENDACIONES

6.1. CONCLUSIONES

- La telefonía convencional basa sus redes en una conmutación de circuitos, en la cual se establece un canal dedicado durante el tiempo de establecimiento de una llamada y se libera el mismo una vez terminada la comunicación. Sobre esta plataforma diseñada para transmitir voz se adaptaba la transmisión de datos.
- La telefonía IP basa sus redes en una conmutación de paquetes de datos, es decir, que no es necesario establecer un canal dedicado para la transmisión de información, sino que los paquetes pueden transmitirse por distintas rutas. La telefonía IP transforma la voz en una trama de datos, por lo cual la voz es transportada por sistemas de transmisión de datos.
- La principal razón por la que la telefonía IP reduce los costos, se basa en el desarrollo y mantenimiento de una sola red de datos y no en el desarrollo de redes de datos y voz por separado.
- No existe un marco legal vigente para la VoIP y Telefonía IP en el Ecuador, por lo que se debe promover un marco legal que propicie la competencia y favorezca a los usuarios con tecnología y bajos costos, y que esta a su vez sea el motor que reduzca la brecha digital y genere oportunidades de educación, empleo, productividad y riqueza.
- Asterisk es una completa solución de una pequeña central telefónica IP por software que se instala sobre plataforma de servidor con sistema operativo

Linux (GNU Linux), y junto con los convenientes interfaces de telefonía tradicional (líneas analógicas o RDSI) que convierte a dicho sistema en una potente centralita telefónica. Proporciona todas las funcionalidades de las grandes centralitas propietarias (buzones de voz, IVR, etc.) y ofrece algunas posibilidades y servicios no disponibles en la mayoría de ellos (grabación de llamadas, extensiones remotas). Además, por su arquitectura abierta y por su tecnología VoIP aporta importantes ventajas en costes y capacidades frente a los sistemas de telefonía convencionales.

- El crecimiento a largo plazo de la telefonía IP se verá impulsado por la mayor flexibilidad en los servicios ofrecidos por las redes IP, comparadas a la telefonía tradicional. Esta flexibilidad se deriva de las capacidades de señalización de los sistemas IP, superiores en versatilidad a los de la telefonía tradicional. Sin embargo, para convertirse en un sistema de comunicaciones de escala global, la telefonía IP debe al menos igualar a la telefonía tradicional en aspectos como: calidad de servicio, soporte a usuarios y soporte a operadores.
- Actualmente se encuentra disponible en internet mucha información acerca de las múltiples herramientas que permiten no solo medir el desempeño de las aplicaciones de telefonía IP, sino que también se oferta una infinidad de productos y soluciones que complementan los sistemas de telefonía IP basados en Asterisk.
- El desarrollo del sistema mediante configuración de archivos de Asterisk, permite visualizar de una manera mas amplia la versatilidad que ofrece este sistema de telefonía IP para numerosas aplicaciones disponibles dentro de los sistemas de telefonía, como por ejemplo: respuestas interactivas de voz (IVR) y aplicaciones específicas mediante emisión de tonos de llamada; sin embargo puede convertirse muy complejo para usuarios que estén iniciando en Asterisk, por lo que existe versiones autoejecutables (*Live CD*) como: TrixBox y AsteriskNow que brindan al usuario final un entorno de programación y administración mucho más amigable, en el cual se puede desarrollar sistemas básicos de telefonía IP,

basados en Asterisk, mediante la manipulación de paginas web que accedan y modifican los archivos de configuración de Asterisk.

- Se debe proyectar correctamente la capacidad del sistema de telefonía IP, debido a que el servidor ejecuta funciones de proceso de datos por cada usuario o terminal, un subdimensionamiento de esto, puede ocasionar varios problemas en la ejecución de llamadas como: retardos en el establecimiento de llamadas o disminuir la calidad de voz.

