



**UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA**

**IMPLEMENTACIÓN DE UNA CENTRAL TELEFÓNICA IP ASTERISK
PARA LAS PRÁCTICAS DE LABORATORIO EN EL CURSO EL629,
SISTEMAS DE CONMUTACIÓN TELEFÓNICA**

**MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL ELECTRICISTA**

BENJAMÍN FRANCISCO ARRATIA PÉREZ

**PROFESOR GUÍA:
LUIS CASTILLO BARROS**

**MIEMBROS DE LA COMISIÓN
HÉCTOR AGUSTO ALEGRÍA
JOACHIM HINTZE GOEBEL**

**SANTIAGO DE CHILE
JUNIO 2011**

RESUMEN DE LA MEMORIA
PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL ELECTRICISTA
POR: BENJAMÍN ARRATIA PEREZ.
FECHA: 28/06/2011
PROF. GUÍA: Sr. LUIS CASTILLO B.

“IMPLEMENTACIÓN DE UNA CENTRAL TELEFÓNICA IP ASTERISK
PARA LAS PRÁCTICAS DE LABORATORIO EN EL CURSO EL629,
SISTEMAS DE CONMUTACIÓN TELEFÓNICA”.

El objetivo general del presente trabajo de título es diseñar e implementar un Sistema Telefónico *IP* (*Internet Protocol*, Protocolo de Internet) utilizando Asterisk, para desarrollar prácticas de laboratorio en el curso EL629, y redactar las correspondientes guías de laboratorio. Estas prácticas se desarrollarán durante el próximo semestre, en 4 laboratorios que permitirán poner en práctica los conceptos estudiados en las clases de telefonía *IP*.

La maqueta de telefonía *IP* que, como producto de este trabajo de título se implementó en el Laboratorio de Postítulo del DIE (Departamento de Ingeniería Eléctrica), cumple con los requisitos planteados inicialmente. Como el sistema telefónico Asterisk es de código abierto y desarrollado en un ambiente colaborativo, y la infraestructura utilizada corresponde únicamente a recursos disponibles en el Departamento, la implementación de la maqueta no requiere de equipamiento adicional, y cumple con las restricciones de presupuesto existentes.

El sistema de telefonía *IP* ó maqueta que se implementó en los *PC's* (Computadores Personales) del laboratorio de Postítulo del DIE, constituido por: 2 Centrales Asterisk (software sobre sistema operativo Linux), 6 líneas *Softphone*, 1 *Analog Telephone Adapter* (ATA) y 1 teléfono *IP*. Se programó, configuró y probó. Se diseñaron experiencias de laboratorio, que se desarrollaron en la maqueta. Con lo anterior se redactaron guías de laboratorio para desarrollar en el curso.

El producto obtenido está constituido por una maqueta y cuatro guías. Las guías tratan los siguientes temas: 1) Implementación de la maqueta y exploración del sistema; 2) Configuración de la maqueta; 3) Habilitación de servicios integrados a la central Asterisk; y 4) Configuración de funciones del plan de discado de Asterisk.

El trabajo cumple con los objetivos propuestos, y satisface los requerimientos del laboratorio del curso EL629. El presente trabajo servirá de apoyo al curso, puesto que permitirá a los alumnos comprender en forma practica los temas de Telefonía *IP* tratados en el curso.

A
*mis queridos padres por todo su sacrificio y esfuerzo
para educarme.*

AGRADECIMIENTOS.

Quería agradecer a todos por acogerme estos años, a todas las personas que hicieron posible mi titulación, en especial al profesor Luis Castillo que me ayudó en el desarrollo de la memoria y que siempre tuvo muy buena disposición. Al profesor Joachim Hintze, por confiar en mi, al profesor Héctor Augusto por su constante apoyo, a Néstor Becerra por apoyarme en el proceso de titulación, al profesor Nicolás Beltrán por su amabilidad y facilidades prestadas para el desarrollo del proyecto de título. A mis compañeros de la U, con los que me desarrollé como ingeniero, y a todos los funcionarios que me ayudaron a llevar a cabo las tareas diarias. Quería agradecer a mi mamá, papá, y familia en general, por toda su ayuda y apoyo durante la carrera. Y a todas las personas que me ayudaron a terminar la carrera.

ÍNDICE DE CONTENIDOS.

RESUMEN	ii
DEDICATORIA	iii
AGRADECIMIENTOS	iv
ÍNDICE DE CONTENIDOS	v
ÍNDICE DE FIGURAS	viii
ÍNDICE DE TABLAS	ix
LISTA DE DEFINICIÓN DE SÍMBOLOS	ix
CAPÍTULO I	
INTRODUCCIÓN	1
CAPÍTULO II	
REVISIÓN BIBLIOGRÁFICA.....	3
2.1 Introducción a la Telefonía Tradicional y a la.....	3
Red Telefónica Conmutada Pública.	
2.1.1 Conmutación de Circuitos y Conmutación de Paquetes	3
2.1.2 Telefonía Tradicional y sus Principales Componentes	4
2.1.2.1 Teléfono	5
2.1.2.2 Codificador de Voz	6
2.1.2.3 Centrales Telefónicas	7
2.1.2.4 Centrales Privadas (PBX)	8
2.1.2.5 Troncales e Interfaces en la PSTN	9
2.1.2.6 Señalización	11
2.2 Telefonía IP	11
2.2.1 Voz sobre IP (VoIP) y Telefonía IP	12
2.2.2 Telefonía IP: Concurrencia de Conceptos	12
2.2.2.1 Conceptos de Telefonía Tradicional	12
2.2.2.2 Conceptos de Redes de Datos	14

2.2.2.3	Componentes de la Telefonía IP	16
2.2.3	Calidad de Servicio (QoS) en Redes TCP/IP	17
2.2.3.1	Sensibilidad del Oído Humano a las Distorsiones que Introduce la Transmisión de la Voz	17
2.2.3.2	Variación de Retardo: <i>Jitter</i> y <i>Wander</i>	18
2.2.3.3	Pérdida de Paquetes	19
2.2.4	Protocolos de Telefonía IP	20
2.2.4.1	Protocolos <i>Call Setup</i> o de Señalización	21
2.2.4.2	Protocolo <i>RTP</i> para la Fase Conversación	27
2.2.5	CODEC	30
2.2.5.1	Principios de Funcionamiento de los CODEC Usados en Telefonía.	30
2.2.5.2	Otras Especificaciones de los CODEC usados en Telefonía. Velocidad Nominal del CODEC y Retardo que Introduce.	33
2.2.5.3	Retardo de Paquetización	35
2.3	Concepto Asterisk	36
2.3.1	¿Qué es Asterisk?	36
2.3.2	Cómo Funciona Asterisk	40
2.3.2.1	Canales Físicos y Canales Lógicos	40
2.3.2.2	Canales ZAP	40
2.3.2.3	Tratamiento de las Llamadas Cuando Ellas se Inician en un Canal Físico ZAP.	41
2.3.2.4	Tratamiento de las Llamadas Cuando se Inician en un Dispositivo que usa Protocolos de Telefonía IP.	41
2.4	Instalación de Linux y Asterisk	42
2.4.1	Instalación de Asterisk sobre CentOS	43
2.4.1.1	Instalación de CentOS	43
2.4.1.2	Instalación de Asterisk sobre CentOS	43
2.4.1.3	Instalación de Asterisk sobre Ubuntu	46

CAPÍTULO III

IMPLEMENTACIÓN	47
3.1 Implementación de los Servidores y de los Softphones	47
3.2 Configuración del Sistema de Telefonía IP	52
3.3 Configuración de los Servicios en el Sistema de Telefonía IP..	52
3.4 Habilitación de Funciones de Asterisk	53
3.5 Diseño de prácticas de Laboratorio Telefonía IP	53
para los Alumnos del Curso EL629.	
3.5.1 Objetivos y Temas a Desarrollar en el Laboratorio N°1	53
3.5.2 Objetivos y Temas a Desarrollar en el Laboratorio N°2	53
3.5.3 Objetivos y Temas a Desarrollar en el Laboratorio N°3	54
3.5.4 Objetivos y Temas a Desarrollar en el Laboratorio N°4	54
3.6 Realización Detallada de las Experiencias Diseñadas	54
3.7 Guías de Laboratorio	54

CAPÍTULO IV

DISCUSIÓN DE RESULTADOS	56
4.1 Cumplimiento de los Requerimientos Iniciales.....	56
4.2 Logros	56
4.3 Análisis Crítico y Posibles Mejoras.....	57

CAPÍTULO V

CONCLUSIONES	58
REFERENCIAS BIBLIOGRÁFICAS	60
ANEXO A: GUÍAS DE LABORATORIO	70
A.1 Guía de Laboratorio N°1. Instalación del Software:	71
SO CentOS, Asterisk sobre CentOS, X Lite y Zoiper	
A.2 Guía de Laboratorio N°2. Configuración Primera Parte: Asterisk	85
con softphones X-Lite y Zoiper, y entre centrales Asterisk.	

A.3	Guía de Laboratorio N°3. Configuración Segunda Parte: Teléfono IP, ATA, Servicios Voice-mail e IVR.	98
A.4	Guía de Laboratorio N°4. Funciones de Llamada: Funciones de Límite de Tiempo para Llamadas y Grabación de Audio.	111

ÍNDICE DE FIGURAS.

		Página.
FIGURA N°1	Componentes y Circuitos de un Teléfono Analógico.	6
FIGURA N°2	Centrales Telefónicas en la RTCP.	8
FIGURA N°3	Conexión de <i>PBX</i> a la RTCP.	9
FIGURA N°4	Puertas <i>FXS</i> y <i>FXO</i> .	10
FIGURA N°5	Fases de una Llamada.	13
FIGURA N°6	Retardo <i>Jitter</i> .	18
FIGURA N°7	Diagrama de Llamada <i>IP</i> .	20
FIGURA N°8	Diagrama de Protocolos y Codecs.	24
FIGURA N°9	Diagrama de Protocolos H.323.	25
FIGURA N°10	Estructura de Paquete RTP.	29
FIGURA N°11	Partes de un Paquete RTP.	29
FIGURA N°12	Diagrama de Codificación.	31
FIGURA N°13	Relación entre V_{Codec} , R y L.	32
FIGURA N°14	Período de tiempo R entre dos datagramas.	34
FIGURA N°15	Topología-Arquitectura de Sistema Telefónico Asterisk 1.	37
FIGURA N°16	Topología-Arquitectura de Sistema Telefónico Asterisk 2.	38
FIGURA N°17	Topología-Arquitectura de Sistema Telefónico Asterisk 3.	38
FIGURA N°18	Topología-Arquitectura de Sistema Telefónico Asterisk 4.	39
FIGURA N°19	Topología-Arquitectura de Sistema Telefónico Asterisk 5.	39
FIGURA N°20	Mapa de Conexiones de <i>LAN</i> del Laboratorio de Postítulo del DIE.	49
FIGURA N°21	Arquitectura de Maqueta de Laboratorio para el Curso EL629.	51

ÍNDICE DE TABLAS.

		Página.
TABLA N°1	Calidad de Voz Según el Retardo.	17
TABLA N°2	Factor de Deterioro.	19
TABLA N°3	Cuadro Descriptivo de CODEC's.	33
TABLA N°4	Tiempo entre Toma de Muestras Sucesivas (TEMS) según CODEC.	34
TABLA N°5	TEMS, R y R/TEMS según CODEC.	35
TABLA N°6	Velocidad de Salida Nominal y Retardo de Paquetización por CODEC.	36

LISTA DE DEFINICIÓN DE SÍMBOLOS.

- ACELP: Predicción Lineal Excitada de Código Algebraico (*Algebraic Code Excited Linear Prediction*).
- ACF: Confirmación de Admisión (*Admission Confirmation*).
- ARQ: Petición de Admisión (*Admission ReQuest*).
- AstDB: Base de Datos de Asterisk (*Asterisk DataBase*).
- ATA: Adaptador de Teléfono Análogo (*Analog Telephone Adapter*).
- BRI: Interfase de Tasa Básica (*Basic Rate Interface*).
- BSD: Sistema Operativo.
- Buffer: Espacio de memoria en el que se almacenan datos para evitar que el programa o recurso que los requiere, ya sea hardware o software, se quede sin datos durante una transferencia.
- B2BUA: Agentes Usuarios Espalda contra Espalda (*Back to Back User Agents*).
- CDR: Registro de Detalles de Llamadas (*Call Detail Records*).
- CentOS: Sistema Operativo de Empresa Comunitaria (*Community ENTERprise Operating System*).
- CO: Central Local, Oficina Central, Central Clase 5 (*Central Office*).
- CODEC: Codificador/Decodificador de audio.
- cRTP: RTP comprimido (*compressed RTP*).
- DAHDI: Interface de Hardware de Asterisk de Digium (*Digium Asterisk Hardware Interface*).
- DHCP: Protocolo de Configuración de Anfitriones Dinámicos (*Dynamic Host Configuration Protocol*).
- DID: Discado Entrante Directo (*Direct Inward Dial*).
- DIE: Departamento de Ingeniería Eléctrica.
- DVD: Disco Versátil Digital (*Digital Versatile Disc*).
- EL629: Curso "Sistemas de Conmutación Telefónica".
- FDM: Multiplexación por División de Frecuencia (*Frecuency División Multiplexing*).
- FXS: Estación de Intercambio Exterior (*Foreign Exchange Station*).
- FXO: Oficina de Intercambio Exterior (*Foreign Exchange Office*).
- GKRCs : Ruteadores de Señalización.
- GPL: Licencia Pública General (*General Public License*).
- GUI: Interfaz de Usuario Gráfica (*Graphical User Interface*).
- Hw: Hardware.
- IAX: Protocolo de Intercambio en Asterisk (*Inter-Asterisk eXchange Protocol*).
- IETF: Cuerpo de Tareas de Ingeniería de Internet (*Internet Engineering Task Force*).

- IP: Protocolo de Internet (*Internet Protocol*).
- ITU-T: Unión Internacional de Telecomunicaciones (*International Telecommunication Union*).
- IVR: Respuesta de Voz Interactiva (*Interactive Voice Response*).
- IXC: *Carrier* de Intercambio (*Interexchange Carrier*).
- LAN: Red de Área Local (*Local Area Network*).
- LEC: *Carrier* de Intercambio Local (*Local Exchange Carrier*).
- MacOSX: Sistema Operativo.
- MCU: Unidad de Control de Multipuntos (*Multipoint Control Unit*).
- Megaco: Protocolo de Control de Compuerta de Media (*Media Gateway Control Protocol*).
- MFC: Tonos Multifrecuencia.
- MGCP: Protocolo de Control de Compuerta de Media (*Media Gateway Control Protocol*).
- MP: Procesador Multipunto (*Multipoint Processor*).
- MPMLQ: Cuantización de Multi-pulsos de Máxima Probabilidad (*Multi-Pulse Maximum Likelihood Quantization*).
- MySQL: Mi SQL, Lenguaje de Peticiones Estructuradas (*SQL, Structured Query Language*).
- NAT: Traducción de Direcciones de Red (*Network Address Translation*).
- PAM: Modulación de Amplitud de Pulso (*Pulse Amplitud Modulation*).
- PBX: Central Privada (*Private Branch Exchange*).
- PC: Computador Personal (*Personal Computer*).
- PCI: Interconector de Componentes Periféricos (*Peripheral Component Interconnect*).
- PCM: Modulación por Código de Pulsos (*Pulse Code Modulation*).
- PCTR: Protocolo de Control de Transporte de Tiempo Real (*RTCP, Real-time Transport Control Protocol*).
- PLC: Conciliamiento de Pérdida de Paquetes (*Packet Loss Concealment*).
- POTS: Servicio de Teléfonos Viejos y Básicos (*Plain Old Telephone Service*).
- PRI: Interfase de Tasa Primaria (*Primary Rate Interface*).
- QoS: Calidad de Servicio (*Quality of Service*).
- RAS: Registro, Admisión, Status (*Registration, Admission, Status*).
- RDSI: Red Digital de Servicios Integrados (*ISDN, Integrated Service Digital Network*).
- RFC: Hacer Peticiones Para Recibir Comentarios (*Request For Comments*).
- RTCP: Red Telefónica Conmutada Pública (*PSTN, Public Switched Telephone Network*).
- RTP: Protocolo de Transporte de Tiempo Real (*Real-time Transport Protocol*).
- RTSP: Protocolo de *Streaming* de Tiempo Real (*Real Time Streaming Protocol*).
- SCCP: Protocolo de Control de Cliente Ligero (*Skinny Client Control Protocol*).
- SER: Enrutador Expreso SIP (*Sip Express Router*).
- SIP: Protocolo de Iniciación de Sesión (*Session Initiation Protocol*).
- SSL: Capa de Soquetes Seguros (*Secure Sockets Layer*).
- SS7: Sistema de Señalización #7.
- Sw: Software.
- TCP: Protocolo de Control de Transmisión (*Transmission Control Protocol*).
- TCP/IP: TCP y IP.
- TDM: Multiplexación por División de Tiempo (*Time-Division Multiplexing*).
- UDP: Protocolo de Datagramas de Usuario (*User Datagram Protocol*).
- VoIP: Voz sobre IP (*Voice Over IP*).
- VPN: Red Virtual Privada (*Virtual Private Network*).
- WAN: Red de Área Amplia (*Wide Area Network*).
- Web: WWW (*World Wide Web*).

CAPÍTULO I:

INTRODUCCIÓN.

El presente trabajo de memoria, titulado "Implementación De Una Central Telefónica IP Asterisk Para Las Prácticas De Laboratorio En El Curso EL629, Sistemas De Conmutación Telefónica", se enmarca dentro de las actividades docentes del curso. Producto de las modificaciones en el plan de estudios, los profesores han considerado conveniente introducir al curso experiencias de laboratorio. Estos laboratorios darán la oportunidad a los alumnos de poner en práctica las materias de telefonía *IP* estudiadas en clases de cátedra.

Para desarrollar el tema de memoria se contó con los apuntes del curso EL629 y gran cantidad de documentación relacionada al tema, la que se especifica en la bibliografía.

Como se indicó más arriba, el objetivo general del trabajo es "Implementar Una Central Telefónica IP Asterisk Para Las Prácticas De Laboratorio En el Curso EL629, Sistemas De Conmutación Telefónica", lo que significa implementar una maquetaⁱ de telefonía *IP* basada en Asterisk para desarrollar experiencias de laboratorio más guías de laboratorio a desarrollar sobre la maqueta.

Los objetivos específicos del proyecto son:

- 1) Implementar un sistema de telefonía *IP* Asterisk (maqueta), constituido por dos servidores, un teléfono *IP*, una *ATA* (*Analog Telephone Adapter*) para la conexión de teléfonos analógicos, y varios *softphones*;
- 2) Configurar el sistema implementado para constituir una maqueta para realizar prácticas de laboratorio de acuerdo a las necesidades del curso EL629;
- 3) Diseñar prácticas de laboratorio para el curso en coordinación con el profesor de telefonía *IP* del curso;
- 4) Desarrollar completamente las experiencias diseñadas sobre la maqueta; y
- 5) Con la experiencia de todo el trabajo realizado, redactar las Guías de Laboratorio de telefonía *IP* para el curso EL629.

A continuación una breve descripción de los siguientes capítulos:

El capítulo dos, consiste en una revisión bibliográfica de los temas tratados en el trabajo de memoria. Se revisan las materias necesarias para contextualizar el tema y abordar la implementación. Se revisa la telefonía tradicional (red telefónica pública conmutada),

ⁱ Entre los ingenieros y técnicos especialistas en telefonía se conoce con el nombre de "maqueta" a una instalación compuesta de hardware y software que se utiliza para experimentar y probar servicios y dispositivos usados en comunicaciones, en un ambiente lo más parecido posible al ambiente real.

funcionamiento, estructura y partes; luego se estudia la telefonía *IP* (telefonía a través de internet), protocolos, componentes, calidad de servicio, arquitecturas, CODECs (Codificador/Decodificador) y terminales; a continuación se revisa el concepto Asterisk, descripción, características, diagramas de topología-arquitectura y funcionamiento; Después se estudia la instalación de Linux y Asterisk, pasos y procedimientos a seguir para la instalación y finalmente se estudia la configuración y programación de Asterisk.

El capítulo tres, llamado “Implementación”, presenta el desarrollo del trabajo de título. Se presenta el trabajo desarrollado, se enumeran los pasos que se deben seguir para dejar la maqueta operativa, también se informa sobre las guías redactadas para realizar las experiencias de laboratorio.

En el capítulo cuatro, titulado “Discusión de resultados”, se analiza y discuten los resultados obtenidos del trabajo realizado. Incluyendo cumplimiento de los requerimientos iniciales, logros, análisis crítico y posibles mejoras.

En el capítulo número cinco, llamado “Conclusiones”, se presentan las conclusiones finales y se responden preguntas como: ¿Fueron satisfactorios los resultados obtenidos?, ¿Cómo se puede continuar el desarrollo de este trabajo?

CAPÍTULO II:

REVISIÓN BIBLIOGRÁFICA.

Para llevar a cabo este trabajo de título fue necesario recurrir a numerosa bibliografía, que puede ser clasificada de la siguiente forma:

- Telefonía Tradicional.
- Telefonía *IP* (*Internet Protocol*, Protocolo de Internet).
- Concepto Asterisk.
- Instalación Linux, Configuración y Programación del Sistema Asterisk.

Considerando que el objetivo del presente trabajo de memoria es implementar una central telefónica *IP* y redactar guías para prácticas de laboratorio en el curso EL629 “Sistemas de Conmutación Telefónica”, la bibliografía para el fundamento teórico de este trabajo (telefonía general, telefonía *IP*, concepto Asterisk, programación Asterisk) corresponde principalmente a los apuntes del curso, redactados por el profesor Luis Castillo B. La parte práctica (instalación y programación de Asterisk, está basada en material facilitado por el profesor guía y numerosa bibliografía disponible en sitios *web*, la que es actualizada regularmente. A continuación se hace una revisión bibliográfica de los temas anteriormente mencionados.

2.1 Introducción a la Telefonía Tradicional y a la Red Telefónica Conmutada Pública.

2.1.1 Conmutación de Circuitos y Conmutación de Paquetes.

Haremos una revisión de dos importantes conceptos:

- Telefonía mediante conmutación de circuitos.
 - Telefonía mediante conmutación de paquetes.
- 1) Características de la conmutación de circuitos:
 - i) Durante el establecimiento de la comunicación se establece un circuito extremo-extremo que está disponible durante toda la llamada para intercambiar la información entre los abonados A y B.
 - ii) Cada nodo de conmutación (central telefónica) colabora estableciendo secciones del circuito extremo-extremo.

- iii) Las distintas secciones del circuito pueden ser de muy distintas naturalezas: par de cobre (conexión galvánica), enlace de radio *FDM* analógico (*Frequency Division Multiplexing*, Multiplexación por División de Frecuencia), enlace de radio *TDM* digital (*Time-Division Multiplexing*, Multiplexación por División de Tiempo), redes de datos en que se establece un circuito virtual, etc.
 - iv) La existencia de un circuito dedicado a la comunicación mientras ella esté establecida, garantiza baja probabilidad de pérdida de información.
- 2) Características de la conmutación de paquetes:
- i) Se efectúa en forma distribuida en los múltiples dispositivos y computadores que conforman la red de datos *IP*.
 - ii) En una llamada telefónica, la voz se “paquetiza” en múltiples bloques de datos que contienen muestras digitales de la voz de cada uno de los interlocutores.
 - iii) Cada paquete contiene muestras de la información que intercambian los abonados A y B, y además información de direccionamiento, con indicación de los dispositivos de origen y de destino, que es utilizada por los routers para encaminar los paquetes hasta sus destinos.
 - iv) Los paquetes correspondientes a la información de una misma comunicación pueden seguir caminos diferentes.
 - v) La probabilidad de que se pierda información o que llegue retrasada es mayor que en la conmutación de circuitos.

La tecnología de conmutación usada mayoritariamente hoy por hoy en el servicio telefónico es la conmutación de circuitos lo que se conoce como Telefonía Tradicional. La conmutación de paquetes es usada por la Telefonía IP. Si bien con la Telefonía IP por el momento no se pueden ofrecer todos los servicios que son entregados por la Telefonía Tradicional, se espera que una vez resuelto este problema, la Telefonía IP pueda entregar el servicio telefónico a un menor costo.

2.1.2 Telefonía Tradicional y sus Principales Componentes.

Antes de empezar el estudio de la Telefonía *IP*, se debe conocer los conceptos más importantes de la Telefonía Tradicional. Para referirse a telefonía tradicional se usan diferentes términos: Telefonía, Telefonía de Conmutación de circuitos, *POTS* (*Plain Old Telephone Service*, Servicio Telefónico Básico Tradicional). A veces, impropriamente, se le llama Telefonía Analógica, lo que es incorrecto puesto que las centrales telefónicas tradicionales y la red que las interconecta, actualmente son completamente digitales.

Por otra parte, a veces para referirse a la Telefonía *IP*, se habla de Voz sobre *IP*, lo que en rigor también es impropio. A continuación se describirá en forma cualitativa el los componentes, funcionamiento y conceptos mas importantes de la telefonía tradicional.

La red usada por la Telefonía Tradicional se conoce como *Public Switched Telephone Network, PSTN* (Red Telefónica Conmutada Pública, RTCP).

La infraestructura (teléfonos, red externa, red telefónica conmutada) provee una serie de componentes que permiten establecer las llamadas en forma rápida y confiable. Una breve descripción de algunos de estos componentes será útil para entender la telefonía tradicional, y para saber que debe “copiar” la telefonía *IP* de la telefonía tradicional para lograr proveer el mismo desempeño, confiabilidad y seguridad que ella. Los componentes principales de la RTCP son:

- 1) Teléfono.
- 2) Codificador de voz.
- 3) Central Telefónica.
- 4) Central Pequeña Privada (*PBX, Private Branch Exchange*).
- 5) Troncales y puertas en la RTCP: Interfaces de interconexión con la RTCP Señalización.
- 6) Señalización

2.1.2.1 Teléfono.

En la RTCP se utilizan teléfonos analógicos y digitales:

- i) Teléfono analógico:

Es el tipo más popular en la actualidad. Se conecta a las centrales locales y *PBX's* de la RTCP a las que el teléfono analógico envía señales analógicas estandarizadas a través del par de hilos de cobre. En la central local estas señales se digitalizan. En la figura N°1 se muestra el circuito y conexión de un teléfono analógico.

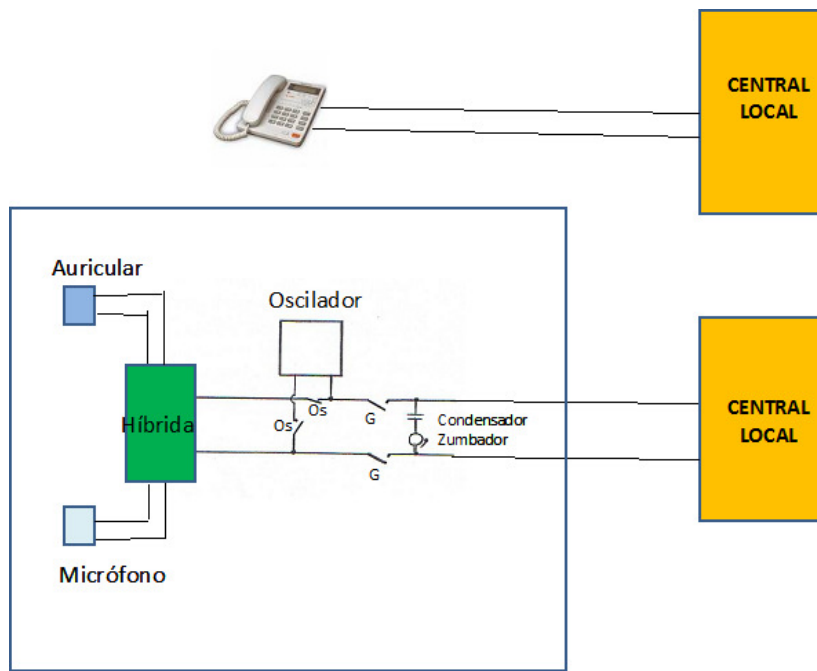


Figura N°1: Componentes y Circuitos de un Teléfono Analógico.

ii) Teléfono digital:

Muchas Corporaciones tienen *PBX* a las que se puede conectar teléfonos analógicos y además teléfonos digitales. Estos últimos digitalizan la voz y entregan y reciben de la *PBX* directamente señales digitales (señalización y voz), a través del par de cobre. Tienen limitaciones de distancias entre la ubicación del teléfono y la *PBX*. Las señales digitales tienen formatos propietarios.

2.1.2.2 Codificador de Voz.

Cuando hablamos frente al micrófono del teléfono, se produce una señal eléctrica analógica que inicialmente viaja en esa forma por el par telefónico hasta la central local. Allí esta señal es digitalizada (convertida en sucesión de ceros y unos) y transmitida en esta forma hasta el otro extremo de la red, donde vuelve a convertirse en analógica. Las señales analógicas de audio se convierten a formato digital, y viceversa, mediante los *CODEC*, que emplean protocolos estandarizados para realizar la conversión. El estándar más usado es el G.711, emplea una técnica de codificación llamada *PCM* (*Pulse Code Modulation*, Modulación de Código de Pulsos).

Dentro de G.711 hay dos variantes: G.711u (usado principalmente en Norteamérica) y G.711a (usado fuera de Norteamérica). G.711 convierte las señales analógicas de audio entrantes en señales digitales a una tasa de 64.000 bits/seg, a la salida. Un canal simple es normalmente denominado IT, un banco *PCM* o trama E1 está constituido por 30 IT + 1 IT para la señalización + 1 IT para sincronismos y alarmas. Este punto será tratado en mayor profundidad en el punto 2.2.5 *CODEC*.

2.1.2.3 Centrales Telefónicas.

Son los nodos que conforman el núcleo de la Red Telefónica Conmutada Pública, conmutan las llamadas de circuito a circuito, estableciendo el camino requerido para las llamadas por la red telefónica de circuitos. Un grupo de circuitos entre centrales se denomina Troncal y su capacidad se mide en cantidad de tramas E1. Las centrales se categorizan en base a las funciones que realizan. Es útil conocer la nomenclatura usada en EE.UU. para entender textos de telefonía editados en ese país:

- La central a la que llegan las líneas de los usuarios (Central Local), en EE.UU. son conocidas como “*Central Office*” (Oficina Central), “*CO*” o “Central Clase 5”.
- Las centrales locales son operadas por las compañías telefónicas locales, operadores a los que en EE.UU. se les denomina “*Local Exchange Carrier*” (*LEC*, Empresa de Intercambio Local).
- Las centrales locales entregan, a través de troncales, las señales digitalizadas a otras centrales de la red RTCP.
- Las centrales Tandem, son operadas en EE.UU. por compañías de larga distancia a las que se llama “*Interexchange Carrier*” (*IXC*, Empresa de Intercambio). En Chile a veces se da a Tandem una acepción mucho más amplia, ya que incluye a distintos tipos de centrales.
- Las centrales Tandem se conectan a centrales locales y a otras centrales Tandem para proveer caminos de circuitos a través de la RTCP.
- En EE.UU. se distinguen distintas clases de Tandems:
 - “Clase 4” Interconecta áreas de una ciudad o pueblos en una región.
 - “Clase 3” Interconecta grandes centros de población en una región particular.
 - “Clase 2” Interconecta todas las regiones.
 - “Clase 1” Interconecta centros internacionales fuera de EE.UU.
- Las centrales Tandem cursan tráfico masivo y están diseñadas para ser muy escalables y confiables.
- En telefonía *IP* los enrutadores vienen a ser símil de las centrales telefónicas.

A continuación se ilustra en la Figura N°2 la organización de la RTCP:

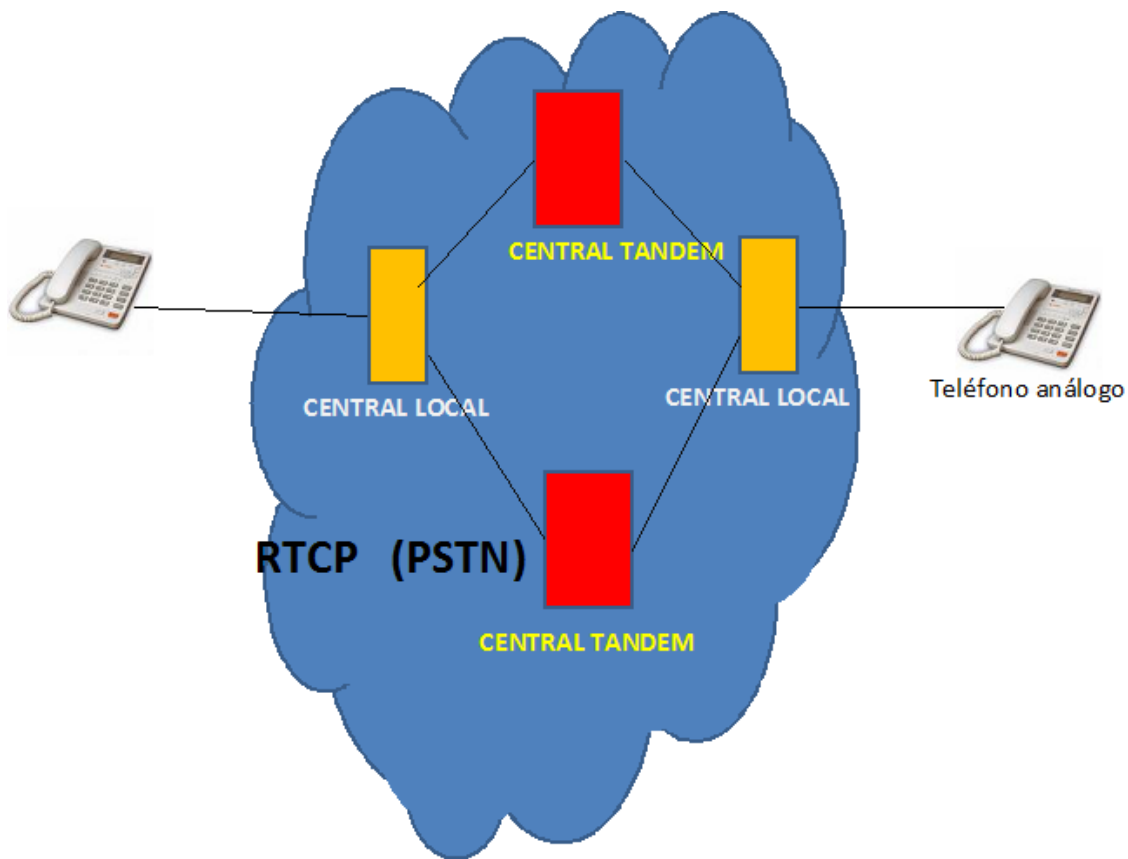


Figura N°2: Centrales Telefónicas en la RTCP.

