



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
FACULTAD DE ESTUDIOS SUPERIORES
ARAGÓN

**Implementar VoIP para realizar llamadas con arquitectura de Software Libre bajo
plataforma Linux**

TESIS

**QUE PARA OBTENER EL TÍTULO DE
INGENIERO MECÁNICO ELECTRICISTA**

PRESENTAN:

**ANA LINA TREJO MORALES
JACINTO VARGAS AVENDAÑO**

ASESOR:

ING. JOSÉ LUIS PÉREZ BÁEZ



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Dedicatoria y agradecimientos de Jacinto

Quiero agradecerte a ti Señor Dios, y al mismo tiempo dedicarte este logro en mi vida, ya que tú has dado la pauta para que todo suceda, este éxito es tuyo, ya que me colocaste en el momento y con las personas adecuadas para ello.

Comenzando con ponerme en camino de mis primeros maestros, que son mis padres, ya que ellos me guiaron en la vida para poder conseguir estos y más logros, y mis hermanos que son mis compañeros en este sendero y el resto de mi familia.

También Señor, por poner en mi camino a personas que fueron mis compañeros y hermanos en esta batalla, como Erick Israel Landeros Rodríguez, y Héctor Manuel Méndez Díaz, que ellos fueron de manera separada y a su modo fueron una gran guía, no solo en la parte académica, también en grandes aspectos de la vida.

Señor Bendice a mi amiga y compañera de Tesis, Ana Lina Trejo Morales, ya que ella es la principal causante de este logro, con sus motivaciones y constantes empeños, ha venido llevando y guiando este camino.

Y agradecer a otras personas que estuvieron y fueron pieza clave en este logro como Ing. Giovanni Nopal Pascual, que fue la primer persona que nos inició en el camino del software libre y un

excelente amigo. Ing. Humberto Martuscelli Villalba, que es la primer persona que me formó laboralmente y me tuvo confianza permitiendo un crecimiento laboral, Ing. Benito Barranco que por su ayuda y experiencia nos encamino a la elaboración de esta tesis. Ing. Anton Krall el cual aportó grades experiencias y ayudó a un crecimiento profesional. Lic. Sergio Rangel que es un gran amigo y guia de tu palabra. Ing. José Luis Pérez Báez, el cual fue nuestro asesor y con su paciencia y guia pudimos llevar a cabo la elaboración de esta tesis. Ing. Mauricio Martínez Ramírez, mi actual jefe pero sobre todo excelente amigo, del cual sigo aprendiendo bajo su guia. Por todos ellos Señor te doy gracias, ya que esta tesis solo es el conjunto de todas las bendiciones que me has dado al poner en mi vida y en el momento oportuno a cada una de estas personas.

Dedicatoria y agradecimientos de Ana

Esta tesis se la dedico a mi papá que aunque no logró ver el inicio y el fin de esta carrera, él me apoyó y lo sigue haciendo desde donde se encuentre. (estoy segura que hubiéramos hecho experimentos con esto).

A mi mamá que siempre ha estado conmigo en todas mis decisiones por buenas o malas que sean y que por fin logra ver la tesis hecha realidad.

A MIS familias: Morales Soto, Rosales Trejo, Avendaño Morales, Peña Morales, Esquivel Trejo, Morales Abundez. (Abue, tías, tíos, primos, sobrinos, hermanas, cuñados), que nunca quitaron el dedo del renglón para que llegara a este cometido, y porque a pesar del tiempo transcurrido me tuvieron paciencia.

A la Lic. Lizbeth Silva por el apoyo brindado durante la elaboración de este trabajo.

Le agradezco al Ing. Jose Luis Pérez Báez, que nos dio una segunda oportunidad para formar parte de sus alumnos de tesis.

Al Ing. Benito Barranco por ser parte de este proyecto en un inicio.

Al lugar donde conocí por primera vez Asterisk, del que no tenía ni la menor idea de que en un futuro llegaría a ser mi tema de tesis: DGSCA (ahora TIC UNAM) con el Ing. Giovanni Nopal porque me dejó ser parte de su equipo y enseñarme las bases de lo que sé.

A mi primer jefe laboral que a pesar que mi conocimiento era de nivel básico, me dio la oportunidad de desarrollarme profesionalmente en este tema: Ing. Christian Cabrera.

*A Jacinto (el hermano que nunca tuve) por llevarme en este camino y hacer
que este proyecto llegara a su meta.*

A los sinodales por sus aportaciones.

*A toda la gente que he conocido y que me han dado de alguna u otra forma
parte de su experiencia.*

A Dios por permitirme estar en este punto.

Gracias.

Índice de figuras

1.1. FXS	33
1.2. FXO	33
1.3. FXS y FXO	34
1.4. Señalización analógica	35
1.5. Señalización E y M	36
1.6. Tono de descolgado	37
1.7. Conmutación	38
1.8. Timbrado	38
1.9. Loop Start	39
1.10. Ground Start	40
1.11. Señal Muestreada	42
1.12. Cuantificación	42
1.13. Trama E1	46
1.14. Señalización Canal Asociado	47
1.15. Señalización de Canal Común	51
1.16. Canales BRI	52
1.17. Canales PRI	52
1.18. SS7	53
1.19. Modo Asociado	54
1.20. Modo No Asociado	54
1.21. Modo Cuasi Asociado	54
1.22. Niveles SS7	55
1.23. Modelo OSI	58
1.24. Protocolos Modelo OSI	60

1.25. Modelo TCP/IP	61
1.26. Protocolos Modelo TCP/IP	64
1.27. Diferencias entre Modelo OSI y TCP/IP	65
1.28. Direcciones IP	69
1.29. Datagrama IP	69
1.30. Protocolo TCP	72
1.31. Protocolo UDP	73
2.1. Protocolo RTP	80
2.2. Encabezado RTCP Sender Report (SR)	83
2.3. Protocolo SIP	84
2.4. Sesión SIP	86
2.5. Mensajes SIP	87
2.6. Solicitudes o Métodos SIP	88
2.7. Respuestas SIP	89
2.8. Registro de un UA SIP	90
2.9. SIP 401 Unauthorized	90
2.10. Petición de credenciales	91
2.11. Respuesta OPTIONS	91
2.12. Respuesta 200 OK	92
2.13. Descripción del mensaje INVITE	93
2.14. Traza SIP con Wireshark	93
2.15. SDP	95
2.16. Protocolo H.323	98
2.17. Gatekeeper	100
2.18. Típica red H323	101
2.19. Establecimiento de llamada	103
2.20. Flujo de datos	103
2.21. Desconexión	104
2.22. Full Frame	105
2.23. Mini Frame	106
2.24. Topología	114

3.1. Sistema Operativo	117
3.2. GNU-Linux	118
3.3. Terminal o Shell	120
3.4. CAT	120
3.5. LS	121
3.6. CD	121
3.7. PWD	121
3.8. NANO	122
3.9. Asterisk Logotipo	123
3.10.DAHDI	124
3.11.Arquitectura de Asterisk	127
3.12.apt-get install build-essential.	132
3.13.Código de área	133
3.14.WGET	134
3.15.tar -zxf	134
3.16.rasterisk	135
3.17.Configuración de Extensión SIP (Template)	138
3.18.Configuración de Extensiones SIP	139
3.19.Configurando el softphone	139
3.20.Dialplan	144
3.21.Troncal SIP	146
3.22.Contexto dialout	147
3.23.IVR	148
3.24.Parámetros de Conferencias	150
3.25.Conferencia	151
3.26.Voicemail en el extensions.conf	153
3.27.Voicemail.conf	153
3.28.Costos	155
3.29.Telmex	156
3.30.Axtel	157
3.31.Cablevisión	158
3.32.Virtualphone	158

Índice de cuadros

1.1. Pulsos	30
1.2. Codificación	43
1.3. Velocidad de transmisión de la información	44
1.4. Señalización en R2	50
1.5. Puertos para UDP	74
2.1. Códecs y su velocidad en bits	110
2.2. Overhead	110
2.3. G.729 y su Overhead	111
2.4. G.711 y su Overhead	111
3.1. Comandos para el uso de nano	122
3.2. Versiones de Asterisk	125
3.3. Versiones de Asterisk (Continuación)	126
3.4. Comandos para compilar Asterisk	135
3.5. Simbología Asterisk	136
3.6. Opciones de configuración SIP	137
3.7. Opciones de configuración SIP (Continuación)	138
3.8. Aplicaciones en Asterisk	141
3.9. Aplicación DIAL	142
3.10. Aplicación DIAL (Continuación)	143
3.11. Parámetros de Voicemail	152

Índice general

Dedicatoria y agradecimientos de Jacinto	3
Dedicatoria y agradecimientos de Ana	5
Índice de figuras	7
Índice de cuadros	11
Objetivos	17
Planteamiento del problema	19
Justificación	21
Hipótesis	23
Introducción	25
1. Fundamentos Teóricos y Señalización	27
1.1. Origen de la Telefonía	27
1.1.1. Los principios de la transmisión de voz	28
1.1.2. Operación del sistema telefónico básico	29
1.2. Telefonía Análoga	31
1.2.1. La PSTN	31
1.2.1.1. FXS	33
1.2.1.2. FXO	33
1.2.2. Señalización analógica	34
1.2.2.1. Señalización E&M	35

1.2.2.2. Loop Start o Por corriente de bucle	36
1.2.2.3. Ground Start o Aterrizada	39
1.3. Telefonía Digital	40
1.3.1. Modulación por pulsos codificados (PCM)	41
1.3.2. Teorema de Nyquist	44
1.3.3. Enlaces Digitales	45
1.3.4. Señalización Digital	46
1.3.4.1. CAS o Señalización de Canal Asociado	47
1.3.4.2. CCS o Señalización de Canal Común	50
1.4. Modelo OSI	57
1.5. Modelo TCP/IP	61
1.5.1. IP(Protocolo de Internet)	65
1.5.2. TCP (Protocolo de Control de Transmisión)	69
1.5.3. UDP (Protocolo de Datagrama de Usuario)	72
2. Entorno de la Telefonía sobre IP	75
2.1. Conceptos de Voz por IP	75
2.2. Señalización, Transporte y audio	77
2.2.1. Protocolos de Transporte	78
2.2.1.1. RTP (Protocolo de Transporte en Tiempo Real)	78
2.2.1.2. RTCP(Protocolo de Control de RTP)	80
2.2.2. Protocolos de Señalización	83
2.2.2.1. SIP (Protocolo de Inicio de Sesión)	84
2.2.2.2. H.323	97
2.2.2.3. IAX2 (Inter-Asterisk Exchange2)	101
2.2.3. Codificación y decodificación de voz (Códex)	106
2.2.4. Ancho de banda y códex	109
2.3. Elementos de Red	112
3. Estándares abiertos y código libre para Telefonía IP	115
3.1. Código libre	115
3.2. Sistemas operativos.	116

3.2.1. Linux	117
3.2.1.1. Estructura de directorios	118
3.2.1.2. Shell y comandos básicos	119
3.3. Asterisk	123
3.3.1. Arquitectura	126
3.3.2. Canales en Asterisk	128
3.3.3. Estructura de directorios	128
3.4. Implementación. Procedimiento para la instalación de Asterisk 1.8.15 . . .	131
3.4.0.1. Simbología Asterisk.	136
3.4.0.2. Configuración de la plataforma.	136
3.4.1. Recursos Adicionales	148
3.5. Comparativo	154
Conclusiones	159
Recomendaciones	161
Anexo	163
3.6. Plan de Marcación	163
3.7. Plan de Interconexión	164
3.8. Plan de Señalización	166
Bibliografía	169

Objetivos

Objetivo General

Proponer una solución de conmutadores basada en OpenSource para pequeñas y medianas empresas y de esta manera se ahorren en costos pero teniendo los beneficios de escalación que el mismo software ofrece. Porque a pesar de ser gratuito se adapta a las nuevas tendencias tecnológicas como dispositivos móviles, marcadores predictivos, mensajería instantánea, etc..Compitiendo a la par con marcas establecidas en el mercado pero licenciadas.

Objetivo Particular

- Dar a conocer la evolución que ha tenido la telefonía a través del tiempo, y sus avances, aplicaciones y adaptación a las distintas tecnologías de la actualidad.
- Explicar brevemente aspectos teóricos de la telefonía IP.
- Utilizar software libre (Debian, Asterisk) para la instalación del conmutador y algunas de sus aplicaciones.
- Mostrar un comparativo de costos con respecto a la telefonía tradicional vs Telefonía sobre IP.

Planteamiento del problema

Debido a su limitación económica, las pequeñas y medianas empresas casi no invierten en su infraestructura de voz ya que sólo conocen las marcas que predominan en el mercado (Avaya, CISCO, Alcatel, etc.), donde su costo de inversión es demasiado alto y la mayoría de sus aplicativos se venden por separado, sin considerar que es un sistema cerrado que no ofrece la posibilidad de ajustarlo a la medida de sus requerimientos por lo que buscan alternativas comunes y baratas (Panasonic, Telmex, etc.), pero que tampoco ofrecen mayores beneficios y por ende su aprovechamiento es mínimo.

Justificación

Hace apenas unos años, el hablar de Telefonía sobre IP solo era para grandes empresas, sin embargo, en la actualidad se han venido dando diferentes formas de poder implementarlo para todo tipo de usuarios.

Su implementación es variada y puede abarcar desde una empresa de grandes dimensiones, en la que se instala el conmutador en conjunto con cualquier tecnología como SIP, líneas digitales, líneas análogas, hasta haciéndolo de manera virtual, la cual es más barata y no necesita mas que un enlace de internet y de teléfonos o adaptadores telefónicos para poder conectarlo, de esta manera esta tecnología llega al alcance de pequeñas empresas, las que por sus recursos limitados, obtienen casi los mismos beneficios que como si tuvieran un conmutador físico.

El alcance de este tipo de plataforma es escalable porque de esta manera se adecua a trabajar con todas las tecnologías existentes, por ejemplo, el poder dejar un mensaje de voz desde tu línea telefónica y recibirlo en tu correo electrónico y consultarlo en diferentes dispositivos (celular, tabletas, pc) y de esta manera se demuestra la convivencia entre las distintas tecnologías, siendo completamente transparente para el usuario final y teniendo un máximo aprovechamiento de los diferentes recursos.

Esta tesis se realiza en equipo ya que ambos integrantes estamos involucrados en el mismo entorno de trabajo del Código libre (Asterisk), por lo que decidimos realizar este proyecto en conjunto, aportando cada quien sus conocimientos que fueron adquiridos en el área.

Hipótesis

Los avances de la tecnología han permitido que la telefonía vaya creciendo a su mismo ritmo, teniendo más demanda en distintos campos de aplicación como mensajes de texto, sistemas de acceso, vídeo y audio conferencias, etc., y abarcando distintos mercados, como call-center, casa-habitación, PyMES¹, oficinas virtuales, y corporativos, adaptándose y en muchas ocasiones sustituyendo a las tecnologías ya existentes.

Por lo tanto vemos que Asterisk es una solución adaptable, escalable y accesible en cualquier tipo de esquema que se requiera y que va a abrir un mercado para soluciones tanto en instalación como en soporte a problemas que la misma aplicación vaya presentando.

¹Pequeñas y Medianas Empresas.

Introducción

La comunicación ha ido evolucionando a través de los años; específicamente la telefonía es algo que ha venido creciendo de manera considerable, tanto en el aumento de usuarios como el mejoramiento de tecnologías. Casi toda la gente aún utiliza líneas de telefonía análogas, sin embargo estamos en una etapa donde el auge sobre el protocolo de Internet esta creciendo de manera desmesurable, se ha ido agregando la Voz sobre IP (VoIP), que no necesariamente es telefonía, pero que si cumple con ciertos requisitos, como aplicaciones que corren sobre la web, chat, *facebook*, *skype*.

La telefonía es algo que no se pudo quedar atrás y hoy en día también puede agregarse en el plus de la Voz por IP trayendo como principales bondades, por ejemplo, el abaratamiento de llamadas de larga distancia, o el poder transmitir vídeo y voz al mismo tiempo, ya que estos son enviados como datos a través de la nube, lo que reduce el costo de la comunicación en cualquier punto remoto.

Es importante indicar la diferencia entre Voz por IP y Telefonía sobre IP. La primera es el conjunto de normas, dispositivos, protocolos, o en pocas palabras, la tecnología que permite la transmisión de las voz sobre el Protocolo de Internet (IP), mientras que lo segundo es el conjunto de nuevas funcionalidades de la telefonía, esto significa que la telefonía tradicional finalmente puede llegar a ofrecer servicios "extras" gracias a poder portar la voz sobre el protocolo IP en redes de datos.

En un principio para poder establecer esta comunicación lo difícil era garantizar la calidad en una llamada, ya que la voz era tratada como datos, por lo que en caso de tener pérdidas de paquetes la voz es algo que no se puede retransmitir, trayendo con esto mala calidad de audio en cualquiera de los puntos remotos, pero al paso de los años esto se ha venido mejorando ya que en las redes que manejan este tipo de tecnología

se aplica el QoS (Calidad de Servicio), lo que da prioridad a cualquier paquete de voz antes que de datos.

Actualmente existen muchas compañías que trabajan con este tipo de tecnología, tanto proveedores de servicio como proveedores de equipos para poder trabajar con la misma, esto hace que el beneficio sea muy alto, principalmente a nivel empresarial, ya que acerca más a sus sucursales de diferentes estados o países de manera económica.

En el capítulo 1 se tratará una reseña de la telefonía a través del tiempo, así como los elementos que conformaban en un inicio al aparato telefónico describiendo sus funciones, se mencionarán las dos formas de telefonía que hasta nuestros tiempos existe (análoga y digital), y los conceptos teóricos que conforman el estándar IP.

En el capítulo 2 se adentrará en lo que es la tecnología de Voz por IP, abarcando sus componentes y protocolos que lo conforman.

En el capítulo 3 se desarrollará la implementación del conmutador, tanto en forma teórica como práctica, y se realizará un comparativo de costos entre 4 proveedores de telefonía como Axtel, Telmex, Cablevisión y Virtualphone, este último siendo de tecnología SIP.

Capítulo 1

Fundamentos Teóricos y Señalización

1.1. Origen de la Telefonía

La palabra telefonía se define como la ciencia que tiene por objetivo la transmisión de sonidos a distancia, donde se incluyen todos los medios y procedimientos empleados para la transmisión, transporte y recepción de sonidos.

La historia del teléfono se remonta hacia 1854, cuando Charles Bourseul de origen francés construyó un aparato, al que se tomaría como el primer predecesor del teléfono. Posteriormente, en 1861 un segundo aparato fue construido por el alemán Philipp Reis, pero ninguno de ellos fue exitoso. La invención del teléfono como tal, se le atribuye al científico estadounidense de origen escocés, Alexander Graham Bell, quien lo hizo una realidad el 10 de marzo de 1876.

En junio de 2002, el Congreso de Estados Unidos, le adjudicó el mérito de dicha invención al también estadounidense pero de origen italiano, Antonio Meucci, quien en 1860 había descubierto que la transformación de las vibraciones sonoras en impulsos eléctricos, lograban la transmisión de la voz a través de un cable.

En 1871, en Nueva York, Meucci solicitó la patente por su invento, al que llamó *teletrófono*; pero por falta de dinero no pudo renovar su solicitud de patente. En 1874 presentó su prototipo a la compañía de telégrafos *Western Union*, la que en un principio no tuvo el menor interés en su aparato. Pero dos años después, Meucci se enteró que Graham

Bell inventó el teléfono, quien era patrocinado precisamente por Western Union, por lo que se inició con una batalla legal y, aunque un tribunal de Nueva York le dio la razón en 1887, Meucci no pudo reclamar la parte de los beneficios económicos porque su patente del *teletrófono* ya había caducado.

Gracias a la patente, Bell pudo hacer de la idea del teléfono un negocio rentable y tiene el mérito de haber desarrollado la idea y convertirla en algo práctico para la sociedad.

1.1.1. Los principios de la transmisión de voz

La voz humana está compuesta por ondas acústicas que viajan a través del aire a la velocidad del sonido, (340 m/s). Pero no se puede comunicar fácilmente con puntos distantes porque se va atenuando rápidamente, perdiendo energía a medida que va viajando.

La voz humana es una onda acústica pero existen otros tipos de ondas como las eléctricas, éstas últimas son de una naturaleza diferente a las ondas acústicas, y pueden ser transmitidas a través de un conductor metálico viajando a la velocidad de la luz, es decir aproximadamente 300,000 Km/s. A este tipo de ondas se le puede controlar la atenuación y hacer que recorran grandes distancias.

Con lo anterior, a mediados del siglo 19, muchos persiguieron la idea de transformar las ondas acústicas en ondas eléctricas para luego poder transmitir las a grandes distancias a través de conductores metálicos. La cuestión radicaba en inventar un dispositivo para hacer dicha transformación. Este dispositivo, conocido como *micrófono* en nuestros días es una parte importante de cualquier aparato telefónico.

Una característica de la voz humana es que las cuerdas vocales modulan la voz en un amplio espectro de frecuencias que van de los graves a los agudos en un rango aproximado de 200Hz a 7kHz. En la actualidad para que se pueda transmitir una voz que sea entendible no es necesario transmitir todas las frecuencias sino un rango que sea mucho menor, esto es de 400Hz a 4kHz, que aunque se distorsione un poco la voz es posible entenderla.

1.1.2. Operación del sistema telefónico básico

Aunque hoy en día surgen mas formas de comunicación las cuales resultan ser más eficientes y novedosas, el teléfono a diferencia del correo o el telégrafo, no ha desaparecido, por el contrario ha sufrido cambios en su implementación, pero es necesario saber cuáles son los principios del teléfono y sus componentes para entender la evolución y adaptación con la tecnología actual.

El teléfono está constituido por varias partes básicas, entre las que se deben destacar:

1. *El Microteléfono*, es la pieza que contiene tanto el *micrófono* (elemento transmisor), como el *auricular* (elemento receptor), esta pieza se desprende del resto del equipo, sin embargo puede o no estar comunicado por un cable. Cuando el usuario levanta el *microteléfono* se inicia el proceso de comunicación entre el aparato y la central telefónica a donde se encuentra conectado.

Quitando el peso del *microteléfono*, el "gancho conmutador" se levanta, y como tenía apagado el circuito eléctrico del teléfono, esta acción hace que la corriente eléctrica comience a circular por dicho circuito; entonces el aparato recibe de la central la señal que le indica al usuario que puede marcar al número con el que desea comunicarse. Luego el teléfono transmite a la central las señales definidas por los dígitos marcados por medio de la unidad de disco o teclado, dependiendo del tipo de aparato que se use.

- *Micrófono o Transmisor*. El *micrófono* es un elemento de suma importancia, ya que aquí se da todo un proceso de transformación de señal, es un transductor que convierte, codifica y modula la señal acústica de la voz en una señal eléctrica. Existen muchos tipos de micrófonos, uno de los mas usados consta de una placa llena de granitos de carbón entre dos placas metálicas. Una de las placas es una membrana que vibra con las presiones de las ondas sonoras generadas por la voz, comprimiéndolas dependiendo de la intensidad y frecuencia de la voz, ocasionando una variación en la señal eléctrica.

Otro tipo consiste de una bobina de hilo de cobre enrollado sobre núcleo de ferrita, sujetado a la vez a un diafragma el cual vibra con la presión de la onda emitida por la voz, mandando una señal eléctrica al medio.

- *Auricular o Receptor*. Muy parecido al *micrófono*, ya que cuenta con un nú-

cleo magnético enrollado por hilo de cobre (bobina), de igual manera tiene una membrana, la cual vibra a causa de la frecuencia e intensidad de la corriente que pasa, haciendo variar el campo magnético atrayendo o repeliendo la membrana convirtiendo, decodificando y demodulando la señal eléctrica en acústica para el entendimiento humano

2. *Unidad de Marcación.* Para la marcación existen dos tipos de discado, por pulsos y otro por tonos.

- *Tonos.* El discado de tonos se lograba gracias a que el teléfono poseía un disco rotativo, el cual tenía 10 agujeros para los dedos que correspondían del 0 al 9, estos abrían y cerraban el circuito, dependiendo el número a marcar ya que el disco giraba más o menos dependiendo cual fuera el número, de esta manera el conmutador interpretaba cual era la marcación.
- *Pulsos.* La marcación por pulsos se da gracias a la implementación de un teclado en donde se suman las frecuencias dadas en Hz (*Cuadro 1.1*), por ejemplo si alguien marca el numero dos, tendría que sumar 1 336 Hz + 697Hz dando un total de 2 033Hz.

Cuadro 1.1: Pulsos

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

Como nota, podemos mencionar que la transmisión de un dígito tarda 0.7 s, mientras que por el tono tarda 1.5 s.

3. *Timbre.* Es el elemento que produce una señal auditiva fuerte para avisar que hay una llamada entrante. Este se activa con una corriente alterna proveniente de la

central telefónica a la que está conectado el aparato. La señal eléctrica se origina en esta central cuando su equipo conmutador recibe la señal procedente del aparato que inició la comunicación.

El teléfono tiene una corriente continua siempre que se encuentra colgado, cuando recibe una llamada también recibe 75V de tensión a una frecuencia de 25 Hz y es cuando se escucha el timbre del teléfono.

4. *Circuito de regulación de la transmisión.* Su función consiste en mantener la intensidad acústica dentro de límites predeterminados para que la calidad de la comunicación sea la adecuada. Se trata de varios componentes eléctricos o electrónicos, como condensadores, varistores y resistencias. Este circuito opera automáticamente, permitiendo que las señales en el teléfono siempre tengan una intensidad adecuada para escuchar normalmente, en forma independiente de la intensidad que provenga de la línea telefónica.

1.2. Telefonía Análoga

A través del tiempo el ser humano siempre ha tenido la necesidad de comunicarse y han existido diferentes formas de hacerlo, por ejemplo el correo que se daba por medio de mensajeros, después surgió el telégrafo el cual se utilizaba mediante el envío y recepción de señales analógicas a través de la corriente eléctrica, y por supuesto el teléfono, por mencionar solo algunos.

1.2.1. La PSTN

Los antecedentes de la *PSTN o Red Pública Telefónica Conmutada*)¹ comienzan por el año 1876, con la primer llamada que se hizo sobre un circuito conocido como Ring-Down, que consistía en un circuito de dos alambres y en cada extremo de los mismos, un teléfono, por lo que no existía una forma de discar para realizar la llamada ya que el que primero descolgara comenzaba con la conversación; a esto se le denomina *half-duplex* ya que solo una persona podía hablar a la vez.