6.2. RECOMENDACIONES

- Es importante seleccionar el codec adecuado para el sistema de telefonía IP, con el fin de garantizar una calidad de voz a un ancho de banda adecuado y poder estimar el tráfico cursado en el canal de comunicación. En este caso en particular, el codec utilizado es G.723, debido a que brinda una alta calidad de voz.
- Existen diversos enfoques en la evaluación de los sistemas de telefonía IP, por lo cual se recomienda un adecuado análisis de los recursos a utilizar dentro de los sistemas, los cuales nos garantizaran un mejor desempeño y una adecuada robustez de las aplicaciones existentes en el mercado que contribuirán a que la telefonía IP siga sumando adherentes y se posicione como una alternativa cotidiana al sistema telefónico tradicional.
- La importancia que han tomando los sistemas de telecomunicaciones a nivel empresarial, obliga a la implantación de tecnologías que reduzcan costos de implementación y mantenimiento por lo que la utilización de sistemas de telefonía IP basados en software de código abierto es una buena opción para la pequeña y mediana industria (PYME's) frente a sistemas de telefonía tradicional.

BIBLIOGRAFIA

- [1] <http://www.rfc-es.org/rfc/rfc0791-es.txt>
- [2] http://www.geocities.com/une_chikhani/Paginas/INTERNET.htm
- [3] http://www.cisco.com/global/ES/solutions/smb/avvid_solutions/iptel_home.shtml
- [4] <http://www.uol.com.ar/noticias/tecnologia/datadelanet/20050310/nota2.html>
- [5] <http://www.monografias.com/trabajos16/telefonía-senalización/telefonía-senalización.shtml>
- [6] <http://users.tkk.fi/~lhuovine/study/iwork99/voip.html>
- [7] <http://www.aslan.es/boletin/boletin30/acterna.shtml>
- [8] MEGGELEN, SMITH, MADSEN; Asterisk™: The Future of Telephony, 1ra. edición, O'Reilly Media, Inc; Estados Unidos de América 2005, 358 p.
- [9] <http://www.voipforo.com/voip.css>
- [10] <http://www.conatel.gov.ec/website/baselegal/leyes.php>
- [11] <http://www.alfa-redi.org>
- [12] <https://ourproject.org/projects/asterix>
- [13] <http://www.vozdigital.org/modules.php?op=modload&name=PNrowiki&file=index&page=Codecs>
- [14] <http://profesores.elo.utfsm.cl/~agv/ipd438/2s03/projects/Protocols4Multimedia/Informes/Informe%203.pdf>

ANEXO 1

**PLAN DE MERCADO DEPARTAMENTO DE
ELECTRICA Y ELECTONICA**

PLAN DE MARCADO DEPARTAMENTO DE ELECTRICA Y ELECTONICA

De acuerdo al personal que labora en el departamento de eléctrica y electrónica de la ESPE se elaboró es siguiente plan de marcado:

Extensión Asignada	Usuario	Nombre de usuario	Contraseña	Contraseña Correo de Voz
1001	Fabian Saenz	fabian.saenz	1001	1001
1002	Carlos Romero	carlos.romero	1002	1002
1003	Rubén León	ruben.leon	1003	1003
1004	Román Lara	roman.lara	1004	1004
1005	Gonzalo Olmedo	gonzalo.olmedo	1005	1005
1006	Xavier Martínez	xavier.martinez	1006	1006
2001	Darío Duque	dario.duque	2001	2001
2002	Rodrigo Silva	rodrigo.silva	2002	2002
2003	Ramiro Ríos	ramiro.rios	2003	2003
2004	Evelio Granizo	evelio.granizo	2004	2004
2005	Dérlin Morocho	derlin.morocho	2005	2005
3001	Jaime Andrango	jaime.andrango	3001	3001
3002	Julio Larco	julio.larco	3002	3002
3003	Víctor Proaño	victor.proano	3003	3003
4001	Rodolfo Gordillo	rodolfo.gordillo	4001	4001
4002	Alejandro Chacón	alejandro.chacon	4002	4002
4003	Hugo Ortiz	hugo.ortiz	4003	4003
4004	Wilson Yépez	wilson.yepetz	4004	4004
5001	Paúl Ayala	paul.ayala	5001	5001
5002	Edwin Chávez	edwin.chavez	5002	5002
5003	David Andrade	david.andrade	5003	5003
5004	Paúl Vizcaíno	paul.vizcaino	5004	5004
5005	Don Berniz	don.berniz	5005	5005
6001	Jose Vaca	jose.vaca	6001	6001