2.1.2.4 Centrales Privadas (*PBX, Private Branch Exchange*).

La *PBX* es la base de las redes de voz corporativas de empresas y organismos de diversas índoles. Las redes corporativas pueden estar constituidas por una o varias *PBX* interconectadas entre ellas mediante troncales *TIE*. Los teléfonos conectados a las *PBX* se conocen con el nombre de “extensión” ó “anexo”. Las redes corporativas tienen diferencias con aquellas que atienden residencias, tienen que servir a múltiples usuarios que requieren de características avanzadas (Identificación de usuario, transferencia de llamadas, desvío de llamadas, etc.). En estas redes las llamadas entre los usuarios de la corporación, desde el punto de vista de los usuarios, no distinguen si ellos están en la misma ciudad o en otra muy lejana. Es decir toda llamada dentro de la Corporación es llamada local. Para las comunicaciones con el exterior (con la RTCP) las *PBX* se interconectan a centrales locales mediante grupos de circuitos que son de uso común para todos o grupos de anexos, consiguiéndose así ahorros de costos, al aprovechar más eficientemente las conexiones con la RTCP.

Servicio “Discado Entrante Directo” (*DID, “Direct Inward Dialing”*).

Para una *PBX* que disponga del equipamiento necesario, se puede contratar con la empresa de telefonía local el servicio Discado Entrante Directo. Este servicio permite que las llamadas desde la RTCP lleguen directamente a la extensión (o anexo), sin pasar por operadora. Para ello la central local asigna troncales especiales, analógicas o digitales, por las que envía a la *PBX* los últimos dígitos marcados por el abonado llamante, los que corresponden al número del anexo.

En el servicio *DID* la central local de la RTCP señala a la *PBX* para que inicie el proceso *DID*, que consiste en enrutar la llamada entrante directamente al anexo correspondiente a los últimos dígitos marcados por el llamante. Las *PBX* también proporcionan los características tradicionales, como llamadas en espera, conferencias, desvíos.

Muchas grandes corporaciones tienen redes de *PBX*. Como dijimos, las troncales que interconectan las *PBX* entre ellas se les conoce como troncales *TIE*. En sistemas de telefonía *IP* las *PBX IP* realizan las mismas funciones de las *PBX* de la RTCP. En la Figura N°3, que sigue a continuación, se ilustra la conexión de una *PBX* a la RTCP.

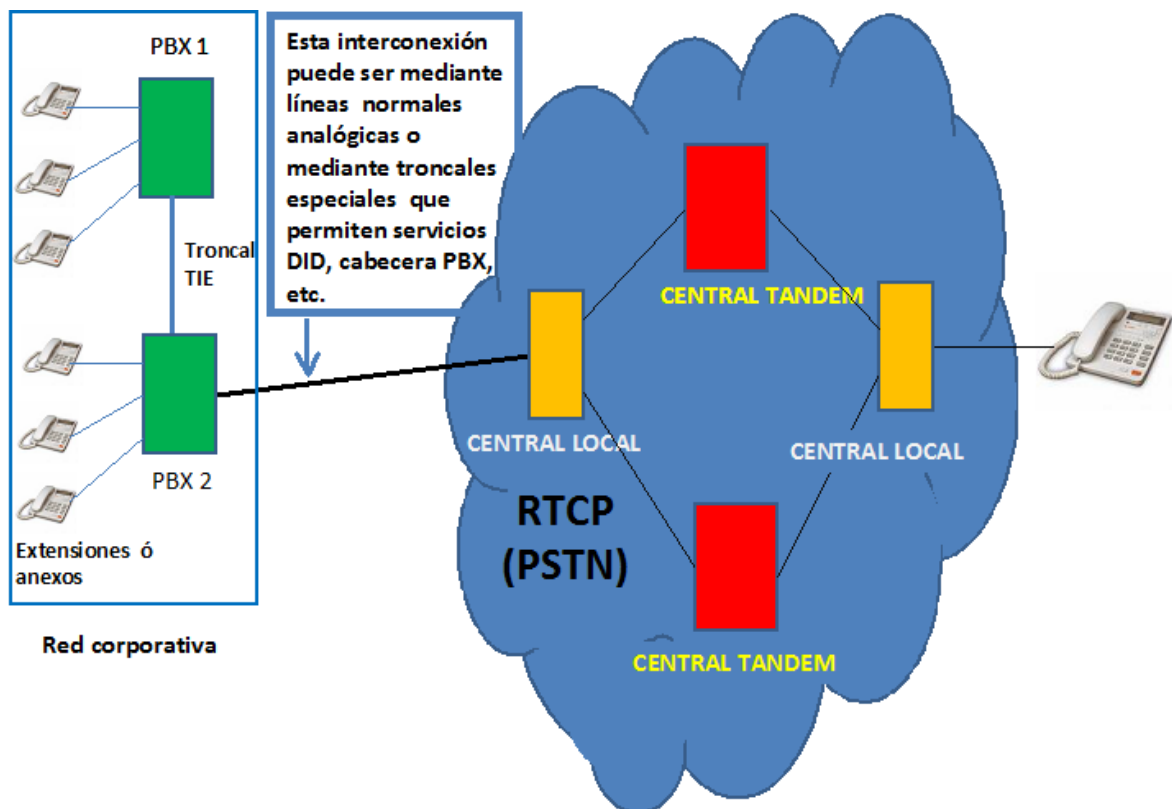


Figura N°3: Conexión de *PBX* a la RTCP.

2.1.2.5 Troncales e Interfaces en la RTCP.

Para la conexión a la RTCP existen diferentes tipos de Troncales o Interfaces, que corresponden a puertas por las que se intercambia señalización y las señales de audio correspondiente a la conversación telefónica. Hay puertas digitales y analógicas

Los distintos dispositivos e instalaciones utilizados en los sistemas de telecomunicaciones se interconectan entre ellos a través de “puertas”. Toda puerta tiene dos partes: una que “mira” hacia el interior del dispositivo o instalación y otra que tiene por finalidad recibir o enviar las señales a otro dispositivo con el cual se interconecta. La parte de la puerta que “mira” hacia el interior normalmente está alambrada en forma fija, mientras que la otra parte, que es la que nos interesa, corresponde a un conector, al que se pueden conectar o desconectar cables hacia otros

dispositivos. Entonces, cada vez que hablemos de “puerta” de un dispositivo o instalación nos estaremos refiriendo a la parte del conector. Hay diferentes clases de puertas, las que se definen a continuación:

- i) Puerta FXS (Foreign Exchange Station, corresponde a Phone o tel set): La forma más común de conexión a la RTCP es como usuario con una línea telefónica analógica conectada a una central local. A este tipo de puerta de entrada a la RTCP se le conoce como puerta FXS. Todos tenemos en las paredes de nuestra casa, una o más de estas puertas. Estas puertas proporcionan servicio POTS.
- ii) Puerta FXO (Foreign Exchange Office, corresponde a Line o tel line): Los equipos terminales como teléfonos analógicos, las máquinas fax, los módems, etc. tienen una o más puertas FXO que se conectan mediante un cable a puertas FXS de la RTCP.

Las puertas FXS de la RTCP proporcionan señalización que entiende los equipos terminales: batería de alimentación, tono de invitación a discar y voltaje de ringing, por su parte las puertas FXO proporcionan señalización que entienden las centrales y PBX: códigos multifrecuencia (MFC) y señales de cuelgue y descuelgue. Se tiene entonces que, como ocurre con un *plug* y un *jack*, siempre una puerta FXO se conecta con una puerta FXS. Nunca FXO con FXO ni FXS con FXS. Algunos dispositivos como las PBX, tienen tanto puertas FXS como FXO. En la Figura N°4 se ilustran las características de ambas puertas:

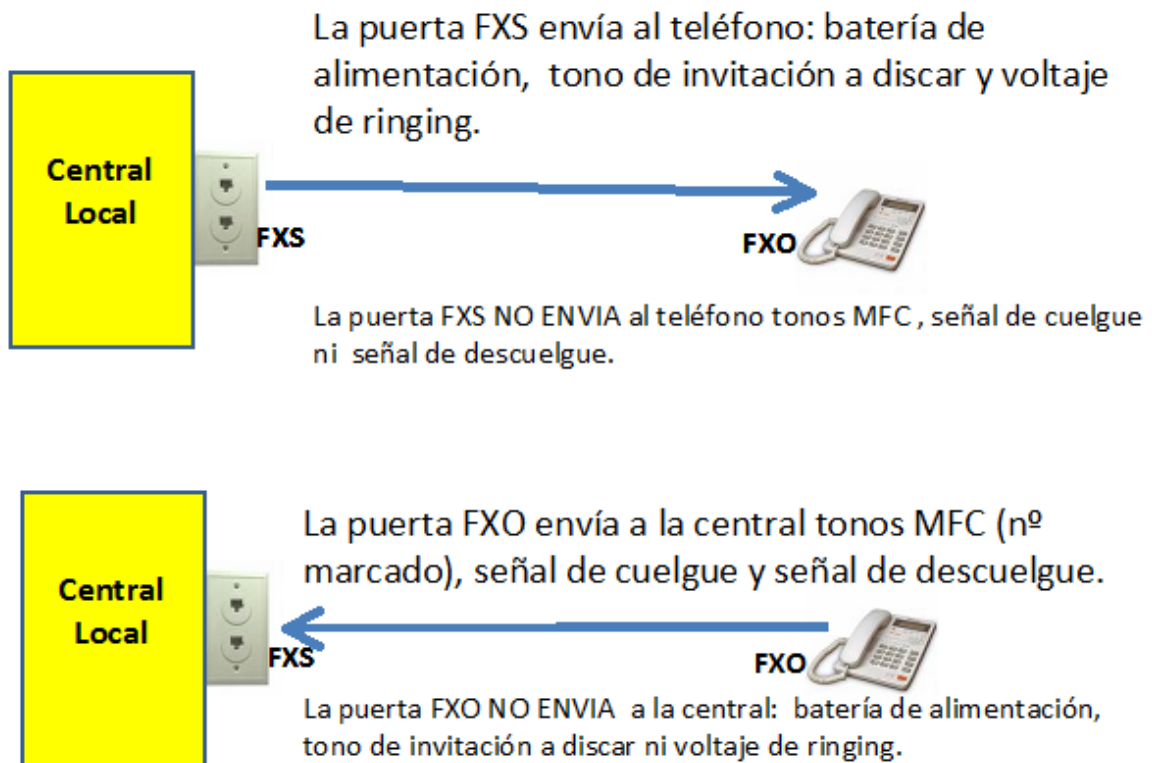


Figura N°4: Puertas FXS y FXO.

Si se conecta una puerta *FXO* con otra puerta *FXO*, no se logra interconexión alguna, ya que corresponde a conectar un teléfono con otro teléfono, sin que nadie proporcione tono de invitación a discar, voltaje de alimentación ni voltaje de ringing.

- i) Puerta Digital *ISDN BRI* (*Integrated Service Digital Network* y *Basic Rate Interface*, Red Digital de Servicios Integrados, RDSI e Interfase de Tasa Básica): Permite la conexión digital de usuarios con servicio RDSI. Esta conexión proporciona dos circuitos denominados *Bearer* (B) y uno Delta (D), por lo que se dice que ésta es una conexión 2B + D. Cada circuito B permite la transmisión de dos flujos de 64 Kbps de información de datos, voz, video, y el circuito D de flujos 16 Kbps de información de señalización y control, y eventualmente también de datos (telemetría).
- ii) Línea Troncal Analógica Especial: Para interconectar *PBX* con servicios especiales proporcionados por la central local (*DID*, número cabecera de *PBX*, etc.). Las troncales que permiten servicio *DID* pueden hacer llegar dígitos a la *PBX*. En estos casos las troncales de la central local no se conectan a una puerta *FXO* normal de la *PBX*. Las troncales analógicas para servicio *DID* cursan tráfico solamente en el sentido entrante a la *PBX*.
- iii) Puerta Digital E1 permite la conexión de 32 circuitos de voz. Normalmente se usa conector RJ45 con dos pares alambrados (Transmisión y Recepción).
- iv) Puerta Digital T1 (no se usa en Chile) permite la conexión de 24 circuitos de voz (24).
- v) Puerta Digital E1 *PRI* (*Primary Rate Interface*, Interfase de Tasa Primaria) permite la conexión digital de una *PBX* con servicio RDSI a la RTCP. Corresponde a una conexión de 30 circuitos B 64 Kbps más 2 circuito D 64 Kbps (30B + 2D).

2.1.2.6 Señalización.

Para la interconexión de centrales entre sí, existen diferentes tipos de troncales (interfaces). Los distintos tipos de interfaces para la interconexión entre centrales se definen con sus respectivos protocolos, cada uno de los cuales se conoce como un “Sistema De Señalización”, siendo SS7 (Sistema de Señalización #7) uno de los mas populares.

2.2 Telefonía IP.

La telefonía IP, el segundo tema a tratar sera abordada en los siguientes puntos:

- Voz sobre IP (VoIP) y Telefonía IP.
- Telefonía IP: Concurrencia de Conceptos.
- Calidad de Servicio (QoS) en Redes TCP/IP.
- Protocolos de Telefonía IP.

- CODEC.

2.2.1 Voz sobre *IP* (*VoIP*) y Telefonía *IP*.

Características esenciales de una comunicación *VoIP*:

- i) La voz se codifica en forma de paquetes de datos (se “paquetiza”).
- ii) Se usa una red de datos *IP* para transferir dichos datos.

Se debe distinguir entre las expresiones “*VoIP*” y “Telefonía *IP*”. Telefonía *IP* es un concepto amplio, que se preocupa de la problemática de reproducir en redes *IP* todos los servicios que ofrece la Red Telefónica Conmutada Pública (incluyendo, como por Ej., fax, *ISDN*, interconexión con todo tipo de redes fijas y móviles). *VoIP* se refiere a la tecnología usada para transmitir voz por redes *IP*, sean estas redes privadas o Internet. No obstante a veces, erróneamente, ambas expresiones se usan como sinónimos.

2.2.2 Telefonía *IP*: Concurrencia de Conceptos.

Para el estudio de la telefonía *IP* se requiere conocer conceptos de:

- 1) Telefonía tradicional.
- 2) Redes de datos *IP*.
- 3) Nuevos conceptos propios de la telefonía *IP*.

Teniendo en cuenta que ya se estudió la telefonía tradicional, en esta parte se revisará principalmente los últimos dos grupos de conceptos.

2.2.2.1 Conceptos de Telefonía Tradicional.

En relación a la telefonía tradicional, serán muy útiles todos los conceptos ya estudiados. Por ahora recordemos las fases y eventos principales que se distinguen en una llamada telefónica. En una llamada telefónica se distinguen fases:

- i) Fases establecimiento y disolución de la llamada (normalmente se llama fase establecimiento a la que incluye ambas): Corresponde a la parte de la llamada telefónica que abarca la secuencia de eventos necesarios para establecer la llamada (*Call Setup*) y deshacerla (*Call Take Down*).
- ii) Fase conversación: Corresponde a la parte de la llamada en que se transmite la voz, es decir en que tiene lugar la conversación telefónica. La fase conversación ocurre después de la fase *Call Setup* y antes de la fase *Call Take Down*.

En la Figura N°5 que se muestra a continuación se ilustran las fases de una conversación telefónica.

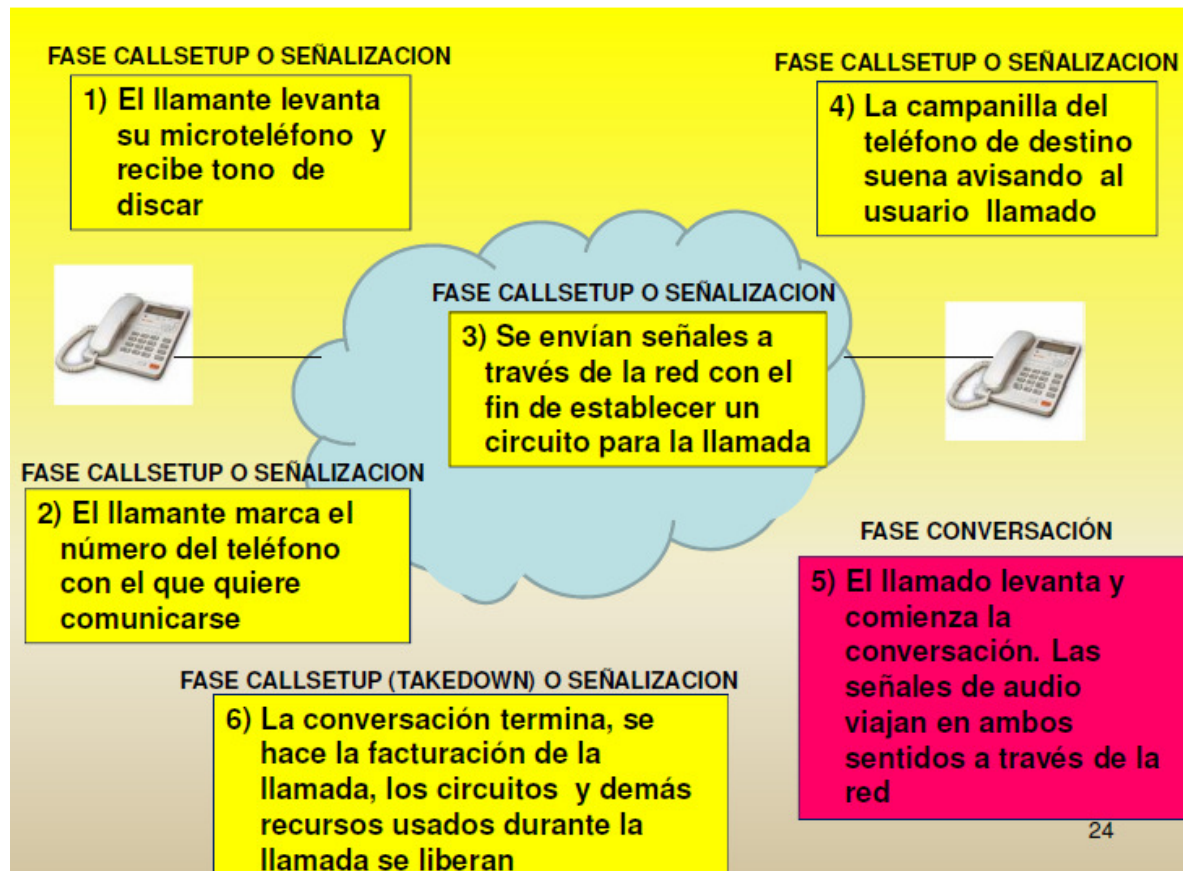


Figura N°5: Fases de una Llamada.

i) Fase de Establecimiento (*Call Setup*).

Durante las fases de establecimiento y disolución de la llamada se desarrollan funciones como:

- Envío de tono de invitación a marcar.
- Envío de *ringing*.
- Envío de tono de ocupado.
- Liberación.

ii) Fase conversación.

Durante la fase de conversación, la voz es sometida a varios procesamientos con el objetivo de:

- En el extremo emisor convertir la voz analógica en muestras digitales.
- Paquetizar la voz: En telefonía *IP* aparece la necesidad de este proceso que en telefonía tradicional no era necesario.
- Transmitir la voz a través de la red en formato de paquetes.
- Reensamblar los paquetes.
- En el extremo receptor volver a convertir, ahora de digital a analógico.
- La conversión de análogo a digital y viceversa se realiza mediante los CODEC ubicados en ambos extremos de la comunicación.

Eventos importantes durante una llamada.

- El llamante levanta el auricular y escucha tono de invitación a marcar.
- El llamante marca un número telefónico.
- Es invocado el protocolo de establecimiento de llamada con el fin de localizar al abonado llamado y enviarle una señal que origine *ringing*.
- El teléfono de destino suena indicando que ha llegado una llamada.
- El llamado levanta el auricular y se inicia la conversación bidireccional.
- En telefonía *IP* durante la conversación las señales de audio correspondientes a la voz son codificadas como datagramas (usando CODEC) y se transmiten por la red *IP* en forma de paquetes.
- La conversación finaliza, la comunicación se da por terminada y se disuelve. Se realiza la tarificación.

2.2.2.2 Conceptos de Redes de Datos.

Protocolos para transmisión de datos son reglas precisas que regulan:

- Como construir los paquetes o bloques de datos.
- Como se deben comportar los extremos que los envían y reciben.

i) Protocolo *IP*:

- Los protocolos para la transmisión de datos se han ido desarrollando durante los últimos 50 años.

- Desde la aparición de Internet, el “*Internet Protocol*”, o *IP*, se ha convertido en el más importante de todos.
- *IP* destaca por su gran escalabilidad y adaptabilidad.
- Se ha ido convirtiendo en ubicuo (está presente al mismo tiempo en todas partes).
- *IP* revolucionó la transmisión de datos y la forma de comunicarse.
- En los últimos años la palabra “convergencia” se ha puesto de moda, abriendo grandes expectativas a la industria de las redes *IP*.
- Convergencia: diferentes tipos de datos - voz, video y aplicaciones de datos - se transfieren usando una misma red *IP*.

ii) Estándares para transmisión de datos:

Los organismos con mayor influencia en la creación de estándares han sido:

- *ITU-T* “*International Telecommunication Union*”, Union Internacional de Telecomunicaciones (ingenieros de telefonía).
- *IETF* “*Internet Engineering Task Force*”, Cuerpo de Tareas de Ingeniería de Internet (ingenieros de transmisión de datos). El *IETF* ha tenido como especial preocupación los estándares *IP*. Las nuevas técnicas para la transmisión de datos se someten a una rigurosa fase de pruebas, consistente en estudio, implementación y revisión, con el fin de verificar la estabilidad y robustez de ellas. La etapa *RFC* (*Request For Comments*) es el último paso para que un borrador de estándar para Internet se transforme en estándar aprobado. Cada componente de los Protocolos Internet *TCP* (*Transmission Control Protocol*, Protocolo de Control de Transmision) *UDP* (*User Datagram Protocol*, Protocolo de Datagramas de Usuario) y *RTP* (*Real-time Transport Protocol*, Protocolo de Transporte de Tiempo Real) que discutiremos, tienen uno o más *RFC* que especifican su operación.

iii) Protocolos *TCP/IP* (Protocolos que utilizan *TCP* y *IP*)

Teniendo en cuenta que la telefonía *IP* se basa en la transmisión de la voz paquetizada por redes de datos, en esta sección revisaremos conceptos fundamentales relacionados con la transmisión de datos y especialmente, con los protocolos *TCP/IP*. La familia de protocolos *TCP/IP* es la base de Internet y de muchas actuales redes corporativas. Tanto el computador local como el distante contienen un grupo de programas que constituyen el software *TCP/IP*, al que se denomina “*stock* de protocolos” (Conjunto de protocolos).

El “*stack* de protocolos” del computador local intercambia información con el “*stack* de protocolos” del computador distante y viceversa, con el fin de lograr la transferencia de

datos entre ambos extremos. La información que intercambian ambos *stock* de protocolos se refiere a:

- El tamaño de las porciones de datos (tamaño de *payload*).
- La identificación asociada con cada *payload*.
- Las acciones que se tomarán si una porción de datos se pierde o daña durante su viaje por la red.

Payload: Unidad de información que intercambian los extremos

- La aplicación residente en el extremo emisor intercambia porciones de datos con la misma aplicación residente en el extremo receptor. Estas porciones de datos, que llamaremos “*payload*”, corresponden a la “carga útil” que se transmite extremo a extremo. Para transmitirlos por la red, los *payload* se encapsulan en otros tipos de unidades de datos.
- Estas porciones de datos son las unidades de información que intercambian los extremos. Puede que sea necesario fragmentarlos para ser transferidos a través de los diversos “segmentos” ó “saltos” que conforman la red, “El *Stack* de Protocolos *TCP/IP*” determina como serán transferidas las porciones de datos desde el programa de envío hasta el programa de recepción, a través de la red *IP*.

iv) Protocolos telefonía *IP*

Los revisaremos en el punto 3.3 de esta revisión bibliográfica. Los *RFC* son documentos que especifican protocolos sobre Internet. Pueden ser propuestos por cualquier persona, sin embargo solamente la *IETF* los reconoce como *RFC* después de estrictos análisis. La página del editor *RFC* es <http://www.rfc-editor.org/>. Como ejemplo, el protocolo *IP* se especificó en *RFC* 791, en el año 1981.

2.2.2.3 Componentes de la Telefonía *IP*.

Para transferir voz en forma de datos proporcionando servicio de telefonía *IP*, por la misma red en que hay tráfico de archivos, e-mail y web, se requiere un conjunto de componentes:

- CODEC.
- Protocolos *TCP/IP*.
- Protocolos telefonía *IP*.
- Servidores de telefonía *IP* y *PBX IP*.
- *Gateways VoIP* y *Routers* (Compuertas VoIP y Enrutadores).

- Teléfonos *IP* y *Softphones*. (Telefono “*Soft*”, de *Software*, Programa de Teléfono *IP*)

Mas adelante se trataran los puntos pendientes.

2.2.3 Calidad de Servicio (*QoS*) en Redes *TCP/IP*.

En este capítulo revisaremos algunos fenómenos que ocurren durante el transporte de los paquetes por las redes *TCP/IP*, que tienen como efecto degradar la calidad de los servicios en tiempo real, como lo es el servicio de telefonía *IP*. Estos fenómenos no afectan mayormente la calidad de los servicios que no son en tiempo real, como lo es la transmisión de e-mail's y de archivos por la red de datos.

2.2.3.1 Sensibilidad del Oído Humano a las Distorsiones que Introduce la Transmisión de la Voz

El oído humano es particularmente sensible a: los retardos de la voz, provoca “eco” y efecto “Radio-Comunicador” (“*Walkie-Talkie*”); las irregularidades del ritmo verbal; y las pérdida de sonidos.

i) Efectos del Retardo sobre la Calidad de la Comunicación.

En una conversación telefónica retardos de la voz iguales o superiores a 150 mseg. son notorios y poco aconsejables, retardos de la voz mayores a 300 mseg. son molestos.

ii) Tipos de Calidad en una Conversación Telefónica según el Retardo de la Voz.

Se tipifica la calidad de diferentes tipos de comunicaciones, según los tiempos de retardo aceptados.

Tabla N°1: Calidad de Voz Según el Retardo.

	Máximo retardo para la voz (un solo sentido)
Calidad Voz:	0 - 150 mseg.
Calidad Satelital:	160 - 500 mseg.
Calidad Banda Ciudadana:	400 - 700 mseg.
Calidad Fax Y Transmisiones <i>Broadcast</i> :	450 - 800 mseg.

Entonces para tener “calidad voz” en telefonía *IP*, todos los procesos *IP* deben realizarse dentro de los 150 mseg. de retardo a que se han acostumbrado por años los usuarios telefónicos.

2.2.3.2 Variación de Retardo: *Jitter* y *Wander* (Fluctuación y Deambulación).

- Si bien el fenómeno de retardo ya degrada la calidad del servicio de telefonía *IP*, éste se degrada más aún cuando el retardo es variable.
- Las variaciones del retardo pueden ser rápidas o lentas. En el primer caso la variación de retardo se conoce como fluctuación y en el segundo caso como deambulación.
- En una conversación telefónica cualquier irregularidad en la llegada de los paquetes (fluctuación, deambulación) es molesta para el que escucha.
- Como veremos, la variación de retardo puede ser corregida, pero a costa de introducir más retardo.

En la Figura N°6 que sigue a continuación se muestra el efecto fluctuación.

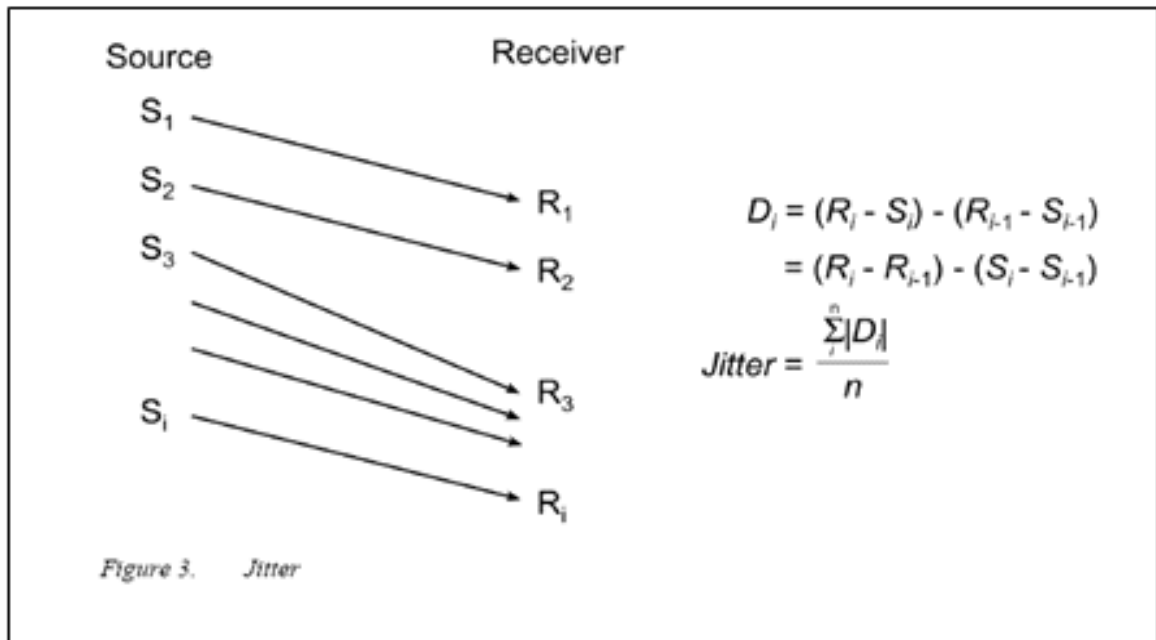


Figura N°6: Retardo *Jitter*.

La forma utilizada para contrarrestar los efectos de la variación de retardo, es almacenar las tramas recibidas en un buffer suficientemente grande, que dé tiempo a que lleguen las tramas que han tenido mayor retardo y luego ordenar las tramas en la secuencia correcta antes de reenviarlas. Es decir se logra corregir la variación de retardo, pero a costa de introducir retardo adicional. Si la variación de retardo es grande se requiere mayor tiempo de *buffering*. Si la variación de retardo es menor, se requiere menor *buffering*. Por esta razón, con el fin de disminuir el retardo adicional introducido, se usa un “buffer fluctuación adaptivo”.

El buffer fluctuación adaptivo permite ajustar el tiempo de almacenamiento de las tramas en el buffer de destino, a la cantidad de fluctuación que está ocurriendo en la red, cantidad que entre otros factores depende del tráfico que instantáneamente presiona sobre la red.

Generalmente, para efectos de los cálculos, se considera que el retardo que introduce la corrección de fluctuación es igual a 2 veces el período R entre datagramas.

2.2.3.3 Pérdida de Paquetes.

La pérdida de paquetes en el trayecto extremo-extremo, provoca como efecto pérdidas de sonidos, que son aún más molestas. La calidad de la voz se hace intolerable por pérdida de sonidos, si la pérdida de paquetes extremo-extremo supera 3%. Normalmente se acepta hasta un 1% de pérdida de paquetes, aunque es preferible que ésta no sobrepase 0,5%. No siempre es fácil lograr este objetivo, si se tiene en cuenta que en las redes IP intencionalmente se eliminan paquetes para evitar congestiones de red. En la siguiente tabla se muestra el factor de deterioro según el CODEC utilizado.

Tabla N°2: Factor de Deterioro.

Codec	Ie (0% de perdida)	Ie (2% de Perdida Aleatorio de Paquetes)	Ie (5% de Perdida Aleatoria de Paquetes)
G.711 sin PLC	0	35	55
G.711 con PLC	0	7	15
G.729 ^a	11	19	26*
G.723.1 (6.3 Kbps)	15	24	32**
* Los valores fueron para 4% perdida aleatoria de paquetes. ** Los valores fueron para 4% de perdida aleatoria de paquetes.			

$I_e = \text{Impairment Factor G.113}$ (Factor de deterioro).

En que la percepción del cliente es como la que sigue:

5 *Very good* (Muy bueno)

10 *Good* (Bueno)

20 *Adequate* (Adecuado)

30 *Limiting case* (Caso limite)

45 *Exceptional limiting case* (Caso Limite excepcional)

55 *Customers likely to react strongly* (Los clientes tienden a reaccionar fuertemente, reclamos, cambios de operador de red)

Conciliamiento de Pérdida de Paquetes (*PLC, Packet Loss Concealment*), es una técnica para enmascarar los efectos de la pérdida de paquetes en comunicaciones *VoIP*, que se revisará más adelante.

2.2.4 Protocolos Telefonía *IP*.

Nos dedicaremos ahora a estudiar protocolos correspondientes a las capas altas (sobre la pila, “*stack*”), o sea “Aplicaciones” para nuestro tema: La Telefonía *IP*. Estos protocolos son los responsables de generar los bloques de datos que hemos llamado carga (“*payload*”) y entregarlos a la pila para su transmisión por la red de datos. Como vimos anteriormente, la implementación de llamadas telefónicas *VoIP* por una red de datos involucra:

- i) El establecimiento (y disolución) de las llamadas (*Call Setup*), es decir las funciones de señalización de la telefonía tradicional: obtener tono de invitación a marcar, marcar un número, obtener un tono de llamado o de ocupado, recibir señal “abonado B contestó”. Se incluye aquí también la disolución de las llamadas (*Call Takedown*).
- ii) La comunicación telefónica misma (fase de conversación).

Durante ambas fases se requieren los protocolos de Telefonía *IP*. En la Figura N°7, que se muestra a continuación, se aprecia un diagrama del funcionamiento de una llamada *IP*.

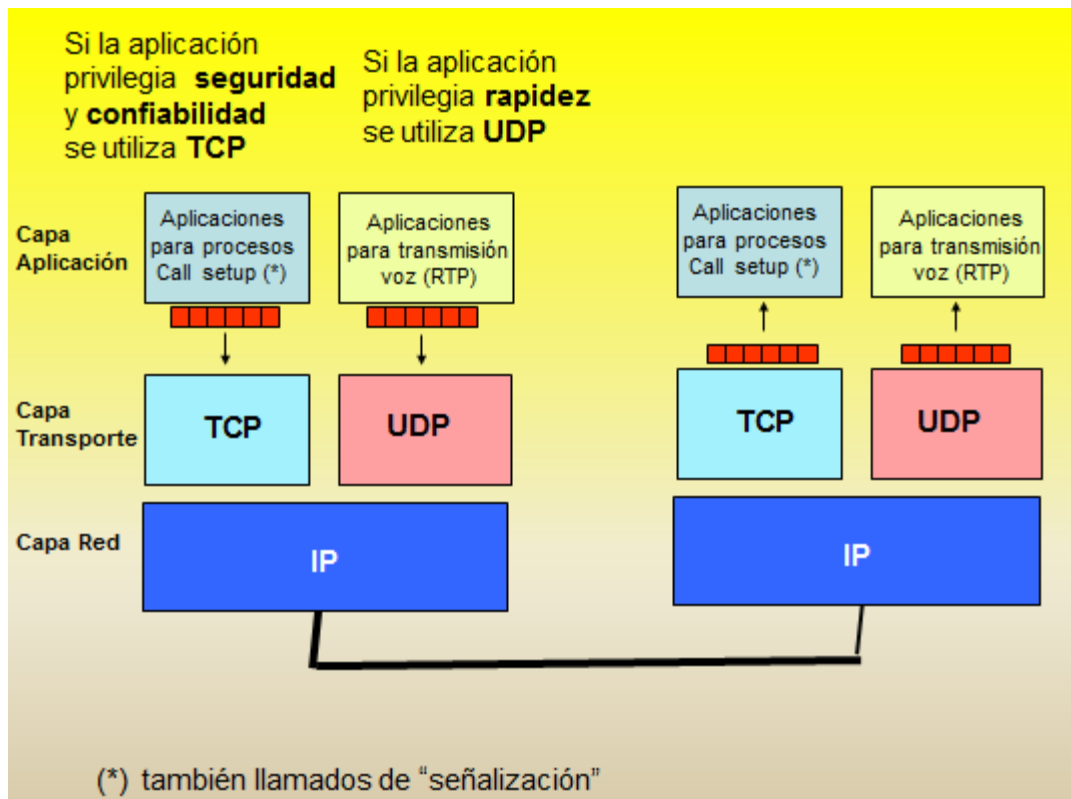


Figura N°7: Diagrama de Llamada *IP*.

i) *Call Setup* (Señalización).