Este tipo de circuitos comenzaba a crecer pero al mismo tiempo generaba problemas

¹También conocida como RTB o Red Telefónica Básica

por el enlace físico que se requería, tanto en costos como en ubicación, por lo que se optó por conectar a un switch el cable del usuario para que se conmutara la llamada. De esta manera, existían unos cuantos usuarios detrás del switch que requerían la conexión para el destinatario, así que con un solo operador era suficiente para cerrar el circuito. La demanda de usuarios iba creciendo que los operadores no se daban abasto, por lo cual, se crearon los conceptos: *Llamada telefónica* y *Conmutador telefónico*, lo que significa: marcar un identificador numérico y enviar la voz a un switch automatizado que conmute la llamada al destino, esto implicó que el sistema trabajara más rápido y se cayó en el concepto *Full-duplex*, que significa que se puede estar hablando en ambos extremos y al mismo tiempo.

La estructura de la PSTN consiste en 4 partes fundamentales:

- *Equipo terminal*. El Teléfono, el identificador de llamadas, la contestadora, o las modificaciones dentro del mismo.
- *La transmisión*. Esto depende si la información es sólo voz, datos o ambos.
- *La conmutación*. Se refiere al modo de conexión de los enlaces y establecer el camino directo para la comunicación.
- *Señalización*. Se controla la red telefónica y se administran las conexiones.

La PSTN es la línea que se tiene en los hogares o las empresas, cuyo uso se ha enfocado fundamentalmente hacia las comunicaciones mediante voz, aunque también se usan para la transmisión de datos como fax, Internet, etc.

Cada línea PSTN tiene asignada una numeración específica, algo así como su dirección telefónica y está físicamente construida por dos hilos metálicos, conocidos como *par de cobre*, que se extienden desde la central telefónica hasta la instalación del usuario, (este circuito es conocido también como *bucle local*). Cada central atiende las líneas de sus usuarios de una determinada área geográfica (a éste se le denomina *Conmutador Primario*). A su vez, las centrales telefónicas están unidas entre sí por sistemas más complejos y basados en tecnología digital (al que se le conoce como *Tandem*). Esta unión de centrales constituye el sistema telefónico nacional que a su vez está enlazado con los restantes del mundo (llamado *Tandem Superior*).

1.2.1.1. FXS

La interfaz *Foreign eXchange Subscriber* o *FXS* es lo que está situado al otro lado de una línea telefónica tradicional. Un *FXS* envía el tono de marcado, la señal de llamada que hace sonar a los teléfonos. En líneas analógicas, un *FXS* utiliza alrededor de 48 voltios DC para alimentar al teléfono durante la conversación y hasta 80 voltios AC (20 Hz) cuando genera el tono de llamada (ring). Los puertos *FXS* (Figura 1.1) son por lo tanto los encargados de:

- Proporcionar tono de marcado.
- Suministrar tensión y corriente al dispositivo final.



Figura 1.1: FXS

1.2.1.2. FXO

La interfaz *Foreign eXchange Office* o *FXO* es cualquier dispositivo que, desde el punto de vista de la central telefónica, actúa como un teléfono tradicional. Un *FXO* debe ser capaz de aceptar señales de llamada o ring, ponerse en estado de colgado o descolgado, y enviar y recibir señales de voz. Se puede decir entonces que un *FXO* (Figura 1.2) es como un teléfono o cualquier otro dispositivo que suena (como una máquina de fax o un módem).



Figura 1.2: FXO

A modo de resumen (Figura 1.3):

- Un *FXS* necesita estar conectado a un *FXO* (como una línea telefónica necesita estar conectada a un teléfono) o viceversa.
- Un *FXS* suministra energía (elemento activo) a un teléfono *FXO* (elemento pasivo)

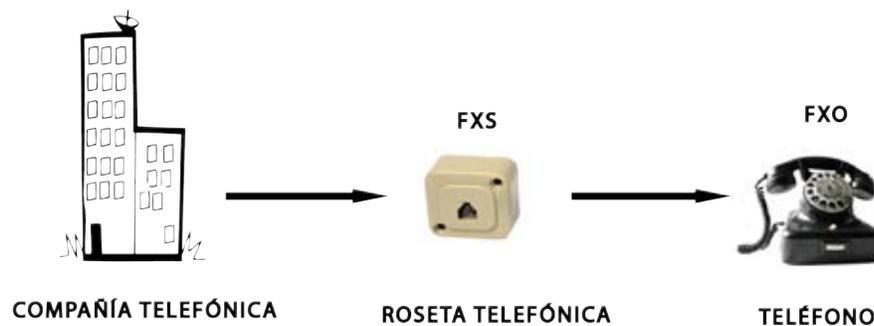


Figura 1.3: FXS y FXO

1.2.2. Señalización analógica

Las señales analógicas son señales que se generan a través de campos electromagnéticos; en la vida cotidiana se pueden percibir señales muy comunes a diario como son la luz, el sonido o algún tipo de energía, pero en el caso de la voz una señal analógica es una señal de audio, donde los seres humanos pueden escuchar entre los 20 hz a los 20 Khz.

La voz puede viajar utilizando como medio el aire, sin embargo la distancia que cubre es muy limitada; para poder ampliar el rango es necesario ocupar un transductor el cual es el encargado de convertir la señal de la voz de entrada a otro tipo de señal, en este caso eléctrica.

La señal analógica es representada por una onda senoidal la cual puede ir variando de acuerdo al sonido con respecto al tiempo.

Cuando se quiere establecer una llamada telefónica en un correcto orden, se manejan ciertos protocolos de señalización que permiten intercambiar la información entre el abo-

nado y la Central telefónica o el PBX².

Cuando se usa una línea telefónica se intercambian un conjunto de "señales". Las señales sirven para ofrecer información del estado de la llamada al usuario. Algunas de esas señales son el tono de marcado o el tono de línea ocupada. Estas señales se transmiten entre el FXS y el FXO haciendo uso de un protocolo conocido como "señalización".

Los protocolos de señalización sirven para transmitir información del estado en que se encuentre el canal de comunicaciones: desconectado, timbrado, etc.. información de control y otra información como los tonos que se envían, el identificador de llamadas, etc. (Figura 1.4)

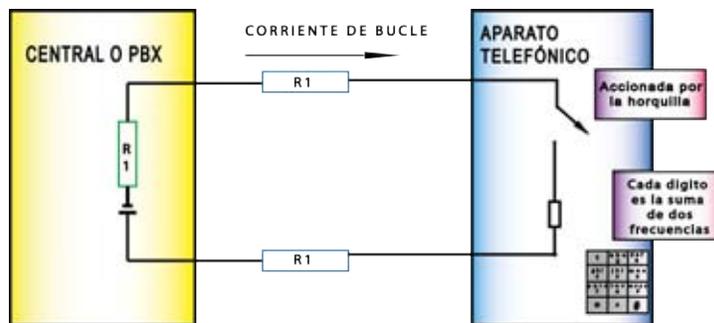


Figura 1.4: Señalización analógica

1.2.2.1. Señalización E&M

Es un protocolo de enlace entre centrales y equipos o entre las propias centrales. Este tipo de señalización es el más utilizado para proveer troncalización analógica. Se indican los cambios de estado de *On-Hook* (colgado) y *Off-Hook* (descolgado), con ello se disminuye la probabilidad de que una línea troncal de dos vías sea tomada al mismo tiempo en ambos puntos finales. Utiliza un par extra de cables en las líneas troncales local y remota, uno de ellos es nombrado **E** mientras que el otro es llamado **M**. El cable **E** recibe señales mientras que **M** las transmite. (Figura 1.5)

²Private Branch Exchange: cualquier central telefónica conectada directamente a la red pública de telefonía (PSTN)



Figura 1.5: Señalización E y M

1.2.2.2. Loop Start o Por corriente de bucle

Loop start signaling o Señalización por corriente de bucle, es quizás la más antigua y la más conocida. En este tipo de señalización se indica en la red las condiciones de: Colgado, Descolgado, Marcación, Conmutación, Timbrado y Conversación. (Figura 1.9)

A continuación se describirán cada uno de los estados:

La Central o PBX conecta una batería de alimentación normalmente de -48V al extremo del par de cobre que viene siendo el teléfono, desde esta conexión se forma un bucle de corriente. En el teléfono se encuentra una llave que se acciona por una horquilla la que puede abrir o cerrar este bucle de corriente.

- *Colgado (On-Hook)*. Cuando el teléfono se encuentra en este estado, el bucle se encuentra abierto, es decir que parece que no estuviera conectado y no fluyera corriente por la línea.
- *Descolgado (Off-Hook)*. Cuando se descuelga el auricular, el aparato telefónico envía una señal a la Central, y el bucle ahora cerrado permite el paso de la corriente. Esto significa que de manera interna el teléfono conecta entre sí el par de cobre (llamados *TIP* y *RING* por nombre de las partes del conector) a través de una resistencia eléctrica. En la Central se dispone de un sensor de corriente para detectar este estado.

El tono tiene una frecuencia de 425Hz, una presencia continua durante 10 segundos después de descolgar el teléfono. (Figura 1.6)



Figura 1.6: Tono de descolgado

- **Marcación.** Cuando la Central se da cuenta de la señal de descolgado, manda un tono de marcado al teléfono, dando la indicación de que ya puede digitar el destino. Al pulsar un dígito, el teléfono genera una señal de audio compuesta por un par de frecuencias *DTMF (Dual-Tone Multi-Frequency)* mezclados y enviados por el par de cobre (Véase cuadro 1.1).

Estas frecuencias son traducidas a números por la Central en donde ella va a escoger una ruta para llegar al destino, la cual puede ser grande o pequeña ya que el destino puede estar en una central primaria diferente, si fuera el caso la central está obligada a conectarse entre centrales de niveles superiores hasta alcanzar a la primaria donde se encuentra el destinatario.

Existen una serie de números que deben ser marcados al inicio de una llamada, y esto es de acuerdo a si la llamada es de Larga Distancia Nacional o Internacional. Para una llamada nacional hablando de México, primero se marca el código 01 después el código de área correspondiente a la zona, para después marcar el número destino. En el caso de larga distancia internacional, primero se digita 001, después el código del país, después el código de área de la localidad y finalmente el número destino. En caso de ser celulares, si son de manera local se debe digitar 044 y después el número a diez dígitos; en caso de ser de larga distancia, este debe ser 045 y los diez dígitos.

También existen otros códigos que sirven para activar o desactivar servicios por parte de la Central si es que se han contratado.

- **Conmutación.** Ya que han sido recibidos los dígitos por la Central, si la línea del destino se encuentra ocupada, ella lo detecta y al abonado le envía el llamado *Tono de Ocupado*, este se distingue porque el tono va seguido de un segundo de

silencio. Si por el contrario, el destino se encuentra disponible, el tono que escuchará el abonado es de aproximadamente un segundo, seguido de un silencio que dura aproximadamente 4 segundos, y la Central le enviará una señal al destino indicando que tiene una llamada. (Figura 1.7)

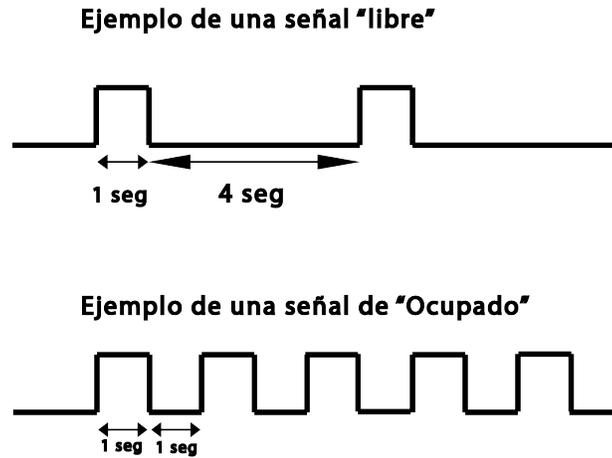


Figura 1.7: Conmutación

- *Timbrado.* Cuando la Central hace llegar la llamada al destino, se lo hará notar con un timbrado o ringado, aunque también puede ser de manera visual. La señal de timbrado es de una onda Sinusoidal de 20 o 25 Hz y 90 Voltios de amplitud. (Figura 1.8)

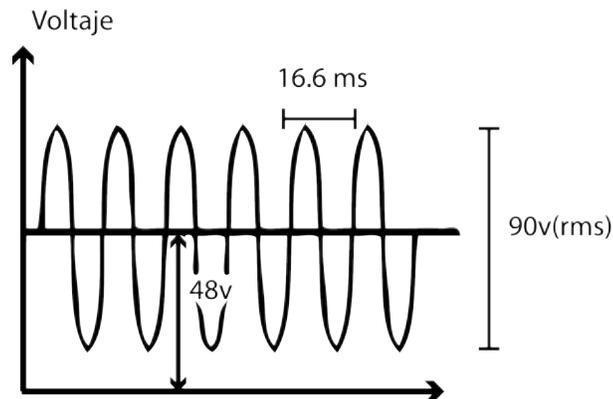


Figura 1.8: Timbrado

- *Conversación.* Cuando se establece la llamada entre ambos sentidos, se cierra el circuito, vuelve al estado de descolgado, por lo que la Central estará avisada que el destino decidió contestar y así está completa la conexión.

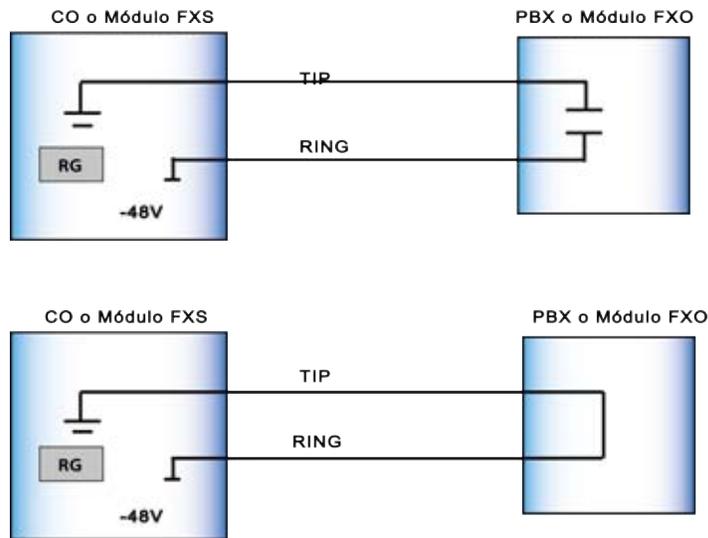


Figura 1.9: Loop Start

1.2.2.3. Ground Start o Aterrizada

Esta señalización se utiliza en el circuito local. Aquí se requiere de detectar un tono de marcado proveniente de uno de los pares de cobre, el Ring, que se encuentra aterrizado. Una vez detectado el contacto Ground Start se abre y el resto de la línea es de manera instantánea activada.

Los pares de cobre Tip y Ring se encuentran desconectados de la tierra y el PBX está constantemente revisando que el Tip se aterrice así como la Central que lo haga el Ring. El conmutador aterriza a Ring para que la Central esté avisada de que hay una llamada entrante. Una vez detectada, la Central hace lo mismo: aterrizar a Tip; el conmutador encuentra al cable Tip aterrizado y responde desactivando la línea a tierra para unirla con Tip.

Si hay una llamada entrante, la Central manda por el cable Ring la señal de timbrado, ésta es de una frecuencia de 20 Hz y de 90V de corriente directa. El PBX revisa que Tip se encuentre aterrizada y que reciba la señal de timbrado; cuando se tengan los recursos entonces el PBX cerrará el lazo entre los pares de cobre. La central deja de mandar el

flujo de corriente y apaga la señal de timbrado. (Figura 1.10)

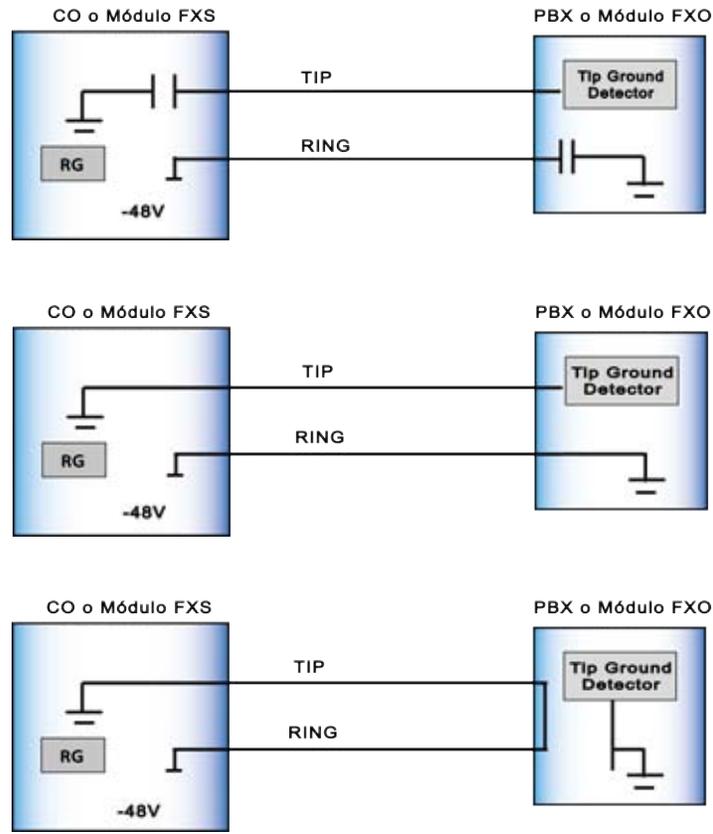


Figura 1.10: Ground Start

1.3. Telefonía Digital

Un sistema analógico utiliza una cantidad física, como el voltaje para representar el comportamiento de otro sistema físico. Las propiedades de esta representación son explotadas para almacenar, transformar, duplicar o amplificar el fenómeno original. Las señales utilizadas en un sistema analógico son continuas en el tiempo y en su amplitud (es decir, en su magnitud), por lo que pueden tomar un número potencialmente infinito de valores.

Para que una onda analógica llegue a otro extremo sin que se pierda o que llegue con un volumen bajo, con estática, entre otros problemas, es necesario muestrear las características de la forma de onda original, almacenar la información medida y enviar esos datos al punto final. Entonces, en el extremo receptor, se puede usar la información transmitida

para generar una señal de audio completamente nueva que tenga las mismas características que la original.

1.3.1. Modulación por pulsos codificados (PCM)

Es el método que permite transformar una señal analógica (la voz) en una secuencia de bits, es decir a una señal digital. Entre la amplitud máxima y mínima de la señal se definen 16 niveles (de 0 a 15) y estos se codifican como números binarios (0000, 0001.. 1111). Para este proceso, se requieren de 4 pasos:

1. *Muestreo*. Consiste en tomar muestras de la señal analógica "n" veces por segundo, es decir, convertir una señal analógica continua a una señal discreta en el tiempo. (Figura 1.11)

El teorema de *Nyquist* dice que es suficiente con tomar muestras de la señal de voz al doble de la frecuencia más alta de la misma, esto es para que la señal muestreada contenga la misma información que la señal original. Debido a que la frecuencia más alta en una señal de voz humana es aproximadamente 4 kHz, la frecuencia de muestreo utilizada en un canal de voz es de 8kHz, esto significa que se toma una muestra cada $125 \mu s$.

Esto es:

$f_M = 4 \text{ kHz}$, frecuencia máxima

$$T = \frac{1}{2} \times 4 = 125 \mu s, \text{ intervalo de muestreo}$$

2. *Cuantificación*. En esta etapa se le asigna un valor cuántico a cada muestra de acuerdo a la amplitud de la señal. (Figura 1.12)

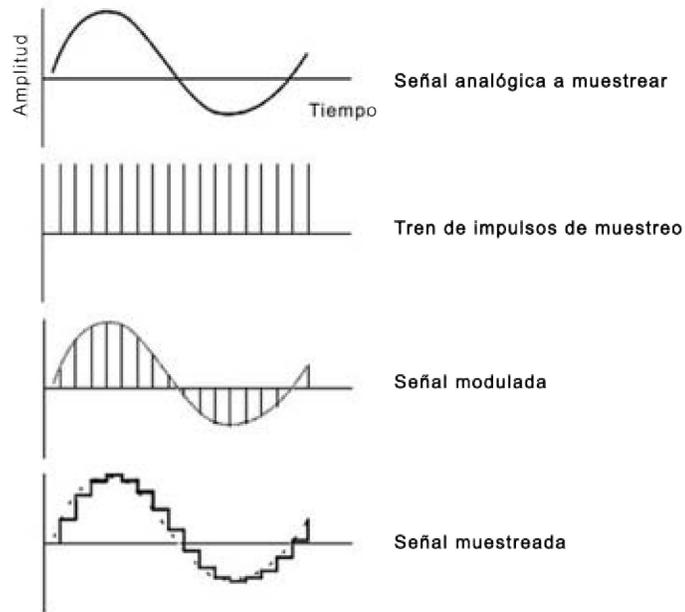


Figura 1.11: Señal Muestreada

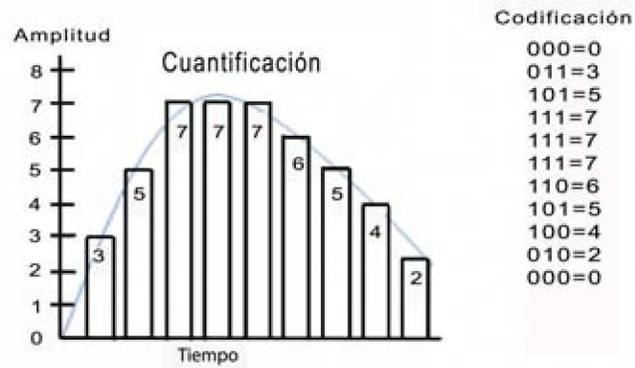


Figura 1.12: Cuantificación

3. *Codificación.* La codificación permite optimizar el canal de comunicación, debido a que envía la mayor cantidad de información por un solo canal de voz con la mínima pérdida de calidad. Existen diferentes técnicas para codificar señales cuantificadas, pero las dos técnicas más comunes utilizadas en la conversión binaria PCM son:

- μ -law, utilizada en Estados Unidos, Canadá y Japón.
- a-law.

En este proceso, a cada nivel se le asigna una codificación digital de bits, se toman los suficientes bits para representar 128 niveles o 256 niveles.(Cuadro 1.2)

Para representar 128 niveles se requiere de 7 bits, es decir: $2^7 = 128$

Para representar 256 niveles se requiere de 8 bits, es decir: $2^8 = 256$

Cuadro 1.2: Codificación

NIVEL	CODIFICACIÓN 7 BITS	CODIFICACIÓN 8 BITS
0	0000000	00000000
1	0000001	00000001
2	0000010	00000010
...
127	1111111	01111111
128		10000000
129		10000001
...		...
254		11111110
255		11111111

Para obtener la velocidad de transmisión de la información se deben de tomar en cuenta los siguientes parámetros:(Cuadro 1.3)

Cuadro 1.3: Velocidad de transmisión de la información

# de bits para representar cada muestra	x	Frecuencia de muestreo	=	Velocidad de transmisión
8bits	x	8KHz	=	64kbits/seg
7bits	x	8KHz	=	56kbits/seg

Después del proceso de codificación, la señal digital puede ser transmitida o comprimida antes de su transmisión.

4. *Compresión*. Es utilizada en las nuevas tecnologías de voz, es el proceso en el cual se transfiere o almacena información reduciendo la cantidad de bits, esto tiene como ventaja la reducción del tiempo de transmisión de la voz. Entre mayor comprimida esté la señal de voz, se pierde más la fidelidad de la voz en el receptor.

1.3.2. Teorema de Nyquist

En 1928 *Henry Nyquist*, un ingeniero Suizo que trabajaba para *AT&T*, resolvió el dilema de cuánto es necesario muestrear una señal como mínimo para después poder reconstruirla de forma exacta a la original.

El teorema que propuso dice que "*como mínimo se necesita el doble de ancho de banda como frecuencia de muestreo*"³. Esto queda reflejado de mejor manera con la siguiente expresión.

$$f_m > 2BW_s$$

Por lo tanto, para que la voz humana sea entendible es suficiente con transmitir un rango de frecuencia de entre 400Hz y 4kHz, entonces como mínimo debemos de muestrear al doble de la frecuencia mayor, es decir a 8kHz.

³Landívar, Edgar, Comunicaciones Unificadas con Elastix. Volumen 1

1.3.3. Enlaces Digitales

Un enlace digital es una interfaz lógica o física que puede tener múltiples interfaces lógicas y es conectada a un solo destino, los enlaces digitales comúnmente utilizados son tres:

- *E1*: Este tipo de circuitos utiliza *TDM*⁴ para transmitir los paquetes, utiliza *CAS*⁵ o *CCS*.⁶ Son utilizados en Europa, Asia, Sudamérica y América Central.
- *T1*: Transmite los paquetes de voz utilizando *TDM* y usando señalización *CAS*.
- *ISDN*: Es un sistema telefónico por circuitos conmutados y utiliza *CCS*. Cuenta con tres tipos de circuitos digitales:
 1. *BRI*⁷
 2. *T1 PRI*⁸
 3. *E1 PRI*

Circuitos T-carrier y E-carrier

DS-0 es un canal digital de 64Kbit/s. Un *DS-0* es por tanto una medida de canal estándar que sirve para definir múltiplos mayores como los circuitos que se mencionaran a continuación.

- **T1**

Los circuitos *T-carrier* (*portadora-T*) fueron diseñados como nomenclatura para circuitos digitales multiplexados. El más conocido de los circuitos *T-carrier* es el popular *T1*. Éste es un circuito digital compuesto de 24 *DS-0*'s o *time slots* de 64 kbps, tráfico 1.544 Mbit/s. Cuenta con un bit más para sincronización de multitrama,

⁴ Multiplexación por División de Tiempo

⁵ Señalización por Canal Asociado

⁶ Señalización por Canal Común

⁷ Basic Rate Interface o Servicio Básico

⁸ Primary Rate Interface o Servicio Primario

lo cual da un total de 193 bits por cada enlace T1.

Luego de los T1's se tienen múltiplos mayores como T2, T3, T4 y T5.

Resumiendo: una línea dedicada T1 proporciona 1.544 mbps que TDM puede dividir en 24 canales por separado, cada uno de 64kbps.