ANEXO 2

COMANDOS CONSOLA DE ASTERISK

COMANDOS CONSOLA DE ASTERISK

!	Execute a shell command
abort halt	Cancel a running halt
add extension	Add new extension into context
add ignorepat	Add new ignore pattern
add queue member	Add a channel to a specified queue
ael debug contexts	Enable AEL contexts debug
ael debug macros	Enable AEL macros debug
ael debug read	Enable AEL read debug
ael debug tokens	Enable AEL tokens debug
ael no debug	Disable AEL debug messages
ael reload	Reload AEL configuration
agent logoff	Sets an agent offline
agi debug	Enable AGI debugging
agi no debug	Disable AGI debugging
answer	Answer an incoming console call
autoanswer	Sets/displays autoanswer
cdr status	Display the CDR status
console	Sets/displays active console
database del	Removes database key/value
database deltree	Removes database keytree/values
database get	Gets database value
database put	Adds/updates database value
database show	Shows database contents
database showkey	Shows database contents
debug channel	Enable debugging on a channel
debug level	Set global debug level
dial	Dial an extension on the console
dnsmgr reload	Reloads the DNS manager configuration
dnsmgr status	Display the DNS manager status
dont include	Remove a specified include from context
dump agihtml	Dumps a list of agi command in html format
dundi debug	Enable DUNDi debugging
dundi flush	Flush DUNDi cache
dundi lookup	Lookup a number in DUNDi
dundi no debug	Disable DUNDi debugging
dundi no store history	Disable DUNDi historic records
dundi precache	Precache a number in DUNDi

dundi query	Query a DUNDi EID
dundi show entityid	Display Global Entity ID
dundi show mappings	Show DUNDi mappings
dundi show peers	Show defined DUNDi peers
dundi show peer	Show info on a specific DUNDi peer
dundi show precache	Show DUNDi precache
dundi show requests	Show DUNDi requests
dundi show trans	Show active DUNDi transactions
dundi store history	Enable DUNDi historic records
extensions reload	Reload extensions and *only* extensions
feature show channels	Show status of feature channels
flash	Flash a call on the console
group show channels	Show active channels with group(s)
hangup	Hangup a call on the console
help	Display help list or specific help on a command
iax2 debug	Enable IAX debugging
iax2 jb debug	Enable IAX jitterbuffer debugging
iax2 no debug	Disable IAX debugging
iax2 no jb debug	Disable IAX jitterbuffer debugging
iax2 no trunk debug	Disable IAX trunk debugging
iax2 provision	Provision an IAX device
iax2 prune realtime	Prune a cached realtime lookup
iax2 reload	Reload IAX configuration
iax2 set jitter	Sets IAX jitter buffer
iax2 show cache	Display IAX cached dialplan
iax2 show channels	Show active IAX channels
iax2 show firmware	Show available IAX firmwares
iax2 show netstats	Show active IAX channel netstats
iax2 show peer	Show details on specific IAX peer
iax2 show peers	Show defined IAX peers
iax2 show provisioning	Show iax provisioning
iax2 show registry	Show IAX registration status
iax2 show stats	Display IAX statistics
iax2 show users	Show defined IAX users
iax2 test losspect	Set IAX2 incoming frame loss percentage
iax2 trunk debug	Enable IAX trunk debugging
include context	Include context in other context
indication add	Add the given indication to the country
indication remove	Remove the given indication from the country
init keys	Initialize RSA key passcodes