Intercambio de datagramas anidados en *TCP's* y *UDP's* simulan todos los pasos de señalización que ocurren en la Red Telefónica Conmutada Pública durante establecimiento y disolución de la comunicación.

ii) Conversación.

Dos flujos de datos correspondientes a la voz digitalizada. Un flujo en cada dirección de conversación. Se distinguen dos familias de protocolos de alto nivel:

- Para transferir la información durante las fases establecimiento y disolución.
- Para transferir la voz digitalizada.

Entonces, hay dos conjuntos de protocolos de alto nivel: Protocolos de Establecimiento de Llamadas o de señalización, utilizados para intercambiar la información necesaria para establecer y disolver la comunicación. Estos protocolos “son servidos” por los protocolos *TCP* o *UDP*. Permiten simular todos los pasos de la señalización de Red Telefónica Conmutada Pública necesarios para establecer y deshacer las comunicaciones. Y el protocolo para el transporte del flujo de voz, llamado Protocolo de Transporte de tiempo-real (*RTP, Real-Time Transport Protocol*). Este protocolo se utiliza para el transporte de cada uno de los dos flujos de voz (uno para cada dirección de conversación). Usa el protocolo *UDP*.

2.2.4.1 Protocolos *Call Setup* o de Señalización (utilizados en las fases de establecimiento y disolución de llamada).

- En telefonía *IP* cuando se habla de protocolos de Establecimiento de Llamadas, se debe entender que estos también realizan las funciones de disolución de las llamadas (*Call Takedown*).
- Los protocolos de establecimiento de Llamadas administran funciones como:
 - Mapeo de los números telefónicos con las direcciones *IP*.
 - Generación de tonos de invitación a marcar y señales de ocupado.
 - Timbre de llamado (*ringing*, Teléfono Sonando).
 - Descuelgue.
 - Liberación de Llamada (*Release*).
- Los Protocolos de Establecimiento de Llamadas o de señalización corresponden a los programas que implementan los procesos para el establecimiento y disolución de las comunicaciones.

- Utilizan tanto *TCP* como *UDP* para encapsular los datos que es necesario intercambiar durante las fases de establecimiento y disolución de las llamadas.

Los Principales Protocolos de Establecimiento de Llamadas o de Señalización.

Los protocolos de Establecimiento de Llamadas impulsados por el mundo de la conmutación telefónica han sido desarrollados por la *ITU-T* y se agrupan en la recomendación H.323: Protocolo H.323. Los protocolos de Establecimiento de Llamadas impulsados por el mundo de la transmisión de datos han sido desarrollados por la *IETF*. Entre ellos destaca: Protocolo *SIP* (Protocolo de Iniciación de Sesión, *Session Initiation Protocol*) *IETF RFC 2543*. Entre los *protocolos* desarrollados en forma conjunta por la *ITU-T* y por la *IETF* están: Protocolo *Megaco* / H.248 (acrónimo también de Protocolo de Control de Compuerta de Media, *Media Gateway Control Protocol*) *IETF RFC 2885*, *MGCP* (Protocolo de Control de Compuerta de Media, *Media Gateway Control Protocol*). *IETF RFC 2705*. *SIP* y *Megaco* son protocolos livianos y por tanto más rápidos.

1) Arquitecturas Centralizada y Distribuida.

En el pasado todas las redes de voz se construían en base a arquitectura centralizada: terminales pasivos (los teléfonos) todos controlados por centrales en que estaba toda la inteligencia de la red. La arquitectura centralizada tiene la ventaja de la simplicidad del manejo de las llamadas y es adecuada al servicio telefónico básico, pero es poco flexible para introducir terminales más sofisticados y nuevos servicios. Cuando hablamos de inteligencia de la red nos referimos al estado de las llamadas, características de las llamadas, enrutamiento de las llamadas, provisión de servicios, tarificación y cualquier otro aspecto relacionado con el manejo de las llamadas. La telefonía *IP* tiene la ventaja que puede ser diseñada en base a arquitecturas centralizadas o distribuidas, aprovechando las ventajas de cada una de ellas. Ejemplos de estos son:

i) Arquitectura Centralizada: Protocolos *MGCP* y *Megaco/H.248*.

Estos protocolos fueron diseñados pensando en la existencia de un dispositivo central llamado Controlador de Compuerta de Media (Controlador de Compuerta de Medio), Controlador o Administrador de llamadas (*Call Manager*), que maneja la lógica de conmutación y el control de la llamada. El dispositivo centralizado indica a las Compuertas (*Gateways*) como transmitir los flujos *RTP* con las muestras de voz, las que sí se transmiten directamente entre compuertas. La inteligencia de red está centralizada y los terminales son relativamente pasivos (con características, *features*, limitados o no residentes en ellos). Aunque la mayoría de las arquitecturas centralizadas usan los protocolos *MGCP* o *Megaco/H.248*, también es posible diseñarlas en base a protocolos *SIP* o H.323, usando Agentes de Usuarios de Espalda contra Espalda (*B2BUAs*, *Back to Back User Agents*) o Guardian de Compuerta (*Gatekeeper*), ruteadores de señalización (*GKRCS*), respectivamente.

- Ventajas: Centraliza el control de las llamadas, la administración y la provisión de los servicios. Simplifica el flujo de las llamadas. Replica las características de la telefonía tradicional. Los Ingenieros de telefonía tradicional la prefieren.

- Desventajas: Sus detractores dicen que ahoga la innovación de los *features* de los terminales y que es un obstáculo para diseñar nuevos servicios que no existen en la telefonía tradicional.

ii) Arquitectura Distribuida: Protocolos H.323 y *SIP*.

Estos protocolos permiten distribuir la inteligencia de la red entre los terminales y dispositivos controladores de llamadas. Los terminales pueden ser Compuertas, Teléfonos *IP*, *PC*, Servidores o cualquier dispositivo que pueda iniciar y terminar una llamada de telefonía *IP*. El controlador de llamadas se denomina Guardian de Compuerta en redes H.323 y Servidores Redireccionadores (*Proxy* o *Redirect Servers*) en redes *SIP*.

- Ventajas: Es más flexible. Permite tratar las aplicaciones de telefonía *IP* como cualquier otra aplicación *IP*. Permite agregar inteligencia a cualquier terminal y dispositivo de control de llamadas, dependiendo de los requerimientos del negocio que se está implementando y de la tecnología de la red. Los ingenieros de redes *IP* se encuentran más a gusto en este ambiente.
- Desventajas: Sus detractores critican la mayor complejidad. Además indican que la infraestructura de RTCP es el único modelo que debiera usarse cuando se trata de replicar los servicios de voz existentes.

La empresa CISCO desarrolló el protocolo propietario llamado Protocolo de Control de Cliente Ligerero (*SCCP*, “*Skinny Client Control Protocol*”), que como su nombre lo indica destaca por ser un protocolo liviano.

SCCP sigue el modelo maestro – esclavo (arquitectura centralizada). Se emplea principalmente para el control de los teléfonos *IP* Cisco. Los teléfonos *IP* Cisco con *SCCP*, son clientes del servidor Administrador de Llamadas y se comunican durante la fase *call setup* de la llamada a dicho servidor utilizando el protocolo *SCCP* para el intercambio de los parámetros de control de la llamada. Los segmentos provenientes de la aplicación, *SCCP* se transmite sobre *IP*. Los clientes livianos (teléfonos *IP*) para establecer, supervisar y deshacer las llamadas pueden también interoperar con terminales H.323, a través de un servidor redireccionador H.323 (*proxy*, generalmente el mismo *Call Manager*). Existen modelos de teléfonos *IP* de bajo costo que funcionan en protocolo *SCCP*. Otra ventaja de *SCCP* es que ya se han desarrollado muchas facilidades para las existentes en la red RTCP.

Los protocolos de señalización o *call setup* se pueden clasificar según ellos sean utilizados para señalización de usuario o para señalización de red. La siguiente figura resume los protocolos según esta clasificación. El protocolo *TRIP* (*RFC 3219*) provee un medio para que los nodos de la red *TCP/IP*, a través de servidores de localización (*Location Servers*), ubiquen el Gateway adecuado. *TRIP* es un protocolo diseñado para ser usado entre operadores. Permite conocer la accesibilidad, negociar las aptitudes y especificar los atributos de las Compuertas que participan en la comunicación. Como subproducto, *TRIP* entrega a los operadores información estadística valiosa para dimensionar las redes. La Figura N°8 que se observa a continuación resume los distintos tipos de protocolos.

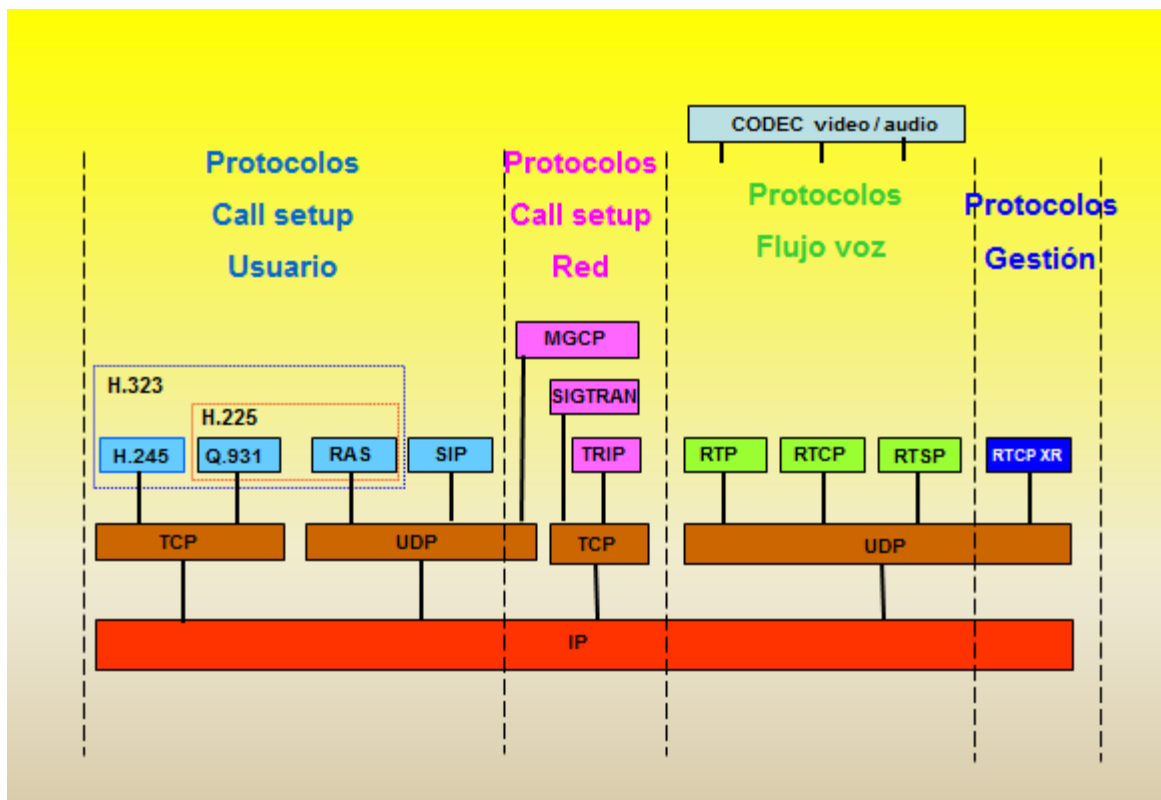


Figura N°8: Diagrama de Protocolos y Codecs.

Real-time Transport Control Protocol (RTCP, Protocolo de Control de Transporte de Tiempo Real, PCTR) se definió en *RFC 3550*, la que reemplazó a *RFC 1889*. RTCP permite controlar la información de flujos *RTP* (control “fuera de banda”). Se usa para transmitir periódicamente control de la transmisión de los paquetes a los participantes de la sesión multimedia. Provee *feedback* (retroalimentación, información) sobre la calidad de servicio que está siendo proporcionada por *RTP*.

Real Time Streaming Protocol (RTSP, Protocolo de Flujos de Tiempo Real), desarrollado por *IETF* y especificado en *RFC 2326* (año 1993), es un protocolo que se usa en sistemas de “*streaming media*”ⁱⁱ (Canales de Media), que permite al cliente controlar remotamente el servidor *streaming media*, en forma similar a lo que hacemos con los comandos “*play*” & “*pause*” (Reproducir y Pausar) en un reproductor de DVD, permitiendo acceso basado en el tiempo (“*time-based*”) al servidor.

Protocolo H.323 (*ITU-T*)

La versión uno de H.323 se publicó en 1996. Se puede utilizar tanto en redes con arquitectura centralizada como distribuida. Especifica varios protocolos para comunicaciones de multimedia por redes de paquetes, es decir por redes que no garantizan completamente la calidad de servicio. H.323 prevé la transmisión en tiempo real de video y audio por redes de paquetes.

ⁱⁱ La tecnología de canales de media (“*streaming media*”) permite procesar contenidos multimedia (música, vídeo) sin necesidad de esperar a que éstos se descarguen completos al disco duro para iniciar su escucha o visualización.

La versión dos de H.323, publicada en 1998 y una versión tres posterior, introducen mejoramientos que agregan facilidades de la telefonía tradicional en la telefonía *IP*, como por ejemplo transmisión de fax, servicios suplementarios (H.450), conferencia de llamados, etc.

Especifica mecanismos para el establecimiento, supervisión y disolución de los flujos de información, incluyendo los flujos de información de audio, entre dos terminales que cumplen H.323. Usa modelo *peer to peer* (par a par). H. 323 es realmente una familia de estándares, basados en la telefonía, para multimedia, incluyendo voz y videoconferencias. H.323 es un conjunto de protocolos robusto y monolítico, es decir contiene todo lo necesario para funcionar. Esto tiene ventajas y desventajas. Desventajas porque es poco flexible, de adaptación lenta a nuevas tecnologías.

La familia H.323 ha ido siendo perfeccionada durante años, y como resultado ya ha alcanzado gran robustez. Pero el costo de su robustez se paga con un alto exceso de *overhead* (procesamiento): una sesión incluye gran cantidad de *handshakes* (saludos) y datos intercambiados para ejecutar cada función. H.323 es ampliamente utilizado. H.323 utilizado en arquitectura distribuida, permite a los operadores construir redes escalables, flexibles, redundantes. H.323 provee mecanismos para la interconexión con otras redes *IP* y permite que la inteligencia de la red resida o en los terminales o en los Guardianes de *Gatekeepers* (Compuertas). La figura que sigue (N°9) muestra un diagrama de los protocolos H.323.

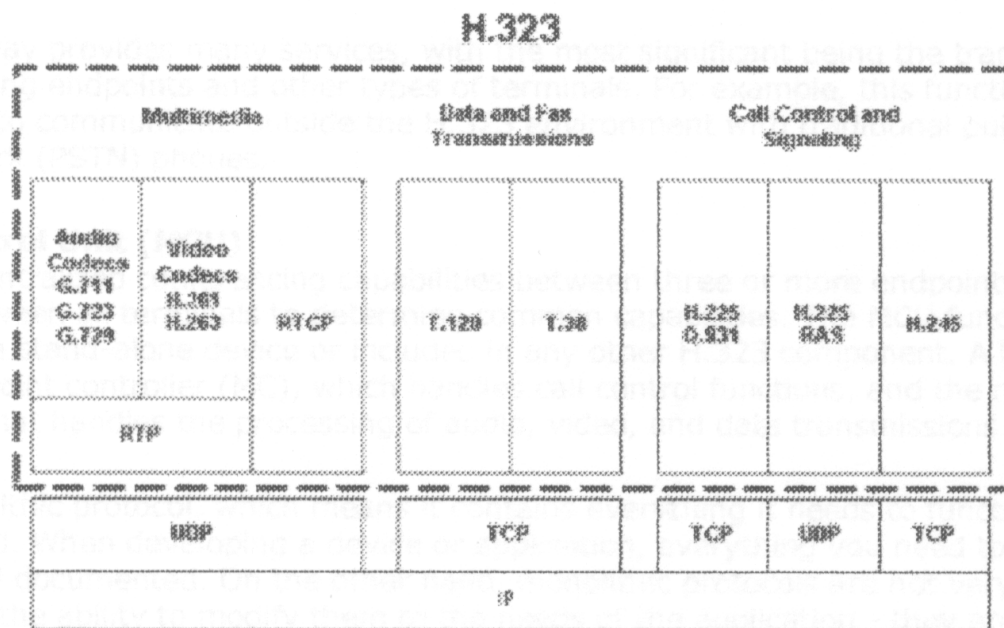


Figura N°9: Diagrama de Protocolos H.323.

H.323 es un conjunto de normas que especifican Componentes, Protocolos y Procedimientos para soportar comunicaciones multimedia sobre redes de conmutación por paquetes.

H.323 cubre varios sub-protocolos que tienen que ver con el Establecimiento de las llamadas y la señalización a través de redes LAN y WAN (*Local Area Network* y *Wide Area Network*, Red de Area Local y Red de Area Amplia).

Componentes de H.323.

Se distinguen cuatro componentes principales:

- **Terminal:** Corresponden a los dispositivos extremos de red capaces de proveer comunicaciones en tiempo real y *full-duplex*. Un terminal puede estar constituido por un simple teléfono *IP* o un computador con software con funcionalidades H.323. Los dispositivos “terminales” deben soportar software de *Call Setup* (Establecimiento de Llamada), protocolo *RTP*, CODECs y *stack TCP/IP*. Como opción pueden disponer de software para comunicaciones de video.
- **Gatekeeper:** Cuando esta entidad está presente, todo dispositivo H.323 debe registrarse en ella antes de iniciar la comunicación con otro dispositivo H.323. El registro se realiza utilizando protocolo *RAS* (*Registration, Admission, Status*, Registro, Admisión, Status) sobre *UDP*, el que es parte de la especificación H.225. Desde el punto de vista lógico, el Guardian de Compuerta (*Gatekeeper*) está separado de los terminales H.323, pero sus funciones pueden cohabitar en la Compuerta (*Gateway*) y en el *MCU* (Unidad de Control de Multipuntos, *Multipoint Control Unit*). En nomenclatura H.323 el Guardian de Compuertas es el controlador de llamadas. Tiene asignadas varias funciones como control de admisión de los usuarios, traducción de número telefónico a dirección *IP*, gestión de Ancho de Banda, gestión de zona, etc.
- **Gateway (Compuerta):** Esta entidad provee varias funciones, entre las cuales la más importante es la traducción entre terminales H.323 y otros tipos de terminales. Así por ejemplo tiene la funcionalidad para la comunicación de terminales telefónicos *IP* con terminales telefónicos tradicionales ubicados en la Red Telefónica Conmutada Pública.
- **Multipoint Control Unit (MCU, Unidad de Control de Multipunto):** Soporta en forma centralizada las capacidades de conferencia de tres o más extremos que establecen una conferencia, y maneja las negociaciones entre los terminales para determinar las capacidades comunes. Las funcionalidades *MCU* pueden residir en un teléfono *IP* o estar incluida en cualquier otro componente H.323. *MCU* tiene dos partes principales: Controlador de Multipuntos (*Multipoint Controller, MC*), que maneja las funciones de control de llamada y Procesador de Multipunto (*Multipoint Processor, MP*) que maneja el procesamiento de audio, video y transmisión de datos. H.323 trabaja bien en aquello para lo que fue diseñado: comunicaciones multimedia entre dispositivos conectados a una red de paquetes. Sin embargo cuando se creó, no se tuvo en mente gran cantidad de usuarios, por lo que tiene limitaciones cuando se usa en redes WAN. Hay quienes opinan que todo es cuestión de maduración y que H.323 “se pondrá a la altura de lo requerido”. Los diferentes protocolos a que se refiere la recomendación H.323 son usados en las diferentes fases de la comunicación. Así por ejemplo, en la primera fase, en que el abonado llamante inicia la comunicación, se utilizan los protocolos de registro, admisión y estatus (*RAS*) que especifica la recomendación H. 225.0. La señalización entre terminales A y B se hace de acuerdo a las especificaciones Q.931 y H.225.0. El control de la sesión, que incluye la

negociación entre A y B para decidir el CODEC que se utilizará durante el flujo *RTP*, el intercambio de mensajes para control de flujo y otras funciones de control de sesión, se hace siguiendo las especificaciones H.245. Los protocolos generan mensajes, los que como hemos visto, se parcelan en unidades de datos (*payloads*, carga) para efectos de transmitirlos por la red de datos.

Mensajes *RAS/H.225* usados para Registro, Admisión, Estado.

Usados durante diálogo entre terminal H.323 y *Gatekeeper* orientado al registro, autorización y control de estado del terminal H.323 en la red. Este diálogo se efectúa por el llamado “canal lógico *RAS*” que se establece en toda comunicación.

Ej. *ARQ* Petición de Admisión (*Admission ReQuest*)

ACF Confirmación de Admisión (*Admission Confirmation*)

Mensajes *Q.931/H.225* usados para enrutamiento de la llamada.

Usados para la señalización propiamente tal, la que se basa en el protocolo RDSI *Q.931*. En toda comunicación telefónica se establece un “canal lógico de señalización” por el cual se realiza este diálogo.

2.2.4.2 Protocolo *RTP* para la fase de conversación

- El intercambio de los datos que corresponden a la voz digitalizada, ocurre después del Establecimiento de Llamadas y antes de la disolución de la Llamada (*Call Setup* y *Call Takedown*).
- Los datos a intercambiar provienen de los CODEC. Corresponden a los datagramas que producen los CODEC durante el proceso de muestreo y digitalización de la voz.
- Estos datos son intercambiados mediante dos flujos de datos - uno en cada dirección - haciendo así posible que ambos participantes en la conversación telefónica hablen al mismo tiempo (existen dos “canales lógicos para la voz”).
- Para cada uno de estos dos flujos se usa un protocolo de capa alta, llamado Protocolo de Transporte de Tiempo-Real (*Real-time Transport Protocol*, *RTP*), el que es encapsulado en *UDP* para su viaje por la red.
- El protocolo *RTP* se conoce también como protocolo IETF RFC 1889.
- *RTP* es ampliamente usado para transportar flujos de datos correspondiente a audio (datagramas generados por CODEC *G.7XX*) y video (datagramas generados por CODEC *H.26X*).
- Está diseñado para aplicaciones que envían datos en una dirección sin señal de reconocimiento (sin acuse de recibo, *acknowledgement*).

- El encabezamiento de cada datagrama RTP contiene un “timbre” con la hora (*timestamp*). Este “Timbre de hora” permite que la aplicación en el extremo receptor reconstruya exactamente el “cronometraje” (“*timing*”) de la información original.
- El encabezamiento también contiene un número secuencial, que permite al extremo receptor detectar y tomar medidas cuando los datagramas sufren pérdida, duplicación o deterioro.
- Los dos flujos RTP, es decir la conversación bi-direccional misma, son elementos importantes que determinan la calidad de la llamada, en lo que se refiere a la calidad de la voz durante la conversación.
- Veremos la composición de las unidades de datos RTP cuya carga útil o *payload* corresponde a la voz codificada en forma de datos.
- Cada datagrama más un encabezamiento RTP, se encapsula en un segmento UDP, y éste a su vez se encapsula en un paquete IP.
- El encabezamiento del RTP lleva la información necesaria para el correcto reensamblado y decodificación del *payload*.
- Un campo del encabezamiento IP indica a los dispositivos de la red que necesitan esta información, que la transferencia de la unidad RTP se está haciendo mediante protocolo UDP.
- Como ya lo vimos, el *header* IP contiene la información necesaria para que el paquete “viaje” correctamente por la red. Parte de dicha información corresponde a las direcciones de origen y destino del paquete.
- En el paquete IP el encabezamiento de RTP va a continuación del encabezamiento de UDP y éste a continuación del encabezamiento de IP, como se aprecia en la figura N°10:

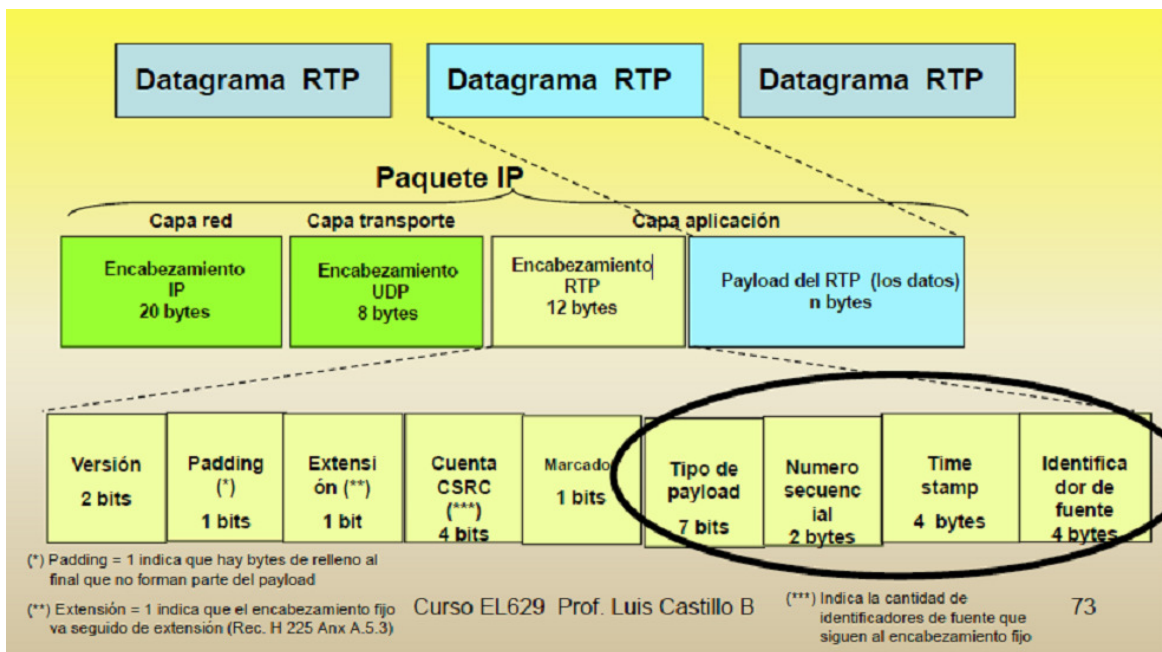


Figura N°10: Estructura de Paquete RTP.

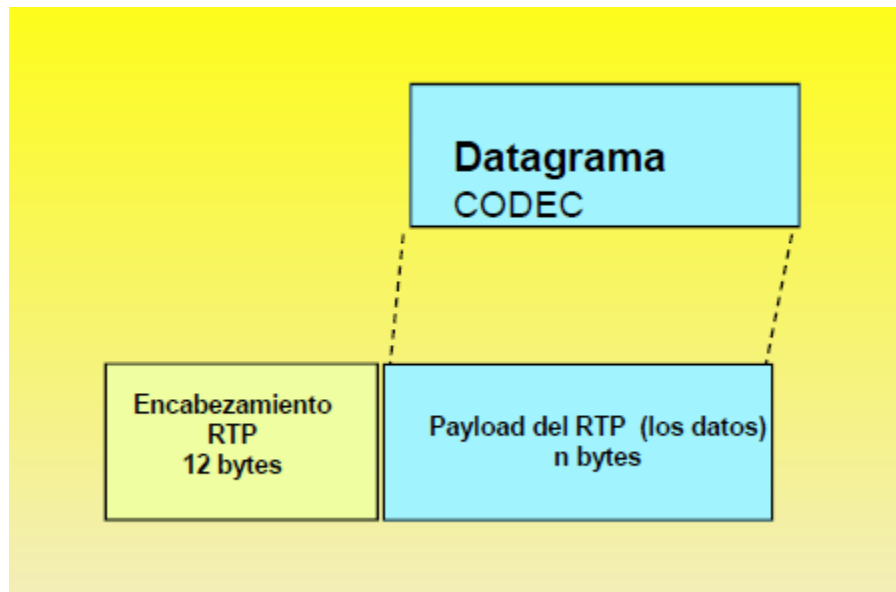


Figura N°11: Partes de un Paquete RTP.

- La unidad de información del Protocolo de Transporte de Tiempo Real corresponde a un datagrama entregado por el CODEC más el encabezamiento de RTP, como se parecía en la figura N°11.

RTCP (Real-time Transport Control Protocol)

- Describe el intercambio de mensajes de control relacionados con la Calidad de Servicio (*QoS*): retardo, *jitter*, tasa de pérdidas, etc.

- Su uso es opcional, pero recomendable ya que entrega información del estado de la comunicación en lo referente a la *QoS*.
- No tiene mecanismos para mejorar la *QoS*.
- Entrega además otros servicios como sincronización entre medios.

cRTP (RTP comprimido, *compressed RTP*)

Con el fin de aumentar la eficiencia en el uso del ancho de banda, este protocolo utiliza el procedimiento de comprimir los encabezamientos RTP. Sin embargo el precio que se paga no es bajo: mayor retardo introducido por los procesamientos de compresión y descompresión.

2.2.5 CODEC.

2.2.5.1 Principios de Funcionamiento de los CODEC Usados en Telefonía.

La función de los CODEC es reducir el volumen de bytes de información a fin de ahorrar Ancho de Banda en la red de datos por la que se transmitirá, y ahorrar espacio de almacenamiento en los dispositivos en que eventualmente se decida grabar la información. Para realizar esta función el diseño de los CODEC tienen en cuenta que los archivos de datos en que está codificada la información contienen bytes que se pueden clasificar como Redundantes (información repetitiva y fácil de predecir), otros que son Irrelevantes (por Ej. frecuencias inaudibles, ruido de fondo durante silencios), y por último otros que corresponden a la información básica o Relevante (la necesaria para reconstruir la información). La compresión puede ser:

- Sin pérdidas reales: Se elimina la información redundante.
- Sin pérdidas subjetivas: Se elimina la información redundante y también la irrelevante.
- Con pérdidas: Se elimina la información redundante, irrelevante y parte de la básica. En este caso se obtiene una información semejante a la original con cierta degradación de su calidad.

Técnicas para Codificar, Comprimir y Evitar Degradación de Calidad.

Los CODEC usan técnicas sofisticadas para codificar y para comprimir, respaldadas por modernas teorías matemáticas y estadísticas, que están fuera del alcance de este curso. Sus nombres indican la forma en que se realiza el trabajo: Cuantización de Multi-pulsos de Máxima Probabilidad (*Multi-Pulse Maximum Likelihood Quantization, MPMLQ*); Predicción Lineal Excitada de Código Algebraico (*Algebraic Code Excited Linear Prediction, ACELP*), disponible para G.729 y G.723.1. La técnica de ocultamiento de pérdida de paquetes (*packet loss concealment, PLC*) es una característica adicional disponible en los CODEC G.711u y G.711a. Como las conexiones *VoIP* son en tiempo real no es posible usar técnicas de control de errores como *ARQ* (Petición de Repetición Automática, *Automatic Repeat-Request*), por lo que se recurre

al enmascaramiento de los paquetes perdidos. *PLC* no agrega retardo ni otros problemas secundarios, pero encarece los CODEC's.

El Conciliamiento de Pérdida de Paquetes (*PLC*), es una técnica para enmascarar los efectos de la pérdida de paquetes en comunicaciones *VoIP*. *PLC* realiza las siguientes acciones para enmascarar las pérdidas:

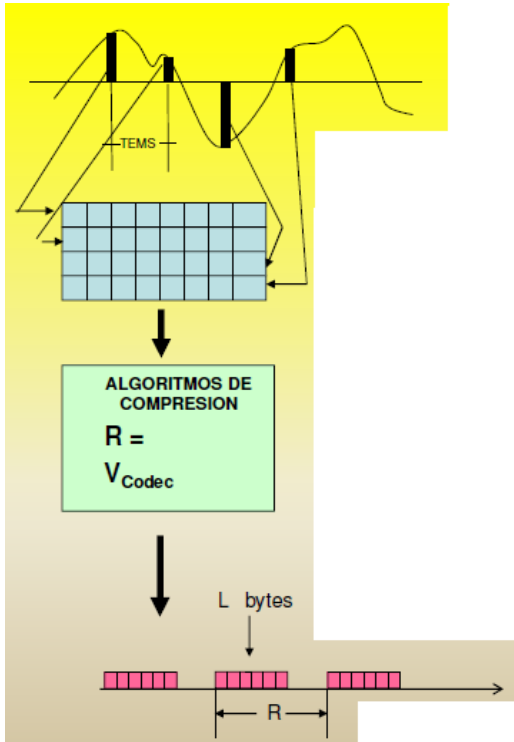


Figura N°12: Diagrama de Codificación.

En la figura N°12 se resumen los procesos y funciones que realizan los CODEC utilizados en telefonía.

El proceso es como sigue: se toman muestras de la señal de voz analógica cada $TEMS$ segs:

$TEMS =$ Tiempo Entre Muestras Sucesivas de la señal de voz analógica.

Las muestras se cuantizan, codifican y con ellas se va creando en tiempo real un archivo digital que contiene la información de la voz. Con el fin de ahorrar Ancho de Banda, el archivo se somete a algoritmos que eliminan la información correspondiente a silencios y la información redundante. Los procesos a que se somete la información se configuran fijando diversos parámetros, entre los que destaca R . En su puerta de salida el CODEC entrega un flujo binario cuya tasa binaria es la "velocidad nominal del CODEC" (V_{codec}). El flujo corresponde a datagramas de L bytes entregados cada R segs.

Cada cierto tiempo $TEMS$ se toma una muestra de la señal de audio analógica y se somete a varios procesos: *PAM* (Modulación de Amplitud de Pulso, *Pulse Amplitud Modulation*),

cuantización, codificación, compresión, etc.ⁱⁱⁱ con el fin de crear bloques de bits que contienen la información correspondiente a muestras de voz, a los que se denomina datagramas. Los datagramas se transmiten por la red al CODEC correspondiente, donde se realiza el proceso inverso. Cada datagrama tienen L bytes y se envía al CODEC correspondiente cada cierto intervalo de tiempo R. Cada datagrama corresponde a una “porción” de información digital que constantemente generan los procesos antes indicados. El tiempo requerido por el CODEC para coleccionar y procesar la información correspondiente a las muestras de voz analógica que se codifican en los datagramas, provoca el “retardo de paquetización” que introduce el CODEC.

Datagramas Entregados por el CODEC.

Relación entre V_{Codec} , R y L.

El CODEC entrega la información cada cierto período de tiempo R fijo (pero ajustable), en forma de datagramas de tamaño L, como se muestra en la figura N°13.