■ E1

Los enlaces *E1* están formados por 32 canales o time slots de 64 kbps, es decir que cada enlace tiene un ancho de banda de 2.048 Mbps. Utiliza una señalización fuera de banda, es decir que utiliza 2 canales completos para control y señalización, el canal 1 es para sincronización de las tramas y el 17 para señalización. (Figura 1.13)

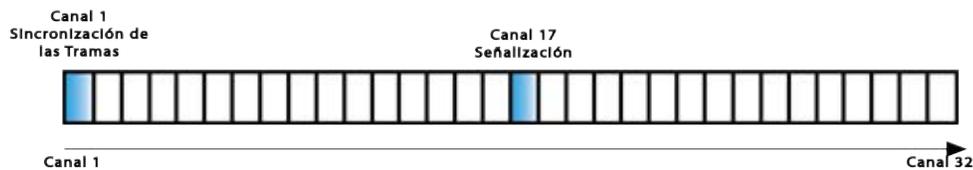


Figura 1.13: Trama E1

Resumendo: Una conexión E1 de un ancho de banda de 2.048 mbps es dividida en 32 canales de 64kbps cada uno.

1.3.4. Señalización Digital

Al igual que en la telefonía analógica que maneja un cierto tipo de protocolos de señalización, lo existe para la digital, y permite establecer, procesar, supervisar y finalizar la comunicación entre dos terminales.

Los protocolos de señalización se pueden agrupar en dos tipos llamados *CAS* (Channel Associated Signaling o Canal Asociado) y *CCS* (Common Channel Signaling o Canal Común). La diferencia es que mientras *CAS* transmite la señalización en el mismo canal en que viaja la información, *CCS* la transmite en un canal separado. Por este hecho es que con *CAS* se reduce ligeramente el ancho de banda disponible o útil para la comunicación ya que una parte de él se está usando para señalización.

1.3.4.1. CAS o Señalización de Canal Asociado

El protocolo *CAS* (*robbed-bit*) es usado en circuitos *T1* y *E1* alrededor del mundo. *Robbed-bit* toma (o "roba", de allí su nombre) el octavo bit de cada canal de comunicación cada seis frames y lo reemplaza por información de señalización. El bit original robado simplemente se pierde. Ese bit es el menos significativo.

En *CAS*, la información y la señalización son transmitidas por el mismo canal.

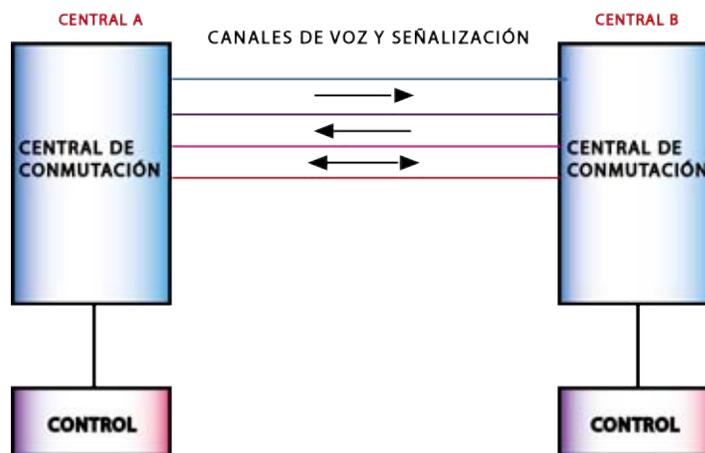


Figura 1.14: Señalización Canal Asociado

Puede trabajar tanto en infraestructura analógica como en la digital, aquí mencionaremos la digital, ya que es la utilizada hoy en día y la que nos compete para una red VoIP.

Existen tres grupos de señales en este sistema de señalización:

- *Señales de supervisión*: Son conocidas como señalización de línea, indican el estado del enlace troncal, los estados del enlace pueden ser diferentes en cada terminal del mismo.
- *Señales de dirección*: Representan los dígitos marcados.
- *Tonos y anuncios*: Se refieren a los tonos de marcado, ocupado, etc.; y a algunos mensajes como los que indican que el número al que se llama está fuera de servicio.

Cuando se transportan datos la pérdida de un bit no puede pasar desapercibida y la calidad de la transmisión se degrada de manera sensible.

Señalización R2

Un protocolo *CAS* que aún subsiste en nuestros días es *R2*. Es uno de los dos sistemas *UIT-T*⁹ regionales y se utiliza ampliamente dentro y entre los países europeos. Fue muy usado en los años 60.

Este protocolo pertenece a una serie de protocolos en donde cada implementación se denomina "*variante*". Existen variantes dependiendo del país o inclusive de la compañía telefónica que lo ofrece.

Toda la información de señalización correspondiente a un canal particular (o circuito) se pasa a través del mismo circuito (es decir, está asociado a él).

Los sistemas *CAS* de multifrecuencia casi siempre tienen 2 partes, la de *Señalización de Línea* y la de *Señalización entre Registros*, cada una con una función distinta. La parte de *Señalización de Línea*, controla la línea y el equipo común, además de que se encarga de enviar las señales de toma de línea y otras señales de supervisión, como por ejemplo, la señal de liberación. La parte de *Señalización entre Registros*, se encarga del transporte entre centrales y de la información referente a los dígitos que se marcan. Esta división en 2 partes de *CAS* ayuda a minimizar el número total de emisores y receptores de código de señalización en una central.

MFC/R2¹⁰ es una señalización telefónica usada ampliamente en México, Brasil y otros países de Latinoamérica y Asia. Es un protocolo de señalización *CAS* sobre tramas digitales E1.

MFC/R2 comparado con protocolos de señalización más recientes como *ISDN PRI/BRI* o *SS7*, *R2* ofrece funcionalidades limitadas ya que la señalización es sólo usada para

⁹Unión Internacional de Telecomunicaciones, es el organismo especializado de Telecomunicaciones de la Organización de las Naciones Unidas encargado de regular las telecomunicaciones a nivel internacional entre las distintas administraciones y empresas operadoras.

¹⁰ *Multi Frequency Compelled R2 (R2 dirigido por multifrecuencia).*

establecer la llamada o para terminarla.

MFC/R2 es definido por la *ITU*¹¹, sin embargo la mayoría de los países que usan *MFC/R2* no siguen las especificaciones de la *ITU* e implementan su propia variante.

Éste es protocolo de señalización peer-to-peer, esto es, que solo hay dos participantes involucrados en un enlace *E1/R2* y ambas puntas se comportan del mismo modo,

En *MFC/R2* se tienen dos tipos de señales:

1. *Señales de Línea*. Sirven para monitorear el estado de la llamada y las señales *MF*¹² son usadas para transmitir información de la llamada durante el establecimiento de la misma *DNIS* (*Dialed Number Identification Service*, que es, el número marcado o número destino), *ANI* (*Automatic Number Identification*, o Identificador de Llamadas), *Calling Party Category* (Categoría de Marcado). Estas señales se envían usando señales *CAS* que viajan usando el canal 16 del enlace *E1*. Todas las señales *CAS* para cada canal del *E1* es multiplexado por éste canal. Cada 2 milisegundos cada punta del enlace actualiza sus 4 bits de señal *CAS* conocidos como los bits *ABCD*.

MFC/R2 usa 2 de esos 4 bits para enviar las siguientes señales: *Idle*, *Block*, *Seize*, *Seize Ack*, *Clear Back*, *Forced Release*, *Clear Forward*, *Answer*.

Como sólo se usan 2 bits para las 7 posibles señales, algunas de ellas tienen el mismo patrón de bits, pero no representa un problema porque aunque el patrón de bits para *Forced Release* y *Seize* son los mismos, el protocolo conoce lo que la otra punta del enlace quiere. La razón de usar sólo 2 bits teniendo 4 disponibles viene de la época donde la versión analógica de *MFC/R2* fue portada para trabajar en un mundo digital.

El cuadro 1.4 describe los patrones de bits usados en la señalización *R2* mediante los bits *ABCD* de *CAS*:

¹¹Unión Internacional de Telecomunicaciones

¹²MultiFrecuencia

Cuadro 1.4: Señales de Línea en R2

Estado de Circuito	Envío AB	Recepción AB
Idle/Released (disponible)	1 0	1 0
Seized (toma)	0 0	1 0
Seizure Acknowledged (control toma)	0 0	1 1
Answered (contestación)	0 0	0 1
Clear Back (cuelgue)	0 0	1 1
Clear Forward (before Clear Back) (desconexión)	1 0	1 1
Clear Forward (after Clear Back)(liberación)	1 0	1 1
Blocked (bloqueo)	1 0	1 1

2. *Señales de Direccinamiento*: son 15 diferentes señales *MF*, que son tonos audibles compuestos por 2 frecuencias que viajan usando el canal de audio, y es por eso que un detector de audio es un componente importante en el paquete *R2*.

La *ITU* define las frecuencias que pueden ser mezcladas para componer los tonos *MF* y asigna los significados de cada uno de ellos. Sin embargo, algunos países dan significados diferentes a los tonos *MF*.

Los tonos de *MF* son usados para transmitir *ANI* (*Automatic Number Identification*, o Identificador de Llamadas), *DNIS* (*Dialed Number Identification Service*, que es, el número marcado o número destino) y la *Categoría de Marcado*. (Probablemente otra información es intercambiada en enlaces internacionales *R2*). Cuando la llamada es aceptada o rechazada, el detector *MF* se apaga y ya no se intercambian más señales *MF*, es entonces que el canal de audio puede ser usado para voz.

1.3.4.2. CCS o Señalización de Canal Común

En este tipo existe un canal dedicado exclusivamente para la señalización de todos los canales. Se le llama también *Señalización fuera de banda*.

Aquí los canales de voz o troncales, sólo son utilizados hasta que se establece una conexión. (Figura 1.15)

El mejor ejemplo de protocolo de tipo *CCS* es el *Integrated Services Digital Network (ISDN)* del cual se hablara en seguida.

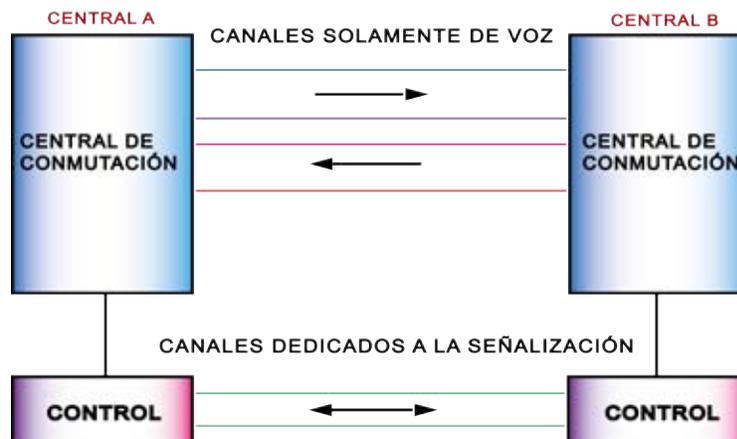


Figura 1.15: Señalización de Canal Común

ISDN o Red Digital Integrada de Servicios

Llamada en inglés *ISDN (Integrated Services Digital Network)*, permite que en una línea coexistan múltiples canales, pudiendo contener cada uno de ellos *DATOS (canales B)* o *SEÑALIZACIÓN (canales D)*. Pero no se limita sólo a la transmisión de voz, ya que cada canal tiene un ancho de banda de 64 Kbps, de forma que pueden emplearse *canales B y D* para la transmisión de datos (éstos últimos siempre que no haya datos de señalización). Ello la hace de una mayor flexibilidad ya que los canales pueden ser reconfigurados sobre la marcha para que transmitan voz o datos.

Maneja dos servicios estandarizados como son el *BRI* o *Servicio Básico (Basic Rate Interface)* y el *PRI* o *Servicio Primario (Primary Rate Interface)*.

- **BRI**

Un *BRI* supone 2 canales útiles del tipo *full-dúplex*, llamados *Canales B* de 64Kbit/s cada uno, más un canal de señalización también *full-dúplex* de 16Kbit/s, llamado *Canal D* que en total suman 144Kbit/s. Es más orientada a hogares. (Figura 1.16)



Figura 1.16: Canales BRI

- **PRI**

Puede ser una opción viable para cuando los usuarios son de un tamaño considerable como negocios o empresas ya que puede aglutinar más *Canales B* necesarios para transmisiones específicas, como audio, videoconferencias, etc.. En la actualidad es muy popular y se transmite sobre circuitos *T-carrier* y *E-carrier*.

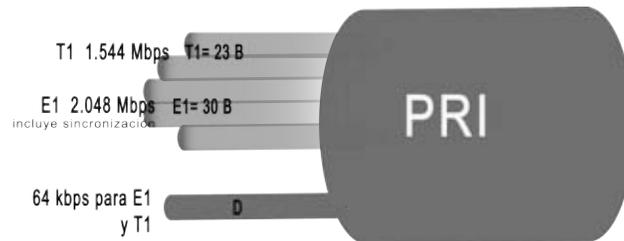


Figura 1.17: Canales PRI

Señalización SS7

Es el estándar de la tecnología *CCS*. Fue creado por la *ITU-T* para proveer señalización en la *PSTN*, permitiendo establecer llamadas, intercambiar información, enrutar, operar y tarifar servicios. Consiste en el uso de un canal de señalización diferente al canal de voz. Dicha separación permite que se tenga un canal que lleve la conversación y por otro el de la señalización. Es un sistema de señalización fuera de banda que se caracteriza por la transmisión de paquetes de datos a alta velocidad y por la posibilidad de permitir la señalización entre diferentes elementos de red, entre los que no se tiene una comunicación directa.

- *Puntos de Señalización.*

Son elementos de red que intercambian paquetes de datos para: el control de dirección, el establecimiento y control del enlace; y la gestión y mantenimiento de la red. Se compone de 3 tipos: *SSP (Service Switching Point)*, *STP (Signal Transfer Point)* y *SCP (Signal Control Point)*.

1. *SSP* o Puntos de Conmutación de la Señalización (*Service Switching Point*): Son switches telefónicos en los que se establece, conmuta y finaliza una llamada. Se comunican con otros *SSP's* por medio de circuitos con mensajes de señalización para gestionar recursos de la red, conectar, desconectar y administrar llamadas de voz.
2. *STP* o Punto de Transferencia de la Señalización (*Signal Transfer Point*): Provee acceso a la red, enruta o conmuta todos los mensajes de Señalización en la red. Se comunica lógicamente con otros *STP's*, es decir que no es necesario los enlaces físicos para su comunicación; son configurados en pares y proveen redundancia a la red.
3. *SCP* o Punto de Control de la Señalización (*Signal Control Point*): Proporciona acceso a las bases de datos para obtener información de enrutamiento adicional, información de operación y mantenimiento de servicios adicionales.

En la figura 1.18 se muestran los componentes dichos anteriormente:

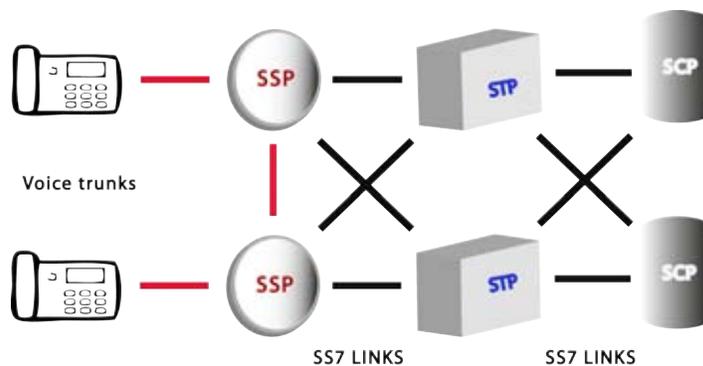


Figura 1.18: SS7

■ *Modelos de Señalización.*

Existen tres tipos de modelos de señalización para una red SS7:

1. *Modo Asociado*: Los canales de señalización y de voz, están directamente

conectados entre dos puntos de conmutación de forma paralela. Es adecuado para grandes cantidades de tráfico, ya que un canal de señalización SS7 puede administrar más de 2000 canales. (Figura 1.19)



Figura 1.19: Modo Asociado

2. *Modo No Asociado*: Es el que se utiliza con más frecuencia por SS7. Aquí, los canales de voz pueden tener una conexión directa entre los puntos de conmutación, mientras que los mensajes de señalización se transmiten a través de múltiples puntos de referencia para llegar al SSP final. (Figura 1.20)

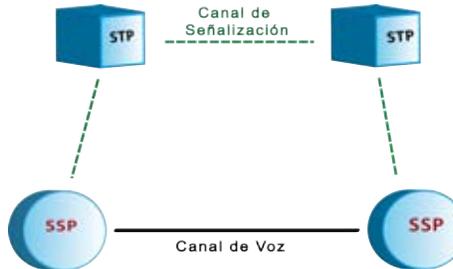


Figura 1.20: Modo No Asociado

3. *Modo Cuasi-Asociado*: En este caso, los mensajes de señalización son transmitidos a través del mínimo número de puntos de transferencia posibles, para así llegar a su punto de conmutación final. Al utilizar menos STP's que el Modo No Asociado, el retardo que introduce a la red es menor. (Figura 1.21)

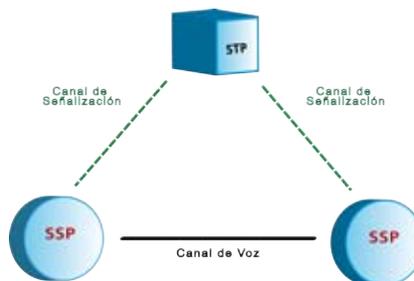


Figura 1.21: Modo Cuasi Asociado

- *Arquitectura SS7*

La parte principal en esta red es el Protocolo de Transferencia de Mensajes de Señalización (*MTP*). Este protocolo comprende los niveles dos y tres del modelo *OSI*¹³, y sobre él se transmite la señalización de todas las redes de conmutación de circuitos tradicionales, tanto de operadores de telefonía fija como de telefonía móvil.

El protocolo de señalización *SS7* se modela en 4 niveles (Figura 1.22):

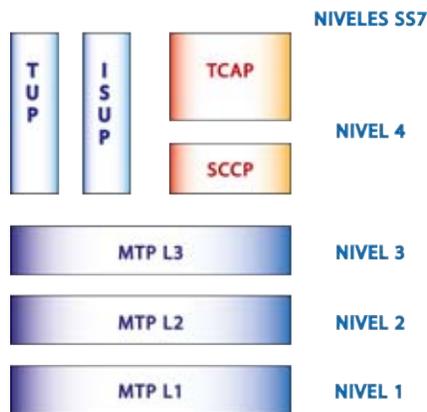


Figura 1.22: Niveles SS7

- *Nivel 1: Físico*

En general, los enlaces de señalización se implementan mediante transmisión *TDM* (*Multiplexación por División en el Tiempo*) utilizando canales de 64 kbps dentro de tramas punto a punto de 2 Mbps (*E1* normalizados).

- *Nivel 2: Enlace de Datos*

Este nivel proporciona la red con la entrega secuenciada de todos los paquetes de mensajes *SS7*. Similar que la capa de enlace de datos del modelo *OSI*, donde sólo se refiere a la transmisión de datos de un nodo a otro, y no a su destino final en la red.

La numeración secuencial se utiliza para determinar si los mensajes se han perdido durante la transmisión.

- *Nivel 3: Nivel de red*

El nivel de la red depende de los servicios de Nivel 2 para proporcionar un

¹³Interconexión de Sistemas Abiertos

encaminamiento, la discriminación de mensajes y funciones de distribución de mensajes.

- El Mensaje de la Discriminación determina a quien se dirige el mensaje.
 - El Mensaje de Distribución ocurre cuando se trata de un mensaje local
 - El Mensaje de Enrutamiento ocurre cuando no es un mensaje local.
- *Nivel 4: Protocolos de usuario y aplicación de piezas*

Este nivel 4 consta de varios protocolos, partes y piezas de usuario de aplicaciones

Acontinuación describiremos cada componente de manera breve:

- *TUP o Telephone User Part* (Parte de Usuario Telefónico): Proporciona los servicios básicos de telefonía.
- *ISUP o ISDN User Part* (Parte de Usuario ISDN): Es un protocolo orientado a la conexión, el que se encarga de que las llamadas telefónicas entre dos nodos se establezcan, se mantengan y finalicen.
- *TCAP o Transaction Capabilities Application Part* (Parte de Aplicación de Capacidades de Transacción): Éste define los protocolos que serán utilizados para la comunicación entre aplicaciones, además de permitir la comunicación entre el nivel de aplicación y los niveles inferiores, como *MTP* y *SCCP*
- *SSCP o Signaling Connection Control Part* (Parte de Control de Señalización): Este protocolo es no orientado a conexión. Provee direccionamiento y funciones adicionales a la *MTP* para transferir información entre las centrales en una red de señalización *SS7* ya sea orientado o no a la conexión.
- *MTP o Message Transfer Part* Parte de Transferencia de Mensaje: Este protocolo permite transmitir información de señalización a través de la red, se forma de 3 niveles:
 1. Enlace de datos de señalización.
 2. Enlace de señalización.
 3. Red de señalización.

1.4. Modelo OSI

El modelo de referencia de Interconexión de Sistemas Abiertos (OSI), fue el modelo de red creado por la Organización Internacioneal de Normalización (ISO) en el año de 1984; el cual es un conjunto completo se estándares funcionales que especifican interfaces, servicios y formatos de soporte para conseguir la interoperabilidad entre los distintos tipos de tecnología de red a nivel mundial.

Su función consiste en la separación de las distintas tareas que son necesarias para comunicar dos sistemas independientes. En él también se puede comprender cómo viaja la información a través de una red, explicando de qué manera los paquetes de datos pasan a través de varias capas a otro dispositivo de una red, aún cuando el remitente y el destinatario posean diferentes tipos de medios de red.

Este modelo consta de siete capas numeradas, cada una de las cuales tiene una función de red específica. A esta división de las funciones de sistema de red se denomina *división en capas*.

Las ventajas de encontrarse en dicha división son:

- Se puede dividir la comunicación de red en partes más pequeñas y sencillas. Lo que también ayuda a simplificar el aprendizaje.
- La normalización los componentes de red para permitir el desarrollo y el soporte de los productos de diferentes fabricantes.
- Permitir a los distintos tipos de hardware y software de red comunicarse entre sí.
- Impide que los cambios en una capa puedan afectar las demás, para que se puedan desarrollar con más rapidez.

La figura 1.23 ilustra las siete capas del modelo de referencia OSI.

Capas Modelo OSI

Capa 1: Capa Física.

Esta capa se encarga de las conexiones físicas de la computadora hacia la red. Tanto en lo que se refiere:

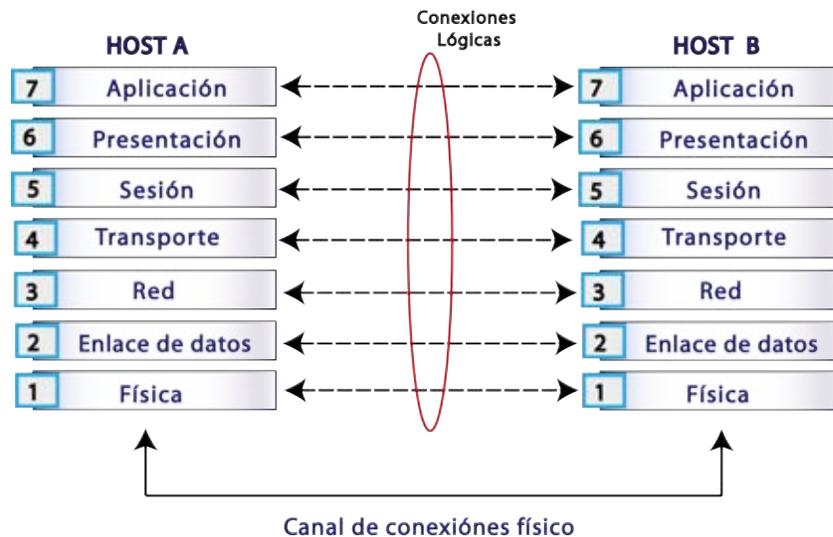


Figura 1.23: Modelo OSI

- El medio físico: Medios guiados: cable coaxial, cable par trenzado, fibra óptica y otros tipos de cables. Medios no guiados: radio, infrarrojos, microondas y otras redes inalámbricas.
- Características del medio: tipo de cable o calidad del mismo, tipo de conectores normalizados o en su caso tipo de antena, etc..
- La forma en que se transmite la información: codificación de señal, niveles de tensión/intensidad de la corriente eléctrica, modulación, etc..

Capa 2: Capa de Enlace.

Es la que se encarga de la integridad de la recepción y del envío de la información, así como de saber donde comienza la transmisión de la trama y donde termina. Garantiza que tanto la computadora transmisora como la receptora se encuentren sincronizadas en su reloj y también que empleen el mismo sistema de codificación y decodificación.

Capa 3: Capa de Red.

Esta es la responsable del direccionamiento de mensajes y de la conversión de las direcciones y de los nombres lógicos a físicos. Determina la ruta del mensaje desde la computadora emisora hasta la receptora dependiendo de las condiciones de la red. Dentro de las funciones del ruteo de mensajes, evalúa la mejor ruta que debe de seguir el

paquete dependiendo del tráfico en la red, etc. A la información proveniente de esta capa, se le agregan componentes apropiados para su ruteo en la red y para mantener un cierto nivel en el control de errores. La información es presentada según el método de comunicaciones para acceder a la LAN, a la WAN (como los enlaces E1), y la conmutación de paquetes.

Capa 4: Capa de Transporte.

Es llamado ocasionalmente nivel de Host to Host o nivel de End to End, debido a que en él se establecen, mantienen y terminan las conexiones lógicas para las transferencias de información entre los usuarios.

Este nivel puede incluir los tipos de datagramas, los servicios de correo electrónico, las prioridades de los mensajes, la recolección de la información y su administración, la seguridad en tiempos de respuesta, las estrategias de recuperación en caso de falla y la segmentación de la información cuando el tamaño es mayor al máximo del paquete según el protocolo.

Al recibir información del nivel de Red, la capa de Transporte verifica que la información se encuentre en el orden adecuado y revisa si existe información duplicada y extraviada. En caso de encontrarse en desorden, este nivel corrige el problema y transfiere la información al nivel de sesión en donde se le dará un proceso adicional.

Los protocolos TCP (Transmission Control Protocol) y UDP (User Datagram Protocol) son característicos de este nivel.

Capa 5: Capa de Sesión.

En esta capa se permite que dos aplicaciones en diferentes dispositivos establezcan, usen y terminen la conexión llamada Sesión. Establece las reglas para iniciar y terminar la comunicación entre los dispositivos y brinda el servicio de recuperación de errores, es decir, si la comunicación falla y es detectada, el nivel de Sesión puede retransmitir la información para completar el proceso de comunicación.

Capa 6: Capa de Presentación.

Garantiza que la información que envía la capa de Aplicación de un sistema pueda ser leída por la capa de aplicación de otro. De ser necesario, la capa de Presentación tradu-

ce entre varios formatos de datos utilizando un formato común.