load	Load a dynamic module by name
local show channels	Show status of local channels
logger reload	Reopens the log files
logger rotate	Rotates and reopens the log files
logger show channels	List configured log channels
meetme	Execute a command on a conference or conferee
mgcp audit endpoint	Audit specified MGCP endpoint
mgcp debug	Enable MGCP debugging
mgcp no debug	Disable MGCP debugging
mgcp reload	Reload MGCP configuration
mgcp show endpoints	Show defined MGCP endpoints
mixmonitor	Execute a MixMonitor command
moh classes show	List MOH classes
moh files show	List MOH file-based classes
moh reload	Music On Hold
mute	Disable mic input
no debug channel	Disable debugging on a channel
pri debug span	Enables PRI debugging on a span
pri intense debug span	Enables REALLY INTENSE PRI debugging
pri no debug span	Disables PRI debugging on a span
pri set debug file	Sends PRI debug output to the specified file
pri show debug	Displays current PRI debug settings
pri show span	Displays PRI Information
pri unset debug file	Ends PRI debug output to file
realtime load	Used to print out RealTime variables.
realtime update	Used to update RealTime variables.
reload	Reload configuration
remove extension	Remove a specified extension
remove ignorepat	Remove ignore pattern from context
remove queue member	Removes a channel from a specified queue
restart gracefully	Restart Asterisk gracefully
restart now	Restart Asterisk immediately
restart when convenient	Restart Asterisk at empty call volume
rtp debug	Enable RTP debugging
rtp debug ip	Enable RTP debugging on IP
rtp no debug	Disable RTP debugging
save dialplan	Save dialplan
send text	Send text to the remote device
set debug	Set level of debug chattiness
set verbose	Set level of verbosity

show agents	Show status of agents
show agi	Show AGI commands or specific help
show applications	Shows registered dialplan applications
show application	Describe a specific dialplan application
show audio codecs	Shows audio codecs
show channel	Display information on a specific channel
show channels	Display information on channels
show channeltypes	Show available channel types
show codecs	Shows codecs
show codec	Shows a specific codec
show conferences	Show status of conferences
show config mappings	Show Config mappings (file names to config engines)
show dialplan	Show dialplan
show features	Lists configured features
show file formats	Displays file formats
show functions	Shows registered dialplan functions
show function	Describe a specific dialplan function
show hints	Show dialplan hints
show image codecs	Shows image codecs
show image formats	Displays image formats
show indications	Show a list of all country/indications
show keys	Displays RSA key information
show license	Show the license(s) for this copy of Asterisk
show manager command	Show a manager interface command
show manager commands	List manager interface commands
show manager connected	Show connected manager interface users
show modules	List modules and info
show modules like	List modules and info
show parkedcalls	Lists parked calls
show queue	Show status of a specified queue
show queues	Show status of queues
show switches	Show alternative switches
show translation	Display translation matrix
show uptime	Show uptime information
show version	Display version info
show version files	Show versions of files used to build Asterisk
show video codecs	Shows video codecs
show voicemail users	List defined voicemail boxes
show voicemail zones	List zone message formats
show warranty	Show the warranty (if any) for this copy of Asterisk

sip debug	Enable SIP debugging
sip debug ip	Enable SIP debugging on IP
sip debug peer	Enable SIP debugging on Peername
sip history	Enable SIP history
sip no debug	Disable SIP debugging
sip no history	Disable SIP history
sip notify	Send a notify packet to a SIP peer
sip prune realtime	Prune cached Realtime object(s)
sip prune realtime peer	Prune cached Realtime peer(s)
sip prune realtime user	Prune cached Realtime user(s)
sip reload	Reload SIP configuration
sip show channels	Show active SIP channels
sip show channel	Show detailed SIP channel info
sip show domains	List our local SIP domains.
sip show history	Show SIP dialog history
sip show inuse	List all inuse/limits
sip show objects	Show all SIP object allocations
sip show peer	Show details on specific SIP peer
sip show peers	Show defined SIP peers
sip show registry	Show SIP registration status
sip show settings	Show SIP global settings
sip show subscriptions	Show active SIP subscriptions
sip show users	Show defined SIP users
sip show user	Show details on specific SIP user
skinny debug	Enable Skinny debugging
skinny no debug	Disable Skinny debugging
skinny show devices	Show defined Skinny devices
skinny show lines	Show defined Skinny lines per device
soft hangup	Request a hangup on a given channel
stop gracefully	Gracefully shut down Asterisk
stop now	Shut down Asterisk immediately
stop when convenient	Shut down Asterisk at empty call volume
transfer	Transfer a call to a different extension
unload	Unload a dynamic module by name
unmute	Enable mic input
zap destroy channel	Destroy a channel
zap show cadences	List cadences
zap show channels	Show active zapata channels
zap show channel	Show information on a channel
zap show status	Show all Zaptel cards status