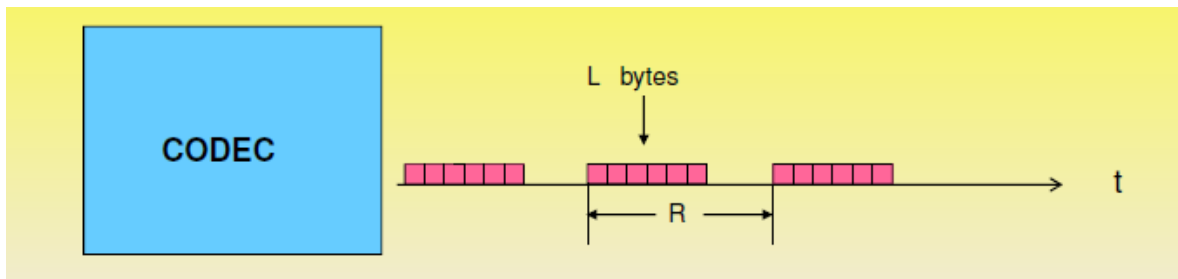


Figura N°13: Relación entre V_{Codec} , R y L.

V_{Codec} = Tasa de datos nominal del CODEC ó velocidad del CODEC (bits/seg.)

R = Tiempo o período entre dos datagramas sucesivos (seg.)= 1/Frecuencia

L = Tamaño del datagrama (bytes)

El flujo binario, o sea la cantidad de bits por segundo es:

$$V_{\text{Codec}}(\text{bps.}) = \frac{1}{R} * L(\text{bytes}) * 8(\text{bits})$$

El largo de los datagramas, o sea la cantidad de bytes de los datagrama es:

$$L = \frac{V_{\text{Codec}} * R}{8}$$

Para una V_{Codec} dada, si se aumenta el tiempo R entre datagramas, aumenta el largo L del datagrama, ya que en un segundo el transporte de la misma cantidad de bits, se realiza en menor

ⁱⁱⁱ Otro algoritmo que se aplica en algunos CODEC permite discriminar si la señal analógica de voz corresponde a voz propiamente tal o si corresponde a tonos, como por Ej. DTMF o de equipos FAX o Modem.

número de envíos (con menor frecuencia). Así por ejemplo, si para el CODEC G.711 el tiempo R entre datagramas aumenta de 0,020 segs. a 0,030 segs., el largo del datagrama aumenta de 160 a 240 bytes.

Algunos teléfonos IP permiten ajustar el “retardo entre paquetes” (*delay between packets*) o el “largo de los paquetes de voz” (*speech packet length*) y la “duración del paquete” (*packet duration*). La unidad utilizada en los tres casos es el msecs. Las tres expresiones indican lo mismo y se refieren al intervalo de tiempo entre dos datagramas, es decir corresponde al período de tiempo R entre entregas de datagramas consecutivos. Por ejemplo, si la tasa de salida del CODEC emisor es 64 Kbps, y el ajuste se pone en “datagrama de voz 0,010 segs.”, se tendrá que el extremo emisor creará y entregará a la red datagramas cada 0,010 segs. Por lo tanto en este caso el largo L del datagrama resultará:

$$L = (V_{\text{Codec}} * R) / 8 = 80 \text{ bytes}$$

2.2.5.2 Otras especificaciones de los CODEC usados en telefonía. Velocidad Nominal del CODEC y Retardo que introduce.

Como vimos anteriormente, los CODEC utilizados en telefonía realizan las funciones de muestrear a intervalos regulares la señal analógica de voz, cuantizar las muestras en valores discretos, codificar, comprimir, producir bytes de 8 bits y ensamblarlos en datagramas para que sean transmitidos por la red de datos. La velocidad nominal a que el CODEC entrega los datagramas en su salida es uno de los factores determinantes del ancho de banda que se ocupa en la red de datos para transmitir los datagramas. De aquí que sea atractivo diseñar CODEC con bajas velocidades de salida. Velocidades de salida bajas se logran mediante procesamientos más complejos que consumen mayor tiempo provocando retardo, el que afecta en parte la calidad del audio. La tabla de la página siguiente muestra comparativamente estas características para los CODEC más utilizados. A continuación se muestra en la tabla N°3 un cuadro descriptivo de CODEC's.

Tabla N°3: Cuadro descriptivo de CODEC's.

CODEC	Velocidad nominal de salida (Kbps)	Calidad del audio	Complejidad del procesamiento	Retardo (ms)
G.711	64	Excelente	Muy baja	Despreciable
G.723.1	5,6	Buena	Muy alta	Importante
G.723.1	6,4	Buena	Muy alta	Importante
G.726	32	Buena	Baja	Despreciable
G.728	16	Buena	Media	Despreciable
G.729 (A)	8	Buena	Alta	Apreciable

Período R entre Datagramas y Consumo de Ancho de Banda en la Red de Datos.

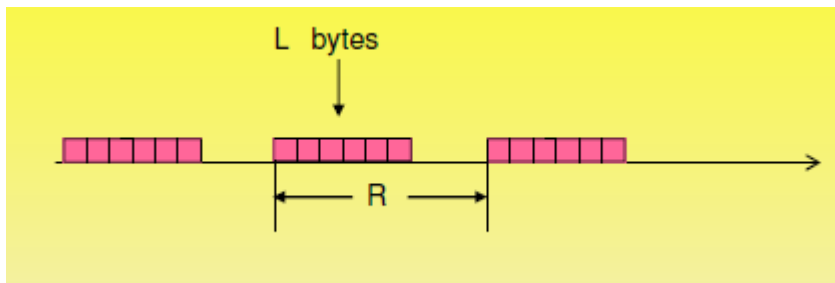


Figura N°14: Período de tiempo R entre dos datagramas.

Por ejemplo si $R = 0,020$ segs. el CODEC entrega $1 / 0,02 = 50$ datagramas por segundo. La selección del valor a utilizar para la frecuencia de los datagramas es un compromiso entre requerimiento de calidad y consumo de ancho de banda. Si se decide alta frecuencia de datagramas, ellos resultarán cortos (pocos bits) y la pérdida de algunos datagramas durante la transmisión hasta el CODEC receptor no afectará mayormente a la calidad. Pero el costo que pagamos es que al existir muchos datagramas, habrá muchos encabezamientos que suben el overhead requiriéndose mayor ancho de banda, como lo veremos más adelante. El valor más utilizado para R es 0,020 segs., salvo para los CODEC G.723.1 en que se utiliza $R = 0,030$ segs. El valor $R = 0,020$ segs. es utilizado en los ejemplos que *IETF (Internet Engineering Task Force)* incluye en la *RFC1889*.

Período TEMS entre Toma de Muestras de la Señal Analógica de Voz.

Una de las características importantes de cada CODEC es el intervalo de tiempo que transcurre entre la toma de dos muestras sucesivas de la señal analógica de voz (TEMS). Para los principales CODEC se tiene:

Tabla N°4: Tiempo entre toma de muestras sucesivas (TEMS) según CODEC.

CODEC	Tiempo entre toma de muestras sucesivas (TEMS)
G.711 PCM	0,000125 seg. = 0,125 ms.
G.723.1 ACELP	0,030 seg. = 30 ms.
G.723.1 MP-MLQ	0,030 segs. = 30 ms.
G.726 ADPCM	0,000125 segs. = 0,125 ms.
G.728 LD-CELP	0,000625 segs. = 0,625 ms.
G.729 (A) CS-ACELP	0,010 seg. = 10 ms.

R / TEMS

A veces se utiliza el parámetro $R / TEMS$, que corresponde al cociente entre el tiempo R entre datagramas sucesivos y el tiempo entre tomas de muestras sucesivas de la señal analógica de voz. La tabla siguiente indica el valor de este parámetro para algunos casos:

Tabla N°5: TEMS, R y R/TEMS según CODEC.

CODEC	TEMS (ms.)	R (ms.)	R/TEMS
G.711	0,125	20	160
G.711	0,125	30	240
G.723.1 ACELP	30	60	2
G.726	0,125	20	160
G.726	0,125	30	240
G.728	0,626	20	32
G.728	0,626	30	48
G.729(A)	10	20	2
G.729(A)	10	30	3

2.2.5.3 Retardo de Paquetización.

Como hemos visto, los CODEC realizan una serie de tratamientos a la información: toman muestras a intervalos regulares de la señal analógica de voz, cuantizan las muestras en valores discretos, codifican, comprimen, producen bytes de 8 bits y los ensamblan en datagramas, que finalmente entregan a su salida para ser transmitidos por la red de datos hasta el CODEC correspondiente, donde se realizan los procesos inversos hasta recuperar la señal de voz. El tratamiento que los CODEC hacen a la información, desde que se toman las muestras a la señal de voz analógica hasta que se obtiene en su puerta de salida el datagrama correspondiente a ellas, obliga a ocupar tiempos de almacenamiento en *buffers* (espacio de memoria, en el que se almacenan datos para evitar que el programa o recurso que los requiere, ya sea hardware o software, se quede sin datos durante una transferencia), los que junto a los tiempos de procesamiento provocan lo que se conoce como el retardo total introducido por el CODEC o “retardo de paquetización”. Este retardo, que es fijo, se introduce tanto por el CODEC transmisor como por el CODEC receptor. Los retardos introducidos por los CODEC se suman a los propios de la red de datos, pudiendo llegarse a valores de latencia que degradan la calidad. La tabla siguiente muestra los retardos de paquetización introducidos por los CODEC más comunes usados en telefonía *IP*.

Retardos de paquetización que introducen los CODEC más comúnmente usados para *VoIP*

Tabla N°6: Velocidad de Salida Nominal y Retardo de Paquetización por CODEC.

Nombre del CODEC	Velocidad de salida nominal (V_{Codec})	Retardo de paquetización ^{iv}
G.711u	64,0 Kbps.	1,0 ms.
G.711a	64,0 Kbps.	1,0 ms.
G.726-32	32,0 Kbps.	1,0 ms.
G.729	8,0 Kbps.	25,0 ms.
G.723.1 MPMLQ	6,3 Kbps.	67,5 ms.
G.723.1 ACELP	5,3 Kbps.	67,5 ms.

2.3 Concepto Asterisk.

2.3.1 ¿Qué es Asterisk?

Asterisk es una aplicación de software libre (bajo licencia *GPL*, Licencia Publica General, *General Public License*) de una central telefónica (*PBX*). Como cualquier *PBX*, se puede conectar un número determinado de teléfonos para hacer llamadas entre sí e incluso conectar a un proveedor de *VoIP* o bien a una RDSI tanto básicos como primarios. Mark Spencer, de Digium, inicialmente creó Asterisk y actualmente es su principal desarrollador, junto con otros programadores que han contribuido a corregir errores y añadir novedades y funcionalidades. Originalmente desarrollado para el sistema operativo GNU/Linux, Asterisk actualmente también se distribuye en versiones para los sistemas operativos BSD, MacOSX y otros, aunque la plataforma nativa (GNU/Linux) es la mejor soportada de todas.

Asterisk incluye muchas características anteriormente sólo disponibles en sistemas costosos propietarios *PBX* como buzón de voz, conferencias, *IVR* (Respuesta de Voz Interactiva, *Interactive Voice Response*), distribución automática de llamadas, y otras muchas más. Los usuarios pueden crear nuevas funcionalidades escribiendo un dialplan en el lenguaje de script de Asterisk o añadiendo módulos escritos en lenguaje C o en cualquier otro lenguaje de programación soportado por Linux. Para conectar teléfonos estándar analógicos son necesarias tarjetas electrónicas telefónicas *FXS* o *FXO* fabricadas por Digium u otros proveedores, ya que para conectar el servidor a una línea externa no basta con un simple módem.

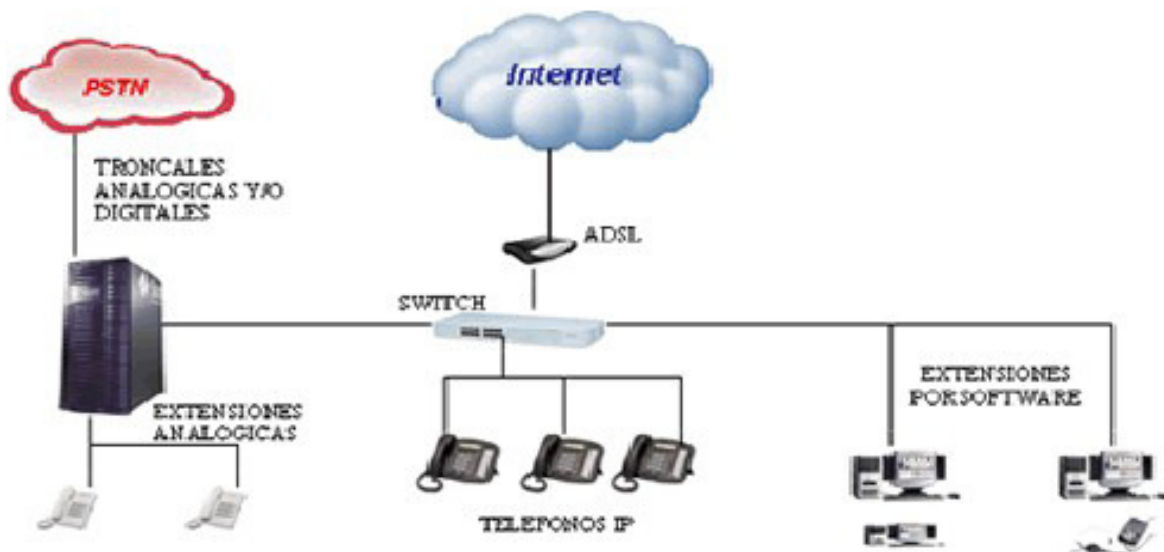
Quizá lo más interesante de Asterisk es que soporta muchos protocolos *VoIP* como pueden ser *SIP*, *H.323*, *IAX* (Protocolo de Intercambio en Asterisk, *Inter-Asterisk eXchange Protocol*) y *MGCP*. Asterisk puede interoperar con terminales *IP* actuando como un registrador y como *gateway* entre ambos. Lejos de poder competir con las compañías que comercializan soluciones de *VoIP* Hardware y Software de alta calidad como AlcatelLucent, Cisco, Avaya ó Nortel, Asterisk se empieza a adoptar en algunos entornos corporativos como solución de bajo coste junto con *SER* (Enrutador Expreso *SIP*, *Sip Express Router*).

^{iv} Retardo fijo debido a los tiempos que se necesita para los procesos *PAM*, cuantización, codificación, compresión y otros. El retardo es introducido tanto por el CODEC transmisor como por el CODEC receptor.

Algunas características de Asterisk:

- Creación de anexos *IP (SIP)*.
- Control, Monitoreo en tiempo real, *CDR* (Registro de Detalles de Llamadas, *Call Detail Records*).
- Acceso a base de datos. *AstDB*, *MySQL* para almacenar el *CDR* (*AstDB*, Base de Datos de Asterisk; *MySQL*, *Mi SQL*, *Structured Query Language*, Lenguaje de Peticiones Estructuradas).
- Interfaz Grafica.
- Respuesta Interactiva de Voz (*IVR*).
- Grabación de conversaciones.
- Correo de voz – email.

Diagramas (Topología – Arquitectura) (Figuras N°15-N°19)



FiguraN°15: Topología-Arquitectura de Sistema Telefónico Asterisk 1.

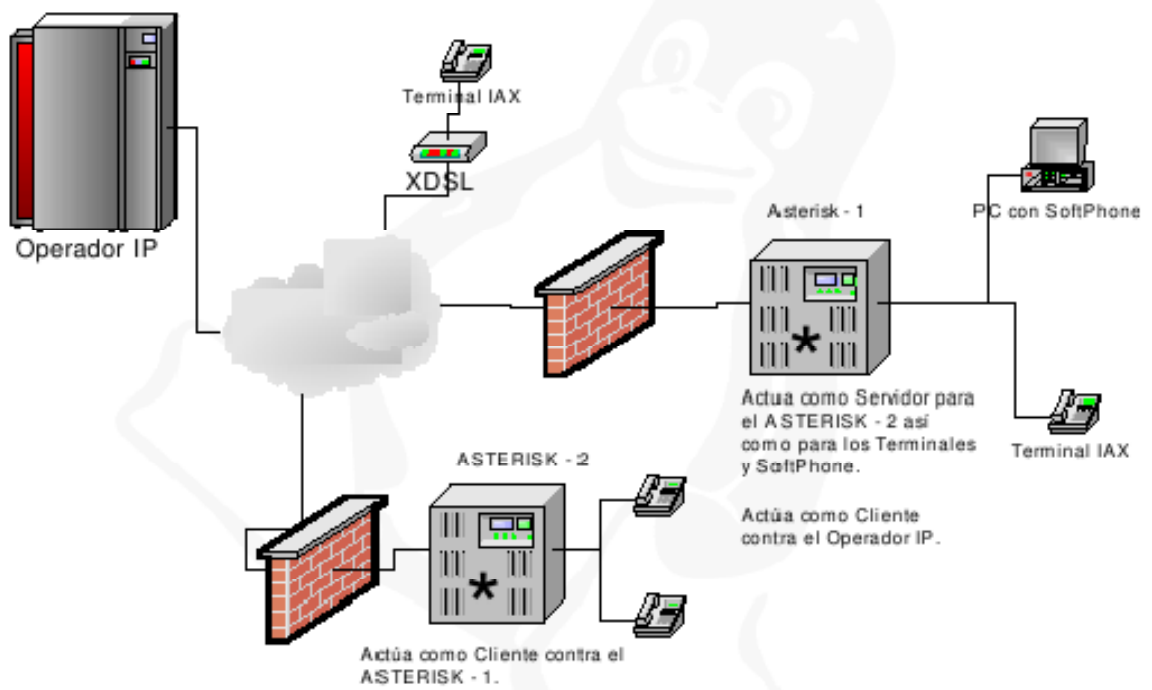


Figura N°16: Topología-Arquitectura de Sistema Telefónico Asterisk 2.

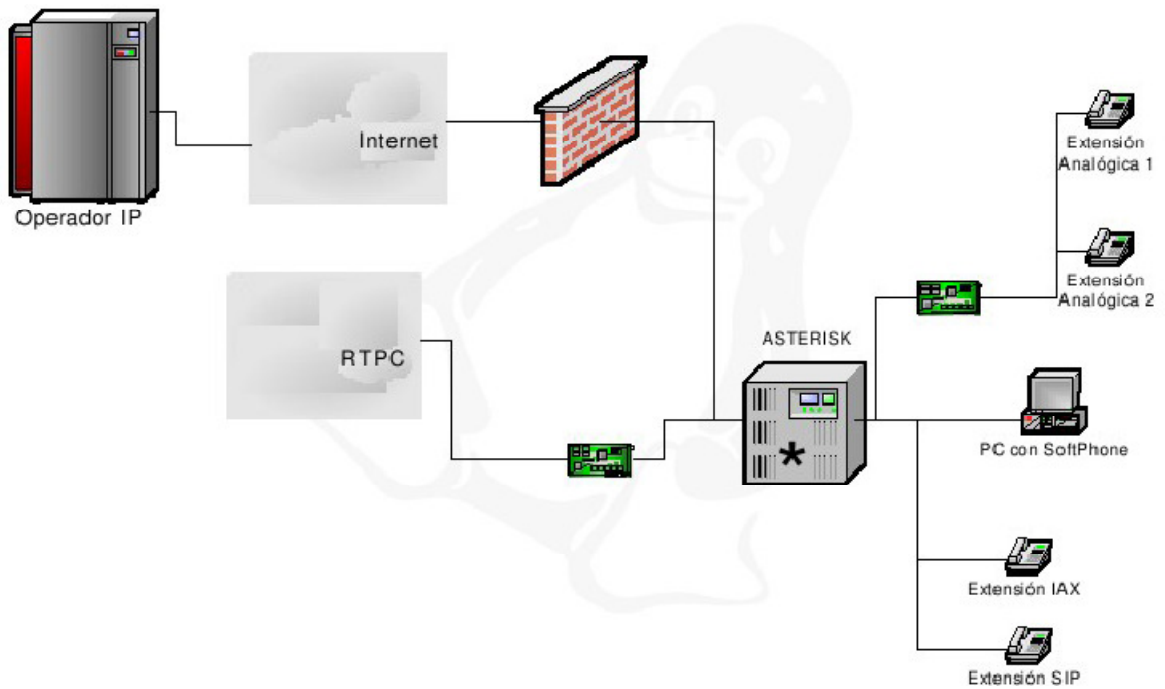


Figura N°17: Topología-Arquitectura de Sistema Telefónico Asterisk 3.

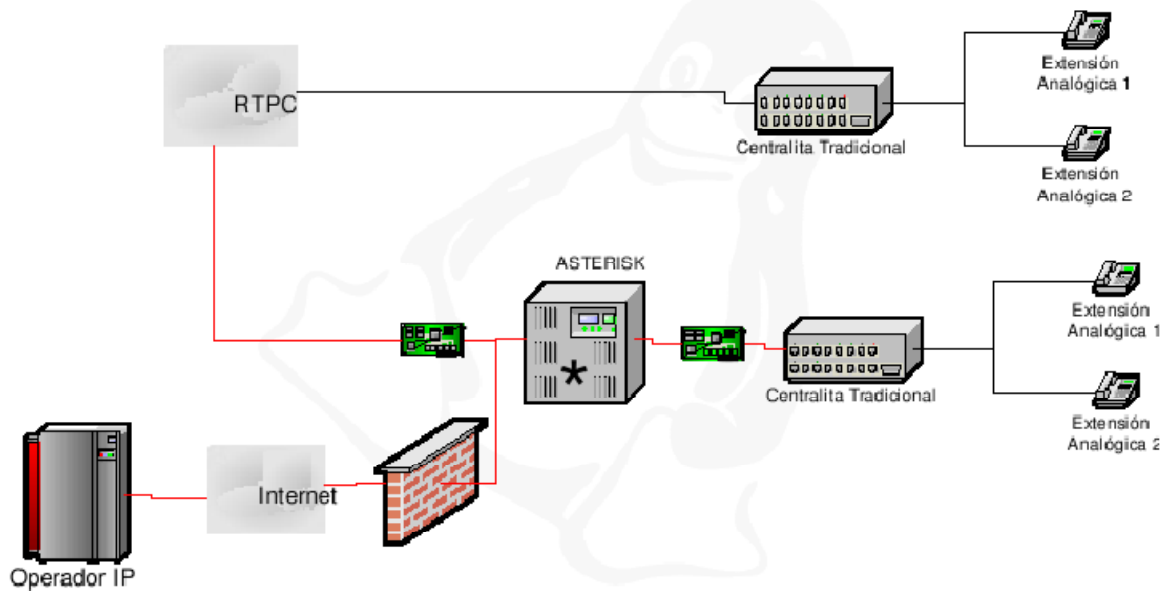


Figura N°18: Topología-Arquitectura de Sistema Telefónico Asterisk 4.

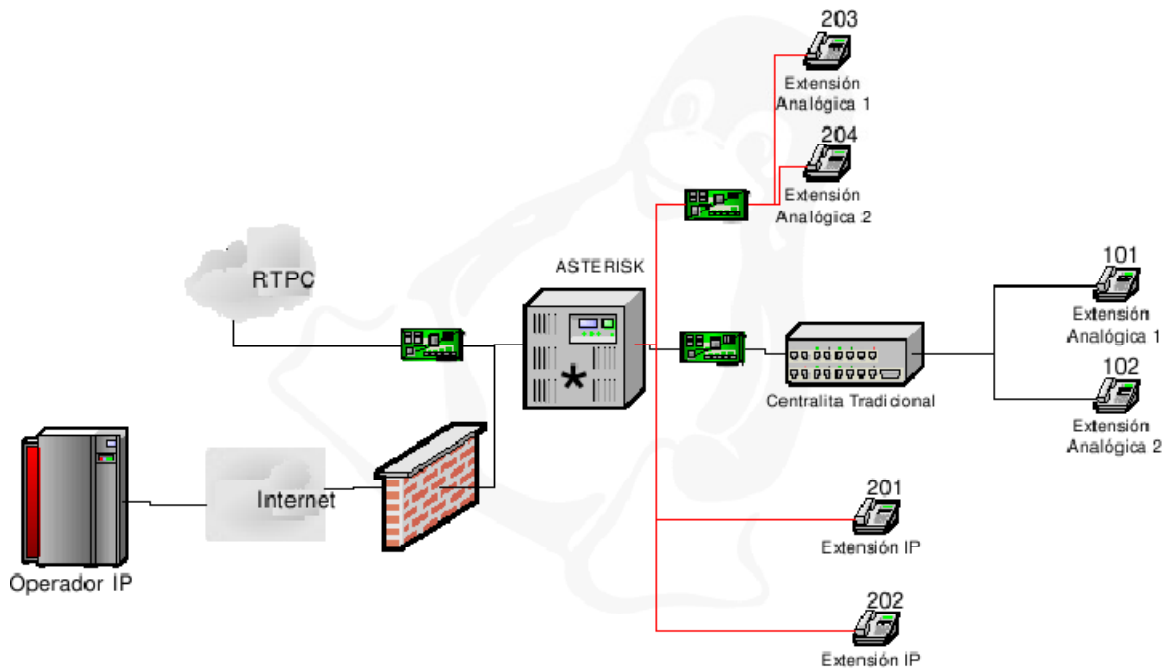


Figura N°19: Topología-Arquitectura de Sistema Telefónico Asterisk 5.

2.3.2 Cómo Funciona Asterisk.

2.3.2.1 Canales Físicos y Canales Lógicos.

Asterisk conmuta diferentes tipos de “canales” entre sí. Los canales pueden ser “canales físicos” o “canales lógicos”^v. Cuando un dispositivo desea iniciar una llamada, lo primero que hace es avisar su intención a la plataforma Asterisk.

2.3.2.2 Canales ZAP.

Los dispositivos típicos de telefonía tradicional se conectan a la plataforma mediante canales físicos, llamados canales ZAP. El hardware de la tarjeta de los canales ZAP, detecta cuando el dispositivo conectado a él tiene la intención de iniciar una llamada, y avisa esta condición a la plataforma. Las puertas para conexión a la telefonía tradicional pueden corresponder a puertas analógicas (*FXO*, *FXS*) o a puertas digitales (trama E1). Ejemplo de las primeras es la tarjeta Digium TDM22B, que tiene 2 puertas *FXS* y 2 puertas *FXO*. Por otra parte, ejemplo de trama E1 es la tarjeta TE120P, que puede configurarse como 1 trama E1, T1 ó J1.

Todas estas tarjetas se conectan en ranuras *PCI* (Interconector de Componentes Periféricos, *Peripheral Component Interconnect*) del servidor. El módulo de software que controla estas tarjetas se conoce como “Zapata”, y en él tienen especial importancia los archivos *zaptel.conf* y *zapata.conf* que veremos más adelante. Cada dispositivo ZAP está permanentemente conectado mediante un canal físico a la plataforma.

Canales Usados por Dispositivos que Emplean Protocolos de Telefonía *IP* (*SIP*, *IAX*, *H323*, *MGCP*, etc.)

Si el dispositivo (ya sea ubicado localmente en el servidor o remotamente en algún *switch* o *PC* de la red de datos) utiliza los protocolos de telefonía *IP* ya mencionados, avisa su intención de iniciar llamada enviando un mensaje a la plataforma Asterisk, por una puerta determinada^{vi}, la que se indica como parámetro general en *sip.conf*, *iax.conf*, etc. según corresponda. Para el caso *SIP*, normalmente los mensajes llegan por la puerta *UDP 5060*. De esta forma, la plataforma establece un canal lógico con el dispositivoⁱⁱⁱ.

Se dice que la plataforma siempre está “escuchando” las puertas ZAP para detectar aquellos dispositivos de telefonía tradicional que quieren iniciar una llamada. A su vez, para detectar si algún dispositivo de telefonía *IP* envía mensaje manifestando intención de llamar, Asterisk “escucha” la puerta *UDP 5060* (dispositivos *SIP*), la puerta *UDP 4569* (dispositivos *IAX*), etc.

^v La forma en que se conecta Asterisk durante la fase señalización con los dispositivos que usan protocolos de telefonía *IP*, es mediante mensajes (*SIP*, *IAX2*, *H323*, etc.) los que para una misma comunicación pueden seguir caminos físicos diferentes. En la fase conversación los dispositivos se comunican entre ellos mediante paquetes *RTP*, los que también pueden seguir diferentes caminos físicos. Por estas razones es necesario tener en cuenta que en este caso nos estamos refiriendo a canales Lógicos. Asterisk está siempre “escuchando” ports por los que deben llegar mensajes que piden iniciar una llamada.

^{vi} Por defecto para *SIP* es la puerta *UDP 5060* y para *IAX* la puerta *UDP 4569*

2.3.2.3 Tratamiento de las Llamadas Cuando Ellas se Inician en un Canal Físico ZAP.

Cuando Asterisk “escucha” intención de iniciar una llamada en un canal físico ZAP, consulta el archivo `zapata.conf` en el cual están programadas la configuraciones de cada uno de los canales. Esta configuración asigna diversos parámetros. Entre los parámetros está contexto (“*context*”) al que deberá acudir en el archivo `extensions.conf`. En contexto está toda la programación de enrutamiento, o bien el inicio de ella, la que puede continuar en otra parte de `extensions.conf`. En el archivo `zapata.conf` también están programados para cada canal parámetros como “grupo” al que pertenece (en `extensions.conf` la llamada se dirige a un grupo de canales y no a cada canal individualmente). En caso que el canal ZAP tenga conectado un terminal telefónico en su extremo (puerta *FXS*), también se declara la Identidad del llamante (“*callerid*”), el cual corresponde al número de anexo, que sirve para identificarlo en la red. La identificación del Llamante (“*callerid*”) permite validar si el anexo está autorizado para hacer llamadas y para que Asterisk lo acceda en caso de llamadas dirigidas a él.

Las puertas *FXS* y *FXO* se declaran en el archivo `zaptel.conf`, respectivamente como `fxoks` y `fxsks`, indicándose así el tipo de señalización con que la tarjeta se intercomunica con el dispositivo. A las puertas de los módulos *FXS* se conectan terminales telefónicos a los que la tarjeta deberá entregar señalización `fxoks`, mientras que a las puertas de los módulos *FXO* se conectan líneas telefónicas, por lo tanto la tarjeta debe comportarse como un terminal telefónico, es decir entregar al dispositivo señalización `fxsks`. Los canales conectados a las puertas E1 se declaran en `zaptel.conf` como `bchan` ó `dchan`

2.3.2.4 Tratamiento de las Llamadas Cuando se Inician en un Dispositivo que usa Protocolos de Telefonía IP.

Cuando Asterisk “escucha” en la puerta 5060, o mejor dicho, cuando recibe un mensaje que indica intención de iniciar una llamada proveniente de un dispositivo *SIP*^{vii}, consulta en el archivo `sip.conf`^{viii} la declaración que se hizo cuando se configuró ese dispositivo, y establece un canal lógico^{ix} para comunicarse con él. La información para identificar el dispositivo (*secret*, *username*, *host*) y leer en `sip.conf` su configuración, viene en el mensaje “escuchado” en la puerta 5060. Al momento de configurar el dispositivo en la central Asterisk, se fijaron diferentes parámetros, entre ellos el contexto (“*context*”), parámetro que indica a que parte del archivo

^{vii} Por simplicidad de la explicación nos estamos refiriendo a dispositivos SIP, sin embargo la misma explicación es válida para dispositivos IAX2, MGCP, MEGACO, H323, y cualquier otro que utilice protocolos de señalización de telefonía IP.

^{viii} Los mensajes enviados por dispositivos que están iniciando una llamada SIP “se escuchan” en la puerta 5060. Cuando llega uno de estos mensajes, lo primero que hace Asterisk es averiguar que configuración tiene programada el dispositivo (supongamos que es el anexo 1010) en el archivo `sip.conf`. Los parámetros particulares con que se configuró el dispositivo se encontrarán en `sip.conf` bajo [1010], entre estos parámetros se encuentran algunos que está enviando el dispositivo (*clave*, *anfitrión*, *nombre de usuario*; *secret*, *host*, *username*) y que ya sirvieron para determinar que justamente es bajo [1010] donde hay que leer las primeras instrucciones para atender al dispositivo.

Los mensajes protocolo IAX “se escuchan” en la puerta 4569. En este caso se sigue un procedimiento similar al explicado para dispositivos SIP, pero ahora se acude al archivo de configuraciones `iax.conf` y no a `sip.conf`.

^{ix} Establecer un canal lógico significa definir las direcciones IP de origen y de destino que deberán llevar los paquetes que fluirán por la red con la información que se intercambiará durante la comunicación. Normalmente en telefonía IP se establecen canales lógicos para las transmisión de la información de señalización (etapa de establecimiento de llamada, *call setup*) y otros canales lógicos diferentes para la transmisión de la información de la voz paquetizada en protocolo RTP

extensions.conf se debe ir a leer las instrucciones para el tratamiento de las llamadas que él genera.

El contexto (“*context*”), que se programa en el archivo extensions.conf, corresponde a una serie de instrucciones que se deben ejecutar, dependiendo de ciertas condiciones, como por ejemplo los dígitos que llegaron al Asterisk por el canal lógico y que quedaron almacenados como variable Exten. Entre los canales lógicos se distinguen las “extensiones” o anexos, los que normalmente funcionan con protocolo *SIP*. Estos canales tienen como terminal un teléfono *IP*, un *softphone* o un adaptador *ATA*^x. Otro tipo de canales lógicos son las troncales (“*trunk*”), que proporcionan conectividad con otros servidores Asterisk. Estos canales pueden funcionar con los diversos protocolos de señalización de telefonía *IP* anteriormente mencionados. Las instrucciones programadas en los contextos (“*context*”) corresponden al plan de discado, enrutamiento (*dial plan*) y están orientadas a establecer una conexión (*bind*) entre el canal lógico por el que llegó la llamada y el canal lógico asociado al dispositivo al que va dirigida la llamada. Cuando se tiene éxito, este *bind* deja establecida la llamada entre el llamante y el llamado a través de los respectivos canales lógicos.

2.4 Instalación de Linux y Asterisk.

Asterisk se puede instalar sobre diferentes distribuciones Linux, como por ejemplo sobre CentOS, Ubuntu, Debian, Fedora Core, SuSe, Slackware. En este documento resumiremos la forma de instalar Asterisk sobre *CentOS* (Sistema Operativo de Iniciativa Comunitaria, *Community ENTERprise Operating System*) sobre Ubuntu.

Ubuntu es una distribución muy amigable que al momento de ser instalada tiene la opción de hacerlo sin necesidad de eliminar *Windows* (Sistema operativo). Una ventaja de *CentOS* sobre Ubuntu es que trae incorporados todos los paquetes adicionales Linux (llamados “dependencias”) que se requieren para instalar Asterisk: paquetes kernel, librerías SSL (Capa de Sockets Seguros, *Secure Sockets Layer*) para encriptación y herramientas. En el punto 3.2 se detallan las dependencias requeridas.

También existen distribuciones que traen incorporado Asterisk y las dependencias que se requieren, como es el caso de *AsteriskNow* (Asterisk Ahora) En este caso la instalación es muy simple, pudiendo incluso hacerse de forma que el *PC* mantenga el *SO Windows* y las aplicaciones residentes. Sin embargo esta solución no permite disponer de todas las potencialidades que se obtienen con otras distribuciones.