Capa 7: Capa de Aplicación

Esta es la capa más cercana al usuario. Suministra servicios de red a las aplicaciones del usuario. No proporciona servicios a ninguna otra capa OSI, sino solamente a aplicaciones que se encuentran fuera del modelo OSI. La capa de aplicación establece la disponibilidad de los puntos de comunicación, sincroniza y establece acuerdos sobre los procedimientos de recuperación de errores y control de la integridad de los datos.

En la figura 1.24 se muestran algunos de los protocolos que se manejan en las siete capas.

CAPAS DEL MODELO OSI	PROTOCOLOS
Aplicación	TELNET,FTP,SNMP, NNTP,SMTP,POP3, DNS,RTP,SSH, HTTP,HTTPS
Presentación	ASN.1
Sesión	NetBIOS
Transporte	TCP,UDP
Red	ARP,IP(IPv4/IPv6) ICMP,X.25
Enlace de datos	Ethernet, fast ethernet, gigabit ethernet,FDDI, ATM,HDLC
Física	CGI, MIME, IEEE

Figura 1.24: Protocolos Modelo OSI

1.5. Modelo TCP/IP

Este modelo fue desarrollado por el DoD (Department of Defense) de los Estados Unidos de América y por DARPA (Defense Advanced Research Project Agency) en la década de los 70. El DoD deseaba que sus paquetes llegaran al destino siempre, bajo cualquier condición, desde un punto determinado hasta cualquier otro. El modelo TCP/IP fue pensado como un estándar abierto para poder conectar dos máquinas cualesquiera, todo el mundo puede utilizarlo y es en el que se basa Internet.

Capas TCP/IP

El modelo TCP/IP (Figura 1.25) define cuatro capas totalmente independientes en las que divide el proceso de comunicación entre dos dispositivos: la capa de Aplicación, la capa de Transporte, la capa de Internet y la capa de Acceso de Red. Debemos destacar que algunas de las capas del modelo TCP/IP poseen el mismo nombre que las del modelo OSI, pero no hay que confundirlas ya que la capa de aplicación tiene diferentes funciones en ambos modelos.



Figura 1.25: Modelo TCP/IP

Capa 1: Acceso a Red

Se encarga de controlar el acceso al nivel físico que se utilice y de enviar la información por el mismo. Transforma toda la información que le llega de las capas superiores (a

bits) y la prepara para que se pueda enviar por el medio. El direccionamiento físico de la red lo hace mediante direcciones MAC. Esta capa incluye los detalles de tecnología LAN y WAN y todos los detalles de las capas: física y de enlace de datos del modelo OSI.

Capa 2: Internet

El propósito de esta capa es enviar paquetes desde cualquier punto entre la red y que estos paquetes a su vez, lleguen a su destino independientemente de la ruta y de las redes que recorrieron para llegar hasta allí. El protocolo que rige esta capa es el Protocolo Internet (IP). Aquí se determina la mejor ruta y la conmutación de paquetes. Esta capa no proporciona la fiabilidad en las conexiones, de esto ya se ocupa la capa de transporte. Pero si realiza un direccionamiento lógico de red mediante las direcciones IP. Es la capa responsable de proporcionar conectividad entre usuarios.

Los Protocolos que se manejan en la capa de Internet son:

- El protocolo IP que sirve como protocolo universal para unir dos máquinas en cualquier momento, lugar y tiempo. No es un protocolo orientado a conexión y no es confiable. Este protocolo determina el formato de la cabecera del paquete IP donde se incluye la dirección lógica y otras informaciones de control.
- El ICMP o Protocolo de Mensajes de Control en Internet, el cual suministra las capacidades de control y envío de mensajes.
- El ARP o Protocolo de Resolución de Direcciones que determina la dirección, de la capa de enlace de datos, la dirección MAC para las direcciones IP conocidas.

Capa 3 de Transporte

La capa de transporte establece, mantiene y termina circuitos virtuales además de que proporciona mecanismos de control de flujo y permite las retransmisiones y detecta o corrige errores.

Dentro de esta capa existen dos protocolos que se utilizan para el envío de segmentos de información:

- TCP o Protocolo para el Control de la Transmisión, que ofrece maneras flexibles y de alta calidad para crear comunicaciones de red confiables, sin problemas de flujo y con un nivel de error bajo. La información que le llega de la capa de aplicación la

divide formando diferentes segmentos. El direccionamiento se realiza a través de puertos.

Sincroniza y regula el tráfico. Controla el flujo para que no se pierdan los paquetes y evitar así una congestión en la red. Es un protocolo orientado a conexión.

Sus funcionalidades básicas son:

- Fiabilidad
 - Control de flujo
 - Corrección de errores
 - Retransmisión
- UDP o Protocolo de Datagrama de Usuario, es un protocolo de transporte no orientado a conexión que intercambia datagramas sin la utilización de ACK ni SYN que se utilizan como acuse de recibo en el caso de TCP. El procesamiento de errores y retransmisiones es soportado por los protocolos de capas superiores.

Capa 4 de Aplicación

Es la capa más cercana al usuario final y la que le proporciona servicios de red. Como es la capa superior, no da servicios a ninguna capa. Es la responsable de traducir los datos de la aplicación o programa para que puedan ser enviados por la red. Sus funciones se resumen en:

- Representación
- Codificación
- Control de diálogo
- Gestión de las aplicaciones de usuario

Los Protocolos de la capa de Aplicación son:

- El protocolo FTP o Protocolo de Transmisión de Archivos, sirve para la transmisión de archivos entre dos máquinas. Utiliza TCP para crear una conexión virtual para la información de control, y luego crea otra conexión para el envío de datos. Los puertos utilizados son el puerto 20 y 21.

- El protocolo HTTP o Protocolo de Transferencia de Hipertexto, es para visualizar la mayoría de páginas web. Sus mensajes se distribuyen como los de correo electrónico. Utiliza el puerto 80.
- El protocolo SMTP o Protocolo Simple de Transferencia de Correo Electrónico, es un servicio de correo que se basa en el modelo de FTP. Transfiere mensajes de correo entre dos sistemas y provee de notificaciones de correo entrante. El puerto que se utiliza es el 25.
- El protocolo DNS o Sistema de Nombres de Dominio, es el que se encarga de reconocer el nombre de la máquina remota con la que se quiere establecer la conexión y traduce el nombre a su dirección IP. Utiliza el puerto 53.
- El protocolo TFTP o Transferencia de Archivos Trivial, tiene las mismas funciones que el protocolo FTP pero funciona sobre UDP, con lo que hay mayor rapidez pero menor seguridad y confiabilidad. El puerto que se utiliza es el 69.

En la figura 1.26 se muestran algunos ejemplos de los protocolos que corresponden a cada capa del modelo TCP/IP.

CAPAS DEL MODELO TCP/IP	PROTOCOLOS
Aplicación	TELNET,FTP,SNMP, SMTP,SNMP,DNS,RIP
Transporte	TCP,UDP,IGMP,ICMP
Internet	IP,IPSEC
Acceso a red	Ethernet, Token Ring, Frame Relay, ATM

Figura 1.26: Protocolos Modelo TCP/IP

Las diferencias y semejanzas entre ambos modelos se muestran en la figura 1.27:

	MODELO OSI	MODELO TCP
DIFERENCIAS	Modelo teórico	Modelo práctico
	Sesión, Presentación y Aplicación son niveles independientes.	Sesión y Presentación son combinados en la capa de Aplicación.
	Las capas Físicas y Enlace de datos son niveles independientes	Las capas Física y Enlace de datos son combinadas en la capa de Subred
	Las redes típicas no se desarrollan normalmente sobre OSI pero se utiliza como guía.	Contiene protocolos sobre los cuales se desarrolla la mayoría de las redes como la Internet.
SEMEJANZAS	Ambos modelos se dividen en capas, el modelo OSI posee 7 niveles o capas, mientras que el modelo TCP/IP solo 4 capas.	
	Los dos modelos utilizan tecnología de conmutación por paquetes y no de conmutación por circuitos.	
	Ambos modelos son útiles y ayudan dentro de la comunicación.	

Figura 1.27: Diferencias entre Modelo OSI y TCP/IP

1.5.1. IP(Protocolo de Internet)

Este protocolo trabaja a nivel de red, solo envía y recibe información entre un origen y destino, sin embargo no garantiza como tal la entrega de los mismos, ya que para realizar dicha tarea divide los datos en paquetes, los cuales pueden tomar diferentes rutas, dependiendo la redundancia que contenga la red.

Datagrama IP

El datagrama IP es la unidad de transferencia en las redes IP. Éste consiste en una cabecera IP y un campo de datos para protocolos superiores.

Se encuentra en la trama de nivel de enlace y suele tener una longitud máxima (MTU) ¹⁴, dependiendo del hardware de red usado, para Ethernet comúnmente es de 1500 bytes.

IP puede tratar la fragmentación y el reensamblado de sus datagramas. El tamaño máximo del datagrama es de 65536 bytes, sin embargo, este valor nunca es alcanzado porque las redes no tienen suficiente capacidad como para enviar paquetes tan grandes. Si el datagrama es demasiado grande para pasar por la red, el router lo dividirá en fragmentos más pequeños que la MTU de la red, de manera tal que el tamaño del fragmento

¹⁴Maximum Transfer Unit

sea un múltiplo de 8 bytes.

Todas las redes deberían ser capaces de manejar al menos 576 bytes. Este valor se especifica en el RFC 791, y fue elegido para permitir un bloque de datos de tamaño razonable. 576 bytes se ha convertido en un valor común de MTU por defecto utilizado para los datagramas IP, ya que es el tamaño mínimo especificado en el estándar IP

Los fragmentos contienen una cabecera, que es una copia del datagrama original y de los datos que la siguen. Los fragmentos se tratan como datagramas normales mientras son transportados a su destino. Es importante mencionar que si uno de los fragmentos se pierde, todo el datagrama se considerará perdido, y los restantes fragmentos también se considerarán perdidos.

La cabecera del datagrama IP se estructura de la siguiente manera:

- **VER (4bits):** Version, es la versión del protocolo IP. La versión actual es la 4. La 5 es experimental y la 6 es IP "next generation"¹⁵. Ésta información solo la utilizan los routers y la capa IP de la conexión.
- **IHL (4bits):** Internet Header Length (*Longitud*), son 4 bits ($2^4 = 16$ posiciones) que indican el número de palabras de 32 bits que ocupa la cabecera. El valor usual de este campo es 5 ($5 * 4 \text{ bits} = 20 \text{ bytes}$).
- **TOS (8bits):** Type Of Service (*Servicio*), El tipo de servicio es una indicación de la calidad del servicio solicitado. Estos parámetros se usarán para guiar la selección de los parámetros de servicios reales al transmitir un datagrama a través de una red en particular. Consta de 8 bits, los tres primeros bits indican la naturaleza y la prioridad del datagrama:
 - 000 (0)- RUTINA
 - 001 (1)- PRIORIDAD
 - 010 (2)- INMEDIATO
 - 011 (3)- URGENTE (FLASH)

¹⁵IPng

- 100 (4)- MUY URGENTE (FLASH OVERRIDE)
- 101 (5)- CRÍTICO
- 110 (6)- CONTROL ENTRE REDES
- 111 (7)- CONTROL DE RED

Los 4 siguientes bits representan el TIPO DE SERVICIO. Cada bit selecciona un tipo de comportamiento (solo uno puede estar activado):

- 1000 El paquete solicita minimizar el retardo
- 0100 El paquete solicita maximizar la tasa de transferencia
- 0010 El paquete solicita maximiza la fiabilidad del transporte
- 0001 El paquete solicita minimizar el costo económico del transporte¹⁶
- 0000 Servicio normal

Y el bit último siempre vale cero.

- **Total Length (16 bits)** (*Longitud Total*), es la longitud total del datagrama, incluida la cabecera y datos. Mide 16 bits e indica cuánto espacio se debe guardar en la memoria para la recepción de cada datagrama, también dice cuantos bytes se deben leer por datagrama. Con esto se puede tener un control muy sencillo de si los datagramas llegan completos o no. También limita el tamaño máximo de los datagramas a 65515 bytes.¹⁷ El tamaño mínimo es de 576 bytes.
- **Identification (16 bits)** (*Identificación*) es un número único de 16 bits que asigna el emisor para ayudar a reensamblar un datagrama fragmentado. Esto permite recomponer el datagrama original en el destino. Un datagrama puede ser fragmentado en un máximo de 65535 fragmentos.
- **Flags (3bits)**: son banderas para el control de fragmentación.
Donde:
 - El bit 0 está reservado y debe ser 0.

¹⁶Esto significa que los datagramas sean encaminados vía la ruta de menor coste

¹⁷ $2^{16} = 65535 \text{ bytes} - 20 \text{ bytes de encabezado} = 65515 \text{ bytes}$.

- El bit DF significa **no fragmentar** (*Do not Fragment*). Con 0 se permite fragmentación y con 1 no.
- El bit MF significa que hay **más fragmentos** (*More Fragments*). Con 0 significa que se trata del último fragmento del datagrama y con 1 que hay más fragmentos.
- **FO (13bits):** Fragment Offset, Nos indica la posición en bytes que ocupan los datos del datagrama original. Sólo tiene sentido si el datagrama forma parte de uno mayor que haya sido fragmentado.
Tiene un campo máximo de 13 bits ($2^{13} = 8192$). De esta forma siempre se puede reconstruir el datagrama original con los fragmentos.
- **TTL (8 bits):** Time To Live especifica el tiempo en segundos que se le permite viajar a este datagrama.
 - Si alcanza el valor 0, el datagrama se destruye y se genera un mensaje ICMP ¹⁸
 - Evita que un datagrama se quede atrapado en un bucle.
- **Protocol (8bits):** El tipo de Protocolo es un valor que utiliza el datagrama (TCP, UDP, ICMP, etc.). Es útil ya que todos los servicios de Internet usan IP como medio de transporte, lo que hace que se necesite un mecanismo de discriminación entre los diferentes protocolos.
- **Header Checksum (16bits):** Chequeo de la cabecera, en este campo se almacena un conteo total de los campos de la cabecera del datagrama IP. Es un mecanismo simple que sirve para detectar posibles errores. Si este chequeo no corresponde con los contenidos, el datagrama se desecha, ya que al menos un bit de la cabecera se encuentra corrupto y el datagrama podría haber llegado a un destino equivocado.
- **Source/Destination IP Address (32bits):** Direcciones IP origen y destino, Están formadas por 4 bytes como muestra el formato de la figura 1.28:

¹⁸Protocolo de Mensajes de Control de Internet O Internet Control Message Protocol

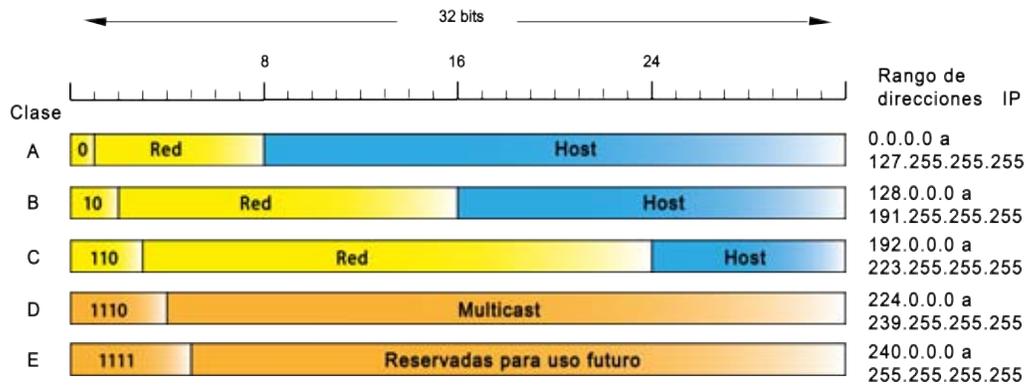


Figura 1.28: Direcciones IP

- Opciones (variable):** pueden o no aparecer en los datagramas. Deben ser implementadas por todos los módulos IP (host y pasarelas). Lo que es opcional es su transmisión en cualquier datagrama en particular, no su implementación.

En la figura 1.29 se muestra la cabecera del Datagrama IP.



Figura 1.29: Datagrama IP

1.5.2. TCP (Protocolo de Control de Transmisión)

Este protocolo está pensado para proporcionar un servicio fiable de comunicación entre procesos en un entorno con múltiples redes.

Es un protocolo de nivel de transporte que proporciona un servicio de transferencia confiable de datos y un método para trasladar datos encapsulados con TCP a un protocolo de nivel de aplicación. Cuando se proporcionan los datos al protocolo IP los agrupa en Datagramas IP.

TCP envía una información extra en el encabezado del paquete, con el cual, controla que los paquetes de datos que fueron enviados a través del protocolo IP se encuentren libres de errores y al ser recibidos se hagan en el mismo orden en el que fueron enviados, así también al haber pérdidas de paquetes, el protocolo TCP utiliza mecanismos para que estos sean reenviados.

Tiene las siguientes características:

- Permite colocar los datagramas nuevamente en orden cuando vienen del protocolo IP.
- Realiza un monitoreo del flujo de datos para así evitar la saturación de la red.
- Hace que los datos se formen en segmentos de longitud variada para entregarlos al protocolo IP.
- Permite multiplexar los datos, esto significa que la información que viene de diferentes fuentes, pueda circular simultáneamente.
- Permite comenzar y finalizar la comunicación amablemente.

Durante una comunicación usando este protocolo, las dos máquinas deben establecer una conexión. La máquina que solicita la conexión se le llama cliente y la máquina receptora se le llama servidor. Las máquinas de dicho entorno se comunican en modo full dúplex, es decir que la comunicación se realiza en ambas direcciones.

Para que se haga posible la comunicación y que funcionen bien todos los controles que la acompañan, los datos se agrupan, es decir que se agrega un encabezado a los paquetes de datos que permitirán sincronizar las transmisiones y garantizar su recepción. El TCP puede controlar la velocidad de los datos usando su capacidad para emitir mensajes de tamaño variable. A estos mensajes se llaman *segmentos*.

Este protocolo se encuentra formado de la siguiente manera:

- **Puerto origen** (16 bits), es el puerto que se relaciona con la aplicación en curso en la máquina origen.
- **Puerto destino** (16 bits), es el puerto que se relaciona con la aplicación en curso en la máquina destino.
- **Número de secuencia** (32 bits), Cuando el indicador SYN está fijado en 0, el número de secuencia es el de la primera palabra del segmento actual. Cuando SYN está fijado en 1, el número de secuencia es igual al número de secuencia inicial utilizado para sincronizar los números de secuencia (ISN)
- **Número de comprobación** (32 bits): Se le conoce como número de descargo y se relaciona con el número del último segmento esperado y no el número del último segmento recibido.
- **OFFSET** (*Margen de datos*) (4 bits): Permite ubicar el inicio de los datos en el paquete. Aquí, el margen es fundamental porque el campo **opción** es de tamaño variable.
- **Reservado** (6 bits): Este campo actualmente no está en uso pero se proporciona para el uso futuro.
- **FLAGS** (*Indicadores*) (6x1 bit): Los indicadores representan información adicional:
 1. URG: Si se encuentra en 1, el paquete se debe procesar en forma urgente.
 2. ACK: Si está en 1, el paquete es un acuse de recibo.
 3. PSH (PUSH): Si se encuentra en 1, el paquete opera de acuerdo con el método PUSH.
 4. RST: Si está en 1, se restablece la conexión.
 5. SYN: El indicador SYN de TCP indica un pedido para establecer una conexión.
 6. FIN: Si se encuentra en 1, se interrumpe la conexión.
- **Ventana** (16 bits): Permite saber la cantidad de bytes que el receptor desea recibir sin acuse de recibo.
- **CHECKSUM** (*Suma de control*) (8 bits) (CRC): Esta se hace tomando la suma del campo de datos del encabezado para poder verificar la integridad del encabezado.

- **Puntero urgente** (16 bits): Indica el número de secuencia después del cual la información se torna urgente.
- **Opciones** (tamaño variable): Diversas opciones
- **Relleno**: Es un espacio restante después de que las opciones se rellenan con ceros para tener una longitud que sea múltiplo de 32 bits.

La figura 1.30 muestra la estructura del Protocolo TCP

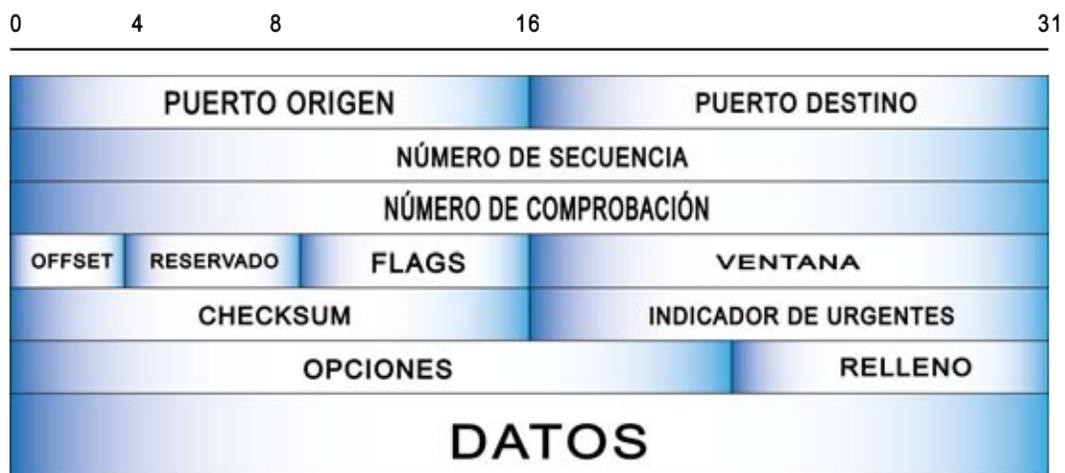


Figura 1.30: Protocolo TCP

1.5.3. UDP (Protocolo de Datagrama de Usuario)

El protocolo de datagrama de usuario es un estándar de TCP/IP y se define en el RFC 768. Perteneció a la capa de Transporte y es un servicio no orientado a conexión. Sirve para que el transporte de los datos sea rápido, compacto pero no confiable entre host TCP/IP. No garantiza la entrega ni comprueba la secuencia de los datagramas, y puede suceder que en la entrega de los datagramas estos lleguen a duplicarse y/o desordenados a su destino. Es por ello que UDP es un protocolo del tipo *best-effort* (máximo esfuerzo), porque hace lo que puede para transmitir los datagramas hacia la aplicación pero no puede garantizar que la aplicación los reciba.

Cuando se detecta un error en un datagrama, en lugar de entregarlo a la aplicación des-

tino, mejor se descarta.

Los mensajes UDP se encuentran encapsulados y se envían en datagramas IP.

Este protocolo se encuentra formado de la siguiente manera:

- **Puerto de origen (16 bits)** Puede ser opcional. En caso de que sí sea ocupado entonces indica el puerto del proceso emisor, y puede que se asuma que ese es el puerto al cual la respuesta debería ser dirigida en ausencia de otra información. En caso de no utilizarse se pone un valor cero.
- **Puerto destino (16 bits):** Es el número de puerto de la máquina destino.
- **Longitud (16 bits):** Representa la longitud en octetos, incluyendo la cabecera y los datos. Lo que significa que el valor mínimo de este campo es ocho.
- **Checksum (16 bits) (*Suma de control*):** Suma de comprobación de errores del mensaje. Para su cálculo se utiliza una pseudo-cabecera que también incluye las direcciones IP origen y destino. Para conocer estos datos, el protocolo UDP debe interactuar con el protocolo IP.
- **Datos:** Aquí viajan los datos que envían las aplicaciones. Los mismos datos que envía la aplicación origen son recibidos por la aplicación destino después de atravesar por internet.

La figura 1.31 muestra la estructura del Protocolo UDP.

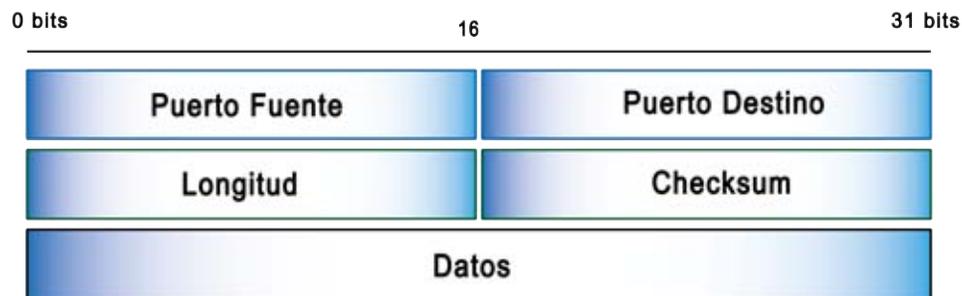


Figura 1.31: Protocolo UDP

Este protocolo maneja número de puertos, los que proporcionan una ubicación para enviar y recibir mensajes UDP. Un puerto funciona como una única cola de mensajes, la que recibe todos los datagramas destinados al programa específico, esto mediante cada número del puerto de protocolo. Lo que significa que los programas basados en UDP pueden recibir varios mensajes a la vez.

El lado de servidor de cada programa que maneja UDP, se encarga de atender los mensajes que llegan a su número de puerto conocido.

Los puertos se identifican mediante un número decimal que va desde el 0 hasta el 65.535. Los fabricantes que implementan UDP disponen de una gran libertad para asignar números de puertos a los procesos, aunque la Autoridad de Números Asignados de Internet (IANA) ha reservado un rango de puertos que va desde el 0 al 1.023 (algunos pueden ser superiores) para servicios estándar, (Well-Known) como DNS, SNMP, o Netbios.

El cuadro 1.5 muestra algunos números de puerto de servidor UDP.

Cuadro 1.5: Puertos para UDP.

Nombre del Servicio	Número de Puerto	Descripción
FTP	21	Transferencia de archivos
SSH	22	Protocolo Secure Shell
Telnet	23	Telnet
HTTP	80	World Wide Web
POP3	110	Protocolo de Correo v.3
MySQL	3306	Protocolo de Base de datos
SIP	5060	Protocolo de inicio de Sesión

Capítulo 2

Entorno de la Telefonía sobre IP

2.1. Conceptos de Voz por IP

El hablar de infraestructura para una solución de Voz por IP, es hablar de soluciones ya sean cerradas, muy costosas, (como Avaya, Cisco, Alcatel, etc.), y que no siempre cumplen con todas las expectativas o necesidades que se requieren, o que para poder abarcar todas las características necesarias, implica el uso de licencias y pagos por implementaciones extras.

Sin embargo existe la parte del Código libre, donde se tiene la facilidad de poder obtener todo lo necesario de manera casi gratuita, ya que se cuenta con un motor de aplicación telefónica (en este caso Asterisk), así como de estándares y protocolos abiertos. Este conjunto permite implementar un equipo con alta escalabilidad a un precio muy bajo, obteniendo todo lo que se necesita para su desarrollo.