ANEXO 3

REFERENCIA DE APLICACIONES

REFERENCIA DE APLICACIONES

General Commands

Aplicación	Descripción	Sintaxis
Authenticate:	Authenticate a user	Authenticate(password[options]) Authenticate(/passwdfile[options]) Authenticate(/db-keyfamily d[options])
VMAuthenticate:	Authenticate a user based on voicemail.conf	VMAuthenticate([mailbox][@context][options]):
ChannelRedirect:	Redirect an existing channel to the dialplan	ChannelRedirect(channel[[context]extension]priority)
DUNDiLookup:	Look up a number with DUNDi	DUNDiLookup(number[[context][options]])
Page:	Page a mobile device (new in Asterisk v1.2)	Page(<i>Technology/Resource&Tech2/Res2...</i> [options])
SendDTMF:	Sends arbitrary DTMF digits	SendDTMF(digits[timeout_ms])
SendImage:	Send an image file	SendImage(filename)
SendText:	Send client a text message	SendText(text)
SendURL:	Send a client a URL to display	SendURL(URL[option])
System:	Execute a system command	System(command)
Transfer	Transfer caller to remote extension	Transfer([Tech/]dest[options])
TrySystem	Execute a system command with always 0 returned	TrySystem(command)
Wait	Waits for some time	Wait(seconds)
WaitExten	Waits for some time for caller to dial a new extension	WaitExten(seconds) WaitExten([seconds][options])
WaitForRing	Wait for Ring Application	WaitForRing(timeout)
WaitMusicOnHold	Wait, playing Music On Hold	WaitMusicOnHold(delay)

Billing

Aplicación	Descripción	Sintaxis
ForkCDR	Fork The CDR into 2 separate entities	ForkCDR()
NoCDR	Make sure asterisk doesn't save CDR for a certain call	NoCDR()
ResetCDR	Reset CDR data	ResetCDR([options])
SetAccount	Sets account code	SetAccount([account])
SetAMAflags	Set the channel AMA Flags for billing	SetAMAFlags(flags)

SetCDRUserField	Set CDR User field	SetCDRUserField(Value)
AppendCDRUserField	Append data to CDR User field	AppendCDRUserField(Value)

Call Management (hangup, answer, dial, etc)

Aplicación	Descripción	Sintaxis
Answer	Answer a channel if ringing	Answer([delay])
Busy	Indicate busy condition and wait for hangup	Busy()
ChanIsAvail	Check if channel is available	ChanIsAvail (Tech/Res[&Tech2/Res2][[options]])
Congestion	Indicate congestion and wait for hangup	Congestion()
Dial	Place a call and connect to the current channel	Dial(type/identifier, timeout, options, URL)
DISA	DISA (Direct Inward System Access)	DISA(<i>passcode</i> [[<i>context</i>]]) DISA(<i>password file</i>)
Hangup	Unconditional hangup	Hangup()
RetryDial	Place a call, retrying on failure allowing optional exit extension	RetryDial()
Ringing	Indicate ringing	Ringing()

Caller Presentation (ID, Name etc)