De acuerdo a la información que entrega la página desde la que se baja *AsteriskNow* (<http://www.asterisk.org/asterisknow>), esta distribución es el camino más rápido para iniciar la construcción de soluciones de telefonía personalizada, con Asterisk. Simplemente hay que descargar un archivo .iso, grabarlo en un *CD* (*Compact Disc*, Disco Compacto), ponerlo en el *PC* (*Personal Computer*) en que alojaremos *AsteriskNow* (Asterisk Ahora) y en menos de 30 minutos tendremos un sistema Asterisk con todas las funcionalidades, listo para realizar aplicaciones de

^x *ATA* = Adaptador de Teléfono Análogo, *Analog Telephone Adaptor*, es decir adaptador o interface que permite por un lado la conexión *TCP/IP* con la red de datos y por el otro la conexión a uno o más teléfonos análogos tradicionales

telefonía personalizada. Quedarán instalados una completa distribución de Linux, Asterisk, el conjunto de drivers Dahdi, un administrador *GUI* (*Graphical User Interface*, Interfaz de Usuario Grafica) la base de datos *MySQL*, el servidor web Apache y una amplia variedad de herramientas de desarrollo y componentes. En la página se indica también que *AsteriskNow* fue construido para diseñadores de aplicaciones, integradores de sistemas, estudiantes, hackers y otros que quieran desarrollar *custom solutions* con Asterisk.

Sin embargo la solución más tradicional y más probada a la fecha es instalar Asterisk sobre *CentOS*. Normalmente cuando se instala Asterisk sobre *CentOS*, se dedica el *PC* exclusivamente a la función servidor Asterisk, quedando solamente este sistema operativo (*SO*) en el *PC*. Pero también es posible particionar el disco duro del *PC* y dejar residentes *CentOS* y otro *SO*, como por ejemplo Windows. Cuando el *PC* que destinaremos a Asterisk tiene ya funcionando el *SO* Windows y no queremos perderlo, es posible hacer una partición del disco duro sin necesidad de formatearlo, usando algún programa utilitario como por ejemplo *Magic Partition* (Partición Magica).

2.4.1 Instalación de Asterisk sobre *CentOS*.

En forma resumida, los pasos a seguir son los siguientes:

2.4.1.1 Instalación de *CentOS*

Se baja de Internet el programa *CentOS* a un *DVD* (*Digital Versatile Disc*, Disco Versatil Digital) ó a varios *CD*, desde el sitio www.centos.org

Por restricción del equipamiento en el laboratorio en que se trabajó, se debió optar por bajar el programa a siete *CD*. La versión elegida fue *CentOS 5.5*

2.4.1.2 Instalación de Asterisk sobre *CentOS*.

Una vez que se tiene el *SO CentOS* en el *PC* en que instalaremos Asterisk, hay por lo menos dos formas para realizar la instalación:

- **Alternativa 1:** Instalación Copiando los Archivos Comprimidos en Directorio `/usr/src`

La instalación se resume en los siguientes pasos:

- acceder a *CentOS*, introduciendo usuario = *root* y *password* = Asignada (raíz, contraseña) en la instalación al usuario *root*.

desde el terminal, utilizar los repositorios *yum* (*Yellowdog Updater Modified*, Actualizador Perro Amarillo Modificado).

- para comprobar las dependencias

```
# yum check-update
# yum -C list ncurses ncurses-devel openssl zlib zlib-devel curl
```

Deben estar todos los paquetes necesarios instalados.

- Copiar al directorio /usr/src los archivos comprimidos (se habían bajado previamente de <http://www.asterisk.org> y estaban grabados en un *pendrive*)

```
libpri-1.4.7.tar.gz
zaptel-1.4.12.1.tar.gz
asterisk-1.4.21.2.tar.gz
asterisk-addons-1.4.7.tar.gz
```

- Descomprimir el código fuente,

```
#cd /usr/src/
```

Comprobar versiones de los programas,

```
# ls -l
libpri-1.4.7.tar.gz
zaptel-1.4.12.1.tar.gz
asterisk-1.4.21.2.tar.gz
asterisk-addons-1.4.7.tar.gz
```

Realizar descompresión,

```
# tar -zxvf libpri-1.4.7.tar.gz
# tar -zxvf zaptel-1.4.12.1.tar.gz
# tar -zxvf asterisk-1.4.21.2.tar.gz
# tar -zxvf asterisk-addons-1.4.7.tar.gz
```

Ahora se tiene los nuevos directorios:

```
#cd /usr/src/
# ls -l
libpri-1.4.7
zaptel-1.4.12.1
asterisk-1.4.21.2
```

Compilar e instalar libpri

```
# cd libpri          (vamos al directorio correspondiente)
# make clean        (borra códigos compilados anteriormente)
# make              (compila)
# make install      (instala)
```

Compilar e instalar zaptel

```
# cd zaptel-1.4.12.1 (vamos al directorio correspondiente)
# ./configure        (comprueba la configuración y que se dispone de toda la
                    biblioteca requerida)
# make menuselect    (compila el programa menuselect y lo inicia)
# make               (compila zaptel)
# make install       (instala)
```

```
# make config          (instala un script de inicio)
```

Compilar e instalar Asterisk

```
# cd asterisk-1.4.21.2 (vamos al directorio correspondiente)
# ./configure          (comprueba la configuración y que se dispone de toda la
                       biblioteca requerida)
# make menuselect      (compila el programa menuselect y lo inicia)
# make                 (compila asterisk1)
# make install         (instala)
# make samples        (instala ejemplos)
# make config          (instala un script de inicio)
```

Reiniciar

en línea de comando linux:

```
# reboot
```

Alternativa 2: Instalación de Asterisk sobre *CentOS* con repositorios *Yum* (*Yellowdog Updater, Modified*, Actualizador Perro Amarillo, Modificado).

Esta es la alternativa que se usó durante el trabajo. Se instaló la versión Asterisk 1.6 con *Yum* desde www.asterisk.org, /downloads /install from yum repo (repositorio). Para agregar los repositorios Asterisk *yum* al *CentOS* y para instalar Asterisk 1.6 se siguió la guía “Instalando Asterisk con *Yum*” (“*Installing Asterisk With Yum*”) incluida en esa misma página web, la que se resume en los siguientes pasos:

- crear archivo de texto “centos-asterisk.repo”, con el texto indicado en la guía.
- poner el archivo en el directorio “/etc/yum.repos.d”.
- crear archivo de texto “centos-digium.repo”, con el texto indicado en la guía.
- poner el archivo en el directorio “/etc/yum.repos.d”.

Con lo realizado en los pasos anteriores, *CentOS* quedó actualizado para utilizar los repositorios Asterisk y Digium además de los repositorios básicos de *CentOS*, quedando en condiciones para que se instale Asterisk. La instalación se inicia con la siguiente línea de comando Linux:

```
[root@localhost~]# yum install asterisk16 asterisk16-configs asterisk16-voicemail
dahdi-linux dahdi dahdi-tools libpri
```

El sistema inicia la instalación e indica en línea las acciones que va realizando. Cuando va a requerir grabar gran cantidad de información en el disco duro, pide previamente autorización antes de proseguir.

Cuando la instalación termina lo indica con:

```
Installed: asterisk16.i386 0:1.6.0.15-1_centos5  
Dependency Installed: asterisk16-core.i386 0:1.6.0.15-1_centos5 ...  
Complete!
```

Y aparece la línea de comando

```
[root@localhost yum.repos.d]#
```

Para iniciar Asterisk se escribe el comando:

```
[root@localhost yum.repos.d]# asterisk -vvvghi
```

2.4.1.3 Instalación de Asterisk sobre Ubuntu.

En forma resumida, los pasos son los siguientes:

Instalación de Ubuntu

El programa se obtiene en el sitio www.ubuntu.com, opción “download ubuntu” y se graba en un *CD*. En www.ubuntu.com/support se encontrará información para la instalación.

Para instalar Ubuntu basta poner el *CD* en el *PC* en que se instalará Ubuntu y hacer doble click en archivo “wubi.ex”. Durante la instalación es posible elegir la opción de dejar residentes en el *PC* ambos *SO*: Ubuntu y Windows. Habrá que ingresar el nombre de usuario y la contraseña que elijamos, y cuando la instalación termine, reiniciar el *PC*. La instalación finaliza solamente después que usemos por primera vez Ubuntu.

Instalación de Asterisk sobre Ubuntu

Cuando se instala Asterisk sobre *CentOS*, como se indicó en la introducción, *CentOS* ya ha instalado todos los paquetes adicionales Linux que requiere la instalación de Asterisk: paquetes kernel (linux-headers-x.x.x-x-generic, linux-image-x.x.x-x-generic, linux-source-x.x.x), librerías *SSL* para encriptación (openssl, libssl-dev, ssh) y herramientas (gcc, g++, make, automake, autoconf, build-essential, bison, flex, libtool, libncurses5, libncurses5-dev, libgsm, libgsm-dev, libnewt-dev, libnewt-pic, curl, libcurl3, tclcurl, libwww-dev, mysql-common, mysql-client, libmysqlclient16, libmysqlclient16-dev, sox, libsox-fmt-all, madplay, libxm12, libxm12-dev, doxygen).

La instalación de *CentOS* y de Asterisk se encuentra en la guía de laboratorio N°1, donde se revisa el tema en mayor detalle.

La configuración y programación de Asterisk se encuentra en las guías de laboratorio N° 3 y 4, donde se revisa el tema en mayor detalle.

CAPÍTULO III:

IMPLEMENTACIÓN.

Los objetivos del presente trabajo se resumen a continuación:

- Implementar una maqueta telefónica *IP* para realizar prácticas de laboratorio de acuerdo a las necesidades del curso EL 629;
- Diseñar y evaluar prácticas de laboratorio para dicho curso;
- Redactar el material docente o Guías para los Laboratorios del curso.

Revisaremos en este capítulo la forma como se cumplieron todos los objetivos antes descritos. Los detalles de cada una de las actividades se explican in extenso en las Guías de laboratorio que se incluyen en el Anexo A: “Guías de Laboratorio Telefonía *IP* diseñadas para el curso EL629”.

3.1 Implementación de los Servidores y de los *Softphones*.

El diseño del sistema a implementar, se definió conjuntamente con el profesor guía, teniendo en cuenta las materias del curso EL 629 que se quieren reforzar mediante el trabajo práctico de los alumnos en laboratorio y la disponibilidad de equipamiento en el DIE para implementar el diseño. El equipamiento del laboratorio utilizado para implementar la maqueta es el siguiente:

➤ 6 *PC*'s (Computadores Personales):

- Marca: Dell.
- Modelo: DHS, Optiplex GX240.
- Sistema Operativo: Windows XP Professional Service Pack 3.
- Procesador: Intel Pentium 4 CPU 1.50 GHz.
- Memoria RAM: 256 MB.
- Disco Duro: Maxtor 5T020H2, 18,61 GB.
- Interfase de Red: Controlador Fast Ethernet Integrado, 3 Com Corporation 3C 905C-Tx/Tx-M [Tornado]
- Tipo de Sistema: x86.
- Dirección IP:
 - Servidor Maestro: 10.10.100.20.
 - Servidor Esclavo: 10.10.100.30.
 - PC1: 10.10.100.50.
 - PC2: 10.10.100.80.
 - PC3: 10.10.100.10.
 - PC4: 10.10.100.110.
 - PC Profesor: 10.10.100.100.

- Dirección MAC:
 - Servidor Maestro:
 - Servidor Esclavo: 00:06:5B:27:9C:DC
 - PC1:
 - PC Profesor:

➤ ATA (1):

- Marca: Leadtek.
- Modelo: BVA 8053R.
- Dirección MAC: 00:30:04:06:C1:7A

➤ Teléfono IP (1):

- Marca: CISCO.
- Modelo: IP Phone 7960 Series.
- Dirección MAC: 000D:BD2C:DBD5

➤ Audífonos con Micrófonos (4):

- 3 Marca: PCTronix.
- 1 Marca: Genius.

➤ Teléfono Análogo:

- Marca: CTC.
- Modelo: Alerce.

➤ LAN:

- Se Detalla en la Figura N°20.

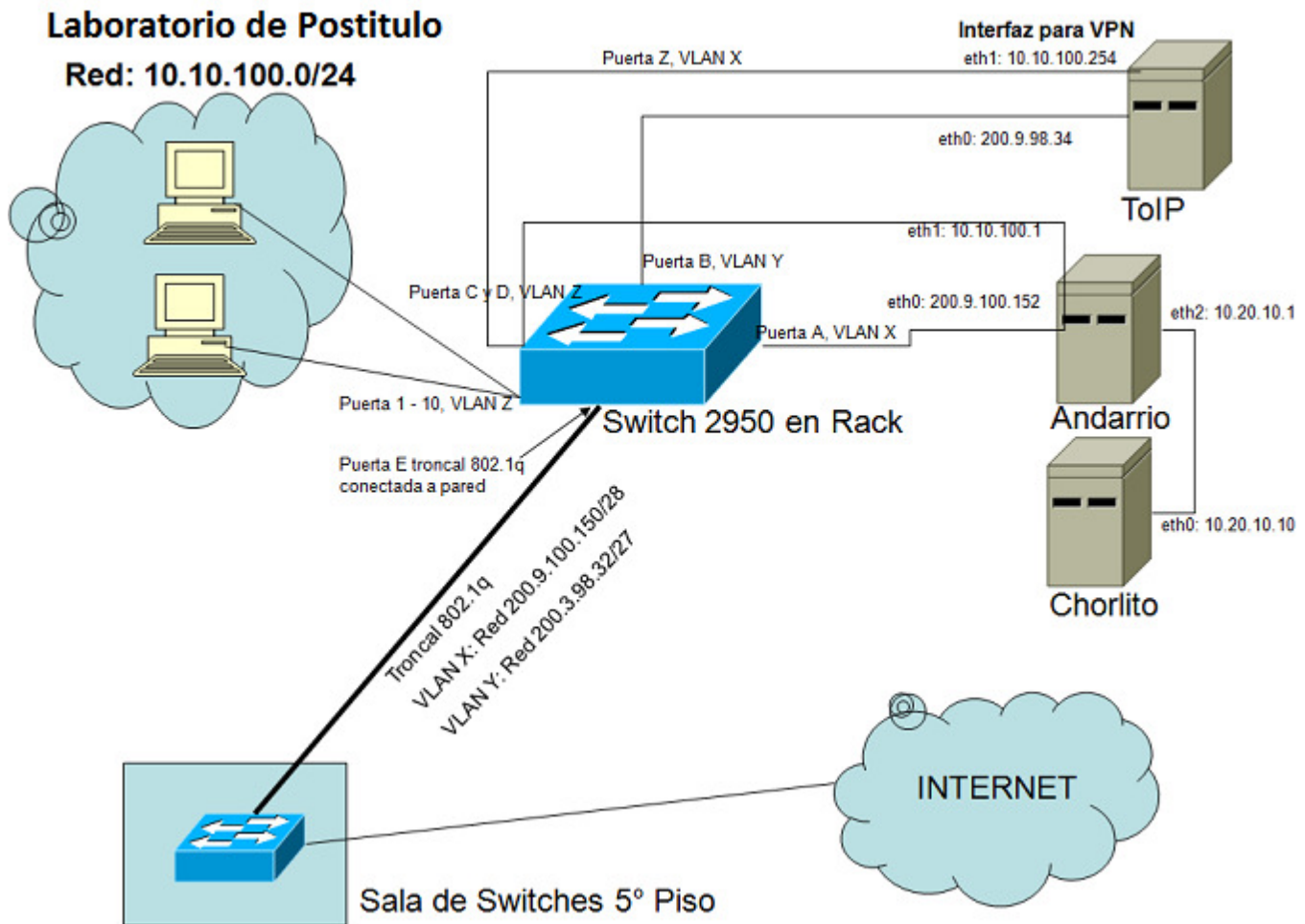


Figura N°20: Mapa de conexiones de LAN del Laboratorio de Postítulo del DIE.

➤ Conexiones:

- Topología: Estrella.

➤ Switch:

- Marca: CISCO Systems.
- Modelo: Catalyst 2900 Series XL.
- Servicio: Provee interconexión entre los dispositivos del laboratorio, y de estos con el exterior, bajo la supervisión y control del sistema de administración de red.

➤ Servidores (Sistema de administración de red):

- Nombre: Andarrio.

- Software:
 - *DHCPd* de Linux, software *DHCP* (Protocolo de Configuración de Huéspedes Dinámicos, *Dynamic Host Configuration Protocol*).
 - *IPTables* (Tablas IP) de Linux, Programa de Cortafuegos (*Software Firewall*).
 - *OpenVPN* de Linux, *software VPN* (Open, Abierto. VPN, Red Privada Virtual, *Virtual Private Network*).
- Servicios: *NAT* (Traducción de Direcciones de Red, *Network Address Translation*) listas de acceso, servidor de redireccionamiento IP (*Proxy IP*), enmascaramiento de red interna, servidor de conexiones remotas.
- Nombre: *ToIP*.
 - Software: *OpenVPN*, *software VPN*.
 - Servicios: *VPN*, servicio de conexiones remotas.
- Nombre: Chorlito.
 - Estado: Fuera de servicio.

La figura N°21 resume la arquitectura de la maqueta diseñada e implementada en el laboratorio de Postítulo del DIE:

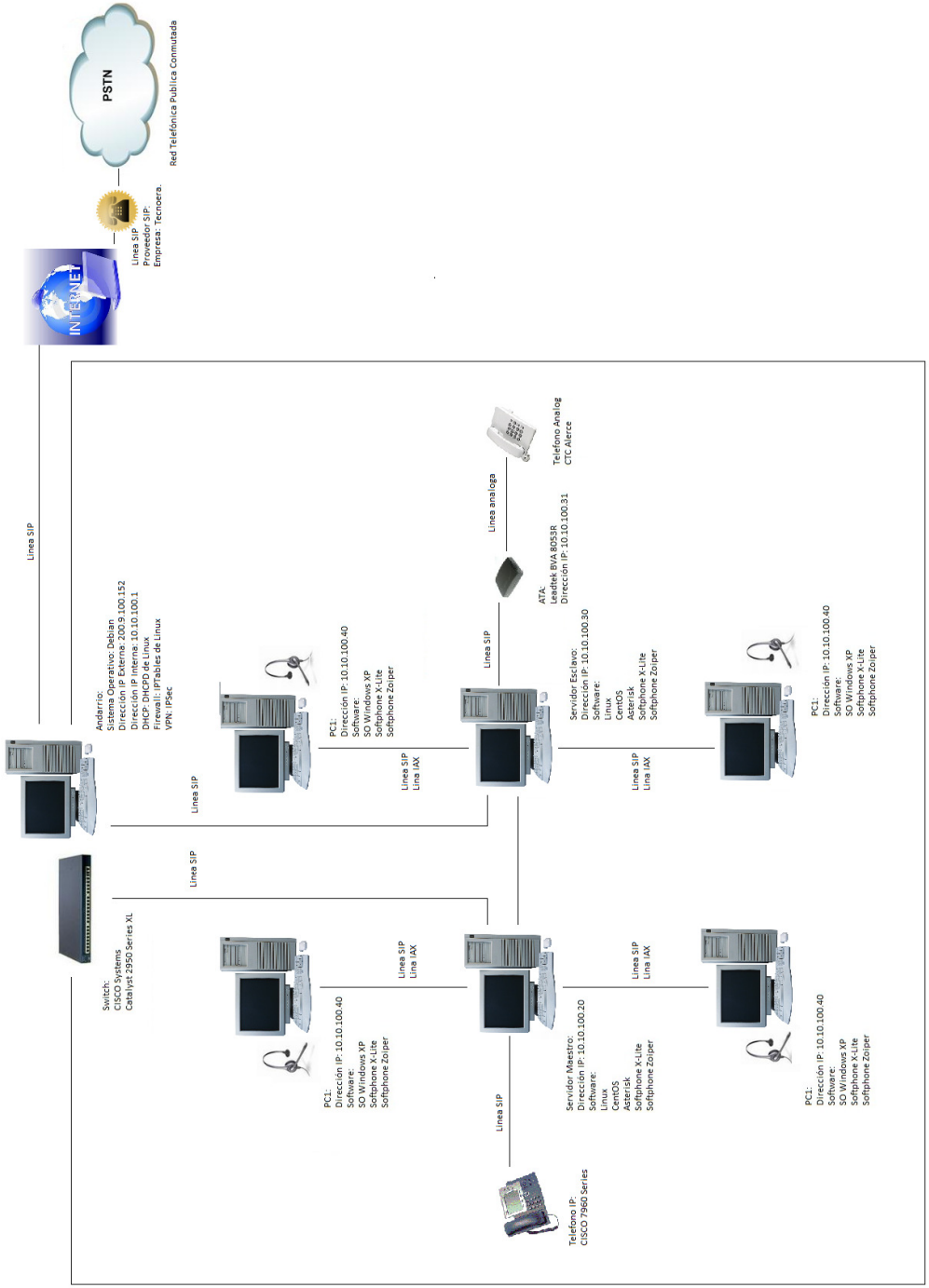


Figura N°21: "Arquitectura de Maqueta de Laboratorio para el Curso EL629".

Los pasos seguidos para implementar la maqueta se resumen en los siguientes:

- Instalación del sistema operativo *CentOS* en dos *PC*'s del laboratorio para constituir los servidores Maestro y Esclavo.
- Instalación de Asterisk sobre *CentOS* en ambos servidores.
- Instalación de *Softphones X Lite* y Zoiper en otros 4 *PC*'s del laboratorio (*PC1* a *PC4*).

3.2 Configuración del Sistema de Telefonía IP.

Se configuró y probó el sistema de telefonía *IP* siguiendo los pasos que se indican a continuación. La forma en que se realizaron las configuraciones, se explican en forma detallada en las Guía de Laboratorio N°1, N°2 y N°3.

- Configuración de Asterisk servidor Maestro con *softphones X Lite* y Zoiper de *PC1* y *PC2*.
- Configuración de Asterisk servidor Esclavo con *softphones X Lite* y Zoiper de *PC3* y *PC4*.
- Configuración de los *softphones X Lite* y Zoiper de *PC1* a *PC4* con los respectivos servidores Asterisk.
- Configuración de la interconexión de los servidores Asterisk entre sí.
- Configuración de un teléfono *IP* conectado al servidor Asterisk Maestro.
- Configuración de *ATA* con un teléfono analógico, conectado al servidor Asterisk Esclavo.

3.3 Configuración de los Servicios en el Sistema de Telefonía IP.

Se configuraron los servicios que se indican a continuación. La forma en que se realizaron las configuraciones, se explica en forma detallada en la Guía del Laboratorio N°3 .

- Configuración de servicio Correo de Voz (*Voice mail*) en ambos servidores.
- Configuración de servicio *IVR* en ambos servidores.
- Configuración de servicio operadora automática (*IVR +Voice mail*) en ambos servidores.
- Configuración de servicio salas de conferencia (*Meetme*, Encuentra me) en ambos servidores.

3.4 Habilitación de Funciones de Asterisk.

Se habilitaron las facilidades que se indican a continuación. La forma en que se realizaron las habilitaciones, se explica en forma detallada en la Guía del Laboratorio N°4.

- Habilitación de límite de tiempo a las llamadas.
- Habilitación de grabación del audio de las llamadas con Mixmonitor.

3.5 Diseño de Prácticas de Laboratorio Telefonía *IP* para los Alumnos del Curso EL629.

Se determinó que durante las prácticas de laboratorio los alumnos deberán implementar el software, configurar y practicar comandos en los temas antes descritos, es decir en: sistema operativo *CentOS*, Asterisk, *softphones*, *ATA* y teléfono *IP*.

De esta forma se diseñaron cuatro laboratorios que cubren los siguientes temas:

3.5.1 Objetivos y Temas a Desarrollar en el Laboratorio N°1.

- Instalar el sistema operativo *CentOS* en dos *PC*'s del laboratorio.
- Practicar y familiarizarse con el terminal de Linux y sus comandos básicos.
- Conocer el editor Vim.
- Practicar y familiarizarse con el terminal gráfico Gnome.
- Instalar Asterisk sobre *CentOS*.
- Instalar *Softphones* de X Lite y Zoiper en 4 *PC*'s del laboratorio.

3.5.2 Objetivos y Temas a Desarrollar en el Laboratorio N°2.

- Configurar Asterisk con los *softphones* X Lite y Zoiper.
- Configurar los *softphones* X Lite y Zoiper.
- Realizar prácticas básicas con comandos Asterisk.
- Configurar la interconexión de dos servidores Asterisk.

3.5.3 Objetivos y Temas a Desarrollar en el Laboratorio N°3.

- Configurar y dejar funcionando un teléfono *IP* conectado a un servidor Asterisk.
- Configurar y dejar funcionando una *ATA* con un teléfono analógico, conectado a un servidor Asterisk.
- Configurar y dejar funcionando servicio *Voice mail*.
- Configurar y dejar funcionando servicio *IVR*.
- Configurar y dejar funcionando servicio operadora automática (*IVR +Voice mail*).
- Configurar y dejar funcionando servicio *Meetme*.

3.5.4 Objetivos y Temas a Desarrollar en el Laboratorio N°4.

- Habilitar facilidad para fijar límite de tiempo a las llamadas.
- Habilitar facilidad para grabar audio de llamadas con Mixmonitor

3.6 Realización Detallada de las Experiencias Diseñadas.

Todas las experiencias diseñadas según lo indicado en el punto 3.5 anterior, se realizaron detalladamente y probaron varias veces en la maqueta de telefonía *IP* implementada en el laboratorio de postítulo del DIE. Esta actividad permitió redactar las cuatro Guías incluidas en el “anexo A”.

3.7 Guías de Laboratorio.

Durante el trabajo práctico se ganó mucha experiencia que facilitó concretar los dos productos finales de esta memoria: la maqueta implementada y la redacción de las cuatro Guías para los Laboratorios de telefonía *IP* que deberán realizar los alumnos como parte del curso EL629, de acuerdo a la nueva programación de este curso.

Las Guías explican detalladamente los objetivos de cada laboratorio, las actividades que deben realizar y los aspectos que deben practicar los alumnos durante el trabajo de laboratorio.

En el Anexo A: “Guías de Laboratorio Telefonía *IP* diseñadas para el curso EL629” se incluyen:

Guía de Laboratorio N°1: “Instalación de software: SO *CentOS*, Asterisk sobre *CentOS*, X-Lite y Zoiper”

Guía de Laboratorio N°2: “Configuración I: Asterisk con *softphones* X-Lite y Zoiper y Asterisk con otro Asterisk”

Guía de Laboratorio N°3: “Configuración II: Teléfono *IP*, *ATA*, Servicios *Voice-mail* e *IVR*”

Guía de Laboratorio N°4: “Funcionalidades: Límite de tiempo y grabación de llamadas”

CAPÍTULO IV:

DISCUSIÓN DE RESULTADOS.

4.1 Cumplimiento de los Requerimientos Iniciales.

En el capítulo “Implementación” se han detallado las actividades realizadas durante el trabajo de memoria. En este capítulo se analizará y discutirá el cumplimiento o incumplimiento de los objetivos planteados.

4.2 Logros.

Los principales logros del trabajo realizado se pueden resumir en los siguientes:

- Este trabajo de título entregó un producto final completo compuesto por una maqueta de Telefonía *IP* Asterisk y cuatro guías de laboratorio.
- Se adquirió mucho conocimiento, tanto teórico como práctico de la telefonía *IP* en general, como de Asterisk.
- Se diseñaron experiencias de laboratorio, que ponen en práctica el conocimiento de telefonía *IP* estudiado en clases.
- Durante la preparación teórica, se investigó e indagó en la web diferentes fuentes de información relacionadas con el tema. Entre la información adquirida se encontró páginas como la página oficial de Asterisk, la página Asterisk Guru, libros sobre el tema, foros, listados de correo, y otros recursos, mucha información importante para el desarrollo del trabajo de memoria, toda ésta se estudió y revisó, y fue muy útil para el desarrollo de la memoria y lo puede ser en el futuro para seguir desarrollando la maqueta, diseñar nuevas experiencias o redactar nuevas guías. Esta información está indicada en la bibliografía.
- Al desarrollar y probar las experiencias se comprobó y verificó que todas están bien elaboradas, y que no presentarán ninguna dificultad en su desarrollo.
- La maqueta cumple con los requerimientos planteados al comienzo.
- El trabajo cumple con las restricciones de equipamiento y presupuesto existentes.
- Se redactaron cuatro guías basadas en las experiencias evaluadas, que utilizarán los alumnos para realizar las sesiones de laboratorio de telefonía *IP*, que se adjuntan en el Anexo A. Cada guía corresponde a una sesión de laboratorio y las cuatro en conjunto cubren las materias de telefonía *IP* en que se tiene interés haya trabajo práctico por parte de los alumnos.

- La calidad de las llamadas, a percepción directa del oído es aceptable. Se prevé para la próxima etapa del trabajo evaluar mediante algún parámetro más exacto la calidad del audio.

4.3 Análisis Crítico y Posibles Mejoras.

- Si bien los objetivos propuestos se lograron, se hubiera deseado investigar e incorporar otras potencialidades del sistema Asterisk a la maqueta. No obstante la maqueta podrá seguir siendo enriquecida posteriormente.
- Será necesario verificar si es factible realizar en las dos horas de cada sesión de laboratorio las prácticas detalladas en las guías adjuntas en el Anexo A, puesto que es probable que sean muy extensas. Las prácticas de laboratorio de telefonía *IP* diseñadas tratan más bien de las materias más básicas del curso. A futuro podría ser interesante diseñar otras prácticas más avanzadas. Las prácticas serán desarrolladas en una próxima etapa, para evaluar cuanto tiempo demora desarrollarlas, y hacer los arreglos necesarios.
- Este trabajo puede pasar a formar parte de un trabajo a largo plazo que trate el tema de la telefonía *IP* y Asterisk en forma más profunda, un trabajo orientado a la telefonía *IP* y no solamente a prácticas de laboratorio, que permita hacer desarrollos e introducir las actualizaciones que va teniendo Asterisk.
- En relación a las experiencias de laboratorio, probablemente será necesario optimizar el uso del tiempo, ya que como se dijo anteriormente, puede que los alumnos no alcancen a desarrollar en cada sesión todas las actividades descritas en la respectiva guía. Es así como puede ser necesario que algunas de las actividades, sobre todo las más rutinarias, se encuentren ya realizadas cuando los alumnos inicien la sesión.
- Con el objetivo de analizar esta situación y mejorar las experiencias de laboratorio, el autor cumplirá la función de ayudante de laboratorio del curso EL629 el semestre que viene. De acuerdo a lo indicado anteriormente, se propone continuar la investigación de servicios que se pueden implementar, formas de programar los servicios, funcionamientos en diferentes configuraciones y topologías de redes telefónicas, tipos de dispositivos que soporta el sistema, limitaciones, etc.

CAPÍTULO V:

CONCLUSIONES.

- Este trabajo ha dejado como resultado un conjunto de herramientas que servirán para desarrollar prácticas de laboratorio en el curso EL629, Sistemas de Conmutación Telefónica. El conjunto de herramientas creadas está constituido por un sistema telefónico *IP Asterisk* (maqueta) y guías de laboratorio para realizar en dicho sistema de telefonía *IP Asterisk* las experiencias de laboratorio del curso EL629.
- Las guías de laboratorio desarrolladas como parte del trabajo, permitirán a los alumnos entender claramente de que se trata cada experiencia, cómo realizarlas y los objetivos que con ella se persiguen.
- La maqueta creada como parte del trabajo de memoria está constituida por dos centrales (servidores Asterisk), cuatro *softphones*, una *ATA* y un teléfono *IP*. Sobre la maqueta se desarrollaron las experiencias de laboratorio diseñadas para el curso EL629.
- En la maqueta se pueden programar servicios como Buzón de Voz (*Voicemail*), Respuesta de Voz Interactiva (*IVR*), Salas de Conferencia (*MeetMe*) y diversas funciones y facilidades como manejo de llamadas, grabación de llamadas, llamadas con duración de tiempo limitado y autenticación. Estos servicios y funcionalidades serán programados por los alumnos durante las sesiones de laboratorio siguiendo las indicaciones de las guías de laboratorio. Se ha obtenido como resultado cuatro laboratorios realizables en cuatro sesiones, durante el semestre, las cuales se describen a continuación.
- Aunque Asterisk hoy en día es un sistema de telefonía *IP* con grandes ventajas sobre otros similares, se debe considerar que en el futuro probablemente aparecerán otros sistemas que lo superarán. Las sesiones de laboratorio que realizarán los alumnos en la maqueta Asterisk, les será una muy buena experiencia y capacitación para desenvolverse en diferentes tipos de sistemas de telefonía *IP*.
- El laboratorio N°1 trata principalmente sobre la instalación de todo el software: CentOS y Linux en los PC's que constituirán las centrales y los *softphones* Zoiper y X Lite en los PC's que construirán los terminales. Además se desarrollan prácticas para familiarizarse con el sistema y sus programas: La terminal Linux, el editor de texto Vim y el terminal grafico Gnome.
- En el laboratorio N°2 se lleva a cabo la configuración básica del sistema, lo que abarca: La configuración de Asterisk para registrar los *softphones* X Lite y Zoiper, configurar los *softphones* mismos, y configurar la interconexión de las centrales. Además se llevan a cabo prácticas básicas con los comandos de Asterisk para conocer el funcionamiento de Asterisk, probar el sistema recién configurado y hacer algunas variaciones en la configuración de este, a fin de conocer en mayor detalle sus facilidades.

- En el laboratorio N°3 se deja operativa la maqueta, implementando los principales servicios de Asterisk, como: Buzón de voz (Voice Mail), Operadora de Voz Interactiva (IVR), Operadora automática, salas de Conferencias (Meetme). Además se instalan y configuran un teléfono IP y una ATA.
- En el laboratorio N°4 se configura en el sistema algunas funcionalidades como grabación de llamadas y llamadas con duración de tiempo limitada. Además se configura una línea de conexión a la red de telefonía tradicional.
- La maqueta permitirá a los alumnos explorar, durante las experiencias de laboratorio, uno de los *softwares* de código abierto de telefonía más conocidos en el mundo, adquiriendo importante experiencia en esta tecnología de código abierto que es cada vez más utilizada, desarrollada y difundida.
- Como se desprende de lo anterior, las herramientas producto del trabajo de memoria realizado, serán un valioso aporte para la formación de los alumnos del curso EL629. Como se indicó en el capítulo Discusión de Resultados, pensamos que este trabajo puede seguir siendo desarrollado, agregando nuevas aplicaciones a la maqueta, diseñando nuevas experiencias de laboratorio y redactando las correspondientes nuevas guías.
- Otra proyección del trabajo es que él puede constituir un punto de partida para continuar investigando el sistema Asterisk y realizar estudios en que converjan tecnologías, como por ejemplo conectar el sistema telefónico Asterisk a la red celular, o conectar a los servidores Asterisk terminales vía bluetooth, etc.
- Como conclusión general podemos decir que los productos finales del trabajo: la maqueta de telefonía *IP* y las guías para los laboratorios del curso Sistemas de Conmutación Telefónica, cumplen plenamente con los objetivos impuestos al inicio de él.
- Adicionalmente, vemos que lo realizado tiene las proyecciones arriba indicadas: puede ser el punto de inicio para realizar otras experiencias de laboratorio y para continuar la investigación de la telefonía *IP* Asterisk.

REFERENCIAS BIBLIOGRAFICAS.

Aparicio Córdova, Tomás Andrés. 2004. Diseño e implementación de un curso práctico avanzado de telefonía IP. Para optar al grado de Ingeniero Civil Electricista. Santiago, Chile. Universidad de Chile.

AsteriskGuru.com. <<http://www.asteriskguru.com/>>. [25 de Mayo de 2011].