La telefonía en conjunto con algunos medios de comunicación ha también evolucionado, ya que no solo se transmite de forma análoga-digital, sino que también ahora puede viajar a través de la nube, es decir usando el protocolo IP.

Sin embargo esta evolución no fue fácil, en un principio presentó muchos inconvenientes ya que en realidad la red no había sido creada para fines de voz, pero con el tiempo y apoyados en muchos protocolos y códecs, se ha dado un gran avance en el área, convirtiendo la Voz por IP en un avance tecnológico bastante significativo.

La Telefonía sobre IP es una de las aplicaciones que ha evolucionado al grado de que se puede originar una llamada a través de la nube pero en forma de paquetes de datos, es decir, en la telefonía tradicional la conmutación se daba a través de circuitos, aquí

la conmutación es por medio de paquetes de datos que viajan a través de la red por protocolo de internet.

La conmutación de paquetes en estricto sentido es el envío de datos o información en bloques a través de la red de un punto a otro. En el origen, la información se divide en pequeñas partes llamadas paquetes, estos paquetes no solo envían la información sino que también envían algo llamado encabezados, los cuales son los encargados de contener información de control (origen, destino, prioridad, etc.). Una vez llegando a su destino, son reensamblados obteniendo el mensaje original.

Sin embargo aunque el Protocolo de Internet es el más significativo para la realización de la Telefonía sobre IP, no es el único, ya que en realidad la Voz por IP requiere de más de un protocolo y de algunos códecs para asegurar su eficiencia.

El hablar de Voz por IP es hablar de la posibilidad de transmitir paquetes de voz sobre una red de datos. Cabe aclarar que no es el mismo Voz por IP que Telefonía sobre IP, aunque el principio básico de ambas sea el mismo. Sin embargo, hablar únicamente de Voz por IP es poder entender que se puede mandar la voz en paquetes de datos, (utilizando para ello algunos recursos), en pocas palabras es: el poder mandar una señal analógica en forma digital.

Existen diferentes aplicaciones que nos dan una muestra de lo que puede ser Voz por IP, por ejemplo el Messenger, éste fue una aplicación muy famosa a finales de los 90, donde en un principio únicamente tenía una función de mensajería instantánea (el llamado chat), el cual fue evolucionando y apareciendo más productos similares, con el paso del tiempo el Messenger daba la posibilidad de poder mandar mensajes de voz y a su vez escucharlos, esto es un principio de Voz por IP, ya que la voz podía viajar en forma de digitalizada por la red de datos.

La Telefonía sobre IP en cambio, aunque también viaja a través de la red de datos, es controlada por protocolos y códecs, los cuales muchos de ellos son gratuitos, el más conocido es el SIP¹, el cual es por así decirlo, el estándar en la comunicación de Telefonía sobre IP, sin embargo también existe el IAX², el cual es propietario de Asterisk, esto es solo hablando de los protocolos de sesión.

La Telefonía sobre IP hoy por hoy presenta grandes ventajas con respecto a la telefonía

¹Protocolo de Inicio de Sesión

²Inter-Asterisk Exchange

tradicional, principalmente en cuestión de costos, donde puede abaratar las llamadas a largas distancias nacionales e internacionales, llegando a ser hasta gratuitos, además de la integración con otros servicios,(Circuito Cerrado de TV, IVR, call-centers, etc.).

2.2. Señalización, Transporte y audio

Ya se ha mencionado un poco acerca de las diferencias entre Telefonía sobre IP y Voz por IP, aunque es un concepto similar sus aplicaciones pueden variar un poco, al igual que en la telefonía tradicional, la Telefonía sobre IP necesita protocolos de señalización en este caso digital.

La señalización tiene, entre otras finalidades, el poder identificar los procedimientos para que un usuario pueda realizar una llamada. Esto implica desde la identificación por parte del sistema de cada uno de los dispositivos o terminales para poder enlazar los diferentes usuarios conectados, definiendo el estatus de una llamada es decir: si ha sido contestada, colgada, si se encuentra ocupada o no disponible, etc., así como los mejores métodos para el establecimiento de la llamada, lo cual implica compatibilidad de códecs, el ancho de banda, el manejo de calidad de servicio y poder dar el plus de servicios adicionales, como es la tarificación, la transferencia de llamadas, las conferencia, etc., por lo que debe coexistir además con otros protocolos y aplicaciones y ser compatibles con ellas.

A través del tiempo se han venido presentando ciertos cambios en cuanto a señalización se refiere. En un principio los protocolos de señalización era de propietarios, ya que al no ser una tecnología tan difundida ni comercializada, los que trabajaban con dicha tecnología poseían sus propios medios de señalización, en otras palabras, solo se podían comunicar entre dispositivos del mismo fabricante, esto naturalmente ocasionó algunos problemas de interoperabilidad con otras marcas, por lo que la UIT-T, comenzó a desarrollar estándares para este fin, como por ejemplo el H.323, del cual se hablará mas adelante. Otro es el SIP, que es uno de los más conocidos. O también con el desarrollo de nuevas tecnologías como Asterisk, se creó el IAX, el cual es un estándar libre pero optimizado para Asterisk. Sin embargo, también empresas de carácter privado desarrollaron sus propios métodos de señalización como CISCO con SCCP(Skinny Client Control Protocol) y Unistim de Nortel.

2.2.1. Protocolos de Transporte

UDP e IP no son suficientes para asegurar el transporte de la voz. Como ya se dijo, UDP es un protocolo que no asegura la llegada de los paquetes en orden según se hayan enviado. Para que el transporte de voz y de video en tiempo real sea buena, es necesario usar dos protocolos adicionales, el RTP (Protocolo en Tiempo Real) y RTCP (Control de Protocolo RTP).

Ambos protocolos se sitúan en nivel de aplicación y se utilizan en conjunto de UDP. Pueden utilizar el modo Unicast (punto a punto) o el modo Multicast (multipunto). Utilizan puertos diferentes. RTP usa un número de puerto par, mientras que RTCP lo usa impar. Cuando una sesión RTP es abierta, se abre al mismo tiempo una sesión RTCP implícita. RTP por defecto tiene el puerto 5004 y RTCP el 5005, pero comprenden entre los rangos de 1025 y 65535.

2.2.1.1. RTP (Protocolo de Transporte en Tiempo Real)

RTP o Protocolo de Transporte de Tiempo Real, es un protocolo de nivel sesión. Publicado por primera vez como estándar en 1996 como la RFC 1889, y actualizado posteriormente en 2003 en la RFC 3550, que constituye el estándar de Internet STD 64.

Se encarga esencialmente del transporte de datos con características de tiempo real como lo es el audio, el video, etc..

El RTP identifica el tipo de información transportada. Para que la aplicación destino pueda sincronizar los flujos y pueda medir los retardos y la fluctuación, RTP añade marcadores temporales para indicar el instante en que se envió el paquete.

RTP también incluye números de secuencia a la información transportada, esto es para que se detecte la pérdida de paquetes y se puedan entregar a la aplicación destino.

Algo importante por mencionar es que este protocolo no es apto para controlar la calidad de servicio, y tampoco garantiza la entrega del paquete en recepción.

Formato del paquete RTP

Los primeros 12 octetos (es decir, los campos V, P, X, CC, M, PT, sequence number,

timestamp y SSRC) siempre están presentes, en tanto que los identificadores de "fuentes contribuyentes" (nodos que generan información al mismo tiempo para, supongamos, una videoconferencia) son utilizados sólo en ciertas circunstancias. Después del header "básico" puede tenerse extensiones opcionales para el header (Extension header).

Finalmente el header es seguido por los datos (payload) que transporta RTP y su formato es definido por la aplicación. El diseño del header de RTP busca llevar sólo aquellos campos que son necesarios para diversos tipos de aplicaciones. (Figura 2.1)

- **Version (V) (2 bits)**. Indica el número de versión RTP utilizada. La actual es la 2.
- **Padding (P) (1 bit)**. Informa que los datos de RTP llevan un "relleno" para complementar un bloque de cierto tamaño.
- **Extension (X) (1 bit)**. Indica si a continuación viene una cabecera de extensión. Su valor sería 1.
- **CSRC Count (CC) (4 bits)**. Este campo contiene el número de identificantes CSRC que siguen la cabecera fija, es decir, el número de fuentes que contribuyen y están ligadas a este paquete.
- **Maker (M) (1 bit)** Es un bit de señalización. Indicada para señalar elementos especiales como salirse de los límites.
- **Payload Type (PT) (7 bits)** *Etiqueta de Flujo*: Aquí se identifica el tipo de contenido (audio, video) que representa el tipo de codificación, de información transportada en el paquete.
- **Sequence Number (16 bits)** *Número de secuencia*: Este valor se va incrementando en 1 por cada paquete RTP enviado, pero su valor inicial es aleatorio. Se pueden detectar paquetes RTP perdidos.
- **TimeStamp (32 bits)** En un protocolo como RTP se utilizan marcas temporales para referenciar los paquetes emitidos. Estas informaciones son la base de los cálculos que permiten evaluar el retardo y la fluctuación introducidos por un sistema de comunicación.
- **Synchronization Source (SSRC) (32 bits)** *Fuente de Sincronización* Identifica la fuente que ha producido el paquete. Al principio de una sesión, cada participante

escoge un número de SSRC. En este caso se habla de sincronización ya que la escala de tiempo establecida por la fuente en estos paquetes va a servir de referencia a los receptores para restituir la información correctamente.

- **Contributing Source (CSRC)** (*Fuente Contribuyente*) De 0 a 15 instancias de este campo pueden estar presentes en la cabecera de paquete RTP. El número está indicado por el campo CC.

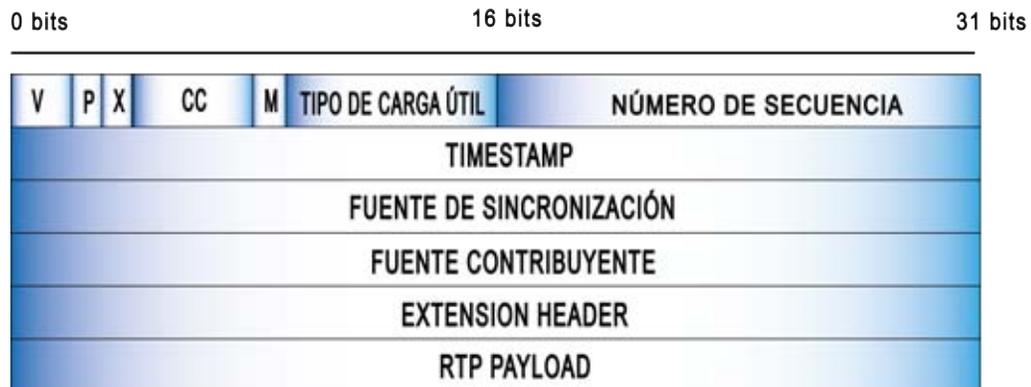


Figura 2.1: Protocolo RTP

De manera concreta RTP funciona así: Cada ciertos instantes de tiempo, dependiendo de la aplicación, se genera un paquete RTP, el cual contiene un encabezado con información acerca de la secuencia de los paquetes, y el codificador utilizado; este paquete se envía a través de UDP a los participantes, utilizando para ello un puerto específico y conocido por todos; una vez recibido se examina la información del encabezado para ver si han llegado los paquetes en la secuencia apropiada o se procede a su reordenación de esta forma se puede descomprimir los datos y reproducirlos en audio.

2.2.1.2. RTCP(Protocolo de Control de RTP)

Este protocolo se basa en transmisiones periódicas de paquetes de control para todos los participantes en la sesión. RCTP controla los flujos, por lo que permite transportar informaciones básicas de los participantes de una sesión y de la calidad de

servicio.

Existen cinco tipos de distintos paquetes RTCP para cada tipo de información:

1. **SR (Sender Report)**. Contiene las estadísticas de transmisión y de recepción para los participantes que son emisores activos.
2. **RR (Receiver Report)**. Contiene estadísticas de recepción para los participantes que no son emisores activos, pero sí receptores de una sesión.
3. **SDES (Source Description)**. Describe la fuente: nombre, email, teléfono, etc.
4. **BYE**. Permite a una estación indicar el fin de su participación en una sesión.
5. **APP**. Es un paquete de señalización específico a una aplicación.

El control de flujo RTP se realiza guardando una evaluación del número de participantes en una sesión (fuentes y receptores). A partir de esta evaluación, se calcula un intervalo de tiempo que sirve de periodo de recurrencia en la difusión de informaciones SR o RR, según el caso. Si se considera una sesión audio con dos participantes, los paquetes RTCP pueden ser emitidos cada 5 segundos, mientras que para 4 participantes son emitidos cada 10 segundos. Los paquetes que se transmiten con más frecuencia son SR y RR.

Encabezado RTCP Sender Report (SR): Tiene 32 bytes y se compone de los siguientes parámetros (Figura 2.2):

- **Version (V) (2 bits)**: indica la versión.
- **Padding (P) (1 bit)***Relleno*: Si está activo quiere decir que el paquete contiene algunos bits de padding al final que no forman parte de la información de control.
- **Reception Report Count (RC) (5 bits)***Conteo de informes de recepción*: Indica el número de informes de recepción contenidos en el paquete SR, considerando un informe para cada fuente.
- **Packet Type (PT) (8 bits)***Tipo de paquete*: Indica el tipo de paquete, si es un paquete SR será representado por el valor 200.

- **Length (16 bits)** *Longitud*: Indica la longitud total del paquete en palabras de 32 bits (cabecera y rellenos comprendidos)
- **SSRC of Sender (32 bits)**: Indica la identificación de la fuente específica del emisor.
- **NTP timestamp (64 bits)**: Marca de Tiempo NTP
- **RTP timestamp (32 bits)**: Marca de tiempo RTP
- **Senders packet count (32 bits)** *Conteo de paquetes del emisor*: Indica el número total de paquetes RTP transmitido por el emisor desde el inicio de sesión.
- **Senders octet count (32 bits)** *Bytes del paquete del emisor*: Indica el número total de octetos RTP (sólo considerados los octetos de datos de usuario y no del de cabecera o relleno), transmitidos por el emisor desde el principio de la sesión.
- **SSRC_n (32 bits)**: Especifica la identificación de la fuente en la sesión, que se refiere a los datos incluidos en el bloque RR.
- **Fraction Lost (8 bits)** *Paquetes perdidos*: Indica la fracción de paquetes perdidos desde el último informe emitido por este participante. La fracción representa la relación entre el número de paquetes perdidos y el número de paquetes esperados.
- **Cumulative number of packet lost (24 bits)** *Número acumulativo de paquetes perdidos*: Indica el número total de paquetes RTP de la fuente en cuestión que han sido perdidos desde el principio de la sesión RTP.
- **Extended highest sequence number received (32 bits)** *Extensión del número de secuencia más alto recibido*: Especifica el número de secuencia del último paquete RTP recibido desde la fuente SSRC_n.
- **Interarrival jitter (32 bits)** *Intervalo de la variación de retardo (jitter)*: Informa de la variación del retardo de transmisión de los paquetes RTP.
- **Last SR timestamp (LSR) (32 bits)** *Marca de tiempo del último informe de envío*: Este campo representa los 32 bits del medio del campo NTP Timestamp utilizado en el primer paquete SR recibido desde la fuente en cuestión.

2.2.2.1. SIP (Protocolo de Inicio de Sesión)

Este tipo de protocolo está basado en texto el cual fue definido por el *IETF*³. La última revisión de SIP fue realizada en el RFC 3261 de Junio de 2002. Se encarga como tal de iniciar, modificar o terminar la llamada de manera interactiva entre los usuarios, tiene la capacidad de saber donde se encuentra el usuario en todo momento, así como puede negociar parámetros para la sesión, como los códecs, las direcciones IP y tráfico media. Puede también integrarse con otros protocolos ya mencionados como el RTP.

Este protocolo tiene ciertas funcionalidades parecidas al HTTP y SMTP, por la forma en que identifica a los puntos terminales mediante el **URI** (*Uniform Resource Identifier*), ejemplo: *sip:usuario@dominio* o *direccion ip [:port]*, además de que SIP a través del tiempo ha sido extendido de tal manera que puede soportar mensajería instantánea y otros servicios como transferencia de llamadas, conferencias, parqueo, etc..

Sin embargo hay que hacer énfasis en que SIP solo es un protocolo de señalización, es decir que una vez establecida la sesión, los usuarios intercambian paquetes de audio y vídeo directamente a través del protocolo RTP, por lo que SIP como tal no puede asegurar la calidad de servicio, para ello se vale de otros protocolos como los que ya dijimos TCP y UDP. (Figura 2.3)

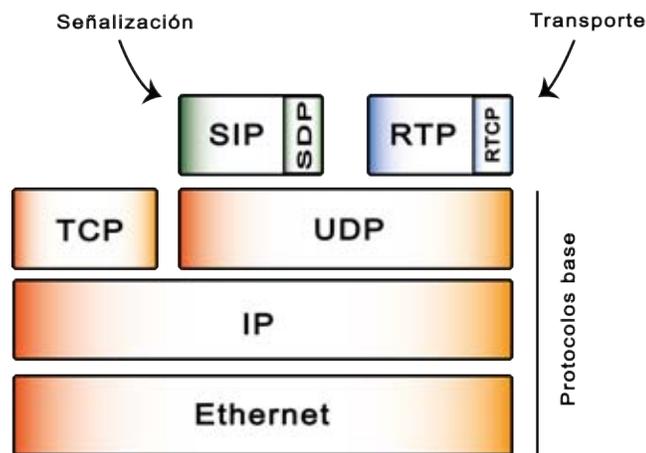


Figura 2.3: Protocolo SIP

³Internet Engineering Task Force

SIP tiene algunos componentes o entidades como son:

- **Agentes de Usuario**

UAC. *User Agent Client.* Que se identifica como la terminal o cliente encargada de iniciar o terminar la sesión SIP.

UAS. *User Agent Server.* Es el servidor que obedece a las peticiones que genera el UAC.

UA. *User Agent.* Simplemente es el dispositivo o terminal de red usado (teléfonos, softphone, etc.)

- **Servidores de red**

Servidor Proxy. Actúa como otros clientes y contiene funciones tanto de cliente como servidor. Un servidor de este tipo es capaz de interpretar y reescribir los encabezados antes de enviarlos a otros servidores.

Servidores de Redirección. Acepta las solicitudes SIP y envía una respuesta redirigida hacia el cliente con la dirección del siguiente servidor. Estos servidores no aceptan llamadas ni procesan o envían solicitudes SIP.

- **Direcciones SIP**

También llamadas *URL (Universal Resource Locators)*. Existen de la forma: *usuario@dominio*. Donde usuario puede ser un nombre o un número telefónico y el dominio puede quedar con un nombre de dominio o una dirección IP.

- **Localización de un servidor y de un usuario**

Para la localización de un servidor, un cliente puede enviar una solicitud SIP de dos maneras. La primera es muy sencilla ya que la aplicación conoce el servidor proxy. La segunda, va de la siguiente forma:

- El cliente determina la dirección IP y el número de puerto del servidor para la solicitud requerida.
- Cuando el número de puerto no sea dado en el URL-SIP de la solicitud, se deja el 5060 que es el que se toma por defecto.

- Si el tipo de puerto no está siendo dado en el URL-SIP, el cliente debe de intentar conectarse usando UDP y después TCP.
- El cliente consulta el servidor DNS para la IP de dominio, por lo que si no encuentra registros, éste es incapaz de localizar el servidor y continuar con su solicitud.

Ya que conocemos las entidades de las que se compone SIP, es importante conocer cómo se llevan a cabo las peticiones, por primera instancia es saber cómo se registra un usuario ante una base de localización, es decir, el usuario o UAC muestra sus credenciales ante el servidor para que posteriormente reciba o realice llamadas.

En este caso es el número de extensión **200** que tiene un dominio **voip.com**, aunque también podría ser un nombre y una dirección ip es decir podría quedar así:

ana@192.168.1.223 es quien está mostrando las credenciales. (Figura 2.4).

Más adelante se dará la interpretación de los mensajes que ahí aparecen.

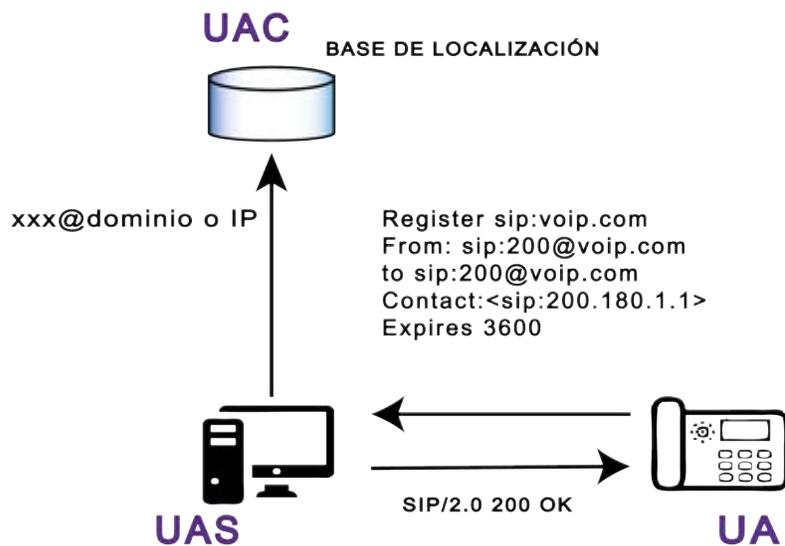


Figura 2.4: Sesión SIP

Mensajes SIP

El protocolo SIP es similar al HTTP, es a base de texto. Los *User Agent Client*(UAC) realizan las "solicitudes" y los *User Agent Server* (UAS) retoman las "respuestas" a las solicitudes que los clientes hacen. Existen dos tipos de mensajes: las *solicitudes o métodos* iniciadas por los clientes y las *respuestas o códigos de estado* enviadas por los servidores. Cada mensaje emplea un formato genérico indicado en el *RFC 2822* que consiste en una línea inicial seguida de uno o más campos de cabecera, una línea vacía que indica el final de las cabeceras, y por último, el cuerpo del mensaje que puede ser opcional. Éstos mensajes se envían sobre TCP o UDP.

GENERALES	ENTIDAD	SOLICITUD	RESPUESTA
Accept	Content-Encoding	Authorization	Allow
Accept-Encoding	Content-Length	Contact	Proxy-Authenticate
Accept-Language	Content-Type	Hide	Retry-After
Call-ID		Max-Forwards	Server
Date		Proxy-Authorization	www-Authenticate
Encryption		Proxy-Require	
Expires		Route	
From		Require	
Record-Route		Response-key	
Timestamp		Subject	
To		User-Agent	
Via			

Figura 2.5: Mensajes SIP

Los encabezados especifican la parte que llama, la parte llamada, la ruta o el tipo de mensaje de la llamada. Hay 4 grupos de estos encabezados (Figura 2.5):

1. Encabezados generales. Se aplican para solicitudes y respuestas.
2. Encabezados de entidad. Los que dan información acerca del tipo de mensaje y longitud.

3. Encabezados de solicitud. Permiten incluir al cliente información adicional de solicitud.
4. Encabezados de respuesta. Permite incluir al servidor información adicional de respuesta.

Solicitudes o Métodos

Estas solicitudes como ya se menciona arriba son iniciadas por los UAC. Son caracterizadas por la línea inicial del mensaje, de nombre *Request-line* la cual contiene:

- Nombre del método
- El identificador del destinatario de la petición (*Request URI*)
- Y la versión del protocolo

En la figura 2.6 se muestran los 6 métodos básicos SIP que están definidos en el *RFC 254* que describen las peticiones de los clientes:

Petición SIP	Descripción
INVITE	Petición enviada al usuario para poder establecer la comunicación. Contiene la información de quien genera la llamada y del destino de la misma así como el flujo de datos a intercambiar.
ACK	Es una petición o respuesta que se obtiene cuando el UAC que genera el INVITE ha recibido una respuesta 200 OK al final de la invitación y lo confirma con una respuesta ACK.
BYE	Finaliza la conexión entre usuarios.
CANCEL	Esta petición es usada para cancelar otra petición, por ejemplo, si un usuario envía un INVITE y aún no se ha concretado la llamada, el origen puede colgar y entonces SIP manda el CANCEL
OPTIONS	Se usa para saber las capacidades con las que cuenta un usuario, por ejemplo, el códec usado, el idioma, etc.
REGISTER	El REGISTER es usado por un usuario para poder ser registrado en la base de localización.

Figura 2.6: Solicitudes o Métodos SIP

Respuestas o Códigos de estado.

Una vez que ya se recibió e interpretó el mensaje de solicitud el UAC, el receptor del mismo responde con un mensaje, siendo casi similar a la descripción anterior, ya que en la línea inicial que se denomina *Status-line*, contiene la versión del SIP, el código de la respuesta conocido como *Status-code* y una descripción nombrada *Reason Phrase*. Estas respuestas están compuestas por tres dígitos que hacen la clasificación de los tipos que se tienen, el primero de ellos define la clase de respuesta que se trata.

En la figura 2.7 se muestra una lista general de las respuestas SIP.

Respuestas SIP	Descripción
1xx	Respuestas informativas
2xx	Respuestas de éxito
3xx	Respuestas de dirección
4xx	Errores de solicitud
5xx	Errores de servidor

Figura 2.7: Respuestas SIP

En la Figura 2.4 se ha referenciado una respuesta *200 OK*, este tipo de respuesta es una de entre muchas otras que el protocolo nos indica dentro de cada una de las solicitudes, *200 OK* significa que la sesión ha sido establecida.

Ya se tiene conocimiento de las peticiones y de las respuestas SIP, sin embargo aún falta saber los mensajes que también envía el protocolo SIP. Siempre envía mensajes por cada cosa que hace, desde el registro de una extensión, hasta la generación de llamadas. A continuación se muestra el proceso que ocurre para el registro de un usuario y una llamada a otra extensión, con la ayuda de un analizador de protocolos de red como *Wireshark*⁴ y desde la consola de Asterisk, se obtendrá la información que nos indicará cada punto del mismo:

En la figura 2.8 se indica en la primer respuesta que la dirección IP 192.168.11.54 perteneciente a un usuario con extensión 104 quiere registrarse al servidor con la dirección IP 192.168.11.130. Su User-Agent es un softphone marca Jitsi que se tiene instalado en un sistema operativo Windows XP.