Aplicación	Descripción	Sintaxis
CallingPres	Change the presentation for the callerid	CallingPres(<i>number</i>)
LookupBlacklist	Look up Caller*ID name/number from blacklist database	LookupBlacklist
LookupCIDName	Look up CallerID Name from local database	LookupCIDName
PrivacyManager	Require phone number to be entered, if no CallerID sent	PrivacyManager
SetCallerID	Set CallerID. Deprecated in favor of CALLERID.	SetCallerID("Name" <Number>[[a]])
SetCallerPres	Channel independent setting of caller presentation	SetCallerPres(presentation):
SetCIDName	Set CallerID Name. Deprecated in favor of CALLERID.	Set(CALLERID(name)=name[[a]])

SetCIDNum	Set only the Caller ID number (not name). Deprecated in favor of CALLERID.	Set(CALLERID(number)=<number>)
SoftHangup	Request hangup on another channel	SoftHangup(Technology/resource[[a]])
Zapateller	Block telemarketers with SIT	Zapateller(options)

ADSI

Aplicación	Descripción	Sintaxis
ADSIProg	Load Asterisk ADSI Scripts into phone	ADSIProg(script)
GetCPEID	Get ADSI CPE ID	GetCPEID()

Database Handling

Aplicación	Descripción	Sintaxis
DBdel	Delete a key from the database.	DBdel(family/key)
DBdeltree	Delete a family or keytree from the database.	DBdeltree(family/keytree)
DBget	Retrieve a value from the database. Deprecated in favor of DB.	DBget(varname=family/key)
DBput	Store a value in the database. Deprecated in favor of DB.	DBput(family/key=value)
MYSQL	Perform various mySQL database activities	MYSQL(): Basic MYSQL Database Functionality
DBQuery	Execute predefined queries against MySQL Servers, and get the result back into the dialplan.	
RealTime	Populate variables with details from database using RealTime	RealTime(<family> <colmatch> <value>[<prefix>])
RealTimeUpdate	Update a field in a database using RealTime	

Application Integration

Aplicación	Descripción	Sintaxis
AGI	Executes an AGI compliant	

	application	
DeadAGI	Executes AGI on a hung-up channel	
EAGI	Executes an AGI compliant application with sound channels	
EnumLookup	Lookup number in ENUM	
ExternallVR	Executes an ExternallVR generator	
Macro	Macro Implementation	Macro(macroname,arg1,arg2...)
MacroExclusive	Only one channel at a time may call this macro, all others have to wait (1.4)	
MacroExit	Exit the macro as if it had fully completed (1.4)	
NoOp	No operation. Can print values to console for debugging.	NoOp()
Perl	res_perl is the mod_perl of Apache, only for Asterisk	
PHP	res_php integrates PHP into Asterisk without AGI	PHP(some.php hello world)
Read	Read a variable with DTMF	Read(variable[[filename]][[maxdigits]][[option]])
TXTCIDName	Lookup caller name from TXT record	TXTCIDName(CallerID[[options]])
UserEvent	Send an arbitrary event to the manager interface	UserEvent(eventname[[body]])

Control Flow & Timeouts

Aplicación	Descripción	Sintaxis
AbsoluteTimeout	Set absolute maximum time of call	AbsoluteTimeout(seconds)
DigitTimeout	Set maximum timeout between digits	DigitTimeout(seconds)
Gosub	Jump to a subroutine and return	Gosub([[context]]extension[[priority]])
Gosublf	Conditional jump to a subroutine and return	Gosublf(condition?label1:label2)
Goto	Goto a particular priority, extension, or context	Goto([[context]]extension[[priority]])
Gotolf	Conditional goto	Gotolf(condition?label1[:label2])
GotolfTime	Conditional goto on current time	GotolfTime()

Random	jump to a specified location based on a random probability	Random(probability:label)
ResponseTimeout	Set maximum timeout awaiting response	ResponseTimeout(seconds)
Return	Return from a Gosub or Gosublf	Return()
StackPop	Remove a return address without returning	StackPop()
While	Start A While Loop	While(condition)
EndWhile	End A While Loop	EndWhile()
ExecIf	Conditional exec	ExecIf (<expr> <app> <data>)