Asterisk Guru. [200-]. Cisco 7960 IP Phone - SIP firmware versión. <http://www.asteriskguru.com/tutorials/cisco_7960_ip_phone_configuration.html>. [25 de Mayo de 2011].

ATA BVA8053R. *Installation guide* (Guía de Instalación).

Bryant, Russell. 2011. Asterisk 1.8 Documentation. Estados Unidos. Open Source Asterisk Community. Consulta en línea: <https://wiki.asterisk.org/wiki/display/AST/Home>

Castillo B., Luis. 2011. Documentos Resumen de Asterisk.

Castillo B., Luis. 2011. Apuntes del curso EL629, Sistemas de Conmutación Telefónica. <https://www.u-cursos.cl/ingenieria/2009/1/EL629/1/material_docente/>. [25 de Mayo de 2011].

Cisco System, Inc. 2002. Cisco IP Phone 7960 and 7940 Series At a Glance (Una Mirada al Teléfono IP CISCO Serie 7960 y 7940. Estados Unidos. <http://www.cisco.com/application/pdf/en/us/guest/products/ps379/c1614/ccmigration_09186a00800a85d5.pdf>. [25 de Mayo de 2011].

Cisco System, Inc. *Cisco IP Phone 7960 and 7940 Series At a Glance* (Una Mirada al Teléfono IP CISCO Serie 7960 y 7940. <http://www.cisco.com/application/pdf/en/us/guest/products/ps379/c1614/ccmigration_09186a00800a85d5.pdf>. [25 de Mayo de 2011].

Cisco Systems, Inc. [200-]. Catalyst 2900 User Guide. <http://www.cisco.com/en/US/products/hw/switches/ps606/products_installation_and_configuration_guide_chapter09186a00802ec0d8.html>. [25 de mayo de 2011].

Cisco systems, Inc. [200-]. Cisco IP Phone 7960/7940 User Guide for SIP. <http://www.cisco.com/en/US/docs/voice_ip_comm/cuipph/7960g_7940g/sip/english/user/guide/user.html>. [25 de Mayo de 2011].

CounterPath Corporation. 2010. *X-Lite 4.0 for Windows User Guide* (Guía de Usuarios para X-Lite 4.0 para Windows). Canadá. <http://www.counterpath.com/assets/files/191/X-Lite_Windows_4.0_User_Guide_R1.pdf>. [25 de Mayo 2011].

Dedoimedo. [200-]. GRUB bootloader - Full tutorial. <<http://www.dedoimedo.com/computers/grub.html#mozTocId233239>>. [25 de Mayo de 2011].

Digium, Inc. [2010]. Asterisk- The Open Source Telephony Project. <<http://www.asterisk.org/>>. [25 de Mayo de 2011].

FonoSIP.com. [2010]. FonoSIP Your Internet Phone. <<http://www.fonosip.com/>>. [25 de Mayo de 2011].

Hernández Torres, Jorge Rafael. 2005. Construcción de un call agent económico con fines docentes. Para optar al grado de Ingeniero Civil Electricista. Santiago, Chile. Universidad de Chile.

Huepe Rebolledo, Cristian Andrés. 2010. Diseño y validación de las componentes de un proveedor de servicios de academia virtual económico y con fines docentes. Para optar al grado de Ingeniero Civil Electricista. Santiago, Chile. Universidad de Chile.

Jesam Gaete, Alvaro Manuel. 2009. Implementación de una plataforma sobre IP utilizando Freeswitch como Testbad para tecnología por voz. Para optar al grado de Ingeniero Civil Electricista. Santiago, Chile. Universidad de Chile.

Meggelen, Jim Van “et all”. 2007. *ASTERISK: THE FUTURE of Telephony* (ASTERISK: EL FUTURO de la Telefonía). 2^a, ed. *United States* (Estados Unidos). O’rreily Media, Inc. Descargable de: <http://cdn.oreilly.com/books/9780596510480.pdf>

Rottmann Chávez, Kurt Rainer. 2010. Diseño e implementación de un laboratorio de IPTV, medición y gestión. Para optar al grado de Ingeniero Civil Electricista. Santiago, Chile. Universidad de Chile.

Tchernitchin Lapin, Nikolai. 2007. Diseño e implementación de una IP-Contact Center distribuida económica y con fines docentes. Para optar al grado de Ingeniero Civil Electricista. Santiago, Chile. Universidad de Chile.

VOIP-Info.org. [2003-2010]. <<http://www.voip-info.org/wiki/index.php>>. [25 de Mayo de 2011].

Voip-Info.org. [200-]. Asterisk. <<http://www.voip-info.org/wiki/index.php?page=Asterisk>>. [25 de Mayo de 2011].

Voip-Info.org. [200-]. Cisco 7940/7960 SIP with BroadVoice. <<http://www.voip-info.org/wiki/index.php?page=Cisco+7940%2F7960+SIP+with+BroadVoice>>. [25 de Mayo de 2011].

Voip para novatos. [2008]. 101 cosas que puedes hacer con Asterisk. <<http://www.voipnovatos.es/item/2008/06/101-cosas-que-puedes-hacer-con-asterisk>>. [25 de Mayo de 2011].

Wikimedia Foundation, Inc. [2011]. Wikipedia The Free Encyclopedia. <<http://en.wikipedia.org/>>. [25 de Mayo de 2011].

Wintermeyer, Stefan. 2007. Practical Asterisk 1.4. (Asterisk Práctico 1.4). Alemania.
Addison-Wesley Professional. Consulta en línea: <http://www.the-asterisk-book.com/unstable/>

APENDICE A:
GUIAS DE LABORATORIO.

GUÍA DE LABORATORIO N°1.

INSTALACIÓN DEL SOFTWARE: SO CENTOS, ASTERISK SOBRE CENTOS, X LITE Y ZOIPER.

1.- Objetivos del laboratorio:

Durante esta sesión de laboratorio los alumnos realizarán las siguientes actividades:

- Instalar el sistema operativo CentOS en dos PC's del laboratorio: PC1 y PC2.
- Practicar y familiarizarse con el terminal de Linux y sus comandos básicos en PC 1 y PC2.
- Conocer el editor Vim en PC 1 y PC2.
- Practicar y familiarizarse con el terminal gráfico Gnome en PC 1 y PC2.
- Instalar Asterisk sobre CentOS en PC 1 y PC2.
- Instalar softphones X Lite y Zoiper
 - Instalar Softphone X Lite en PC 3 y PC 4.
 - Instalar Softphone Zoiper en PC 5 y PC 6.

Los alumnos se dividirán en dos grupos, para descargar e instalar CentOS y Asterisk en sendos PC's del laboratorio, siguiendo el procedimiento que se describe a continuación. Asimismo, posteriormente los alumnos trabajarán en grupos para instalar softphones en otros 4 PC del laboratorio.

2.- Procedimiento para instalación de CentOS.

CentOS puede instalarse como único sistema operativo (en este caso se borra toda la información que tiene el disco duro), o como segundo SO, es decir dejando la posibilidad de usar Windows o CentOS. Si se requiere usar el segundo SO se indica esta opción durante el proceso de partida del PC. La instalación de CentOS como único SO es más simple y rápida.

En esta sección explicaremos la instalación de CentOS dejando utilizable también el anterior SO.

2.1 Pasos previos

Antes de proceder a la instalación de CentOS, se debe averiguar algunas características del computador en que se instalará y llevar a cabo las tareas que se enumeran a continuación:

- Averiguar el tipo de arquitectura del procesador.
- Averiguar el tamaño de la memoria RAM.

- Desfragmentar el disco duro.
- Crear una partición para CentOS en el disco duro.

Una vez realizado lo anterior, se procederá a:

- Obtener CentOS desde la Web
- Instalar CentOS

A continuación se explican los puntos indicados anteriormente:

Arquitectura del procesador

El software CentOS a instalar debe corresponder con la arquitectura del PC en que se instalará. Por esta razón, durante la descarga de CentOS se deberá indicar la arquitectura del PC en que se hará la instalación, pudiendo ser ésta i386 ó x86_64. Para conocerla, ingrese a Windows XP, escritorio de Windows, parte izquierda inferior, pinchar con el botón izquierdo del mouse en inicio, todos los programas, accesorios y pinchar con el botón izquierdo del mouse en información de sistema. Se abrirá una ventana que en la parte superior izquierda muestra el texto "resumen de sistema". Abralo. A la derecha aparecerá el campo llamado "tipo de sistema", donde se encuentra la información buscada. Anotela.

Determinación del tamaño de la memoria RAM

Durante el proceso de instalación se deberá crear una memoria swap, con doble tamaño que la memoria RAM. Por tanto hay que averiguar el tamaño de la memoria RAM, lo que se logra entrando al ambiente de Windows XP, posicionando el cursor sobre Mi PC, presionando el botón derecho del mouse, esto desplegará un menú, del cual se debe seleccionar con el botón izquierdo del mouse la opción de propiedades. En la viñeta "general" encontrará el tamaño de la memoria RAM.

Desfragmentación del disco duro

Antes de crear la partición en que se instalará CentOS, se recomienda desfragmentar el disco duro, de modo que la información esté ordenada y compactada a la hora de particionar. Debe tenerse en cuenta que el proceso es lento, pudiendo tomar hasta hora y media. Para abrir el programa que lleva a cabo esta tarea, diríjase al escritorio de Windows, pinche con el botón izquierdo de su mouse sobre inicio, vaya a todos los programas, accesorios y pinche con el botón izquierdo de su mouse sobre desfragmentador de disco. Esto abrirá el programa. Con el mouse seleccione la unidad de disco duro a procesar y presione con el botón izquierdo de su mouse el botón indicado para ello, esto dará comienzo al proceso. Al terminar el proceso recibirá un mensaje, dando la posibilidad de ver un informe del proceso realizado.

Creación de una partición en el disco duro

Como hemos dicho, instalaremos el sistema operativo CentOS sin eliminar el sistema operativo Windows XP. Una vez instalado, durante el proceso de encendido de la máquina se ofrecerá la opción de usar uno u otro sistema operativo (dualboot). Para ello es necesario crear una partición en el disco duro en la que se instalará el sistema operativo CentOS. Versiones vigentes de Windows como Windows Vista o Windows 7, incluyen los programas necesarios para crear una partición sin necesidad de formatear el disco duro sin embargo Windows XP no. Dado que los PC's del laboratorio cuentan con Windows XP instalados en ellos, es necesario recurrir a un programa externo para crear la partición requerida, sin tener que formatear el disco duro. Para el desarrollo de este proyecto de título se utilizó Partition Magic de Norton (Version 8.0), aunque también se puede utilizar otro programa. Normalmente las particiones instaladas en los discos duros abarcan todo el espacio de disco, por lo que para crear una nueva, es necesario liberar espacio de otras. Por ello es necesario redimensionar las particiones existentes para que se libere espacio y se pueda crear una partición para instalar CentOS. La proporción utilizada para crear la partición es de "dos es a uno", eso significa que Windows XP se queda con dos tercios del disco duro y CentOS con un tercio. El procedimiento para crear una partición con Partition Magic 8.0 es el siguiente:

- Abra el programa Partition Magic 8.0 de Norton.
- En la parte superior izquierda encontrará "crear una nueva partición". Selecciónela presionándola con el botón izquierdo del mouse.
- Seleccione el lugar de creación de la partición. Prefiera la recomendada.
- Luego el asistente pregunta de que particiones desea tomar espacio. Prefiera la partición principal.
- Después el asistente pide el tamaño de la partición, asignele 7 Gb, la etiqueta: "Asterisk-CentOS", Crear como: Logica, Tipo de sistemas de archivos: no es importante y letra de unidad: K. Y a continuación presionar siguiente.
- Confirmar opciones.
- Presionar terminar.
- Abajo a la izquierda encontrará el botón "Aplicar" para aplicar los cambios, presiónelo.
- Para concretar los cambios reinicie.
- Luego de reiniciar, ingrese al programa nuevamente y verifique que la barra del disco duro tenga la partición efectivamente incluida.

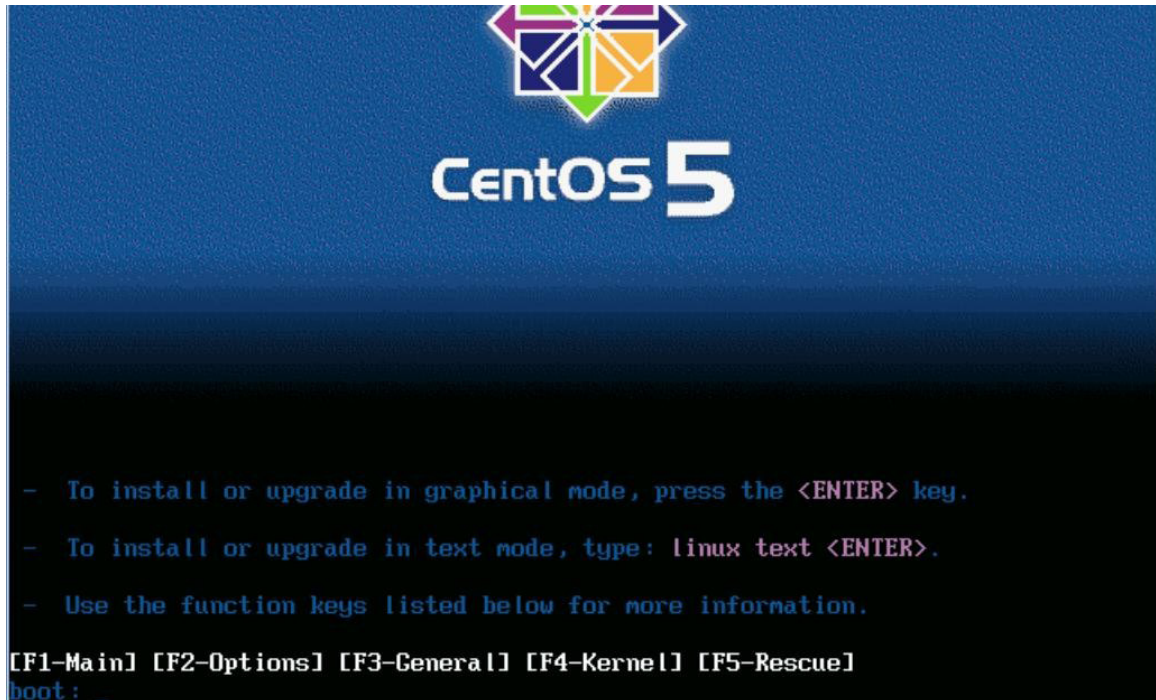
2.2 Obtención de CentOS

Para descargar CentOS desde la web, ingrese a la pagina www.centos.org, diríjase a la sección "CentOS releases" de la última versión lanzada de CentOS. Donde dice download, seleccione la arquitectura del procesador que su computador utiliza. Se abrirá una página con servidores "espejo" ("*mirror*"). Seleccione un servidor de Chile, o de un país cercano por defecto, uno rápido (se aprecia por método prueba-error, a veces son muy lentos). Esto abrirá otra ventana con los archivos de descarga de CentOS. Descargue los archivos ".iso" CentOS del 1 al 7. Si desea puede utilizar Bit Torrent. Grabe las imágenes de CD en CD-

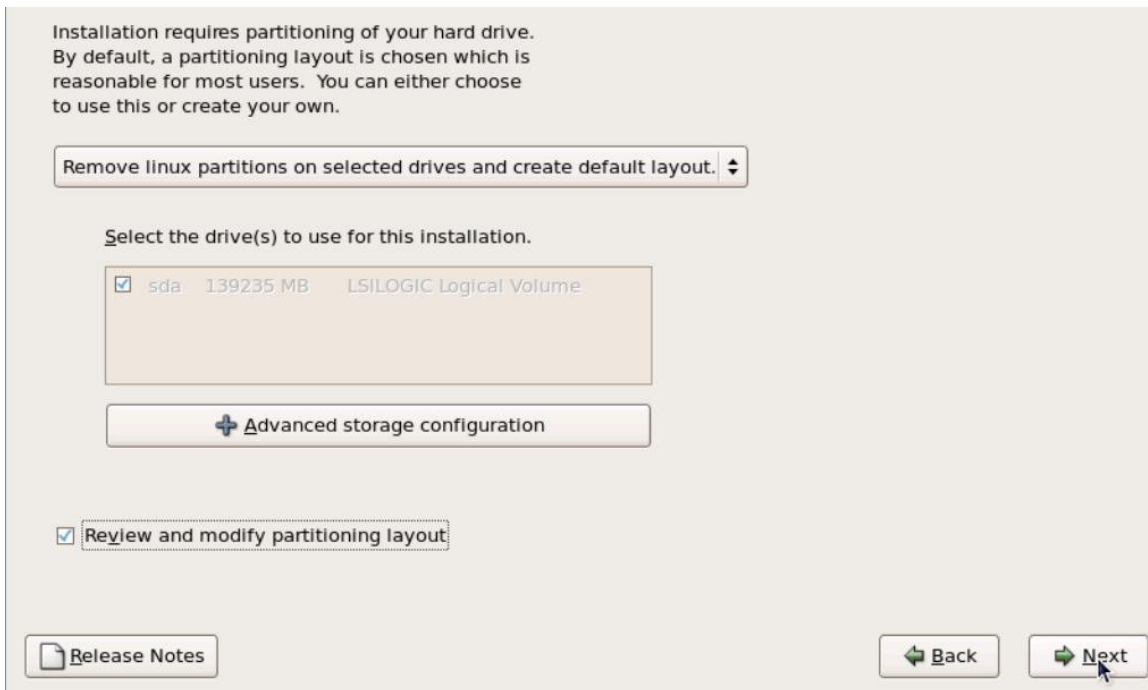
ROM's. Debemos hacer notar que también es posible bajar CentOS a un DVD, sin embargo los PC's del laboratorio actualmente no disponen de lectores de DVD.

2.3 Instalación de CentOS

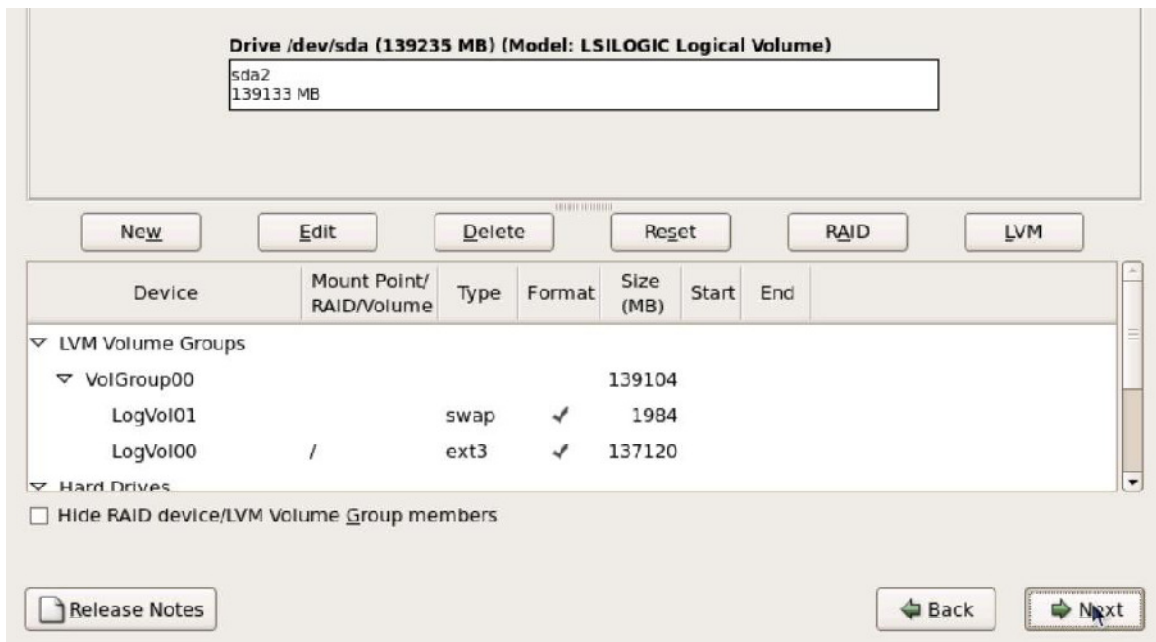
La instalación de CentOS es guiada por un programa asistente de instalación. Las opciones de configuración se detallan a continuación. Para comenzar el proceso de instalación se debe insertar el CD número 1 de instalación de CentOS y reiniciar. El computador leerá el CD de instalación y mostrara una ventana como la siguiente:



- Elija instalación en modo gráfico, presionando el botón <ENTER>.
- Después, aparecerá una ventana dando la posibilidad de revisar los discos de instalación, paso que podemos omitir si no consideramos necesario revisar los discos.
- A continuación aparecerá un menú para ver las notas de ediciones (release notes). Presione "Next" para continuar.
- Después aparecerá un listado de idiomas para elegir el idioma del proceso de instalación. Seleccione Spanish (Español). Presione Next para continuar.
- A continuación seleccionamos el idioma del teclado, elegimos español. Presione Next para continuar.
- Aparecerá un cuadro como el que sigue, informando que la instalación requiere la partición del disco duro.



- Seleccione del menú desplegable "crear diseño personalizado". Presione Next para continuar.
- Como se muestra en la siguiente figura, verá el particionamiento del disco duro, seleccione la partición de Linux creada previamente y elimínela.



- Luego se deben crear particiones adecuadas para la instalación de Linux. Se crean 3 particiones. Para crear cada partición se debe presionar el botón "new", lo que abrirá un cuadro como el que sigue:

Add Partition

Mount Point: /boot

File System Type: ext3

Allowable Drives: sda 139235 MB LSILOGIC Logical Volume

Size (MB): 100

Additional Size Options

Fixed size

Fill all space up to (MB): 1

Fill to maximum allowable size

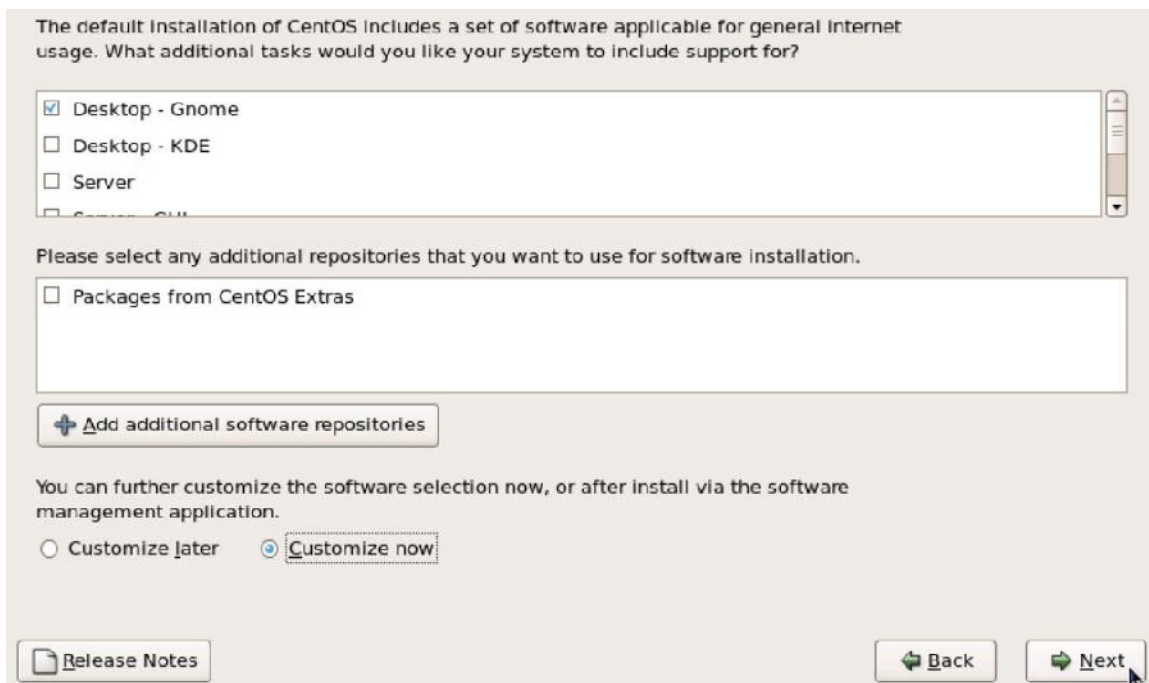
Force to be a primary partition

Cancel OK

- Crear las tres particiones, cada una con las siguientes características:
 - 1) boot (100 Mb): En "Mount Point" seleccionar la opción "/boot", y en tamaño asignar 100 Mb, presionar el botón Ok.
 - 2) swap (doble de Memoria Ram): Se busca en "File System Type" la opción "swap" y se asigna el doble del tamaño de la memoria RAM en "Size (MB)" (la memoria RAM fue averiguada anteriormente). Presione el botón OK para continuar.
 - 3) / (Raiz) (el resto del disco duro disponible): Buscar en Mount Point: "/" y seleccionar completar hasta el tamaño máximo disponible y presionar botón Ok. ¿Que es una partición o root? ¿cual es su utilidad?
- Una vez creadas las particiones, presione Next para continuar.
- A continuación verá un menú relacionado a la configuración del menú Grub que aparece al encender o reiniciar el PC, desde el cual se puede elegir que sistema operativo desea utilizar. En el cuadro al centro verá los dos sistemas operativos,

CentOS y Otro. Modifique el nombre de "Otro" y llámelo "Windows XP", y seleccione el sistema operativo que desea que arranque por defecto. En este caso elegimos Windows XP. Presione Next para continuar.

- En el siguiente cuadro verá la sección "dispositivos de red", que se puede modificar para seleccionar la configuración IP. Asigne manualmente el nombre del Host: Servidor_Asterisk_Maestro ó Servidor_Asterisk_Esclavo para cada computador. Presione Next para continuar.
- A continuación elija la zona horaria, America/Santiago, y marque la casilla: El reloj del sistema utiliza UTC. Presionar botón Next para continuar.
- Asignar la contraseña: lel629maestro o lel629esclavo para el usuario root según corresponda. Presionar Next para continuar.
- A continuación aparece el siguiente recuadro:



The screenshot shows a window titled "The default installation of CentOS includes a set of software applicable for general internet usage. What additional tasks would you like your system to include support for?". It features a list of checkboxes: "Desktop - Gnome" (checked), "Desktop - KDE", "Server", and "Server - GUI". Below this is a section "Please select any additional repositories that you want to use for software installation." with a checkbox for "Packages from CentOS Extras" and a button "Add additional software repositories". At the bottom, there are radio buttons for "Customize later" and "Customize now" (selected), a "Release Notes" button, and "Back" and "Next" navigation buttons.

- Seleccione la opción "Personalizar ahora" y presione Next lo que mostrará un recuadro como el siguiente:



- En la sección "Entorno de Escritorio" seleccione: Entorno de Escritorio de Gnome.
- En la sección Aplicaciones se selecciona solamente: Editores – Internet gráfica.
- En la opción Desarrolla se selecciona: Bibliotecas de Desarrollo – Herramientas de desarrollo.
- Desarrollo de Software Anticuoado.
- En la opción Servidores se selecciona: Servidor de correo.
- En la opción Sistema base se selecciona: Base – Soporte de red mediante discado – Sistema X Windows.
- Una vez seleccionado los paquetes de instalación, presionar Next para continuar con la instalación.
- Presionar Next para iniciar la instalación de CentOS.
- A continuación se muestra una lista de los CD's requeridos para la instalación. Presionar Next para continuar.
- Al terminar el proceso de instalación se debe reiniciar el computador, presionar forward.
- Despues de reiniciar el computador se puede configurar el firewall, deshabilítelo. Continuar.
- Del mismo modo se deshabilita SELinux tambien. Continue.
- Ajusta la hora y fecha de ser necesario. Continuar.
- Crear un usuario llamado "invitado" con contraseña "invitado".
- Se mostrará una ventana relacionada a la tarjeta de sonido, presione forward, puesto que el servidor no posee.
- Presione Finish y confirme el reinicio. Si luego de reiniciar puede llegar al entorno de escritorio CentOS, entonces se puede concluir que la instalación de CentOS ha finalizado exitosamente.

3.- Práctica y familiarización con el terminal (o consola) Linux

Esta parte del laboratorio tiene por objetivo que los alumnos que no hayan trabajado anteriormente con Linux, adquieran un mínimo de familiarización con los comandos de este sistema operativo, utilizando la consola de comandos (o terminal) que proporciona CentOS.

Para abrir un terminal en ambiente CentOS, diríjirse a la barra superior del escritorio, aplicaciones, accesorios y abra una terminal.

Inicie la práctica realizando las siguientes acciones:

- Abra una terminal Linux.
- Lea el manual de la terminal ubicado en ayuda, contenido.
- Escriba "man cd" para ver el manual del comando "cd".
- Mediante el comando "cd", ubíquese sobre el directorio raíz: "/", directorio del cual nacen el resto de los directorios. Use el comando "cd /", o "cd .." para descender de directorio uno a uno.
- Escriba "man ls" en la terminal y presione enter para ver el manual de "ls".
- Use el comando "ls" para ver el sistema de archivos del directorio raíz. Anotelos y averigüe para que sirve cada uno.
- Mediante el comando "cd usr" ubíquese en el directorio "usr", y mediante el comando "ls" vea los archivos en el directorio. ¿Para que sirve este directorio?.
- Vaya al directorio "/var" con el comando "cd /var", y vea los archivos en el directorio usando el comando "ls". ¿Para que sirve este directorio?.
- Vaya al directorio: /home, con el comando "cd/home", y cree un directorio llamado "temp_lab_1" con el comando "mkdir temp_lab_1", y entre en él mediante el comando "cd temp_lab_1".
- Crear un archivo llamado "archivo_a" con el comando "vim archivo_a". Presione "i" después que abra Vim para pasar del modo "normal" al modo "insertar", escriba en programa el texto: "texto de archivo_a", luego presione "esc" para volver al modo Normal, finalmente ingrese el comando "ZZ" (en mayúsculas), que guarda los cambios y cierra el editor.
- Cree un "archivo_b" con las mismas características, mediante el mismo procedimiento.
- Revise el contenido del directorio sobre el cual se encuentra mediante el comando "ls".
- Revise el manual de "cat" mediante el comando "man cat".
- Escriba el comando "cat archivo_a archivo_b > archivo_c", presione enter.
- Revise el contenido de la misma carpeta mediante el comando "ls".
- Con el comando "vim archivo_c" revise el contenido del documento. ¿Que se ve?
- Descienda de directorio con el comando "cd ..".
- Comprima la carpeta temp_lab_1 en el archivo "archivos_lab.tar" con el comando "tar cvf archivos_lab.tar temp_lab_1".
- Cree un directorio llamado "temp_lab_final" en la carpeta "/home" con el comando "mkdir temp_lab_final".

- Con el comando "cp archivos_lab.tar /home/temp_lab_final" copie el archivo recién comprimido a la carpeta creada.
- Con el comando "cd /home" vaya al directorio /home.
- Mediante el comando "rm -rf temp_lab_1" remueva las carpeta temp_lab_1 con sus contenidos.
- Mediante el comando "ln archivo_a link_archivo_a" cree en el mismo directorio un vinculo (link) del archivo_a llamado "link_archivo_a".
- Mediante el comando "ln -s archivo_b link_archivo_b" cree un vinculo "link" del "archivo_b" llamado "link_archivo_b".
- Mediante los comandos "rm archivo_a" y "rm archivo_b" remueva los archivos originales: archivo_a y archivo_b.
- Ejecute el comando "ls". ¿Que se observa?.
- Mediante los comandos "vim link_archivo_a" y "vim link_archivo_b" intente ingresar a los vinculos . ¿Que sucede?.
- ¿Que sucede con los links? Revise "man ln". Explique brevemente.
- Ingrese el comando "ls -l", presione enter. ¿Que son esas letras al comienzo de cada registro? Mediante el comando "man ls" revise el manual ls. Explique brevemente que son.
- Mediante el comando "chmod g+rw archivo_a" modifique los permisos del "archivo_a", de modo que los usuarios pertenecientes al mismo grupo lo puedan ver y modificar.
- Ingrese el comando ls -l, ¿Observa algún cambio?.

4.- Práctica con el editor Vim

Esta parte del laboratorio tiene por objetivo que los alumnos que no hayan trabajado anteriormente con Linux, adquieran un mínimo de familiarización con el editor de texto Vim, incluido en CentOS.

Inicie la práctica realizando las siguientes acciones:

- Escriba "man vim" y presione enter, lea el manual de Vim.
- Salga del manual presionando q.
- Ubíquese en una terminal sobre el directorio /home/temp_lab_final, ingrese el comando "vim prueba_vim", lo que dará origen al archivo de texto en blanco llamado "prueba_vim" en Vim.
- Escriba :help y presione enter. Lea el manual. Salga escribiendo :q y presionando enter.
- Para aprender sobre el funcionamiento básico del editor escriba ":help usr_02.txt" y presione enter, esto le mostrará el uso básico del editor. Ponga especial atención en el tema de los modos, y en los comandos esenciales para trabajar con el editor.
- Explore el programa, pruebe sus distintos comandos y aplicaciones. En el archivo ingrese texto.
- Presione "esc", ingrese :exit y presione enter para guardar y cerrar el terminal.

5.- Práctica para familiarizarse con el terminal gráfico Gnome

Esta parte del laboratorio tiene por objetivo que los alumnos practiquen utilizando el terminal gráfico Gnome de CentOS.

Valiéndose de Gnome, los alumnos ubicarán y modificarán los archivos que guardaron durante el desarrollo de los puntos anteriores.

- Sobre el escritorio CentOS, hacer doble click en el ícono "equipo", después sobre sistema de archivos, a continuación sobre home y finalmente sobre temp_lab_1. Ahí encontrará los archivos con los que se trabajó en la parte anterior (archivos_lab.tar y prueba_vim).
- Descomprima el archivo "archivos_lab.tar", pinchando con el botón derecho del mouse y seleccionando "Abrir con <<Gestor de archivadores>>". Abierto el "Gestor de archivadores", seleccione el archivo a descomprimir y presione extraer. Cierre el gestor de archivadores. Revise el resultado de la descompresión.
- Abra cada uno de los archivos haciendo doble click sobre ellos.

6.- Instalación de Asterisk sobre CentOS.