⁴<https://www.wireshark.org/>

```

SIP read from UDP:192.168.11.54:5060
REGISTER sip:192.168.11.130 SIP/2.0
Call-ID: 95e3fe467d2b32968f178024cdbdb263@0.0.0.0
CSeq: 1 REGISTER
From: "104" sip:104@192.168.11.130;tag=886b0cfb
To: "104" sip:104@192.168.11.130
Via: SIP/2.0/UDP
192.168.11.54:5060;branch=z9hG4bK-323430-101efa437867a9b12795651f1ff5
Max-Forwards: 70
User-Agent: Jitsi2.4.4997Windows XP
Expires: 600
Contact: "104" sip:104@192.168.11.54:5060;transport=udp;registering_acc=192_168_11_130;expires=600
Content-Length: 0

```

Figura 2.8: Registro de un UA SIP

En la figura 2.9 se indica que el servidor le contesta mediante la respuesta *SIP 401 Unauthorized*, ya que para poder registrarse en su base de localización, es necesario contar con ciertas credenciales como un *Usuario* y una *Contraseña*

```

SIP/2.0 401 Unauthorized
Via: SIP/2.0/UDP
192.168.11.54:5060;branch=z9hG4bK-323430-101efa437867a9b12795651f1ff5;received=192.168.11.54;rport=5060
From: "104" <sip:104@192.168.11.130>;tag=886b0cfb
To: "104" <sip:104@192.168.11.130>;tag=as139d1e94
Call-ID: 95e3fe467d2b32968f178024cdbdb263@0.0.0.0
CSeq: 1 REGISTER
Server: pbx-asterisk
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, SUBSCRIBE, NOTIFY,INFO, PUBLISH
Supported: replaces, timer
WWW-Authenticate: Digest algorithm=MD5, realm="asterisk", nonce="57c209eb"
Content-Length: 0

```

Figura 2.9: SIP 401 Unauthorized

La figura 2.10 muestra que el usuario de la extensión 104 le regresa las credenciales que requiere con una petición de *REGISTER* nuevamente.

```
SIP read from UDP:192.168.11.54:5060
REGISTER sip:192.168.11.130 SIP/2.0
Call-ID: 95e3fe467d2b32968f178024cdbdb263@0.0.0.0
CSeq: 2 REGISTER
From: "104" <sip:104@192.168.11.130>;tag=886b0cfc
To: "104" sip:104@192.168.11.130
Max-Forwards: 70
User-Agent: Jitsi2.4.4997Windows XP
Expires: 600
Contact: "104" <sip:104@192.168.11.54:5060;transport=udp;registering_acc=192_168_11_130>;expires=600
Via: SIP/2.0/UDP
192.168.11.54:5060;branch=z9hG4bK-323430-3cd058658a5c2ad098458d2dcc0e
Authorization: Digest
username="104",realm="asterisk",nonce="57c209eb",uri="
sip:192.168.11.130",response="34d18d7962020b5c1b35106a32"
Content-Length: 0
```

Figura 2.10: Petición de credenciales

La figura 2.11 indica que el servidor le regresa una respuesta de *OPTIONS* cuya función es saber las capacidades o cualidades del servidor. Como por ejemplo: tipo de media que es soportado, el idioma o el hecho de que el UA no este disponible.

```
OPTIONS sip:104@192.168.11.54:5060;transport=udp;registering_acc=192_168_11_130
SIP/2.0
Via: SIP/2.0/UDP 192.168.11.130:5060;branch=z9hG4bK367ec330;rport
Max-Forwards: 70
From: "Unknown" <sip:Unknown@192.168.11.130>;tag=as1ddb3f89
To: sip:104@192.168.11.54:5060;transport=udp;registering_acc=192_168_11_130
Contact: sip:Unknown@192.168.11.130:5060
Call-ID: 67d32ca71337407d1698e80b4ad04f54@192.168.11.130:5060
CSeq: 102 OPTIONS
User-Agent: pbx-asterisk
Date: Mon, 29 Sep 2014 21:55:09 GMT
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, SUBSCRIBE, NOTIFY,INFO, PUBLISH
Supported: replaces, timer
Content-Length: 0
```

Figura 2.11: Respuesta OPTIONS

Por último, en la figura 2.12 se ve que le da una respuesta *200 OK* lo que significa que el usuario ha sido registrado.

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP
192.168.11.54:5060;branch=z9hG4bK-323430-3cd058658a5c2ad098458d2dcc0e
received=192.168.11.54;rport=5060
From: "104" <sip:104@192.168.11.130>;tag=886b0cfb
To: "104«sip:104@192.168.11.130>;tag=as139d1e94
Call-ID: 95e3fe467d2b32968f178024cdbdb263@0.0.0.0
CSeq: 2 REGISTER
Server: pbx-asterisk
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, SUBSCRIBE, NOTIFY,INFO, PUBLISH
Supported: replaces, timer
Expires: 600
Contact: <sip:104@192.168.11.54:5060;transport=udp;registering_acc=192_168_11_130>;expires=600
Date: Mon, 29 Sep 2014 21:55:09 GMT
Content-Length: 0
```

Figura 2.12: Respuesta 200 OK

Un mensaje de establecimiento de llamada SIP INVITE (Figura 2.13) está emitido por parte del *UA SIP (IP 10.33.6.101)*, el cual llama al Proxy Server que tiene la IP 10.33.6.102. Este último va a la base de datos de localización para identificar a quién se está llamado (IP 10.33.6.100) y encamina la llamada a su destino.

El mensaje *INVITE* contiene distintos "headers" o encabezamientos obligatorios, entre los que se encuentran: la dirección SIP de la persona que llama, identificándola por un *From : 107 sip:107@10.33.6.101*, la dirección SIP de quien recibe la llamada, que se distingue por un *To : 104 sip:104@10.33.6.100*, una identificación de la llamada, que se define por *Call-Id: 23456789@10.33.6.102*, un número de secuencia: *CSeq: 1 INVITE*, un número máximo de saltos: *Max-Forwards : 20*.

El encabezamiento *Via* está actualizado por todas las entidades que participaron al enrutamiento del requerimiento INVITE. Eso asegura que la respuesta seguirá el mismo camino que el requerimiento. Por otra parte, el requerimiento SIP INVITE contiene una sintaxis *SDP (Session Description Protocol)*. Esta estructura consiste en varias líneas que describen las características del media que el que llama necesita para establecerla.

```

INVITE sip:104@10.33.6.101 SIP/2.0
Via : SIP/2.0/UDP 10.33.6.102:5060
Max-Forwards : 20
To : 104 sip:104@10.33.6.101
From : 107 sip:107@10.33.6.100
Call-Id: 23456789@10.33.6.102
CSeq: 1 INVITE
Contact: 107@10.33.6.108
Content-Type: application/sdp
Content-Length:162

```

Figura 2.13: Descripción del mensaje INVITE

La figura 2.14 muestra una captura de la traza obtenida con *Wireshark* de la llamada descrita en la figura 2.13.

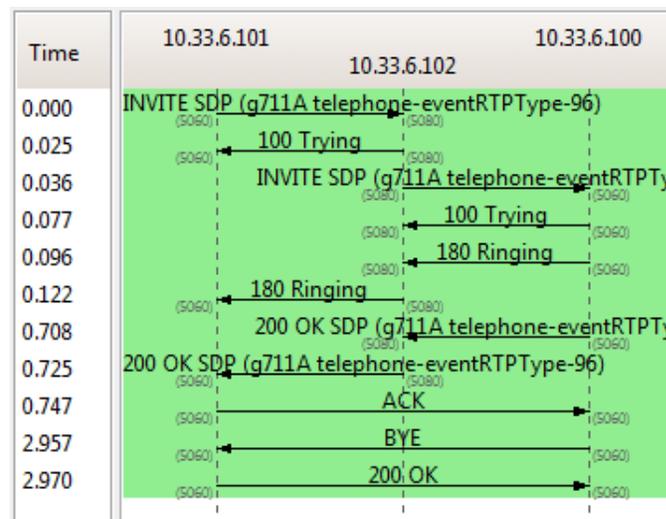


Figura 2.14: Traza SIP con Wireshark

La respuesta *180 RINGING* es enviada por el destinatario (10.33.6.100) al UA del que genera la llamada (10.33.6.101). Cuando el destinatario acepta la sesión, la respuesta *200 OK* esta emitida por el UA y encaminada hacia el UA de quien genera la llamada.

El UA de quien hace la llamada devuelve un método *ACK* al destinatario, relevada por la entidad Proxy Server (10.33.6.102) . La entidad Proxy Server participa al encaminamiento de la señalización entre UAs mientras que los UAs establecen directamente canales RTP para el transporte de la voz o del video en forma de paquetes sin implicación del Proxy Server en este transporte.

Cuando 10.33.6.100 cuelga, su UA envía un requerimiento *BYE* para terminar la sesión. Este requerimiento esta entregado al Proxy Server quien lo encamina al UA de la IP 10.33.6.101. Este último devuelve la respuesta 200 OK.

SDP (Protocolo de Descripción de Sesión)

SIP utiliza SDP para definir el tipo de media de la comunicación. Usan un modelo de oferta/respuesta, en donde en la oferta se ofrecen distintos tipos de protocolos, direcciones, puertos y payloads para establecer la comunicación. Y en la respuesta se definen los que serán usados.

SDP, es un protocolo que define el formato para describir los parámetros de inicialización de una comunicación multimedia. Se describe en el RFC4566 por la IETF.

Incluye la siguiente información:

- Nombre y propósito de la sesión
- Tiempos de inicio y fin de la sesión
- Los tipos de medios que comprende la sesión
- La información necesaria para establecer la sesión. Como la dirección IP a la que se le enviarán los datos, los números de puertos involucrados y los esquemas de codificación.

SDP entrega la información en formato ASCII usando una secuencia de líneas de texto. Usando la forma: `<tipo>=<parámetro1><parámetro2>...<parámetroN>`.

La figura 2.15 muestra una secuencia completa de SDP:

```

v=0
o=root 21928 21928 IN IP4 201.152.8.65
s=session
c=IN IP4 201.152.8.65
t=0 0
m=audio 16392 RTP/AVP 18 101
a=rtpmap:18 G729/8000
a=fmtp:18 annexb=no
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16
a=silenceSupp:off
a=ptime:20
a=sendrecv
+-----+

```

Figura 2.15: SDP

■ Protocol Version

v=0

Número de versión del protocolo. La versión actual es 0. Un mensaje válido SDP siempre empieza de esta manera.

■ Origin

o=200.76.114.240 1394689842 1394689842 IN IP4 200.76.114.240

o=<username><session id><version><network type><address type><address>):

- El campo username puede contener el nombre del creador de la conexión, el host o un guión(-) si ninguno de los anteriores.
- Session id es un número aleatorio usado para asegurar la unicidad.
- Version es otro campo numérico necesario para poder descubrir cuál de varios anuncios para la misma sesión es el más reciente.
- El network-type es siempre IN para Internet.
- Address type es la versión del protocolo IP a utilizar, IP4 o IP6.

- Address es la dirección ya sea en forma decimal con puntos o un nombre de host completo.

■ **Session Name**

s=Session SDP

Es el nombre de sesión

■ **Connection Data**

c=IN IP4 200.76.114.213

(c=<network type><address type><connection address>):

Muestra la información de conexión. Network type IN para Internet y address type, IP4 o IP6. Connection address indica la dirección IP que enviará los paquetes unidifusión.

■ **Times, Repeat Times and Time Zones**

t=0 0

(t=<start time><stop time>):

Son los campos que especifican el inicio y final de sesión. Si stop time se pone a cero, entonces el período de sesiones no se limita, a pesar de que no se convertirá en activa hasta después de start time. Si el tiempo de inicio es también cero, la sesión se considera como permanente.

■ **Media Description, name and address**

m=audio 10750 RTP/AVP 18 8 0 17 96

(m=<media><port><transport><format list>)

Aquí se tiene la información sobre el tipo de datos que se transporta.

- Media corresponde al tipo de datos a transportar (audio)
- Port, al puerto ofrecido para establecer la sesión. (10750)
- En transport se indica el protocolo de transporte a utilizar, RTP/AVP (AVP: Audio Video Profiles).

- Format list enumeran los payloads ofrecidos para la comunicación, ya que se permite ofrecer múltiples opciones de códecs. Hay algunos ya definidos, como por ejemplo: 8 G.711 PCMA, 18 G.729, 4 G.723, 0 G.711 PCMU

■ Attribute rtpmap

a=rtpmap:18 G729/8000

a=rtpmap:8 PCMA/8000

a=rtpmap:0 PCMU/8000

a=rtpmap:17 T38/8000

a=fmtp:96 0-15

a=rtpmap:96 telephone-event/8000

a=ptime:20

(a=rtpmap:<payload type><encoding name>/<clock rate>):

Se trata de una lista de los formatos de códecs con información de Sample rate o frecuencia de muestreo, Fieldname, etc.

- En payload type se encuentra un número que indica el payload ofrecido. En caso de estar ya definido no es necesario este campo, hay algunos que pueden utilizar un payload dinámico dentro del rango 97-127 como son el caso de G.726 y Speex.
- En encoding name y clock rate, encontramos el nombre y la frecuencia de muestreo del códec, respectivamente.

■ Media Attribute

a. SendRecv.

Modo envío/recepción.

2.2.2.2. H.323

El protocolo H.323 fue diseñado por la *ITU*, con el objetivo de poder transmitir audio, video y datos a través de las redes basadas en la conmutación de paquetes pero sin garantizar la calidad de servicio. En un principio tan solo estaba soportado para

funcionar en redes de área local, sin embargo con el tiempo fue habilitado para poder funcionar en redes de área amplia como el Internet.

Como tal, el H.323 mantiene una estructura muy completa en cuanto a la arquitectura y diseño, ya que en ello almacena diferentes protocolos y códecs para poder garantizar la compatibilidad entre los diversos dispositivos. (Figura 2.16)

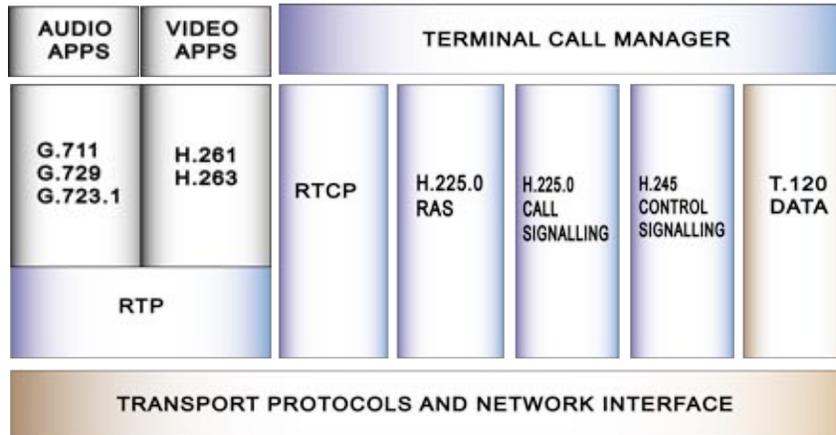


Figura 2.16: Protocolo H.323

Los protocolos con los que cuenta son los siguientes.

- *El H.225*: es utilizado para tener cierto control de llamadas, ya que puede describir la forma del audio y datos en información, en redes basadas en conmutación de paquetes.

En el H.225 se manejan dos partes importantes,

1. La que se encarga de la señalización de las llamadas en formato Q.931, la cual configura las conexiones entre los extremos de H.323 (gateways y/o terminales), que al igual que SIP se consigue mediante intercambio de mensajes entre el protocolo.
2. La que se encarga del registro, admisión y estado de puntos terminales (RAS).

Éste maneja un canal de control de llamadas el cual está establecido a través del puerto 1720 en TCP, aquí es donde se crean los mensajes entre el protocolo para poder iniciar, mantener y finalizar la llamada entre terminales.

La señalización de la llamada se puede crear de manera directa entre terminales, o bien puede hacerse por medio de un *Gatekeeper*, que funciona como el gestor

de las conexiones entre las terminales; éstos pueden ser 1 o más.

A continuación se mencionan algunos de los mensajes que aparecen por *Q.931*.

- *Setup*
 - *Setup Acknowledge*
 - *Call Proceeding*
 - *Progress Alerting*
 - *Connect*
 - *User Information*
- *El H245*. Este protocolo es utilizado principalmente para controlar los canales multimedia, de tal manera que se encarga de intercambiar las distintas capacidades entre los puntos terminales, además de determinar cuál de los puntos finales o terminales asumirá el papel de maestro y cuál el de esclavo, de tal modo que debe realizarse de manera paralela con el *H.225* para poder tener mejor control de los datos transmitidos.

Como tal *H.245* podría identificarse en fases.

1. *Fase TCS o Terminal Capability Set*, es la capacidad de envío de mensajes entre puntos terminales, que es donde se lleva a cabo la negociación de los códecs, DTMF, y demás, entre ambos puntos terminales.
 2. *Fase MSD o Determinacion de Maestro esclavo*, como ya se ha mencionado antes, el protocolo se encarga de determinar que terminal será el maestro y cuál el esclavo, de tal modo de que por ejemplo existan diferentes códecs habilitados, el maestro es el que determina el códec a escoger, el maestro es elegido o determinado en base a la función del *Gateway*.
 3. *Fase OLC Open Logical Chanel o Apertura de canal lógico*, es donde se hace la negociación de puerto, dirección IP, *RTP/RTCP*.
- *El H.235*. Es el protocolo encargado de la seguridad y cifrado.

El *Gatekeeper* es el punto central en una red *H.323*, que aunque técnicamente no es obligatorio tenerlo, se puede decir que sí cumple tareas muy importantes en la comunicación y que como tal, es un conjunto de *MCU*⁵, *Gateways* y terminales.

El *Gatekeeper* maneja tareas como la de traductor de direcciones, es decir que interpreta números, secuencias de caracteres, direcciones *URL*, emails y las relaciona con el otro punto de la terminal a través de direcciones IP; sin un *Gatekeeper*, se necesitaría forzosamente conocer la dirección IP de cada terminal.

Así mismo el *Gatekeeper* también se encarga de controlar los accesos de las terminales, es decir, para estar en función deberán estar autenticadas ante el *Gatekeeper*, o sea que deben pertenecer a su zona para poder realizar o recibir algún tipo de llamada. El *Gatekeeper* también tiene un control sobre el ancho de banda, de esta manera el administrador puede controlar el número de llamadas simultáneas evitando el congestionamiento de la red. Además de que tiene la función de intercambiar tablas de ruteo, es decir, si existieran dos *Gatekeepers* en dos zonas distintas, para poder intercomunicarse es necesario poder intercambiar sus tablas de ruteo entre ambos *Gatekeepers*. (Figura 2.17)

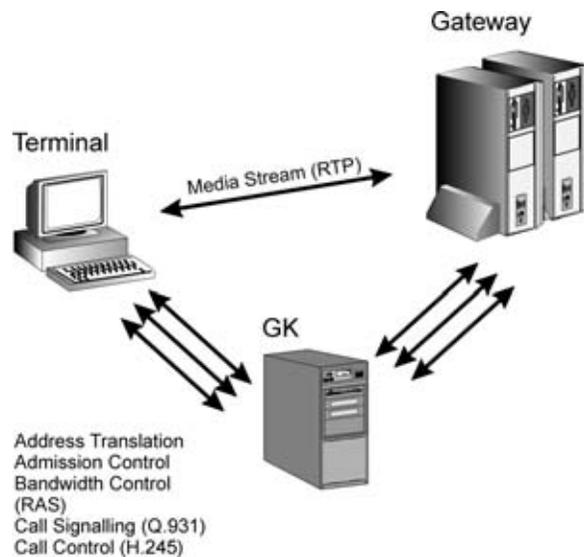


Figura 2.17: Gatekeeper

⁵Unidades de Conferencia Multimedia. Es responsable de controlar las sesiones y de efectuar el mezclado de los flujos de audio, datos y video.

En modo de resumen, podemos decir que el H323 trata de lo siguiente:(Figura 2.18)

- Fue diseñado por la ITU.
- Transmite audio y video a través del protocolo de Internet.
- No maneja calidad de servicio.
- Ofrece señalización, control y transporte multimedia.
- La señalización es muy rápida por usar formato binario.
- H.323 contiene una suite de protocolos como el H245, el H.235.

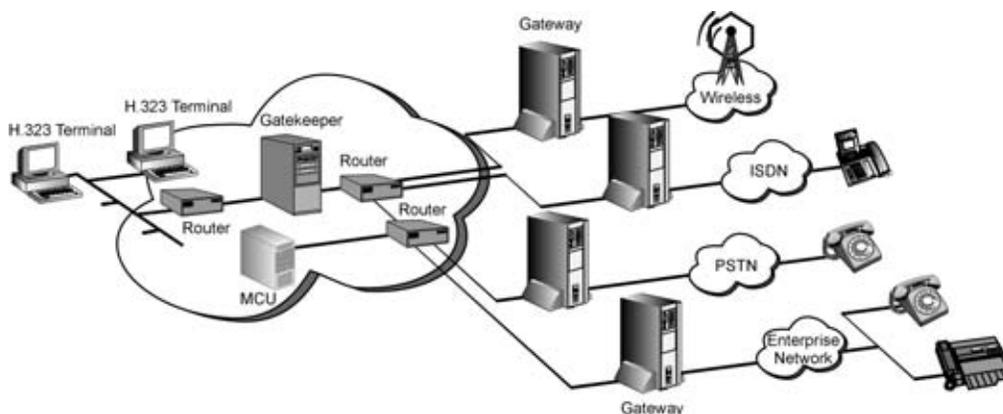


Figura 2.18: Típica red H323

2.2.2.3. IAX2 (Inter-Asterisk Exchange2)

Asterisk como tal puede soportar los protocolos ya mencionados como *SIP* o *H.323*, sin embargo también es poseedor de un protocolo propio el cual está dirigido para poder interconectarse con otros conmutadores de Asterisk llamado *IAX* o *Inter-Asterisk Exchange*, como tal el protocolo se encuentra en su versión 2.

IAX2 puede proporcionar control y transmisión de voz y video sobre tecnología IP, utilizando antecedentes de los otros protocolos *SIP*, *MGCP* y el *RTP*, por lo que :

- Minimiza el uso de ancho de banda.
- Transparencia en *NAT*.
- Soporta la implementación de otros recursos como *paging* e intercomunicación.

IAX2 hoy por hoy es ya el *RFC 5456* que es aprobado por el *IETF*, por lo que su aplicación puede estar a la par con *SIP* o *H.323*.

Las principales características que tiene son:

- Utiliza un único puerto *UDP (4569)* entre terminales, señalización y datos, ya que en otros protocolos se usa un puerto distinto para señalización y otro para la voz.
- La transmisión de datos es en *In-Band*, evitando así problemas con los cortafuegos o el *NAT*.
- *Trunking*, que no es otra cosa mas que una troncal donde un mismo flujo de datos puede conducir información de muchas conversaciones al mismo tiempo, con esto se reduce el consumo de ancho de banda.
- A diferencia del protocolo *SIP*, este es un protocolo binario y por tanto, cada bit o conjunto de bits tiene un significado.

Como puntos negativos se puede decir lo siguiente.

- Ya que la señalización y el tráfico de voz pasan por el mismo canal, tiene que pasar forzosamente por el servidor, a diferencia de *SIP*, que los teléfonos usan la señalización por medio del servidor pero el tráfico de voz puede irse directo entre teléfonos, lo que puede dar un ahorro de ancho de banda considerable.
- *IAX2* aún no es muy extendido, ya que solamente Asterisk y algunos otros dispositivos lo controlan, por lo que no es muy común troncalizar con proveedores de telefonía IP mediante este protocolo.

Funcionamiento de IAX2.

Al igual que *H.323* o *SIP*, *IAX2* también maneja su comunicación a través de un diálogo de flujo de datos. Básicamente se compone de tres etapas:

- **Conexión o Establecimiento de la llamada.**

En la figura 2.19 se mencionan dos terminales, la A y B, donde A desea realizar una llamada con B, en el primer paso A manda un mensaje *NEW* hacia B, entonces

B responde con un mensaje de *ACCEPT*, A contesta con un *ACK* o "acknowledgement", que es la confirmación de respuesta de *ACCEPT*, posteriormente contesta con un *RINGING*, y nuevamente A contesta de recibido con un *ACK*, y ya establecida la llamada B le contesta con un *ANSWER* y nuevamente se contesta con un *ACK*, que quiere decir que la llamada ya esta completa.

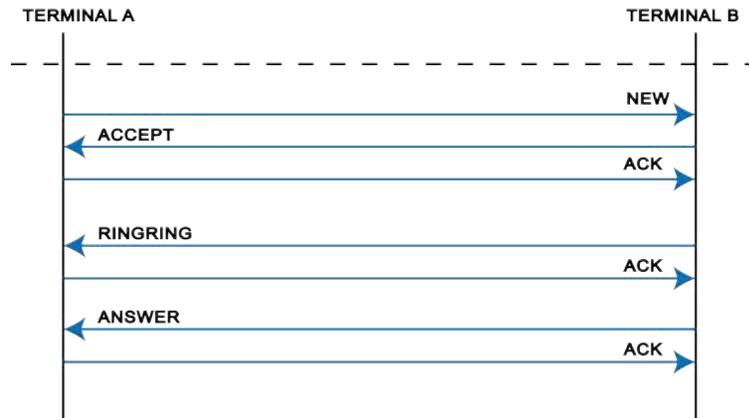


Figura 2.19: Establecimiento de llamada

■ Flujo de Datos o de Audio.

Una vez que ya ha sido establecida la llamada, se hace el intercambio de audio a través de *frames*, los "M" se refiere a *Mini-Frames* que son usados únicamente para la reducción de ancho de banda, y "F" se refiere a *Full-Frames*, los cuales maneja la información necesaria de sincronización. (Figura 2.20)

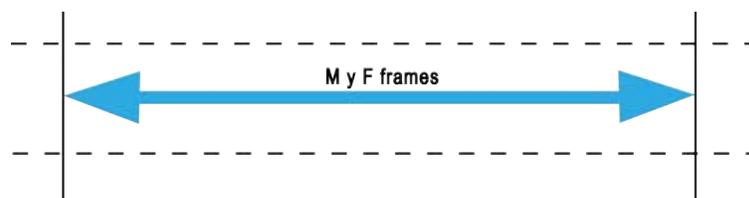


Figura 2.20: Flujo de datos

■ Desconexión.

La finalización de la llamada es tan simple como mandar un mensaje de *Hangup* y el otro lado contesta solo con un *ACK*. (Figura 2.21)



Figura 2.21: Desconexión

Los Frames.

Como se mencionó anteriormente el uso de "frames" en el audio es de suma importancia para el flujo de datos de audio, éstos son binarios, por lo que cada bit tiene un significado en particular.

- **Full-Frames.** Pueden enviar señales o datos de los medios. Generalmente se utilizan para controlar el inicio, configuración y resolución de un IAX, pero también pueden ser utilizados para transportar datos de la secuencia (aunque esto es generalmente no óptima). (Figura 2.22)

El *Full Frame* se envían de forma fiable, por lo que todos requieren de una respuesta inmediata a la recepción. Esta puede ser a través de un mensaje *ACK*. La longitud de la cabecera completa del marco estándar es de 12 octetos.

- **F:** Este bit indica con un 1 si la trama es *Full frame* y con un 0 en caso de no ser.
- **Source Call Number:** (*Número de llamada de origen*). Está compuesta por 15 bits, que son los que identifican el número de conversación origen, pueden existir varios multiplexados por la misma línea.
- **R:** Este bit es puesto a 1 si el frame está siendo retransmitido, si el bit está puesto a 0 quiere decir que es la primera vez que se transmite.
- **Destination Call Number:** (*Número de llamada de destino*) De la misma forma que *Source Call Number*, también está compuesto por 15 bits, solo que en esta ocasión el número indetificador es de el destino.

- **Timestamp:** Marca el tiempo de cada paquete en forma incremental desde la primera transmisión, utiliza 32 bits.
- **OSeqno:** (*Sec. de salida*). Es un número de secuencia del flujo de salida, está compuesto con 8 bits, comienza en 0 y se va incrementando progresivamente, cuando el contador se desborda se resetea a 0.
- **ISeqno:** (*Sec. de entrada*). Es un número de secuencia del flujo de entrada, está compuesto con 8 bits, comienza en 0 y se va incrementando progresivamente, cuando el contador se desborda se resetea a 0.
- **Frame Type:** (*Tipo de trama*). Este campo indica el tipo de trama que trata.
- **C.** Este bit determina como los 7 bits restantes del campo subclase se codifican. Puesto a 0 indica que el campo debe tomarse como 7 bits en un solo mensaje, y puesto a 1 indica que el campo debe tomarse a 14 bits es decir 2 mensajes.

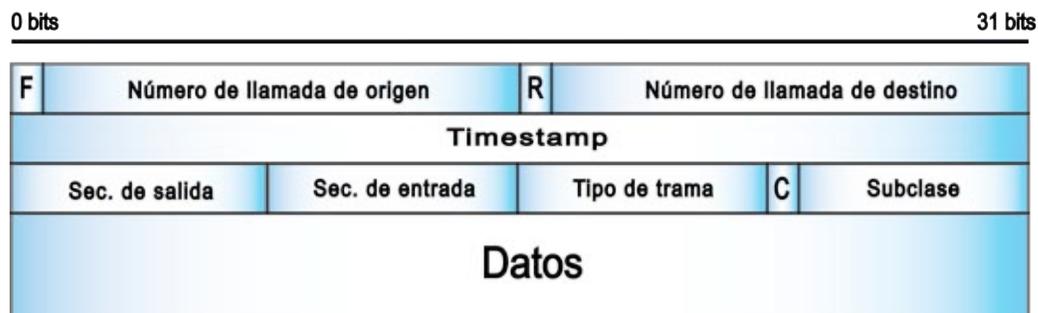


Figura 2.22: Full Frame

- **Mini-Frames.** Su cabecera es de cuatro octetos, no manejan ningún control o señalización de los datos, estas tramas no tienen por que ser respondidas, a diferencia de *Full-Frame* si alguna de estas tramas se pierde simplemente se descarta. (Figura 2.23)

- **F.** Debe tener un valor de 0 para especificar que no se trata de un Full-Frame.
- **Source Call Number:** (*Número de llamada de origen*). Es el número de llamada origen.
- **Time Stamp.** En esta ocasión está truncado a solo 16 bits para poder aligerar la cabecera, los clientes son los responsables de llevar un *timestamp* de 32

bits y una trama "F" para poder sincronizar.

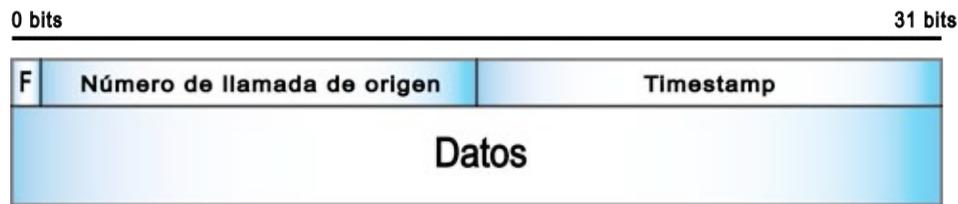


Figura 2.23: Mini Frame

2.2.3. Codificación y decodificación de voz (Códex)

Un códec es un conjunto de transformaciones utilizadas para digitalizar la voz. Los códec convierten tanto la voz en datos (bits) como los datos en voz. El códec tiene como función tomar una señal analógica para convertirla en una señal digital, o sea, en un formato binario (0s y 1s), y por último regresarla a una señal analógica. Existen diversas formas de digitalizar el audio y cada una de esas resulta en un tipo de códec. Cuando se usa una mayor compresión el resultado es una mayor distorsión (peor calidad). Se puede decir que un códec es mejor cuando es capaz de ofrecer mejor calidad de voz usando la misma cantidad de ancho de banda.

Códec G.711

G.711 es un estándar de la *ITU-T* que se usa para la compresión de audio. También se le conoce como *PCM (Pulse Code Modulation)*. Se usa principalmente en telefonía y salió en el año 1972.

Éste es un códec que se usa para representar señales de audio con frecuencias de la voz humana, mediante muestras comprimidas de una señal de audio digital con una tasa de muestreo de 8000 muestras por segundo. Cada muestra es de 8bits, por lo que al multiplicar 8000×8 , da 64kbps, esto significa que no usa compresión. Es una alternativa

cuando existe suficiente ancho de banda pues ofrece un alto rendimiento en redes de área local (LAN) donde existe una gran cantidad de ancho de banda disponible, por lo que si en los puntos en donde se pretenda tener una conversación no se tiene suficiente ancho de banda la voz se puede entrecortar.

Tiene 2 estándares de compresión: *el microlaw (ulaw)* y *el alaw*. También les conoce como *G711u* y *G711a* respectivamente. El microlaw se usa normalmente en Norteamérica y el alaw en Europa.

La familia de códecs *G711* no requieren de gran procesamiento y por eso están disponibles en la mayoría de los equipos de voz por IP. Pero hay veces que el uso del *G.711* no es muy viable porque requiere demasiado ancho de banda como ya se había mencionado.

Códec ILBC (Código de Internet de Ancho de Banda Bajo)

Éste es un códec libre y gratuito que trabaja en anchos de banda muy reducidos. Fue desarrollado por *Global IP Sound (GIPS)*. Se define en el *RFC3951*.

Está diseñado para aplicaciones VoIP, envíos de audio, archivos y mensajes. Es uno de los códecs utilizados por *Gizmo5*, *WebRTC*, *Ekiga*, *Google Talk*, etc.

El códec funciona con una codificación de longitud de 20 o 30 ms, tiene una frecuencia de muestreo de 8kHz/16bits y genera un archivo comprimido de flujo de bits con un bit-rate de 15.2 a 20ms o 13.3 kbps a 30ms.

Se utiliza un bloque independiente de la técnica de codificación de predicción lineal que evita la propagación de errores a través de las tramas. Este códec permite la degradación suave de la voz ocasionada por pérdida o retraso de paquetes. Los bloques comprimidos tienen que estar encapsulados en un protocolo capaz de ser transportados, por ejemplo *RTP*.

Una desventaja es que por su complejidad y reducido consumo de ancho de banda requiere una cantidad importante de procesamiento, por lo que al mantener numerosas llamadas recurrentes puede ocasionar el agotamiento de ciclos del procesador fácilmente.

Códec G.729

Es un algoritmo estándar que comprime y descomprime una señal de audio digital. Desarrollado por la *ITU*. Este códec está bajo licencia de cobro, sin embargo, Intel, como empresa creadora (entre otras), liberó hace años una versión de éste códec que ha ido evolucionando.

Se usa mayoritariamente en aplicaciones de Voz por IP, debido a que ofrece una alta compresión y bajos requerimientos en ancho de banda. Este estándar opera a una tasa de bits de 8 kbit/s, pero existen extensiones que suministran también tasas de 6.4 kbit/s y de 11.8 kbit/s para peor o mejor calidad en la conversación respectivamente.

El códec es licenciado por un canal, este canal se define como una única conexión de un punto de extremidad para una aplicación Asterisk, o una llamada bidireccional entre dos puntos conectados al Asterisk.

Existen varias versiones de este códec:

- **G729**: es el códec original
- **G729A o anexo A**: es una simplificación de G729 y es compatible con G729. Es menos complejo pero tiene algo menos de calidad.
- **G729B o anexo B**: Es G729 pero con supresión de silencios y no es compatible con las anteriores.
- **G729AB**: Es g729A con supresión de silencios y sería compatible solo con G729B.

G729 usa una técnica llamada *CS-ACELP*⁶ la que se basa en métodos alternos de muestreo y expresiones algebraicas como "libro de códigos" para predecir la representación numérica real. Estas expresiones algebraicas se envían al sitio remoto, donde son decodificadas y el audio es sintetizado para imitar el audio original; la predicción y síntesis de forma de onda de audio degrada la calidad de la señal de voz haciendo que la voz suene robótica.

Su compresión permite hacer más llamadas sin necesidad de aumentar la capacidad de

⁶Conjugative-Structure Algebraic-Code-Excited Linear Prediction

red y permite que la voz viaje en conexiones de banda ancha limitadas. Además de esto, ofrece una reducción de ruidos, compresión de silencios, no requiere un dispositivo de gran capacidad para su uso pese a su compresión.

Códec GSM

GSM o Global System for Mobile communications, o Sistema Global para las comunicaciones móviles, es un códec que se utiliza en los canales de las líneas móviles. Lo que importa más en este códec es la cantidad no la calidad, ya que el flujo de datos en una conexión full-rate es de 13kb/s.

Con esto se ahorra ancho de banda ya que casi consume 5 veces menos que el alaw, pero si se trabaja con conexiones de mala calidad a nivel de tiempo de respuesta (el tiempo que tarda en ir y regresar un paquete enviado), la voz podría llegar a distorsionarse tanto hasta hacerse incomprensible.

2.2.4. Ancho de banda y códecs

De la mano de los códecs se encuentra el ancho de banda, esto significa que se debe calcular el ancho de banda necesario para el funcionamiento óptimo de las llamadas, o en su defecto para escoger el códec adecuado, dependiendo el ancho de banda que se tenga. En ambos casos, se debe tener en cuenta que la Telefonía sobre IP se divide prácticamente en dos partes:

- La señalización, que es de lo que se encarga *SIP*, es decir, el establecimiento de las llamadas, la desconexión de las mismas, timbrados, etc..
- El audio. El consumo de ancho de banda, que por parte de la señalización es prácticamente despreciable, pero el flujo de *RTP*, que es donde viaja el audio, el cual su consumo si se debe de tomar en cuenta para su óptimo funcionamiento.

Como ya se ha mencionado, la voz no se puede manejar directamente ya que es demasiado peso para su envío, por lo que requiere una codificación y decodificación (un códec), como los indicados anteriormente. Al viajar por la red el paquete de voz debe ser

encapsulado por diferentes cabeceras haciendo que su peso se incremente, por lo que se debe tomar en cuenta el códec y su velocidad en bits, como el Cuadro 2.1:

Cuadro 2.1: Códecs y su velocidad en bits

Codec	Bandwith	Costo
G711	64 Kbps	Gratuito
G729	8 Kbps	US\$10.00 por licencia
iLBC	13.3 Kbps	Gratis

El cuadro 2.1 hace la mención de algunos de los códecs que se usan comúnmente. Pero hay algo importante que también se debe de tomar en cuenta: el *Overhead total* o la Cabecera. (Cuadro 2.2)

Cuadro 2.2: Overhead

Capa	Velocidad
RTP	4.8 Kbps
UDP	3.2 Kbps
IP	8 Kbps
Ethernet	15.2 Kbps

El *Overhead* o Cabecera, trata de que como se envían datos a través de la red, éstos deben de estar empaquetados. Donde dicho paquete contiene información que permite que los datos sean enviados al destino y puedan ser reconstruidos correctamente. El empaquetamiento se maneja con los protocolos de las capas de transporte, red y enlace del modelo TCP/IP, por consiguiente el audio codificado se empaqueta en *RTP*, el que a su vez se empaqueta dentro de *UDP*, para después ser empaquetado en *IP*. Ethernet es el tipo de red más común y en éste también se empaqueta, aunque en la parte de Ethernet puede variar, ya que puede ser PPP, Ethernet o Frame Relay, sin

embargo la mas usada es Ethernet. Considerando lo anterior, *el Overhead total* es igual a 31.2 Kbps.

Cuando ya se conoce el Overhead y la velocidad del códec, entonces ya se puede hacer el cálculo del ancho de banda como lo muestra el cuadro 2.3 para el G.729 y el cuadro 2.4 para el G.711:

Cuadro 2.3: G.729 y su Overhead

Enlace	Ethernet(15.2 Kbps)
Red	IP(8 Kbps)
Transporte	UDP(3.2 Kbps) RTP(4.8 Kbps)
	G.729(8 Kbps)
Total	39.2 Kbps

Cuadro 2.4: G.711 y su Overhead

Enlace	Ethernet(15.2 Kbps)
Red	IP(8 Kbps)
Transporte	UDP(3.2 Kbps) RTP(4.8 Kbps)
	G.729(64 Kbps)
Total	95.2 Kbps

Estos son algunos ejemplos de ancho de banda por llamada, es decir que si se realizan 10 llamadas simultáneas con códec g711, por ejemplo, el ancho de banda necesario es de $95.2 \times 10 = 952$ Kbps.

2.3. Elementos de Red

- *Servidor SIP:*

Es un dispositivo o aplicación que permite crear y gestionar cuentas *SIP* y hace que los Usuarios *SIP* se puedan registrar, almacenando la dirección IP donde deben acceder para realizar la comunicación con este usuario.

- *Proxy SIP:*

Es una aplicación que permite enviar los paquetes necesarios desde uno de los usuarios a otro usuario para que se empiece a establecer una comunicación. Éste únicamente gestiona el estado de una llamada cuando se realiza.

- *B2BUA (Back 2 Back User Agent)*

Es una aplicación cuya función es controlar llamadas entre usuarios *SIP*. Mantiene el estado de las llamadas para conseguir información valiosa en determinados entornos como facturación, redireccionamiento de llamadas en caso de caída de un proveedor *SIP*, etc..

- *Media Gateway (MGW)*

El *Media Gateway* es una aplicación o dispositivo que convierte la señalización *SIP* y el audio streaming recibidos por *SIP*, en el formato necesario para que sea transportado por otra tecnología como puede ser las líneas analógicas, digitales, diferentes protocolos IP, etc..

- *Media Server*

Es un dispositivo o aplicación que permite almacenar contenido multimedia (audio, vídeo, imágenes, etc.) y que puede enviarla mediante algún tipo de protocolo sin importarle a quien.

- *PBX o Private Branch Exchange*

Un *PBX* es un conmutador basado en la red telefónica ya sea analógica, digital o

incluso móvil, que realiza las siguientes acciones: gestionar llamadas, transferencias, programar menús IVR, grabar conversaciones, etc..

Asterisk es un PBX que engloba un Servidor SIP, un Proxy SIP, un Back 2 Back User Agent, un Media Gateway, un Media Server pero lo más importante es que incluso puede llegar a realizar acciones que ni un Proxy SIP ni un B2BUA pueden realizar como: grabaciones de llamadas, sistemas de buzón de voz, reproducción de locuciones, ofrecer menús IVR, reproducir música en espera, guardar datos en una base de datos recibidos directo de un teléfono, etc..

- *Usuarios SIP:*

Un usuario SIP puede ser cualquier dispositivo o software que soporte el protocolo SIP como por ejemplo un *softphone*, un teléfono IP. El punto es que tenga la capacidad de registrarse con una cuenta SIP.

Estos usuarios reciben una URI formada por *usuario@dominio* donde el campo dominio se corresponde con el *Servidor SIP* donde se encuentra registrado.

- *ATA*

En ocasiones cuando se tienen líneas SIP de manera virtual o una solución de PBX y se cuenta con teléfonos analógicos pero no IP, una opción viable para utilizar estos dispositivos es un ATA, que básicamente es un adaptador de SIP a líneas analógicas, su función es conectar uno de sus puertos Ethernet al segmento de IP y declarar la extensión o línea SIP que se tenga, mientras que en el otro puerto se conecta un teléfono convencional. El ATA convierte todo el tráfico de voz en paquetes para su transmisión en redes IP

- *Softphone*

Es un software que se utiliza para la Telefonía sobre IP, puede manejar protocolos como SIP, IAX y codecs como g711, g729, g722, etc.. Puede ser instalado en computadoras o en smartphones para que su uso sea móvil. Existen en el mercado gran cantidad de marcas libres y comerciales, como Zoiper, Bria, Csipsimple, Telephone, etc..

- *Telular*

Es un dispositivo que funciona como Gateway para poder manejar telefonía celular, es decir, en un PBX se manejan líneas SIP para teléfonos fijos y para las salidas

a celular se manejan por un Telular. Este dispositivo puede contener uno o varias tarjetas SIM⁷ de cualquier compañía telefónica, y de esta manera poder abaratar la telefonía celular dentro de una empresa o hasta llegar a brindar servicios como mensajes de texto.

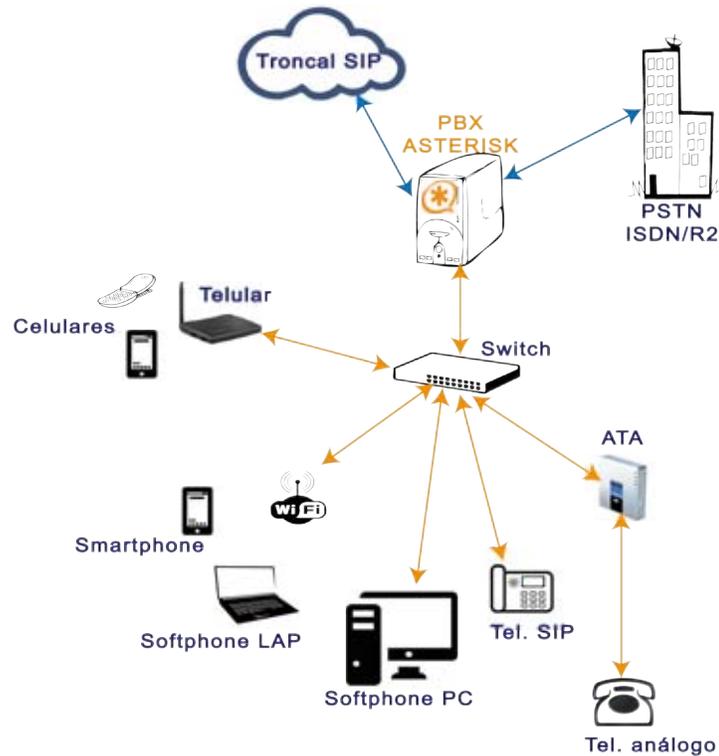


Figura 2.24: Topología

En la figura 2.24 se hace referencia a un diagrama de como estaría conformada una red de Telefonía sobre IP dentro de una empresa. Se muestra en conmutador o PBX con Asterisk configurado y de ahí se desprenden los servicios tanto para los teléfonos SIP, (para PC y Softphone), así como la conexión con un Telular para llamadas a celulares. El Asterisk está conectado a la PSTN mediante una Troncal SIP e incluso puede tener una tarjeta E1 o análoga para tener tanto llamadas entrantes como salientes hacia ella.

⁷Subscriber Identity Module, o Módulo de Identificación de Abonado

Capítulo 3

Estándares abiertos y código libre para Telefonía IP

3.1. Código libre

Código libre se refiere la libertad que se tiene para que se ejecute, copie, distribuya, cambie y se mejore el software. Esto significa que se tiene la libertad de usar el programa para cualquier propósito, estudiar el funcionamiento de la aplicación y poder adaptarlo dependiendo a lo que se requiera, distribuir las copias del software, además de que se puede mejorar el programa y hacer públicas las actualizaciones. Pero esto no significa que sea necesariamente **gratuito**. Actualmente, existen varias modalidades de desarrollo y distribución de software, que son:

- **Software propietario:** estos programas y aplicaciones suelen estar desarrollados por empresas que licencian el código fuente del programa y no permiten su redistribución.
- **Shareware:** el objetivo de esta modalidad es el de *probar antes de comprar*. Generalmente, estos programas no dejan de ser propietarios, pero su distribución es reducida o con límite de uso temporal. En caso de que sea del agrado, entonces se puede pagar un precio por la versión completa, la que tampoco incluye el código fuente (generalmente). En otras palabras, esta versión es de libre distribución, no así la versión completa.

- **Freeware:** se desarrolló en conjunto con el shareware; aquí las cosas son un poco mejores para el usuario final, ya que este tipo de programa está completo y puede ser utilizado y distribuido libremente, pero no tienen la calidad de uno propietario, de Software Libre y que tampoco se incluye el código fuente, por lo que tienen las limitaciones mencionadas en las otras modalidades.

- **Software Libre:** es el punto máximo de libertad tanto para el desarrollador como para el usuario. Para que un programa sea Software Libre, debe cumplir con cuatro requisitos básicos:
 1. Puede ser utilizado sin ningún tipo de limitación.
 2. Ser distribuido libremente y copiado a cuantas computadoras sea necesario.
 3. Debe estar siempre con el código fuente, ya que al disponer de éste, los usuarios pueden hacerle modificaciones para adecuarlo mejor a sus necesidades.
 4. No es un requisito pero un programa de esta modalidad se puede vender, incluso una versión modificada de un programa de Software Libre. Siempre y cuando se respeten los nombres de los autores originales y los tres puntos anteriores, no hay ninguna restricción para hacer algo de dinero con un programa de Software Libre.

3.2. Sistemas operativos.

Un sistema operativo es el software que se encarga de la administración, el control y la gestión del uso del hardware entre los diferentes programas de aplicación y los usuarios. (Figura 3.1)

En el caso de una computadora sencilla, el sistema operativo cuenta con tres funciones principales:

- Vigilar y dirigir el funcionamiento de los componentes externos de la computadora.
- Saber y reconocer en qué parte del disco duro se localizan los archivos del equipo.

- Revisar de manera periódica que la conexión tanto del teclado como la del monitor se encuentren funcionando correctamente para que la transferencia de datos se realice de manera satisfactoria.

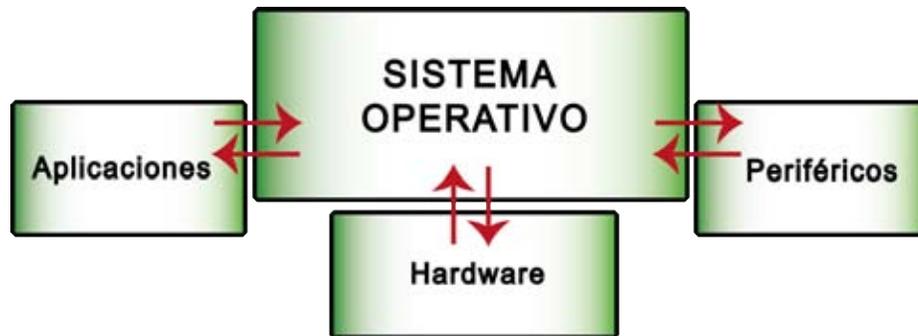


Figura 3.1: Sistema Operativo

3.2.1. Linux

GNU/Linux es el primer sistema operativo basado en UNIX que es 100 % Software Libre. Anteriormente había otros sistemas operativos de libre distribución pero no eran totalmente Software Libre, ya que estaban regidos por licencias más restrictivas.

La idea principal que originó el desarrollo de este sistema era que mantuviera un grado de similitud con el conocido UNIX sin que compartieran una sola línea de código fuente. Por lo que se creó un kernel llamado Linux, desarrollado originalmente por Linus B. Torvalds (en el año de 1991) en conjunto con las aplicaciones de sistema creadas por el proyecto GNU (GNU es un acrónimo recursivo que significa GNU No es Unix (GNU is Not Unix); liderado por Richard Stallman.

El resultado fue una estructura general igual a la de cualquier sistema UNIX: un *núcleo*, más un *intérprete de comandos y aplicaciones*. Tiene una interfaz gráfica llamada Xfree86 además de las muchas aplicaciones para realizar diversas tareas, desde procesamiento de textos hasta montaje de servidores de red, pasando por aplicaciones multimedia y juegos. Lo que también importante es que se distribuye bajo la Licencia Pública General de GNU (GPL) , es decir que es software libre.

Para una comprensión mas práctica de lo que es GNU y lo que es LINUX, diremos:

- GNU: es representado por las aplicaciones Open Source como Firefox, Gnome KDE, Thunderbird, The GIMP y OpenOffice.
- Kernel Linux: es representado en forma de Engrane, el cual está constituido por aproximadamente 10,000 líneas de código en lenguaje C.

(Figura 3.2)



Figura 3.2: GNU-Linux

Aunque Asterisk puede funcionar en muchos sistemas operativos, GNU/Linux es la plataforma más estable en la que existe un mayor soporte y que permite adaptar los requerimientos necesarios para el funcionamiento del sistema de telefonía convirtiéndose en un sistema de comunicaciones económico, avanzado, y completo.

3.2.1.1. Estructura de directorios

En el sistema de ficheros de Linux, existen varias subjerarquías de directorios que poseen múltiples y diferentes funciones de almacenamiento y organización en todo el sistema. Se clasifican en:

- **Estáticos:** Contiene archivos que no cambian sin la intervención del administrador (root), pero pueden ser leídos por cualquier otro usuario. Entre otros están: */bin*,

/sbin, /opt, /boot, /usr/bin, etc.

- **Dinámicos:** Contiene archivos que son cambiantes, y pueden leerse y escribirse, por el root y algunos sólo por su respectivo usuario. Para estos directorios, se recomienda una copia de seguridad con frecuencia. Entre otros están: */var/mail, /var/spool, /var/run, /var/lock, /home, etc.*

- **Compartidos:** Contiene archivos que se pueden encontrar en una computadora y utilizarse en otra, o incluso compartirse entre usuarios.

- **Restringidos:** Contiene ficheros que no se pueden compartir, solo son modificables por el administrador. Como son: */etc, /boot, /var/run, /var/lock.*

3.2.1.2. Shell y comandos básicos

Shell es una terminal o un intérprete de comandos. En Windows se está muy acostumbrado a utilizar las ventanas gráficas, las que permiten por medio de colores o botones interactuar con los programas para una determinada tarea.

En Linux, la manera más habitual de administrar sus programas desde la instalación, la creación o modificación de usuarios, entre otras actividades es por medio de una terminal o intérprete. Esto es en modo texto y generalmente se muestra un *Prompt*, éste es lo que el intérprete de comandos escribe automáticamente en cada línea antes de que se le pueda dar la instrucción por medio de comandos, éste por lo general termina presionando la tecla Intro, para que éste sea recibido por la máquina y ejecutado.

Con la terminal se pueden editar textos, archivos de configuración, apagar o reiniciar la máquina, la instalación de nuevos sistemas, usar un navegador, etc..

La figura 3.3 muestra una terminal de Linux:

```
[root@localhost ~]# help
GNU bash, version 3.2.25(1)-release (i686-redhat-linux-gnu)
These shell commands are defined internally. Type `help' to see this list.
Type `help name' to find out more about the function `name'.
Use `info bash' to find out more about the shell in general.
Use `man -k' or `info' to find out more about commands not in this list.