String & Variable manipulation

Aplicación	Descripción	Sintaxis
ImportVar	Set variable to value	ImportVar(newvar=channelname variable)
Math	Perform (rather simple) calculations. Deprecated in favor of MATH.	math(returnvar,operation,number)
SetGlobalVar	Set variable to value. Deprecated in favor of GLOBAL.	SetGlobalVar(foo=value)
Set	Set channel variable(s) or function value(s)	Set(variablename=value[[variable2=value2]][options])
DBRewrite	Execute perl compatible regular expression and substitution out of a MySQL Database.	

Sounds: Playback

Aplicación	Descripción	Sintaxis
Background	Play a sound file while waiting an extension	Background(filename1 options)
BackgroundDetect	Background a file with talk detect	BackgroundDetect(filename[[silence[[min[[max]]]])
ControlPlayback	Play a sound file with fast forward, rewind and exit controls	ControlPlayback(filename)
DateTime	Say the date and/or time (Obsolete)	DateTime()
Echo	Echo audio read back to the user	Echo()

Festival	Say text with the Festival voice synthesizer	Festival(<i>text,intkeys</i>)
Flite	Say text with the Festival Lite voice synthesizer (faster response than Festival)	
Milliwatt	Generate a Constant 1000Hz tone at 0dbm (mu-law)	Milliwatt()
MP3Player	Play an MP3 sound file or stream	MP3Player(<i>location</i>)
MusicOnHold	Play Music On Hold indefinitely	MusicOnHold([<i>class</i>])
Playback	Play a sound file	Playback(<i>filename,options...</i>)
Playtones	Play a tone list while executing other commands	Playtones(<i>tonename</i>)
Progress	Play early audio to the caller before answering the line	Progress()
SayUnixTime	Say the date and/or time	SayUnixTime(<i>unixtime,timezone,format</i>)
SayAlpha	Say Alpha	SayAlpha(<i>string</i>)
SayDigits	Say Digits	SayDigits(<i>digits</i>)
SayNumber	Say Number	SayNumber(<i>number, gender</i>)
SayPhonetic	Say Phonetic	SayPhonetic(<i>text</i>)
SetMusicOnHold	Set default Music On Hold class	SetMusicOnHold(<i>class</i>)
SetLanguage	Change language setting for sound playback	SetLanguage(<i>language</i>)
StopPlaytones	Stop playing a tone list	StopPlaytones()

Sounds: Recording and monitoring (listening-in)

Aplicación	Descripción	Sintaxis
ALSAMonitor	Monitor the ALSA console	ALSAMonitor(<i>password</i>)
ChangeMonitor	Change monitoring filename of a channel	ChangeMonitor(<i>basename</i>)
ChanSpy	Universal channel barge-in	Chanspy([<scanspec>][<options>])
Dictate	Records and plays back a dictation	Dictate([<base_dir>[<filename>]])
ExtenSpy	Listen/whisper to a specific extension	ExtenSpy(<i>extension@context,options</i>)
MixMonitor	Record and mix call legs natively (unlike Monitor)	MixMonitor(<file>.<ext>[<options>[<command>]])
Monitor	Record a telephone conversation to a sound file	Monitor(<i>ext,basename,flags</i>)
Record	Record user voice input to a file	Record(<i>filename.format[<silence>][<maxduration>][<opt>]</i>)

StopMonitor	Stop monitoring a channel	StopMonitor
StopMixMonitor	Stop monitoring a channel monitored with MixMonitor	StopMixMonitor

SIP Commands

Aplicación	Descripción	Sintaxis
SIPdtmfMode	Change DTMF mode during SIP call	SIPDtmfMode(inband info rfc2833)
SIPGetHeader	Pick any header from a SIP invite message (replaced by SIP_HEADER())	SIP_HEADER(<name>)
SIPAddHeader	Add header to outbound SIP invite	SIPAddHeader()