El proceso de instalación de Asterisk se lleva a cabo de la siguiente manera:

En el ambiente de escritorio Gnome de CentOS, se pincha con el botón izquierdo del mouse el ícono del programa de Internet que tiene la forma de planeta tierra con un mouse rodeándolo, que se ubica en la parte superior del escritorio. Como es la primera vez que se usa el programa, éste mostrará un recuadro informando que no hay un navegador predeterminado, dándose además la posibilidad de elegir el deseado, seleccione Elinks. Después aparecerá un recuadro invitando al usuario a ingresar una URL. Ingrese a la página www.asterisk.org, presionando a continuación enter. El browser que se usa es basado en texto, por lo que la página se visualizará distinta. En la página principal, haga click sobre "downloads". En la sección "Install From Yum Repo", haga click sobre "yum installation". Esto lo llevará a otra página, que corresponde a una guía para instalar Asterisk con Yum ("Installing Asterisk with Yum"). De acuerdo a dicha guía, siga el siguiente procedimiento de instalación:

- Crear un archivo de texto llamado "centos-asterisk.repo" en el directorio "/etc/yum.repos.d" mediante el editor de texto, ubicado en aplicaciones, accesorios y escriba el siguiente texto en el archivo creado:

```
[asterisk-tested]
name=CentOS-$releasever - Asterisk - Tested
baseurl=http://packages.asterisk.org/centos/$releasever/tested/$basearch/
enabled=0
gpgcheck=0
#gpgkey=http://packages.asterisk.org/RPM-GPG-KEY-Digium

[asterisk-current]
name=CentOS-$releasever - Asterisk - Current
baseurl=http://packages.asterisk.org/centos/$releasever/current/$basearch/
```

```
enabled=1
gpgcheck=0
#gpgkey=http://packages.asterisk.org/RPM-GPG-KEY-Digium
```

- Guardar los cambios y cerrar el documento. Crear otro archivo llamado "centos-digium.repo" en el mismo directorio y escriba el siguiente texto en él:

```
[digium-tested]
name=CentOS-$releasever - Digium - Tested
baseurl=http://packages.digium.com/centos/$releasever/tested/$basearch/
enabled=0
gpgcheck=0
#gpgkey=http://packages.digium.com/RPM-GPG-KEY-Digium
```

```
[digium-current]
name=CentOS-$releasever - Digium - Current
baseurl=http://packages.digium.com/centos/$releasever/current/$basearch/
enabled=1
gpgcheck=0
#gpgkey=http://packages.digium.com/RPM-GPG-KEY-Digium
```

- Guardar los cambios y cerrar el documento. Abra una terminal Linux e ingrese el siguiente comando:

```
[root@localhost~]# yum install asterisk16 asterisk16-configs asterisk16-voicemail dahdi-linux dahdi-tools libpri
```

- Presionar enter. Se mostrarán muchos datos, después de los cuales se mostrara un texto como el que sigue:

```
...
...
...
```

```
Total download size: 35 M
Is this ok [y/N]:
```

- Confirme afirmativamente ingresando "y", luego presione Enter. Después de unos cuantos minutos, se pedirá otra confirmación, a la que también debe responder afirmativamente. Después de unos cuantos minutos más aparecerá un texto como el de a continuación:

```
Installed: asterisk16.i386 0:1.6.0.15-1_centos5
Dependency Installed: asterisk16-core.i386 0:1.6.0.15-1_centos5 ...
Complete!
```

Con lo cual se puede concluir que la instalación de Asterisk ha finalizado satisfactoriamente.

Para comprobar inicie Asterisk mediante el comando:

- asterisk -vvvpci.

Si ve la CLI, entonces puede concluir que la instalación se llevó a cabo exitosamente. Salga de Asterisk con el comando:

- Exit

Estudie el directorio /etc/asterisk y compruebe que están los archivos de configuración sip.conf, extensions.conf, iax.conf, etc.

7.- Instalación de softphones X Lite y Zoiper.

7.1 X Lite

Los alumnos instalarán X Lite en 4 PC's del laboratorio siguiendo el siguiente procedimiento:

- Descargar la última versión gratuita del Softphone X-Lite, de la página www.counterpath.com/x-lite.html.
- Ejecutar el programa descargado haciendo doble click sobre éste y seguir las instrucciones de instalación de la guía virtual.
- Se mostrará un menú de bienvenida. Presione *Next* para comenzar la instalación.
- Aceptar los términos de licencia. Presione *Next* para continuar.
- Seleccionar el directorio en el cual desea instalar el programa, prefiera el directorio sugerido. Cree un acceso directo del programa en el escritorio. Presione *Next* para continuar.
- Presione *Install* para comenzar la instalación del programa en el computador.
- Cuando el proceso de instalación haya terminado marque la casilla para iniciar X Lite (*Launch X Lite*). Presionar "*Finish*" para terminar.
- Espere hasta que el softphone se inicie y abra, entonces se habrá concluido que la instalación del softphone X Lite ha sido exitosa.

7.2 Zoiper

En forma similar a lo que se hizo para X Lite, lleve a cabo el siguiente proceso de instalación de Zoiper en los mismos 4 PC's del laboratorio:

- Ingrese a la página www.zoiper.com. Descargue la última versión gratuita de Zoiper.
- Ejecute el programa descargado y siga las instrucciones de instalación de la guía virtual.
- Aparecerá un menú de bienvenida. Presionar *Next* para continuar con la instalación.
- Se mostrarán los terminos de licencia, los cuales se deben aceptar para continuar. Presione el botón "*I Agree*" con el botón izquierdo de su mouse para continuar.
- Se ofrece opción para elegir el directorio en el que se instalará el programa,

- prefiera el directorio recomendado. Presione Next para continuar.
- Escoger el directorio del menu de partida. Presionar Next para continuar.
 - Elegir los componentes Zoiper y acceso directo de escritorio. Presione "*Install*" para instalar.
 - Al terminar aparecerá un cuadro informando que Zoiper ha sido instalado en su computador. Presione "*Finish*" para terminar.
 - Inicie Zoiper y compruebe que quedó correctamente instalado.

GUÍA DE LABORATORIO N°2.

CONFIGURACIÓN I: ASTERISK CON SOFTPHONES X-LITE Y ZOIPER Y ASTERISK CON OTRO ASTERISK.

1.- Objetivos del laboratorio:

Durante esta sesión de laboratorio los alumnos realizarán las siguientes actividades:

- Configurar Asterisk con los softphones X Lite y Zoiper.
- Configurar los softphones X Lite y Zoiper.
- Realizar prácticas básicas con comandos Asterisk.
- Configurar la interconexión de dos servidores Asterisk.

Los alumnos se dividirán en dos grupos para continuar el trabajo iniciado en la sesión pasada hasta dejar configurados dos servidores Asterisk en sendos PC del laboratorio, e instalarán softphones en otros 4 PC's del laboratorio. Los servidores Asterisk se llamarán Maestro y Esclavo respectivamente. Los alumnos terminarán la sesión realizando prácticas con los comandos básicos de Asterisk.

2.- Configuración de Asterisk con los softphones X Lite y Zoiper.

Para establecer la comunicación entre los softphones y Asterisk, es necesario editar los archivos sip.conf, iax.conf y extensions.conf todos los cuales se encuentran en la carpeta /etc/asterisk del servidor.

En cada servidor Asterisk se creará la carpeta /etc/asterisk/backup y moveremos a ella los archivos extensions.conf, sip.conf y iax.conf, como una forma de protegerlos de posibles equivocaciones.

En la carpeta /etc/asterisk crearemos tres archivos de texto en blanco llamados sip.conf, iax.conf y extensions.conf, en los cuales ingresaremos la información que indicamos a continuación:

2.1 Edición del archivo sip.conf

En cada servidor, en el archivo en blanco sip.conf ingresamos la siguiente información:

```
; *****  
; *** CONFIGURACIÓN GLOBAL PARA EL CANAL SIP DE ASTERISK ***  
  
[general]  
context=default ; Contexto por defecto de recepción de  
llamadas.  
realm=XXX.com ; Cambiar XXX por le1629maestro o le1629esclavo según  
; corresponda.  
bindport=5060 ; Puerto UDP para conectarse.
```



```

bindaddr=0.0.0.0          ; Dirección IP a conectarse (0.0.0.0 se conecta con
todos).
srvlookup=yes            ; Habilita revisiones de DNS SRV para llamadas
externas.
disallow=all             ; Deshabilitar todos los codecs.
allow=ulaw               ; Habilita codecs en orden de preferencia.
allow=gsm
language=en              ; Lenguaje por defecto

; *****
; *** DECLARACIONES DE REGISTRO ***

; *****
; ** USUARIOS Y DISPOSITIVOS ***

[601]                    ; (X Lite 1, Extension 601 para el servidor
maestro y 701            ; para el servidor
esclavo).
type = friend            ; Define el tipo de conexión para el cliente.
host = dynamic           ; Define la dirección IP o nombre del cliente
resolvable del          ; dispositivo.
secret = 601             ; Contraseña del cliente.
context = users          ; Contexto por defecto para este cliente.

[602]                    ; (X Lite 2, Extension 602 para el servidor maestro y
702 para                ; el servidor esclavo).
type = friend
host = dynamic
secret = 602
context = users

[603]                    ; (Zoiper SIP 1, Extension 603 para el servidor
maestro y 703            ; para el servidor esclavo).
type = friend
host = dynamic
secret = 603
context = users

[604]                    ; (Zoiper SIP 2, Extension 604 para el servidor
maestro y 704            ; para el servidor esclavo).
type = friend
host = dynamic
secret = 604
context = users

```

Una vez terminada la edición, grabamos y cerramos el archivo sip.conf

2.2 Edición del archivo iax.conf

Ingresamos en los archivo iax.conf de cada servidor los siguientes datos:

```
; iax.conf
```

```

[general]

context=inband           ; Contexto por defecto.
bindaddr=0.0.0.0        ; Direccion IP donde servidor escucha conexiones IAX.
bindport=4569           ; Puerto donde se escuchan las conexiones entrantes.
language=es             ; Lenguaje español.
tos=0x18                ; Indicamos que marque los paquetes para QoS.
bandwidth=low           ; Preferibles codecs con bajo consumo.
allow=all               ; Usaremos todos los codecs

```

```

[650]                   ; Zoiper IAX 1. [750] para el servidor esclavo.

```

```

type=friend
host=dynamic
context=users
qualify=yes
autokill=yes
requirecalltoken=no

```

```

[660]                   ; Zoiper IAX 2. [760] para el servidor esclavo.

```

```

type=friend
host=dynamic
context=users
qualify=yes
autokill=yes
requirecalltoken=no

```

Una vez terminada la edición, grabamos los cambios y cerramos el archivo iax.conf que seguiremos editando más adelante

2.3 Edición del archivo extensions.conf

En cada servidor, en el archivo en blanco extensions.conf ingresamos la siguiente información:

```

; *****
; ** CONFIGURACIÓN GENERAL PARA LAS EXTENSIONES

[general]
static=yes
writeprotect=no
autofallthrough=yes
clearglobalvars=no
priorityjumping=no

; *****
; ** VARIABLES GLOBALES
[globals]

; *****
; ** CONTEXTOS DE EXTENSIONES DEBAJO DE ESTA LINEA

```

```

[users]

exten => 10x0,1,Dial(SIP/60${EXTEN:2:1},20)           ; Para el servidor
maestro.
exten => 10x0,2,Hangup()

; exten => 20x0,1,Dial(SIP/70${EXTEN:2:1},20)       ; Para el servidor
esclavo.
; exten => 20x0,2,Hangup()

exten => 100x,1,Dial(IAX2/6${EXTEN:3:1}0@6${EXTEN:3:1}0,20)
exten => 100x,2,Hangup()                           ; Para el servidor
maestro.

; exten => 200x,1,Dial(IAX2/7${EXTEN:3:1}0@7${EXTEN:3:1}0,20)
; exten => 200x,2,Hangup()                         ; Para el servidor
esclavo.

```

Una vez terminada la edición, grabamos los cambios y cerramos el archivo extensions.conf

2.4 Activación de la configuración

Iniciamos un terminal y activamos Asterisk. Recargamos los archivos sip.conf, iax.conf y extensions.conf con nuestra programación, usando los siguientes comandos:

```

- # asterisk - vvvr
- CLI> sip reload
- CLI> iax2 reload
- CLI> dialplan reload

```

Nos aseguramos que la configuración fue guardada correctamente usando los siguientes comandos:

```

- CLI> sip show peers

Name/username      Host                Dyn    Nat ACL Port Status
1010                (Unspecified)      D      0   Unmonitored
1020                (Unspecified)      D      0   Unmonitored
2 sip peers [2 online , 0 offline]

- CLI> sip show users

Username    Secret Accountcode Def.Context ACL NAT
1010       1010         users       No RFC3581
1020       1020         users       No RFC3581

- CLI> iax2 show peers

```

Name/Username	Secret Accountcode	Def.context	ACL NAT
605	users		No RFC3581

- CLI> dialplan show users


```
[ Context 'users' created by 'pbx_config' ]
'1010' => 1. Dial(SIP/1010|20) [pbx_config]
'1020' => 1. Dial(SIP/1020|20) [pbx_config]
-- 2 extensions (2 priorities) in 1 context. ==
```
- CLI> show dialplan
- CLI> exit

Tomamos nota de los resultados que arrojan los comandos sip show peers, sip show users, iax2 show peers y dialplan show users.

3.- Configuración de los softphone X Lite y Zoiper.

Para configurar los softphones es necesario conocer las direcciones IP de los servidores. Esto se logra abriendo en ambiente CentOS del servidor un terminal y escribiendo el comando "ifconfig", presionando luego enter. Esto mostrará información de las interfaces de red. Debemos anotar la información que aparece después de "inet addr:", la cual corresponde a la dirección IP del servidor.

3.1.- Configuración softphone X-Lite

Se corre el programa para llevar a cabo la configuración del softphone. En la barra superior del menu del softphone aparecerá una sección llamada softphone. Ingresamos a ella presionando el boton izquierdo de su mouse sobre ella, observaremos un menú desplegable, que entre sus opciones muestra "account settings". Ingresamos presionando el botón izquierdo del mouse. En la pestaña "Account" encontraremos una serie de cuadros para configurar el aparato. Los esenciales y necesarios son User ID y Domain, los cuales completamos de la siguiente manera:

- User ID: 601 y 602 (servidor maestro) ó 701 y 702 (servidor esclavo).
- Domain: La dirección IP del servidor (por ejemplo: 10.10.100.20 ó 10.10.100.30), para la extension 601, 602, 701 y 702, asociada al servidor maestro o esclavo, o según corresponda.

Se pueden llenar el resto de las casillas. En particular, se define una clave:

- Password: 601 y 602 ó 701 y 702, ya sea para la extension 601 y 602 ó la 701 y 702.
- Para terminar la configuración presione Ok.

Completado lo anterior, el softphone intentará registrarse en Asterisk. Si se logra quiere decir que la conexión entre Asterisk y el softphone está establecida.

3.2.- Configuración del softphone Zoiper:

El softphone Zoiper se pueden configurar de modo que señalice con protocolo SIP o con protocolo IAX.

3.2.1.- Configuración protocolo SIP (línea SIP):

Activamos el programa softphone Zoiper, entramos a la configuración del softphone. En "Add new SIP account", aparecerá un cuadro llamado add new SIP account, solicitando el nombre de la cuenta. Ingresamos el nombre. A continuación aparecerá un cuadro solicitando los siguientes datos, que ingresamos uno a uno:

- Domain: 10.10.100.20, 10.10.100.30 o según corresponda.
- Username: 603 y 604 ó 703 y 704, o según corresponda.
- Password: 603 y 604 ó 703 y 704, o según corresponda.
- Caller ID: 603 y 604 ó 703 y 704, o según corresponda.

3.2.2.- Configuración protocolo IAX (línea IAX):

Activamos el programa softphone Zoiper, entramos a la configuración del softphone. En "Add new IAX account", aparecerá un cuadro llamado add new IAX account, solicitando el nombre de la cuenta. Ingresamos el nombre. A continuación aparecerá un cuadro solicitando los siguientes datos, que ingresamos uno a uno:

- Server Hostname/IP: 10.10.100.20, 10.10.100.30 o según corresponda.
- Username: 650 y 660 ó 750 y 760, o según corresponda.
- Password: 650 y 660 ó 750 y 760, o según corresponda.
- Caller ID Name: 650 y 660 ó 750 y 760, o según corresponda.
- Caller ID Number: 650 y 660 ó 750 y 760, o según corresponda.

Finalmente presionamos "Apply" y luego "Ok". Para el servidor maestro la numeración comienza con "6" y para el esclavo con "7".

A continuación, el softphone intentará registrarse con alguna de las cuentas. Verificamos que se completa el proceso de registro con todas las cuentas. Para cambiar la cuenta con la cual se está registrando en Zoiper, ubicamos en Zoiper el menú desplegable

que permite elegir la línea que deseamos utilizar, la seleccionamos y automáticamente Zoiper intentará registrarse con la cuenta seleccionada. Tomamos nota de los resultados.

4.- Configuración de interconexión de los dos servidores Asterisk entre sí.

4.1 Edición del archivo iax.conf en el servidor maestro

En el archivo `iax.conf` del servidor maestro que habíamos empezado a editar en el punto 2.2, agregamos la siguiente información adicional:

```
; iax.conf

[servidor_maestro]

type=friend
secret=123                                ; El parámetro secret debe ser igual en ambos
                                           servidores ;esto para efectos de autenticación
                                           entre ambos servidores.
host=direccion.IP.de.servidor_esclavo ; En este caso 10.10.100.30, o según
corresponda.
permit=direccion.IP.de.servidor_esclavo ; En este caso 10.10.100.30, o según
corresponda.
auth=md5                                  ; Algoritmo de encriptación para transmisión
de las 'secret'.
disallow=all
allow=ulaw
context=anexos
qualify=no
```

Guardamos los cambios y cerramos el archivo.

4.2 Edición del archivo iax.conf en el servidor esclavo

En el archivo `iax.conf` del servidor esclavo que habíamos empezado a editar en el punto 2.2, agregamos la siguiente información:

```
; iax.conf

[servidor_esclavo]

type=friend
secret=123                                ; El parámetro secret debe ser igual en ambos
servidores                                ;esto para efectos de autenticación entre ambos
servidores.
host=direccion.IP.de.servidor_maestro ; En este caso 10.10.100.20, o según
corresponda.
permit=direccion.IP.de.servidor_maestro ; En este caso 10.10.100.20, o según
corresponda.
auth=md5                                  ; Algoritmo de encriptación para transmisión
de las 'secret'.
```

```
disallow=all
allow=ulaw
context=anexos
qualify=no
```

Recargamos la configuración en cada Asterisk y pedimos confirmación de los dispositivos configurados ingresando los siguientes comandos en la terminal:

- asterisk*CLI> iax2 reload
- asterisk*CLI> iax2 show peers

4.3 Edición del archivo extensions.conf en el servidor maestro

Modificamos el archivo extensions.conf de la carpeta /etc/asterisk ingresando la siguiente información que utilizará el Asterisk Maestro cuando necesite conectarse con el Asterisk Esclavo:

```
; extensions.conf

[anexos]

exten => 20xx,1,Dial(IAX2/servidor_esclavo@servidor_maestro/${EXTEN})
exten => 20xx,2,Hangup
exten => 10x0,1,Dial(SIP/60${EXTEN:2:1},20) ; Previamente definido.
exten => 10x0,2,Hangup()
```

Guardamos los cambios y cerramos el archivo.

Para recargar la configuración del plan de discado (extensions.conf) y mostrar el plan de discado, iniciamos una terminal, y ejecutamos los siguientes comandos, presionando enter después de cada uno:

- asterisk*CLI> dialplan reload
- asterisk*CLI> dialplan show users
- asterisk*CLI> show dialplan

Tomamos nota de los resultados que arrojan los comandos recién ingresados (dialplan show users, show dialplan y IAX2 show peers).

4.4 Edición del archivo extensions.conf en el servidor esclavo

Modificamos el archivo extensions.conf de la carpeta /etc/asterisk ingresando la siguiente información que utilizará el Asterisk Esclavo cuando necesite conectarse con el Asterisk Maestro:

```
; extensions.conf

[anexos]

exten => 10xx,1,Dial(IAX2/servidor_maestro@servidor_esclavo/${EXTEN})
exten => 10xx,2,Hangup
```

```
exten => 20x0,1,Dial(SIP/70${EXTEN:2:1},20) ; Previamente definido.  
exten => 20x0,2,Hangup()
```

Guardamos los cambios y cerramos el programa.

Para recargar la configuración del plan de discado (`extensions.conf`) y mostrar el plan de discado, iniciamos una terminal, y ejecutamos los siguientes comandos, presionando enter después de cada uno:

```
- asterisk*CLI> dialplan reload  
- asterisk*CLI> dialplan show users  
- asterisk*CLI> show dialplan
```

Tomamos nota de los resultados que arrojan los comandos recién ingresados (`dialplan show users`, `show dialplan` y `IAX2 show peers`).

5.- Prácticas con comandos básicos Asterisk.

Revisar y practicar con comandos básicos de Asterisk.

Analizar los archivos `sip.conf`, `iax.conf` y `extensions.conf`.

Realizar llamadas entre softphones XL, Z y entre XL y Z.

Practicar cambiando números de acceso XL y Z y cambiando tiempos de ringing. Usar `dialplan show`

Practicar absorción e inserción de dígitos.

5.1.- Revisar y practicar con comandos básicos de Asterisk.

Durante el desarrollo de este Laboratorio hemos trabajado con comandos de Asterisk, que se explicaremos a continuación:

`asterisk -r`: Comando utilizado para acceder a la interfaz de línea de comandos Asterisk (CLI). Para desplegar mayor información sobre lo que Asterisk está haciendo agregue el argumento `-v`, cuantas veces se desee, de la siguiente manera:

`# asterisk -vvvvvvr`: A mayor cantidad de `v`, mayor es la información que entrega Asterisk como respuesta a cada comando (mayor es la verbosidad). La *CLI (command line interface, Interface de Línea de Comando)* permite interactuar con un servidor de Asterisk activo y es muy útil para la corrección de errores y monitoreo.

`extensions reload`: Recarga la configuración del plan de discado del archivo `extensions.conf`. En otras palabras, recarga solamente el plan de discado y nada más. Este comando se puede ejecutar incluso aunque hayan llamadas activas. Cualquier canal nuevo que se cree estará basado en el plan de discado recién recargado.

`show dialplan [context]`: Muestra el actual estado del plan de discado tal como se ha cargado en la memoria. Si se agrega un contexto al final del comando, será mostrado solamente ese contexto. El comando `show dialplan` es útil para verificar el orden de selección de patrones también.

dialplan show users: Muestra los usuarios registrados en el plan de discado.

sip reload: Recarga el módulo del canal *SIP*. Esto equivalente a "load chan_sip.so". Es necesario hacer la recarga *SIP* para que los cambios hechos en "sip.conf" y "sip_notify.conf" se activen. Durante la recarga las comunicaciones establecidas por los canales *SIP* no se interrumpen.

sip show peers: Despliega una lista y muestra el estado de todos los terminales *SIP*.

sip show users: Despliega una lista de todos los usuarios configurados en sip.conf

Práctica con otros comandos Asterisk

Asterisk tiene muchos otros comandos. Inicie la consola Asterisk en una terminal con el comando asterisk -vvvr y ejecútelos para practicar con ellos:

help [command[subcommand]...]: Despliega ayuda para uso de comandos y líneas de comandos. Un simple signo de pregunta o tab hará lo mismo. Se desplegará una lista de comandos para los cuales se proporciona ayuda. Escriba "help" y presione enter. Esto mostrará un listado de comandos con una breve descripción.

sip debug: Este comando enciende el depurador de *SIP*. Ingrésele y presione enter. Se mostrará mucha información relacionada al canal *SIP*.

console dial [extension[@context]]: Disca una extensión dada (opcionalmente en el contexto dado) a través del canal de la consola. Este canal solamente está disponible si chan_oss.so o chan_alsa.so está cargado en el archivo modules.conf (por defecto chan_oss está cargado).

hangup: Termina cualquier llamada actualmente establecida usando el canal consola. Este comando está solamente disponible en el canal chan_oss.so o chan_alsa.so que está cargado en el archivo modules.conf. Termina la llamada.

answer: Contesta una llamada entrante en el canal Console (OSS). El canal OSS debe estar configurado en oss.conf antes de que el comando answer esté disponible.

sip no debug: Apaga el depurador de *SIP*. Desactiva *SIP* Debug.

restart/stop [gracefully/now/when convenient]: Al ejecutar restart o stop, todos los canales se liberan y todos los módulos se recargan. Detiene Asterisk.

exit: Cierra la *CLI (Command Line Interface)*. Si se conectó a la consola Asterisk con la bandera "-r", no puede usar los comandos "exit" y "quit" para apagar la *PBX* (como si estuviese corriendo en primer plano). Para apagar la *PBX* más bien que salir de la consola, vea los comandos stop y restart. Sale de la consola.

Disque una llamada a 1020 o 2020 según corresponda, usando estos comandos.

5.2.- Análisis de archivos sip.conf, iax.conf y extensions.conf.

Lea los archivos extensions.conf, sip.conf y iax.conf guardados en la carpeta backup y responda:

- En cuanto a sip.conf:

¿Cuál es la sintaxis para discar a dispositivos SIP? ¿Qué comandos CLI son útiles para revisar terminales y usuarios? ¿Cuál es la diferencia entre nombres de dispositivos y números de telefonos? ¿Cuál es la expresión que permite registrarse a un proveedor SIP externo y recibir llamadas de él? ¿Cuál es la declaración que permite realizar llamadas al proveedor SIP?

- En cuanto a iax.conf:

¿Qué es IAX y qué significa? ¿Cómo se recarga el archivo? ¿Para qué sirve este archivo?

- En cuanto a extensions.conf:

¿Qué es el archivo extensions.conf? ¿Para que sirve? ¿Cómo se recarga el archivo de configuración? ¿Cuál es el formato de los nombres de extensiones? ¿Qué es la prioridad?, ¿Cómo funciona?, ¿Qué tipos hay?, ¿Cuál es el formato de las expresiones de extension?

A continuación:

Analice el archivo de configuración "sip.conf" ubicado en "/etc/asterisk" del servidor Asterisk, y en base a lo leído indique que significa cada una de las expresiones en el archivo y para qué sirven.

Para concluir, escriba las tareas que está ejecutando la central en general. Apóyese en otras fuentes de información si es necesario. Vea al final de esta guía la bibliografía recomendada. Repita el mismo procedimiento para el archivo iax.conf y extensions.conf.

5.3.- Determinación de la dirección IP del servidor Asterisk.

Dirijase a la central servidor, y en una terminal ingresar el comando ifconfig, presionar enter, esto mostrará información sobre la interfaz de red TCP/IP. En la sección llamada "inet addr:" encontrará la dirección IP del servidor.

5.4.- Hacer llamadas entre los softphones XL, Z SIP y Z IAX.

Hacer llamadas entre los softphones X Lite, Zoiper línea SIP y Zoiper línea IAX, y entre ellos en todas las combinaciones. ¿Se logró la comunicación entre los dispositivos? ¿Todas las líneas funcionaron? ¿Qué diferencia hay entre una línea SIP y una línea IAX?

5.5.- Practicar cambiando números de acceso X-Lite y Zoiper y cambiando tiempos de ringing, usar dialplan show.

Antes de comenzar esta sección, abra el archivo `extensions.conf` y `sip.conf`, guarde una copia bajo el nombre de `extensions.conf.of` y `sip.conf.of` en la carpeta `backup`, esto servirá para poder trabajar con los archivos sin problemas y que al final no se pierdan los cambios.

A continuación, en el archivo `extensions.conf`, donde se declaran las extensiones encontrará las expresiones que permiten a los dispositivos comunicarse y definir los números de acceso y tiempos de repique. Las expresiones tienen la forma:

- `exten => numero_de_acceso, prioridad, Dial(SIP/nombre_de_dispositivo, tiempo_de_repique)`

`numero_de_acceso` es el número con que se accede a la extensión, y es el que se debe modificar para que la extensión sea accesada con otro número. `Tiempo_de_repique` se modifica para variar la duración del tiempo de repique.

Modifique en el archivo `extensions.conf` ubicado en `/etc/asterisk` en el servidor, el número de acceso de `10x0` a `3x33` ó `20x0` a `4x44`. Cambie además el tiempo de repique a 30 segundos. Grabe los cambios, cierre el archivo, recargue el plan de discado, y revise el plan de discado configurado, mediante los siguientes comandos:

- `asterisk*CLI> dialplan reload`
- `asterisk*CLI> show dialplan`

5.6.- Absorción e inserción de dígitos.

`{EXTEN}` es la variable que contiene los dígitos discados por el llamante. La absorción e inserción de dígitos se programa “manipulando” u obteniendo un sub-string de la variable `{EXTEN}` en el `dialplan`. En general se pueden obtener substrings de cualquier variable usando el siguiente formato:

```
{NOMBREVARIABLE[:start[:length]]}
```

start posición del carácter a partir del cual se construirá el substring (la posición se cuenta desde posición 0, 1, 2,). También se puede interpretar como la cantidad de caracteres iniciales de la variable, que se ignoran (absorción de caracteres)

length longitud que tendrá el substring. En caso que se omita este parámetro opcional, el substring tendrá todos los restantes caracteres de la variable

Si queremos limitar el substring a los `n` últimos caracteres de la variable, se usa el formato `{NOMBREVARIABLE:-n}`

Ejemplo.

```
{NOMBREVARIABLE}= curso_el629_sistemas_conmutación_telefónica
```

```
Si se quiere “cortar” la variable a “curso_el629”: {NOMBREVARIABLE:0:11}
```

```
Si queremos extraer “el629”: {NOMBREVARIABLE:6:5}
```

```
Si queremos dejar “telefónica”: {NOMBREVARIABLE:-10}
```

Para el caso que nos interesa, es decir la aplicación de estos formatos para modificar la variable `_${EXTEN}` en el dialplan, veamos el siguiente ejemplo:

Supongamos que el abonado llamante marcó los dígitos 27385310 y está digitando desde un teléfono que tiene como `context=anexos`. En este caso, para establecer la llamada, Asterisk irá a la línea correspondiente al número 27385310 del `context= anexosxi`, del archivo `extensions.conf`.

```
exten => 27385310,1,Dial(SIP/678${EXTEN:4},20)
```

Teniendo en cuenta que la variable `_${EXTEN}` tiene el valor 27385310 (dígitos marcados por el llamante), lo programado en esta línea dice lo siguiente:

Cuando el abonado llamante digita 27385310, la llamada se trata de la siguiente forma: establecer la conexión usando tecnología SIP (canal SIP) con el destino 6785310. Los primeros dígitos 678 están directamente indicados en la línea de instrucción y los cuatro finales 5310 se obtienen quitando los cuatro primeros dígitos a los almacenados en `_${EXTEN}`. Es decir en este caso se insertan tres dígitos al comienzo (678) y se absorben los cuatro primeros dígitos almacenados en `_${EXTEN}` (2738). Los últimos dos dígitos de la línea (“20”) indican que se aplicará tensión de llamada (ringing) durante 20 segundos.

Actividad: Diríjase al archivo `extensions.conf` del servidor ubicado en `/etc/asterisk`, y absorba el dígito “x” de las extensiones discadas en 3x33 y 4x44 y inserte “60” o “70” seguido del dígito almacenado en “x” en el nombre del dispositivo al cual desee llamar. Lo cual debería entregar una expresión de la forma:

```
exten => 3x33,1,Dial(SIP/60${EXTEN:1:1},30)
```

Nota: Recuerde que en servidor Asterisk Maestro es 60 y en Esclavo es 70.

Grabe los cambios y cierre el archivo; Diríjase a la consola CLI de Asterisk; ingrese los siguientes comandos, presionando enter despues de cada uno:

```
- asterisk*CLI> dialplan reload
- asterisk*CLI> show dialplan
```

Tome nota del plan de discado almacenado en la central. Realice llamadas usando los nuevos números configurados. ¿Se llevaron a cabo los cambios satisfactoriamente? ¿Hubo comunicación? ¿Comentarios, dudas, sugerencias?

Reestablezca la configuración original, eliminando los archivos `sip.conf` y `extensions.conf` existentes y renombrando los originales, `sip.con.of` y `extensions.conf.of` a `sip.conf` y `extensions.conf`, y ubíquelos en su directorio original.

^{xi} En caso que el dispositivo llamante no tenga asignado un context, la línea de programa `exten =>27385310,` se buscará en el `context=default` del archivo `extensions.conf`

GUÍA DE LABORATORIO N°3.

CONFIGURACIÓN II: TELÉFONO IP, ATA, SERVICIOS VOICE-MAIL E IVR.

1.- Objetivos del laboratorio:

Durante esta sesión de laboratorio se desarrollarán las siguientes actividades:

- Instalación y configuración de un telefono IP al servidor ASTERISK MAESTRO (Grupo 1).
- Instalación y configuración de una ATA con un telefono analógico al servidor ASTERISK ESCLAVO (Grupo 2).
- Programación y pruebas del servicio Voice Mail.
- Programación y pruebas del servicio IVR.
- Programación y pruebas del servicio operadora automática (IVR +Voice mail).
- Programación y pruebas del servicio Meetme.

Los alumnos se dividirán en dos grupos, para continuar con el trabajo comenzado anteriormente, uno trabajará entorno al servidor maestro (mas el PC1 y 2) y el otro entorno al esclavo (mas PC3 y 4), él que trabaja con el maestro, instala el teléfono IP en el PC1, él que trabaja con el servidor esclavo, instala la ATA en el PC3, ambos grupos instalan los demás servicios en sus respectivos servidores.

Para comenzar consiga su dispositivo (telefono IP/grupo 1, ATA-telefono análogo/grupo 2), y encienda su computador, entre a CentOS como usuario root, clave AdminLabEL629. Siga los siguientes procedimientos:

2.- Instalación y Configuración del Telefono IP(Grupo 1).

El objetivo de esta parte del laboratorio es que los alumnos configuren un teléfono IP, para incluirlo en el sistema y familiarizarse con el trabajo de Asterisk con estos dispositivos.

2.1.- Configuración de la extension en Asterisk.

2.1.1.- Definición del anexo.

Abra con el editor de texto el archivo sip.conf, ubicado en /etc/asterisk/, y declarar en la sección de "dispositivos, usuarios y amigos" el dispositivo:

```
[607]
type = friend
host = dynamic
context = users
```

secret=607

Luego, guarde los cambios y cierre el archivo. En una terminal, abra Asterisk, y recargue y revise la configuración con los siguientes comandos:

- [root@servidor.maestroEL629.com~] asterisk -vvvr
- CLI> sip reload
- CLI> sip show peers

Tome nota de los resultado entregados.

2.1.2.- Plan de discado.

Abra con el editor de texto el archivo extensions.conf, ubicado en /etc/asterisk/, y en la sección de contextos de extensiones, agregar las siguientes instrucciones:

```
exten => 1070,1,Dial(SIP/607,20)
exten => 1070,2,Hangup()
```

Luego, guarde los cambios y cierre el archivo. En una terminal, abra Asterisk, y recargue y revise la configuración con los siguientes comandos:

- [root@servidor.maestroEL629.com~] asterisk -vvvr
- CLI> dialplan reload
- CLI> dialplan show users

Tome nota de los resultados entregados.