ZAP commands

Aplicación	Descripción	Sintaxis
Flash	Flashes a Zap Trunk	Flash()
ZapBarge	Barge in (monitor) Zap channel	ZapBarge(channel)
ZapCD	ISDN call deflection (bristuff)	ZapCD (bristuff)
ZapEC	Echo cancellation on/off (bristuff)	
ZapSendKeypadFacility	Send digits out of band over a PRI	ZapSendKeypadFacility(digits)
ZapRAS	Provide ISDN data service	ZapRAS(args)
ZapScan	Scan Zap channels to monitor calls	

Voicemail and conferencing

Aplicación	Descripción	Sintaxis
Directory	Provide directory of voicemail extensions	Directory(vm-context[dial-context[options]])
HasNewVoicemail	Conditionally branches to priority + 101	HasNewVoicemail(vmbox[/folder][@context][varname[options]]) HasVoicemail(vmbox[/folder][@context][varname[options]])
MailboxExists	Checks if mailbox exists	MailboxExists(mailbox[@context])
MeetMe	Simple MeetMe conference bridge	MeetMe([confno][. options][. pin])
MeetMeAdmin	MeetMe conference	MeetMeAdmin(confno,command[,user])

	Administration	
MeetMeCount	MeetMe participant count	MeetMeCount(confno[var])
MiniVM	Mini-Voicemail (new in v1.6)	
VoiceMail	Leave a voicemail message	VoiceMail([flags]boxnumber[@context]&boxnumber2[@context]&boxnumber3)
VoiceMailMain	Enter voicemail system	VoiceMailMain([s]mailbox@context)
VMAuthenticate	Authenticate a user based on voicemail.conf	

Queue and ACD management

Aplicación	Descripción	Sintaxis
AddQueueMember	Dynamically adds queue members	AddQueueMember(queueName[interface][penalty])
AgentCallbackLogin	Call agent callback login	AgentCallbackLogin([AgentNo][exten]@context) AgentCallbackLogin([AgentNo][Options][exten]@context)
AgentLogin	Call agent login	AgentLogin(AgentNo[options])
AgentMonitorOutgoing	Record agent's outgoing call	AgentMonitorOutgoing(options)
ParkAndAnnounce	Park and Announce	
ParkedCall	Answer a parked call	ParkedCall(exten)
PauseQueueMember	Pauses an agent	PauseQueueMember([queueName]agent[options])
Queue	Queue a call for a call queue	Queue(queueName[options][URL][announceoverride][timeout][AGI])
RemoveQueueMember	Dynamically removes queue members	RemoveQueueMember(queueName[interface])
UnpauseQueueMember	Resumes an agent	UnPauseQueueMember([queueName]agent[options])

Bristuff application

Aplicación	Descripción	Sintaxis
PickUp	Mostly channel independent.	PickUp (bristuff)
PickUpChan	Pick up the specified channel	PickupChan(Technology/resource[&Technology2/resource2...])
PickDown	Hang up on a remotely ringing call	PickDown([group])
Steal	Take over a bridged call (leg)	Steal([group])
Devstate	Generate a device state change event (inuse, busy,	devstate(device,state)

	ringing ...)	
Segfault	Crash Asterisk with segfault	
ZapCD	ISDN call deflection	ZapCD (bristuff)
ZapEC	Enable or disable echo cancellation for Zap	
ZapInband	Inband call progress (pre-answer)	
Autoanswer	Autoanswer a call for a specified extension	
AutoanswerLogin	Login to the autoanswer application	

Applications for Sirrix channels

Aplicación	Descripción	Sintaxis
SrxEchoCan	Disable/enable Echo Cancellation	SrxEchoCan(enable)
SrxDeflect	Deflect an incoming call	SrxDeflect(number[pres_allowed]) SrxReroute(number[pres_allowed])
SrxMWI	Set / reset MessageWaitingIndication (MWI) on a Sirrix group	SrxMWI(group[digits_to_append][number_to_set]] sender_number msg_count)