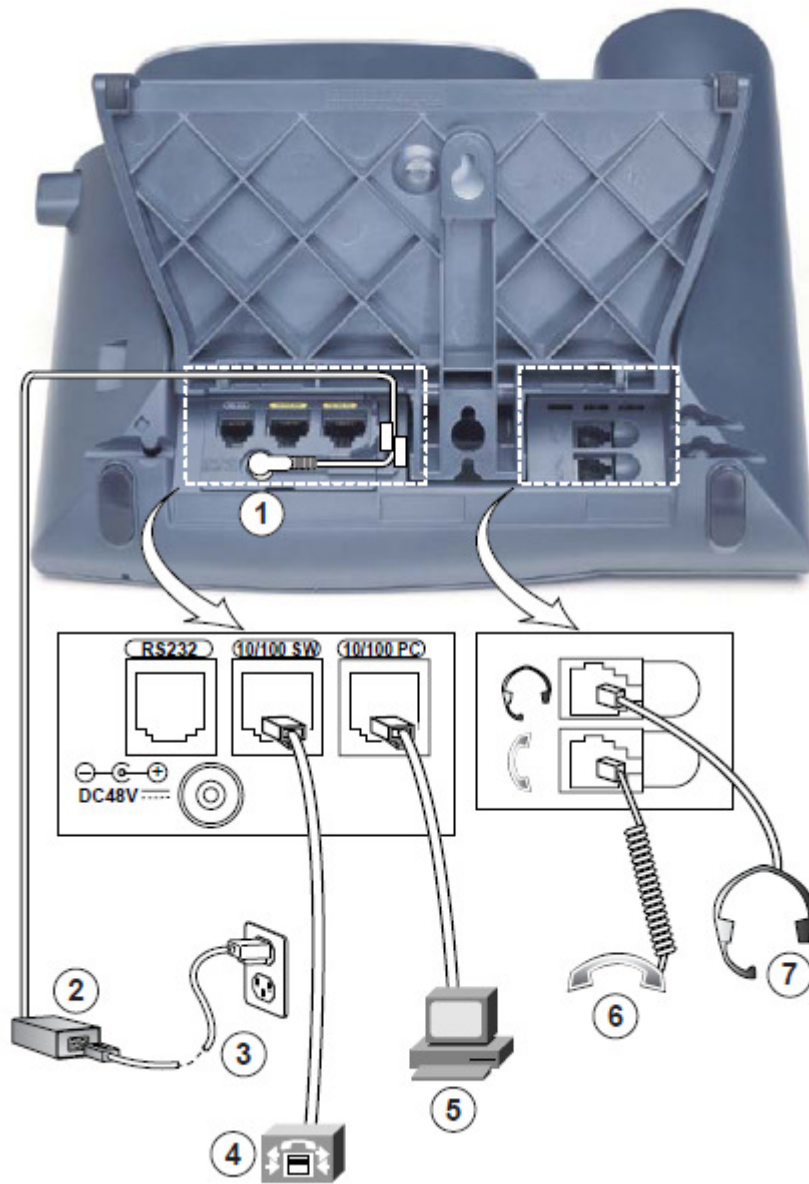
2.2.- Instalación y configuración del Aparato Telefónico.

2.2.1.-Instalación del Teléfono IP.

Siga los siguientes pasos, remítase a las figura N°nn, para localizar puertos en la parte trasera de su teléfono:

- Dirigirse al PC3 y desconecte el cable de Ethernet conectado a él y conéctelo al puerto de “Network” en la parte trasera del teléfono. Use el cable de Ethernet incluido con su telefono para conectar el puerto de acceso en la parte trasera de su teléfono con el PC3.
- Conecte el enchufe de la fuente de poder DC48V AC al enchufe en la parte trasera del teléfono.
- Use el cable de poder para conectar la fuente de poder a un enchufe del laboratorio.

Una vez que conecte el teléfono IP Cisco a la fuente de poder, un proceso de iniciación comienza. Después de unos cuantos minutos, la pantalla de su teléfono desplegará un pantalla de “listo” (“ready” en inglés). Los detalles en la pantalla pueden variar, pero típicamente tienen hora, fecha, extensión, y algunas teclas “blandas” (“soft”). Entonces la iniciación esta completa y su teléfono esta listo para usarse.



- 1 Puerto de adaptador AC (DC48V)
- 2 Fuente de poder con enchufe de puerto de adaptador AC.
- 3 Cable de poder con enchufe de soquete de pared.
- 4 Puerto de Red de trabajo (10/100 SW)
- 5 Puerto de acceso (10/100 PC)
- 6 Puerto de Handset.
- 7 Puerto de Headset.

Para configurar la línea SIP entre el teléfono IP y el servidor, se debe configurar el teléfono IP. Siga el siguiente procedimiento para ello:

- Presionar el botón de configuración en el panel del teléfono, que está abajo y a la derecha del signo de interrogación.
- Esto llevará a la sección de configuraciones. Con las flechas del panel descender hasta "unlock config" y presionar "select". Ingresar la clave "cisco" y presionar "accept".
- A continuación subir en el menú usando las flechas hasta "SIP configuration", y presionar "select". Bajar hasta "Line 3 settings" y presionar "select".
- Sitúese sobre "Name" y presione "edit", en new name ingresar "607", presione accept.
- Sitúese sobre "authentication password", presionar "edit" y ingresar "607" y presionar aceptar.
- Sitúese sobre proxy address, presionar edit y ingresar la dirección del servidor, y presionar aceptar.

Salga del menú, y espere mientras el teléfono se registra. Realizar llamadas desde el teléfono IP hacia las líneas configuradas, y desde las líneas hacia el teléfono IP. ¿Se logra la comunicación?

3.- Instalación y Configuración de ATA (Grupo 2).

El objetivo de esta parte del laboratorio es que los alumnos configuren un teléfono IP, para incluirlo en el sistema y familiarizarse con el trabajo de Asterisk con estos dispositivos.

3.1.- Configuración de extensión en Asterisk.

3.1.1.- Definición de anexo.

Abra con el editor de texto el archivo sip.conf, ubicado en /etc/asterisk/, y declare en la sección de "dispositivos, usuarios y amigos" el dispositivo:

```
[707] ; ATA
type = friend
host = dynamic
secret = 707
context = users
```

Luego, guarde los cambios y cierre el archivo. En una terminal, abra Asterisk, y recargue y revise la configuración con los siguientes comandos:

- [root@servidor.esclavoEL629.com~] asterisk -vvvr
- CLI> sip reload
- CLI> sip show users

Tome nota de los resultados entregados.

3.1.2.- Plan de discado.

Abra con el editor de texto el archivo extensions.conf, ubicado en /etc/asterisk/, y en la sección de contextos de extensiones, agregue las siguientes instrucciones:


```
exten => 2070,1,Dial(SIP/707,20)
exten => 2070,2,Hangup()
```

Luego, guarde los cambios y cierre el archivo. En una terminal, abra Asterisk, y recargue y revise la configuración con los siguientes comandos:

```
- [root@servidor.maestroEL629.com~] asterisk -vvvr
- CLI> dialplan reload
- CLI> dialplan show users
```

Tome nota de los resultados entregados.

3.2.- Instalación y Configuración de la ATA.

A continuación se instalará y configurará la ATA. La ATA, es marca Leadtek, modelo BVA 8053R y que viene en una caja que contiene:

1. Un adaptador de VoIP de banda ancha.
2. Un adaptador de poder.
3. Un cable de ethernet.
4. Una línea telefónica.
5. Una guía de instalación rápida.

Procedimiento de instalación y configuración:

Seguir los siguientes pasos:

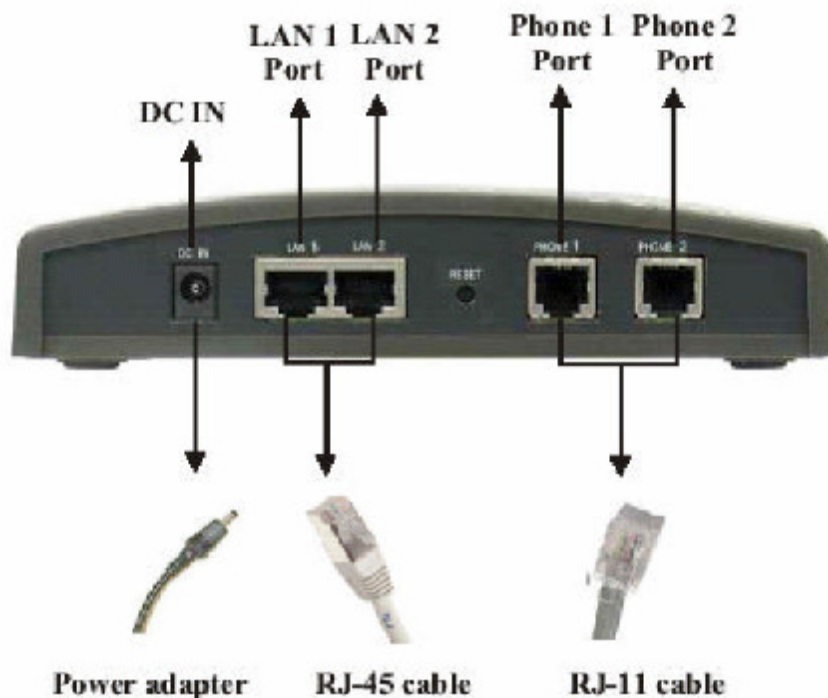
WAN: Conectarlo a un dispositivo accediendo la red (por ejemplo: un ADSL o un cable de modem).

LAN: Conectar al puerto de Ethernet de un PC.

LINE: Conectar a un soquete de pared de línea de teléfono (PSTN) (Opcional).

PHONE: Conectar a un aparato telefónico.

DC IN: Enchufe el adaptador de poder incluido.



Encender el dispositivo:

Cada vez que se enciende el dispositivo (incluyendo la primera vez), se debe dejarlo solo por al menos 5 minutos. Durante este periodo, la información de la configuración puede ser obtenida de su proveedor de servicio y alguna funciones críticas como la actualización del firmware pueden llevarse a cabo por el dispositivo. Apagar el dispositivo durante este periodo puede causar el mal funcionamiento y requerir un servicio de fabrica.

Restituir la configuración de fabrica

1. Desconecte el poder.
2. Conecte el telefono al puerto PHONE.
3. Levante la auricular del telefono (descuelgue).
4. Use un alfiler para empujar el boton RESET en la parte trasera del dispositivo. Sostengale mientras enchufa el poder de vuelta al dispositivo. El LED de poder debiera estar parpadeando rapidamente.
5. Continua sosteniendo el boton RESET hasta que vea cuatro de seis LED's parpadeando varias veces en pares. Suelte el alfiler y espere hasta que todos los LED's paren de parpadear.El proceso completo puede tomar hasta 10 segundos.
6. Ponga el auricular de vuelta sobre el telefono (cuelgue).
7. La configuración de fabrica debiera haber sido resituida.

Cambiar la configuración usando un navegador

Este dispositivo, en su configuración de fabrica, ha sido configurada para funcionar con un servidor DHCP. Para cambiar esto o para hacer cualesquiera otras modificaciones a las

configuraciones de fabrica, puede usar un navegador de internet desde su PC. Primero conecte el dispositivo a un router con DHCP habilitado que su PC tambien comparte. Identifique la dirección IP asignada a este dispositivo examinando la tabla DHCP del router.

Su servicio puede tambien proveer diferentes herramientas o metodos, porfavor revise sus paginas de internet para mayor información. Use la dirección IP identificada en el browser para accederlas paginas de configuración.

Realizar llamadas desde la ATA a todas las lineas configuradas, y desde todas las lineas configuradas hacia la ATA, tanto desde el servidor maestro como desde el esclavo. Informe resultados.

4.- VoiceMail.

Esta parte del laboratorio tiene por objetivo que los alumnos implementen un Voicemail. Un Voicemail consiste en que si un usuario no responde a una llamada telefónica dentro de un determinado numero de segundos, Asterisk desvía la llamada a un VOICEMAIL (o casilla de voz), en donde el usuario del anexo llamante puede grabar un mensaje que luego el llamado podrá escuchar. Existe la opción de que Asterisk cree un archivo con el mensaje de audio y lo envíe adjunto a un e-mail que dirige al usuario del anexo llamado. Con esto los alumnos conoceran y se familiarizaran con esta aplicación de Asterisk, y desarrollaran habilidades de manejo de software.

4.1.- Definición de las casillas de voz

Las casillas de voz se definen con un password, nombre de usuario y el e-mail al que se enviarán los archivos con las grabaciones de audio, en el archivo voicemail.conf que se ubica en el directorio /etc/asterisk. Si se usa la opción envío de e-mail, en la sección general de voicemail.conf debe ponerse el parámetro attach = yes. En la sección particular default de voicemail.conf definimos:

```
;voicemail.conf

[default]

maxmsg = 50          ; define máximo nº de mensajes en carpeta de las
                    ; casillas de voz definidas en default
6001 => 1234, alumno1, alumno1@correo.com
6002 => 5678, alumno2, alumno2@correo.com
6003 => alumno3, alumno3@correo.com
```

Con esto, se crea las casilla de voz 6001, 6002 y 6003, con usuarios alumno1, alumno2, y alumno3. Se asignan las password “1234” para el anexo “6001”, “5678” para el anexo “6002” y el correo de voz para “6003” no posee contraseña. Los archivos de voz se enviarán al e-mail alumno1@correo.com, en el caso del anexo 6001 y a alumno2@correo.com para el anexo 6002 y alumno3@correo.com para el anexo 6003. Una vez modificado el archivo voicemail.conf con todas las nuevas casillas de voz, se debe guardar los cambios y recargar. Pero ¡Atención!, el comando reload ahora tiene una estructura diferente:

```
CLI > reload app_voicemail.so
```

Recuerde presionar Enter despues de ingresar el comando.

4.2 Asociación de las casillas de voz a los anexos.

En el archivo en que se declaró el anexo usuario de la casilla de voz (generalmente sip.conf), ponemos el parámetro mailbox direccionado a la definición hecha en voicemail.conf:

```
; sip.conf

[601]
type = friend
host = dynamic
context = users
secret = 601
mailbox = 6001@default ; el anexo 601 tiene asignado la casilla 6001.

[602]
type = friend
host = dynamic
context = users
secret = 602
mailbox = 6002@default ; el anexo 602 tiene asignado la casilla 6002.

[603]
type = friend
host = dynamic
context = users
secret = 603
mailbox = 6003@default ; el anexo 603 tiene asignado la casilla 6003.
```

Guarde los cambios y cierre el archivo. En una terminal, abra Asterisk, y recargue y revise la configuración con los siguientes comandos:

```
- [root@servidor.esclavoEL629.com~] asterisk -vvvr
- CLI> sip reload
- CLI> sip show users
```

Tome notas de los resultados.

4.3 Plan de discado.

En el context de los anexos llamantes se programa el desvío a casilla de voz ; extensions.conf

```
[users]

exten => 1010, 1, Dial (SIP, 601, 20)
exten => 1010, n, Voicemail(6001@default) ; la casilla de voz 6001.
exten => 1020, 1, Dial (SIP, 602, 20)
exten => 1020, n, Voicemail(6002@default) ; la casilla de voz 6002.
exten => 1000, 1, VoiceMailMain() ; Menu IVR de Voicemail.
```

Grabe los cambios y cierre los archivos, recargue los archivos del plan de discado de `extensions.conf` con los siguientes comandos:

- `dialplan reload`

Recuerde presionar Enter para continuar. Verifique que la configuración fue efectivamente actualizada, mediante los comandos:

- `show dialplan`

Recuerde presionar enter despues de cada uno. Realice pruebas del voicemail, y grabe algunos mensajes, revise su e-mail y busque el mail asociado al mensaje de voz, lea el texto del mensaje y escuche el archivo adjunto. A continuación ingrese al menu de VoiceMail discando 1000, siga las instrucciones de la IVR para acceder a los mensajes de su casilla de voz. ¿funciona Voicemail de acuerdo a lo esperado?

5.- IVR.

Esta parte del laboratorio tiene por objetivo que los alumnos implementen una IVR, maquina de Respuesta de Voz Interactiva, conoscan esta aplicación y posteriormente se hagan pruebas sobre el sistema.

Concepto

La función Interactive Voice Response (IVR), permite que los usuarios del sistema telefónico interactúen con éste. El sistema da indicaciones al usuario mediante mensajes de voz (un menú) y el usuario responde digitando en su teclado las opciones posibles. El sistema encamina la llamada al servicio elegido. Las acciones que toma IVR pueden ser función de la fecha y horario. Además IVR permite poner mensajes corporativos para que el usuario los escuche mientras espera.

Programación de IVR

IVR se programa en el archivo `extensions.conf` definiendo un conjunto de contextos. Los mensajes deben estar grabados como archivos en la carpeta `sounds` que se encuentra en el directorio `/var/lib/asterisk`.

Ejemplos de archivos con mensajes en la carpeta `sounds` son `press-1.gsm` (presione dígito 1), `press-2.gsm`.

El sistema a implementar es el siguiente:

5.1 Descripción del servicio a implementar.

- Para acceder al servicio IVR que programaremos, los usuarios digitan 3000.
- Nuestro menú será: Para comunicarse con ventas marque 1.
- Para comunicarse con repuestos marque 2.
- Si conoce el anexo marque 3.

- Para comunicarse con las salas de conferencia marque 4.
- Para comunicarse con la operadora marque 9.
- Para repetir menú presione *.
- La persona que llama tendrá como tiempo máximo para ingresar un dígito, 6 seg. contados desde finalizada la reproducción del menú.
- Anexo ventas = 1010.
- Anexo repuestos = 2010.
- Anexo operadora = 1030.
- Anexo de la maquina que hace la grabación de voz = 1212.
- En la carpeta sounds deberán existir los files hasta_luego y número_invalido que usaremos en la programación.

5.2 Programación.

El anexo que grabará el menú de voz debe tener acceso al contexto [anexos] en que programaremos la grabación. Para acceder al programa de grabación digitará un nº de anexo conocido solo por él (9876)

```
; extensions.conf
```

```
[anexos]
```

```
; grabación del archivo file_nuestro_menú. Accesan a este programa
; solamente los anexos que conocen el nº 9876. Iniciar grabación después
; del tono. Al finalizar presionar # y se iniciará la reproducción. Para
; modificar, repetir todo el proceso.
```

```
exten => 9876, 1, Answer ()           ; descuelga
exten => 9876, n, Wait (6)             ; espera 6 segundos
exten => 9876, n, Record (file_nuestro_menu.gsm)
exten => 9876, n, Wait (1)
exten => 9876, n, Playback (file_nuestro_menu.gsm)
exten => 9876, n, Hangup()
```

```
; acceso al servicio IVR
```

```
exten => 3000, 1, Goto (nuestro_menu, s,1)
```

```
[nuestro_menu]
```

```
; encaminamiento de las llamadas según la opción solicitada al llamante
```

```
exten => s,1,Answer ()
exten => s,n,Wait (6)
exten => s,n,Background (file_nuestro_menu.gsm)
    ; reproduce menú mientras esperamarcación de una opción
exten => s,n,WaitExtend(6)
    ; espera 6 seg. que se marque un dígito antes de continuar
exten => 1,1,Goto(usuarios,1010,1)
    ; llamante marcó opción 1.
exten => 2,1,Goto(usuarios,2010,1)
    ; llamante marcó opción 2.
exten => 3,1,Goto(menu_anexos,s,1)
    ; Llamante marcó opción 3, se redirije al menú de ingreso de anexo.
exten => 4,1,Goto(menu_meetme,s,1)
```

```

        ; Llamante marcó opción 4, se redirige al menú de ingreso de meetme.
exten => 9,1,Goto(usuarios,1030,1)
        ; llamante marcó opción 9.
exten => *,1,Goto(s, 1)
        ; llamante marcó opción repetir menú.
exten => t,1,Playback(hasta_luego)
        ; Asterisk cuelga por time out.
exten => t,n,Hangup()
exten => i,1,Playback(número_invalido)
        ; Asterisk repite el proceso por nº inválido
exten => i, n, Goto (s, 1)

```

[menu_anexos]

```

exten => s,1,Answer()
exten => s,n,Wait(3)
exten => s,n,Background(file_menu_nro_5.gsm)
        ; reproduce menú mientras espera marcación de una opción
exten => s,n,WaitExtend(6)
        ; espera 6 seg. que se marque un dígito antes de continuar
exten => _x.,1,Goto(anexos,${EXTEN},1)
        ; Dirige las llamadas al contexto anexos.
exten => *,1,Goto(s,1)
        ; llamante marcó opción repetir menú
exten => t,1,Playback(hasta_luego)
        ; Asterisk cuelga por time out
exten => t,n,Hangup()
exten => i,1,Playback(número_invalido)
        ; Asterisk repite el proceso por nº inválido
exten => i,n,Goto(s,1)

```

[menu_meetme]

```

exten => s,1,Answer()
exten => s,n, Wait(3)
exten => s,n, Background (file_menu_nro_4.gsm)
        ; reproduce menú mientras espera
        ; marcación de una opción
exten => s,n,WaitExtend(6)
        ; espera 6 seg. que se marque un dígito antes de continuar
exten => _x.,1,MeetMe(%{EXTEN})
        ; Dirige las llamadas al contexto anexos.
exten => *,1,Goto (s,1)
        ; llamante marcó opción repetir menú.
exten => t,1,Playback (file_menu_nro_3) ; Asterisk cuelga por time out
exten => t,n,Hangup()
exten => i,1,Playback(file_menu_nro_2)
        ; Asterisk repite el proceso por nº inválido
exten => i,n,Goto(s,1)

```

Recuerde guardar y recargar la configuración.

6.- Operadora automática (IVR + Voicemail).

6.1.- Concepto.

La idea de la operadora automática es atender las llamadas y captar mensajes de voz en diferentes casillas de voz, dependiendo del servicio que solicita el llamante, incluso cuando nadie atiende las llamadas. El sistema propone un menú al llamante para que digite una opción, desvíe la llamada al anexo correspondiente a la opción y si éste no responde en un plazo determinado, conecta la llamada a la correspondiente casilla de voz. La implementación de operadora automática se hace mediante las facilidades de IVR y de Voicemail. La programación de la operadora automática, básicamente consiste en agregar las funciones Voicemail al servicio IVR recién programado.

Lo cual es automático, puesto que cada línea posee ya su propio buzón de voz. Por lo que existiendo ya el IVR, solamente se redirige las llamadas, y si no hay respuesta, se desvía al buzón de voz.

7.- Meetme (Salas de Conferencias).

Esta parte del laboratorio tiene por objetivo que los alumnos configuren el módulo de conferencia MeetMe de asterisk que se realiza en el archivo de configuración 'meetme.conf' que está ubicado en el directorio '/etc/asterisk'. Traslade el archivo existente a la carpeta '/etc/asterisk/backup' y cree en la carpeta '/etc/asterisk' un archivo nuevo en blanco con el mismo nombre. Revise el archivo "meetme.conf" original. El archivo de configuración para las salas de conferencias de MeetMe se definiría de la siguiente forma:

La sintaxis del archivo es la siguiente:

Con el contexto [rooms] damos inicio a la definición de nuestras salas de conferencia. Luego para crear una sala de conferencia iniciamos con el comando 'conf =>', seguido del número de la sala de conferencia y la contraseña para dicha sala, si queremos acceder a esta sala sin una contraseña entonces se deja solo el número de la sala.

```
conf => numero_sala,contraseña_sala
```

Posteriormente una vez realizados estos cambios en el archivo 'meetme.conf', veremos la forma de acceder a estas salas. Esto se realiza en el archivo 'extensions.conf':

Con esto, en caso de marcar la extensión 1100 entonces como prioridad número uno ejecutará la aplicación Meetme y accederemos a la sala de conferencia 110. Agregue el siguiente texto a los siguientes archivos:

```
; meetme.conf

[rooms]
;
; Usage is conf => confno[,pin][,adminpin]
;
conf => 610 ; 710 en el servidor esclavo.
conf => 620,1234 ; 720 en el servidor esclavo.

; extensions.conf

[anexos]
```



```
exten => 1100,1,Meetme(610) ; 2100 en el servidor esclavo.  
exten => 1200,2,Meetme(620) ; 2200 en el servidor esclavo.
```

Recuerde guardar los cambios y cerrar los archivos, para luego recargar las configuraciones respectivas con los comandos:

- dialplan reload
- reload

Compruebe que se ha actualizado el plan de discado mediante los comandos:

- dialplan show

Pruebe las salas de conferencia llamando a ellas desde tres softphones.

GUIA DE LABORATORIO N°4

FUNCIONES DE LLAMADA: LÍMITE DE TIEMPO PARA LLAMADAS, GRABACIÓN DE AUDIO, CONEXIÓN SIP CON LA PSTN.

1.- Objetivos del laboratorio:

Durante esta sesión de laboratorio los alumnos realizarán las siguientes actividades:

Configurar el sistema Asterisk para fijar límite de tiempo a las llamadas.

Grabar audio de llamadas con Mixmonitor

Configurar el sistema Asterisk para establecer una conexión SIP con la PSTN

Practicar, configurar y comprobar que desde todos los teléfonos configurados es posible hacer y recibir llamadas desde la PSTN.

2.- Configuración para fijar límite de tiempo a las llamadas mediante parámetro L (x: y: z) en la función Dial

La función que permite fijar límites de tiempo a las llamadas, se programa en la aplicación 'Dial'. Para ello se usa un parámetro adicional que permite fijar un máximo para la duración de las llamadas de determinados anexos. La sintaxis de la aplicación Dial es la siguiente:

```
Dial(type1/identif1[&type2/identif2[&type3/identif3... ] ], timeout)
```

Donde:

type1 > SIP, IAX2, Zap

identified > 4000 or \${EXTEN}

timeout > Opción L(: :)

Mediante el parámetro opcional L(x:y:z), que tiene los subparámetros x, y, z, se programa el límite de duración de la llamada a "x" milisegundos; la notificación de que se cortará la llamada "y" milisegundos antes de que se cumpla el tiempo x; y la repetición de la notificación cada "z" milisegundos. Los subparámetros y, z no son siempre necesarios.

Un ejemplo de cómo se usa L(x:y:z):

Si queremos una duración máxima para las llamadas de 3600000 ms (1 hora), que la primera notificación se concrete cuando falten 61000 milisegundos (1 minuto y 1 segundo) y que la notificación se repita cada 30000 ms (30 segundos), debemos programar:

```
;extensions.conf
```

```
[llamadas-con-tiempo]
```

```
exten => _XXXXXXX,1,Dial(Zap/g1/${EXTEN},L(3600000:61000:30000))
```

```
exten => _XXXXXXX,2,Hangup
```

En este laboratorio crearemos en el archivo extensions.conf un contexto, al que llamaremos "llamadas-con-tiempo", al que dirigiremos las llamadas, para realizar las pruebas.

Entonces, en el archivo extensions.conf del servidor maestro, agregue el siguiente contexto:

```
[llamadas-con-tiempo]
```

```
exten => _10X0, 1, Dial(SIP, 60${EXTEN:2:1}, L(60000:40000:10000))
```

```
exten => _10X0, 2, Hangup()
```

```
exten => _100X, 1, Dial(IAX2, 6${EXTEN:3:1}0@6${EXTEN:3:1}0, L(60000:40000:10000))
```

```
exten => _100X, 2, Hangup()
```

```
exten => _2XXX, 1, Dial(IAX2, servidor\_esclavo@servidor\_maestro, ${EXTEN}, L(60000:40000:10000))
```

```
exten => _2XXX, 2, Hangup()
```

Guarde los cambios y cierre el documento. Del mismo modo en el servidor esclavo, en el archivo extensions.conf agregue el siguiente contexto:

```
[llamadas-con-tiempo]
```

```
exten => _20X0, 1, Dial(SIP, 70${EXTEN:2:1}, L(60000:40000:10000))
```

```
exten => _20X0, 2, Hangup()
```

```
exten => _200X, 1, Dial(IAX2, 7${EXTEN:3:1}0, L(60000:40000:10000))
```

```
exten => _200X, 2, Hangup()
```

```
exten => _10XX, 1, Dial(IAX2, servidor\_maestro@servidor\_esclavo, ${EXTEN}, L(60000:40000:10000))
```

```
exten => _10XX, 2, Hangup()
```

Recuerde grabar los cambios antes de cerrar el documento.

Todas las llamada que el plan de discado haga llegar a estos contextos, tendrán habilitada la función.

Recargue y verifique que los cambios efectivamente han sido introducidos, utilizando los comandos:

- dialplan reload
- dialplan show

Presione Enter después de ingresar cada uno de ellos.

Pruebe la función recién implementada, agregando al archivo extensions.conf, en el contexto [users], el siguiente texto:

```
;extensions.conf  
  
[users]  
  
exten => 4[12]0XX, 1, Goto(llamadas-con-tiempo, ${EXTEN:1}, 1)  
  
exten => 4[12]0XX, 2, Hangup()
```

Guarde los cambios y cierre el archivo, recargue el archivo y compruebe que los cambios son efectivos, usando los comandos:

- dialplan reload
- dialplan show

Haga pruebas llamando a los distintos dispositivos a través de estas extensiones. Compruebe que la función está haciendo su trabajo.

3.- Grabación del audio de las llamadas con Mixmonitor.

Esta aplicación nos permite grabar una conversación telefónica realizada por un anexo X, creando archivos de audio con el contenido de dicha conversación. Los archivos pueden ser formato .wav ó formato .gsm y se guardan como audio de salida y audio de entrada, a menos que se elija la opción “m”, en cuyo caso son mezclados ambos archivos en uno solo. Hay otra opción, “b”, que se usa cuando se quiere grabar solamente desde que la conversación realmente comenzó y solamente hasta que se produce el hung up.

La sintaxis de la función Mixmonitor distingue varios parámetros entre los que citamos los siguientes:

Archivo: nombre del archivo de audio que se creará con la grabación (se puede usar nombre de variables).

Formato audio: especifica el formato para los archivos de audio: wav ó gsm.

Opciones: m = mezcla sonidos de entrada y salida; b = no graba mientras hay una transferencia a otro canal.

La aplicación monitor comienza a grabar un canal mientras éste está en comunicación. Los paquetes de voz de la entrada y de la salida del canal se van almacenando en un archivo de sonido. Si no se especifica una trayectoria como parámetro de la aplicación, el archivo será almacenado en el subdirectorio del “monitor” de la ruta especificada con el astspooldir en asterisk.conf (así que por defecto los archivos de audio serán almacenados en /var/spool/asterisk/monitor).

```
;extensions.conf
```

```
[grabacion-de-llamadas]
```

```
exten => _[12]0XX,1,MixMonitor(grabacion_llamadas.wav,a,StopMixMonitor())
```

```
exten => _[12]0XX,2,Goto(users,${EXTEN},1)
```

```
exten => _[12]0XX,3,Hangup
```

Guarde los cambios. A continuación, para probar la función agregue en el contexto de usuarios [users] del mismo archivo la siguiente extension:

```
[users]
```

```
exten => _5[12]0XX,1,Goto(grabacion-de-llamadas,${EXTEN:1},1)
```

```
exten => _5[12]0XX,2,Hangup()
```

Guarde los cambios y cierre el documento. Recargue el archivo y revise la nueva configuración, utilizando los siguientes comandos:

- dialplan reload
- dialplan show

Recuerde presionar Enter después de ingresar cada comando. ¿Está la configuración actualizada? Realice pruebas llamando a través de este teléfono a los demás teléfonos del

sistema. Escuche las grabaciones de las llamadas. ¿Se grabaron efectivamente las llamadas? ¿Cómo se almacenan las grabaciones?.

4.- Conexión SIP con la PSTN.

CONFIGURACION DE CUENTAS SIP DE UN DETERMINADO PROVEEDOR EN UNA CENTRAL ASTERISK:

En el archivo sip.conf de la misma se debe agregar lo siguiente:

```
; sip.conf
register => usersip:clave_secreta@ast1.tecnoera.net/usersip
[ast1.tecnoera.net]
username=usersip
type=peer
secret=clave_secreta
host=ast1.tecnoera.net
fromuser=usersip
fromdomain=ast1.tecnoera.net
context=ingrese_su_contexto
disallow=all
allow=g729
allow=ulaw
```

Como de costumbre, guarde los cambios. Respecto a “usersip” se refiere al usuario o numeración sip entregada por el proveedor del servicio, que puede ser de 4 o más números, y sobre lo que significa “clave_secreta” es la que se entrega con el usersip.

Para recibir las llamadas desde y hacia la PSTN, Asterisk automaticamente dirige la llamada entrante al contexto llamadas-del-exterior de extensions.conf, al referirse al campo "context" del contexto "[ast1.tecnoera.net]" del archivo "sip.conf" del servidor. En el contexto llamadas-del-exterior en extensions.conf, se maneja la llamada del exterior. Y cuando se desea llamar al exterior se debe establecer comunicación con el canal asociado a la linea SIP. Esto se logra agregando al archivo extensions.conf lo siguiente:

```

; extensions.conf

[llamadas-del-externo]

exten => s,1,Dial(SIP,1010,20)

exten => s,2,VoiceMail()

exten => s,3,hangup()

```

y agregando la siguiente línea al contexto [users]:

```

exten => _X.,1,Dial(SIP,\${EXTEN}@ast1.tecnoera.net,20)

exten => _X.,2,Hangup()

```

No olvide guardar los cambios.

Con la anterior programación, al llegar una llamada desde el exterior, se dirige al contexto "llamadas-del-externo", y luego, según se configuró, se dirige la llamada a la extensión 1010.

Si hay una llamada hacia el exterior, se dirigirá a la línea

"exten => _X.,1,Dial(SIP,[\\${EXTEN}@ast1.tecnoera.net](mailto:${EXTEN}@ast1.tecnoera.net),20)", que discará el número almacenado en la variable `${EXTEN}` a través de una conexión SIP al proveedor SIP.

Haga llamadas a la PSTN y llame desde la PSTN a la central. ¿Se cursaron efectivamente las llamadas? ¿Se logró la comunicación? ¿Algún comentario o sugerencia?

5.- Llamadas hacia y desde la PSTN.

Para que el sistema sea capaz de realizar y recibir llamadas desde el exterior a cualquier teléfono del sistema telefónico, se debe modificar el contexto "llamada-del-externo", reemplazando la información original por la siguiente:

```

[llamadas-del-externo]

exten => s,1,Answer()

exten => s,2,Wait(3)

exten => s,3,Background(/etc/asterisk/llamadas/anexo)

exten => s,4,WaitExten(6)

exten => s,n,Goto(users,\${EXTEN},1)

exten => s,n,Hangup()

```

Dónde `/etc/asterisk/llamadas` es el directorio donde está almacenado el archivo de audio anexo que contiene el menú de recepción para ingresar el anexo al cual se desea comunicar. Y se graba de la siguiente manera:

Se debe ingresar en el contexto [users] el siguiente código:50

```
[users]
exten => 9874,1,Answer()
exten => 9874,2,Wait(3)
exten => 9874,3,Record(/etc/asterisk/llamadas/anexo.gsm)
exten => 9874,4,Wait(3)
exten => 9874,5,Playback(/etc/asterisk/llamadas/anexo)
exten => 9874,6,Hangup()
```

Grabe los cambios y recargue.

Después de ésto, llame al 9874 y grabe el mensaje de recepción para recibir el anexo. Entonces, se está en condiciones de recibir llamadas del exterior. Haga algunas pruebas, llame a la central desde la PSTN, se espera que la IVR conteste y le pregunte el anexo al que se desea comunicar. ¿Funcionó?¿Se logró comunicar? ¿Se estableció la comunicación con los distintos anexos?

Finalmente para llamar de cualquier teléfono del sistema al exterior, tan solo es necesario discar el número al cual se desea llamar y automáticamente Asterisk debiera dirigir la llamada al exterior.

Opcionalmente se puede habilitar un número al cual se llame, que será directo a la línea SIP, como sigue:

En el contexto [users] se define la extensión:

```
[users]
exten => 9,1,Goto(Llamadas-al-exterior,s,1)
exten => 9,2,Hangup()
[Llamadas-al-exterior]
exten => s,1,Answer()
exten => s,2,Wait(3)
exten => s,3,Disa(no-password,Lllamar-exterior)
exten => s,4,Hangup()
[Lllamar-exterior]
exten => _X.,1,Dial(SIP,\${EXTEN}@ast1.tecnoera.net,20)
```



```
exten => _X.,2,Hangup()
```

No olvide guardar los cambios y recargar. Haga ahora las pruebas necesarias. Se espera que al marcar "9", responda el contexto llamadas al exterior, espere tres segundos y posteriormente dé tono de marcado, el número ingresado será discado directamente al proveedor SIP. ¿Se obtuvieron los resultados esperados?