A star (*) next to a name means that the command is disabled.

JOB_SPEC [&]                (( expression ))
: filename [arguments]      :
[ arg... ]                  [ [ expression ] ]
alias [-p] [name[=value] ...] bg [job_spec ...]
```

Figura 3.3: Terminal o Shell

A continuación se mencionarán algunos comandos básicos para moverse entre archivos o para la edición de textos, ya que en Asterisk sus archivos tienen que editarse de esta manera:

CAT (de concatenar).

Permite visualizar el contenido de un archivo de texto sin la necesidad de un editor. Para utilizarlo solo debemos mencionarlo junto al archivo que deseamos visualizar. (Figura 3.4)

```
[root@localhost asterisk]# cat asterisk.conf
[directories]
astetcdir => /etc/asterisk
astmoddir => /usr/lib/asterisk/modules
astvarlibdir => /var/lib/asterisk
astagidir => /var/lib/asterisk/agi-bin
astspooldir => /var/spool/asterisk
astrundir => /var/run/asterisk
astlogdir => /var/log/asterisk
astdatadir => /var/lib/asterisk
```

Figura 3.4: CAT

LS (de listar).

Permite listar el contenido de un directorio o fichero. La sintaxis es como se muestra en la figura 3.5:

```
[root@localhost ~]# ls /etc/asterisk
additional_a2billing_iax.conf
additional_a2billing_sip.conf
ads_i.conf
agents.conf
ais.conf
alarmreceiver.conf
alsa.conf
amd.conf
applications.conf
app_mysql.conf.sample
asterisk.ads_i
asterisk.conf
```

Figura 3.5: LS

CD (de change directory o cambiar directorio).

Es como su nombre lo indica el que hace que se acceda a una ruta distinta de la que se encuentre. Por ejemplo, se está en el directorio /etc y se quiere acceder a /etc/asterisk, sería como lo indica la figura 3.6 :

```
[root@localhost etc]# cd /etc/asterisk
[root@localhost asterisk]# █
```

Figura 3.6: CD

PWD (de print working directory o imprimir directorio de trabajo).

Imprime la ruta o la ubicación al momento de ejecutarlo, de esta manera se evita el perderse si se encuentra trabajando con múltiples directorios y carpetas. Su sintaxis es como lo indica la imagen 3.7:

```
[root@localhost asterisk]# pwd
/etc/asterisk
```

Figura 3.7: PWD

NANO.

Es un editor de textos para Shell, fácil de manejar. Con él, cualquier usuario por poco experimentado es capaz de empezar a usarlo desde el primer momento gracias a las dos líneas de ayuda que mantiene en su parte inferior.

Para editar un archivo con *nano* tenemos que ejecutar lo siguiente: *nano nombre_archivo*
Donde nombre_archivo será el archivo que se quiera editar. En caso de que el archivo

no existiera, se creará un archivo vacío con ese nombre.

La figura 3.8 nos muestra que ya se está dentro del archivo a editar y las dos líneas de la parte inferior es donde se ubican las opciones de este editor:

```
[400](template)
secret=L30n4rd04us+r4114
callgroup=21
pickupgroup=21
dial=SIP/12090450
callerid=5512090450
context=Leonardo-australia
accountcode=numero-de-entrada

^G Get Help   ^O WriteOut   ^R Read File  ^Y Prev Page  ^K Cut Text    ^C Cur Pos
^X Exit       ^J Justify    ^W Where Is  ^V Next Page  ^U UnCut Text ^T To Spell
```

Figura 3.8: NANO

La tabla 3.1 da la explicación de las opciones que se usan en este editor de texto:

Cuadro 3.1: Comandos para el uso de nano

Comando	Función del Comando
ctrl + x	Se encarga de salir del editor
ctrl + o	Guarda el archivo
ctrl + w	Buscar un texto en el editor
ctrl + j	Justificar el párrafo actual
ctrl + y	Moverse a la página anterior
ctrl + w	Moverse a la página siguiente
ctrl + k	Cortar la línea actual y guardarla en el cutbuffer
ctrl + t	Pegar el cutbuffer en la línea actual
ctrl + T	Invocar el corrector ortográfico (si está disponible)

3.3. Asterisk

"Asterisk is an open source framework for building communications applications. Asterisk turns an ordinary computer into a communications server. Asterisk powers IP PBX systems, VoIP gateways, conference servers and other custom solutions. It is used by small businesses, large businesses, call centers, carriers and government agencies, worldwide. Asterisk is free and open source. Asterisk is sponsored by Digium"¹
(Asterisk es un marco de código abierto para construir aplicaciones de comunicaciones. Convierte una computadora normal en un servidor de comunicaciones. Maneja PBX IP, gateways VoIP, servidores de conferencia y otras soluciones personalizadas. Es utilizado por las pequeñas y grandes empresas, call centers, los transportistas y agencias gubernamentales en todo el mundo. Es gratuito y de código abierto. Y es patrocinado por Digium). (Figura 3.9)



Figura 3.9: Asterisk Logotipo

Asterisk es una aplicación de código libre basada en licencia GPL² y por lo tanto es libre para desarrollar sistemas de comunicaciones profesionales de gran calidad, seguridad y versatilidad, además de que usa estándares abiertos como SIP, H323 o IAX, lo cual permite que cualquiera pueda implementar un sistema de red de telefonía con garantías de interoperabilidad esto es de gran beneficio ya que se pueda hacer conexión con la PSTN. Cuando nos referíamos a estándares abiertos es porque Asterisk, en este caso, cumple con las reglas, condiciones o requerimientos en materia de telefonía garantizando que todas las centrales telefónicas puedan operar entre si, ya que de

¹<http://www.asterisk.org/get-started>

²Licencia Pública General

lo contrario, si cada sistema de telefonía contara con sus propios estándares, no habría la posibilidad de comunicarse unos con otros.

Asterisk es un sistema telefónico que no se entrega como los sistemas telefónicos clásicos si no que se puede descargar como un software desde Internet. Todo lo que se necesita básicamente para probar la funcionalidad de este sistema es de una simple computadora, el software de Asterisk y el sistema operativo base. Este sistema puede tanto trabajar con señalización analógica, como E1 y SIP, e incluso combinar las tres cosas fácilmente.

En 1999, Mark Spencer³ fue el que creó Asterisk y actualmente es su principal desarrollador, junto con otros programadores que contribuyen a corregir errores y a añadir novedades y funcionalidades. Se desarrolló en un inicio para el sistema operativo GNU/Linux pero en la actualidad se hacen versiones para sistemas operativos como BSD, MacOSX, Solaris y Microsoft Windows, aunque como ya se dijo, Linux es la mejor plataforma donde se puede montar Asterisk.

*Zapata*⁴ iniciado por *Jim Dixon* es proyecto donde el objetivo era el de diseñar tarjetas específicas para convertir la señal analógica que provenía de la PSTN a una señal digital, y ahorrar costos en el desarrollo de dispositivos de telefonía avanzados. Actualmente el proyecto *Zapata* fue integrado totalmente en Asterisk, y éste a su vez es patrocinado por una empresa que construye dispositivos de telefonía digital, llamada **Digium**, resultando una parte importante de Asterisk con el nuevo nombre: *DAHDI* (son las siglas de *Digium Asterisk Hardware Device Interface*). (Figura 3.10)



Figura 3.10: DAHDI

³http://es.wikipedia.org/wiki/Mark_Spencer

⁴<http://www.zapatatelephony.org/>

Hace 9 años surgió la primera de las versiones Asterisk 1.0. y con el tiempo, la interfaz de telefonía Zapata de Asterisk ha sido modificado y mejorado.

Existen varias versiones de Asterisk, en las que varían las funcionalidades incluidas. Cuando se hace el lanzamiento de una nueva versión se le da soporte durante un periodo de tiempo determinado. En este periodo, harán versiones menores que incluyen cambios para corregir los errores que vayan siendo reportados. En algún momento la versión llegará a ser obsoleta y sólo mantendrá las correcciones de problemas de seguridad. Después llegará a su final de vida donde la versión no recibirá cambio alguno.

Hay que tomar en cuenta, respecto a la versión, estos puntos para conocer la diferencia entre cada uno:

- La versión de largo plazo *LTS (Long Term Support)* será totalmente vigente durante 4 años, con un año adicional de mantenimiento de parches de seguridad.
- Las versiones estándar están soportadas por un período de tiempo más corto, que será por lo menos un año más un año adicional de mantenimiento de parches de seguridad.

Los cuadros 3.2 y 3.3 muestran las versiones estables que se han dado para Asterisk.

Cuadro 3.2: Versiones de Asterisk

Nombre	Tipo de versión	Año	Renombrada como
Asterisk 1.0		2004	
Asterisk 1.2.X		2005	
Asterisk 1.4.X	TLS	2006	
Asterisk 1.6.X	STANDAR	2008	
Asterisk 1.6.1.X	STANDAR	2009	
Asterisk 1.6.2.X	STANDAR	2009	
Asterisk 1.8.X	TLS	2010	

Cuadro 3.3: Versiones de Asterisk (Continuación)

Asterisk 1.10.X	TLS	2011	Conocida como Asterisk 10
Asterisk 1.11.X	STANDAR	2012	Conocida como Asterisk 11
Asterisk 1.12.X	STANDAR	2013	Conocida como Asterisk 12
Asterisk 1.13.X	TLS	2014	Conocida como Asterisk 13

Asterisk incluye muchas características que anteriormente estaban disponibles en costosos sistemas propietarios de conmutadores, como el buzón de voz, conferencias, IVR, entre otras muchas más. Los usuarios pueden crear nuevas funcionalidades escribiendo un plan de marcado o *dialplan* en el lenguaje de script de Asterisk o añadiendo módulos escritos en lenguaje C o en cualquier otro lenguaje de programación soportado por Linux.

Para conectar teléfonos estándar analógicos son necesarias unas 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 de Telefonía sobre IP como pueden ser *SIP*, *H.323* e *IAX*. Además de que puede interoperar con terminales IP actuando como un registrador y como *Gateway* entre ambos.

3.3.1. Arquitectura

Asterisk fue diseñado de una manera modular, lo que permite que cada usuario pueda seleccionar el módulo que más le convenga, teniendo así dos puntos importantes: el primero que sea una aplicación escalable, esto significa que es posible desactivar los módulos no utilizados en caso de que el equipo donde se quiera instalar Asterisk cuente con pocos recursos; y segundo lugar, que sea extensible, o sea que para que se programe un nuevo módulo, no es necesario tener todo el código de Asterisk.

En la figura 3.11 se muestra como es la estructura modular de la que hablaremos mas a detalle.

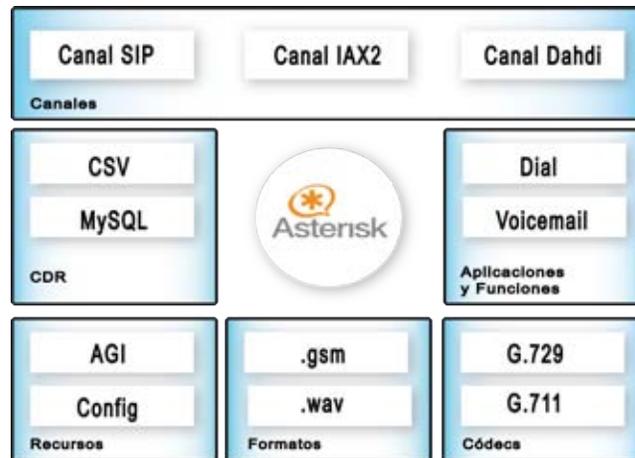


Figura 3.11: Arquitectura de Asterisk

Los módulos se dividen en 7 categorías:

1. **Core.** Corresponde a el núcleo de Asterisk, el cual incluye las funciones más básicas y posibilita la carga de los módulos.
2. **Recursos.** Aportan funcionalidades adicionales al Core, como la posibilidad de leer ficheros de configuración (res config), música en espera (res musiconhold), etc..
3. **Canales.** Su objetivo es que Asterisk pueda manejar un dispositivo de una determinada tecnología. Como ejemplo, tenemos que para manejar los dispositivos que sean SIP utilizaremos el módulo *chan_sip*, para IAX2 el *chan_ixx* y para canales analógicos/digitales el *chan_dahdi*.
4. **Aplicaciones y funciones.** Son los módulos que aportaran las distintas herramientas para configurar el sistema Asterisk.
5. **CDR.** Su objetivo es controlar la escritura del registro telefónico generado por Asterisk a diferentes formatos, por ejemplo a un fichero CSV, una base de datos MySQL, etc..
6. **Códexs.** Asterisk maneja diversos códexs para que pueda codificar y decodificar la información de audio/vídeo que tiene que enviar y recibir.

7. **Formatos.** El objetivo de este módulo es que Asterisk pueda interpretar y manejar ficheros en distintos formatos, como mp3, alaw, ulaw, etc..

3.3.2. Canales en Asterisk

Asterisk usa un canal para comunicarse con los clientes que utilicen un protocolo o una tecnología determinada (SIP, IAX2, etc.), donde los nombres entre canales y protocolos o tecnologías conciden.

Podemos decir que un canal es una conexión que conduce una llamada entrante o saliente en el sistema Asterisk. La conexión puede venir o salir hacia la telefonía tradicional analógica, digital o IP.

Se identifican los siguientes canales:

- **Canal SIP:** Es el canal que se utiliza para las comunicaciones que se envíen o reciban de los clientes SIP.
- **Canal H.323.** Es el canal que se utiliza para las comunicaciones de los clientes que usen H.323.
- **Canal IAX2:** Es el canal que se utiliza para las comunicaciones de los clientes que usen IAX2.
- **Canal DAHDI:** Es el canal que se utiliza para las comunicaciones entre las líneas de la PSTN tanto analógicas y digitales con Asterisk.

3.3.3. Estructura de directorios

Cuando se ha realizado la compilación e instalación de Asterisk, se crean muchos directorios, y cada uno contiene una parte de Asterisk. A continuación se mencionarán los directorios relacionados con Asterisk, así como su función:

[/etc/asterisk/]. Es el directorio más importante ya que contiene los ficheros de configuración, así como el fichero asterisk.conf, donde se indica la ubicación de los demás directorios. Tomaría mucho tiempo revisar todos estos servicios, por lo que nos concentraremos en los más importantes:

- **asterisk.conf**: configuraciones generales de la ubicación de directorios de configuraciones, módulos compilados, voicemails etc. Se recomienda no modificar estas configuraciones, salvo casos especiales.
- **cdr.conf**: Configuraciones referentes al *Call Detail Record*. Los CDR son sumamente importantes para las compañías telefónicas, ya que en ellos se da el detalle de las llamadas que se realizaron tanto de entrada como de salida.
- **codecs.conf**: Es donde se configuran los codecs a utilizar.
- **extensions.conf** : En este archivo se toman las decisiones de ruteo de las llamadas.
- **features.conf**: Este archivo es también muy importante. Permite habilitar y configurar servicios genéricos de un PBX como la transferencia asistida y monitoreo de llamadas.
- **iax.conf**: Importante archivo para el funcionamiento del canal chan_iax que le permite a Asterisk interactuar con otros dispositivos IAX, incluyendo otros PBX Asterisk.
- **logger.conf**: Se maneja el nivel de verbosidad que deben tener los mensajes de log y a donde deben ser enviados.
- **modules.conf**: Archivo sumamente importante. Determina que módulos serán cargados por Asterisk al iniciar.
- **sip.conf**: Parecido al archivo iax.conf pero para el protocolo SIP.
- **dahdi.conf**: Configuración de los canales Dahdi. Las configuraciones de este archivo deben coincidir con el hardware instalado y la configuración del driver dahdi.

[/var/log/asterisk]. En esta carpeta se encuentran los archivos donde se registran las operaciones de Asterisk, así como el CDR en formato CSV:

- **cdr.db**: Este archivo se encuentra disponible si se cuenta con el CDR handler para la base de datos SQLite. El archivo contiene la base de datos de los registros de las llamadas.

- **event log**: Registro de eventos sucedidos en el PBX.
- **full**: Su objetivo es contener todos los mensajes de debug del sistema.
- **messages**: Contiene un listado de los mensajes de warning, debug y demás niveles de logeo.

[/var/lib/asterisk]. Contiene diversos ficheros importantes para Asterisk en distintos subdirectorios, además del `astdb`, la Base de Datos de Asterisk (Berkeley DB2), donde se guarda la información de registro de usuarios, etc.:

- **agi-bin/**: Directorio que contiene los scripts AGI que pueden ser ejecutados desde el plan de marcación con las aplicaciones AGI, EAGI, FastAGI o DeadAGI.
- **firmware/**. Contiene ficheros de firmware necesarios para la comunicación de Asterisk con otros dispositivos como el IAX2.
- **keys/**: Asterisk soporta autenticación mediante RSA en IAX2. En caso de configurar enlaces IAX2 con este tipo de autenticación, las claves públicas y privadas se almacenarán aquí.
- **sounds/**: Directorio con todos los sonidos que serán utilizados por aplicaciones como `Playback()` y `Background()`
- **mohmp3/**: Archivos MP3 para `MusicOnHold`

[/var/spool/asterisk]. Contiene diversos subdirectorios, relacionados con la entrada/salida de ficheros:

- **dictate/**. En este directorio se sitúan los ficheros generados por la aplicación `Dictate`.
- **meetme/**. Contiene los ficheros de audio de las conferencias `MeetMe` que hayan sido grabadas.
- **monitor/**. Contiene los ficheros de audio con las grabaciones realizadas con las aplicaciones `Monitor` y `MixMonitor`.
- **outgoing/**. Asterisk lee periódicamente este directorio en busca de ficheros que permiten generar llamadas automáticamente.

- **system/**. Si utilizamos la aplicación System, Asterisk guarda los posibles ficheros temporales generados en esta carpeta.
- **tmp/**. Contiene ficheros temporales que Asterisk puede necesitar antes de mover un fichero de un sitio a otro.
- **voicemail/**. Asterisk utiliza este directorio para almacenar todos los ficheros con los mensajes de los buzones de voz.

3.4. Implementación. Procedimiento para la instalación de Asterisk 1.8.15

Se ocupará el sistema operativo Debian 6.0.6 Squeeze. Se usará Asterisk en la versión 1.8.15. Se ha optado por instalar versión tipo LTS.

El servidor tiene las siguientes características de Hardware:

- Disco duro: 16GB
- Procesador: Dual core AMR Cortex A7
- Memoria: 2GB

Ya que se instala Debian en el servidor, lo que se debe de hacer es asegurarse que en el archivo:

/etc/apt/sources.list

se tengan las siguientes líneas de repositorios:

```
deb http://ftp.us.debian.org/debian/ squeeze main
deb-src http://ftp.us.debian.org/debian/ squeeze main
deb http://http.us.debian.org/debian/ squeeze main
deb-src http://http.us.debian.org/debian/ squeeze main
```

En caso que no los tenga, habrá que escribirlos para que la compilación de Asterisk no tenga problemas.

Una vez hecho lo anterior, se debe de actualizar los repositorios con la siguiente línea de comando:

```
apt-get update && apt-get upgrade
```

Para que Asterisk funcione de manera correcta, requiere de algunas dependencias del compilador GNU GCC como son: gcc, cpp, glibc-headers, glibc-devel, glibc-kernheaders. Para no instalarlos uno a uno, se hace uso de un paquete único que los engloba: (Figura 3.12)

```
apt-get install build-essential
```

```
Leyendo lista de paquetes...
Creando árbol de dependencias...
Leyendo la información de estado...
Leyendo lista de paquetes...
Creando árbol de dependencias...
Leyendo la información de estado...
Se instalarán los siguientes paquetes extras:
dpkg-dev fakeroot g++ g++-4.8 libalgorithm-diff-perl
libalgorithm-diff-xs-perl libalgorithm-merge-perl libfakeroot
Paquetes sugeridos:
debian-keyring g++-multilib g++-4.8-multilib gcc-4.8-doc libstdc++6-4.8-dbg
libstdc++-4.8-doc
Se instalarán los siguientes paquetes NUEVOS:
build-essential dpkg-dev fakeroot g++ g++-4.8 libalgorithm-diff-perl
0 actualizados, 10 se instalarán, 0 para eliminar y 3 no actualizados.
Necesito descargar 8970 kB de archivos.
Se utilizarán 30.7 MB de espacio de disco adicional después de esta operación.
Desea continuar? [S/n] Des:1 http://mx.archive.ubuntu.com/ubuntu/ trusty/main
libstdc++-4.8-dev i386 4.8.2-19ubuntu1 [1058 kB]
Des:2 http://mx.archive.ubuntu.com/ubuntu/ trusty/main g++-4.8 i386 4.8.2-19ubuntu1 [7023 kB]
Des:3 http://mx.archive.ubuntu.com/ubuntu/ trusty/main g++ i386 4:4.8.2-1ubuntu6 [1500 B]
```

Figura 3.12: apt-get install build-essential.

Lo anterior es un meta-paquete el cual instala los paquetes necesarios para poder programar en C/C++, porque se necesita compilar e instalar Asterisk.

Después se instalarán las dependencias propias de Asterisk con el siguiente comando:

```
apt-get build-dep asterisk
```

El comando `apt-get build-dep` permite instalar las dependencias necesarias para la compilación de un paquete fuente. Cabe destacar que con `build-dep` NO se instala el paquete solicitado, sino sus dependencias necesarias para que funcione correctamente.

Aparecerá una pantalla en azul donde pide un código numérico de área, en la que el sistema de telefonía operará (Figura 3.13). Esto sirve para configurar los estándares regionales predeterminados para que el hardware, en caso de usarlo, funcione. Para México, corresponde el 52:

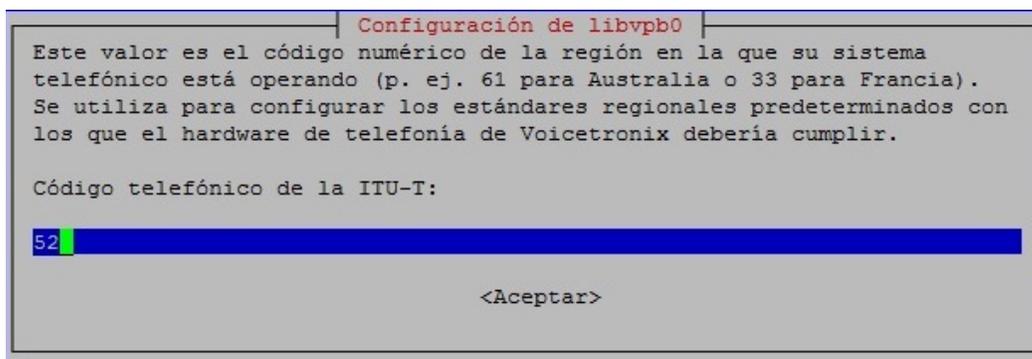


Figura 3.13: Código de área

Lo siguiente será escribir el siguiente comando:

```
apt-get install linux-headers-$(uname -r)
```

Linux headers es un paquete que sirve para realizar instalaciones que tienen que ver con los módulos del kernel. El 'uname-r' es para obtener la versión del kernel.

Se necesitan además un par más de paquetes:

```
apt-get install libxml2-dev
```

```
apt-get install ncurses-dev
```

Una vez que ya se instalaron las dependencias y paquetes, lo siguiente es descargar Asterisk, por lo que se irá a la ruta siguiente:

```
cd /usr/src
```

Hay quienes crean una carpeta para dejar los archivos de asterisk, pero es por cuestiones de gusto.

La descarga se realiza de la página siguiente:

<http://www.asterisk.org/downloads/asterisk/all-asterisk-versions>

Se copia solamente la dirección del enlace y con `wget` se procede a la descarga del archivo. (Figura 3.14)

```
wget http://downloads.asterisk.org/pub/telephony/asterisk/certified-asterisk-1.8.15-current.tar.gz
--2014-06-09 19:40:31-- http://downloads.asterisk.org/pub/telephony/asterisk/asterisk-1.8.15-current.tar.gz
Resolviendo downloads.asterisk.org (downloads.asterisk.org)... 76.164.171.238, 2001:470:e0d4::ee
Conectando con downloads.asterisk.org (downloads.asterisk.org)[76.164.171.238]:80... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 34848342 (33M) [application/x-gzip]
Grabando a: "certified-asterisk-1.8.15-current.tar.gz"
100 %[=====>] 34 848 342 128KB/s en 4m 46s
2014-06-09 19:45:17 (119 KB/s) - "certified-asterisk-1.8.15-current.tar.gz" guardado [34848342/34848342]
```

Figura 3.14: WGET

Una vez descargado se desempaquetará el archivo con: `tar -zxf` (Figura 3.15)

```
root@debian:/usr/src# ls
/usr/src# certified-asterisk-1.8.15-current.tar.gz
/usr/src# tar -zxf certified-asterisk-1.8.15-current.tar.gz
/usr/src# cd certified-asterisk-1.8.15-current.tar.gz
root@debian:/usr/src/certified-asterisk-1.8.15-cert3#
```

Figura 3.15: tar -zxf

El cuadro 3.4 indica los comandos para compilar Asterisk.

Comando	Función del Comando
<code>./configure</code>	Se encarga de correr los Scripts de configuración existentes en asterisk
<code>make menuselect</code>	Selecciona los módulos que se requieren configurar para la instalación, por ejemplo, sonidos, idioma, formato de audio,etc..
<code>make</code>	El make se encarga de compilar todos los archivos que aparecen durante la ejecución de <code>./configure</code>
<code>make install</code>	Instala los módulos seleccionados en el <code>menuselect</code>
<code>make samples</code>	Crea archivos de configuración de ejemplo, con su respectiva explicación de funcionamiento de cada archivo, algo así como un manual
<code>make config</code>	Configura los módulos seleccionados en el <code>menuselect</code> .

Cuadro 3.4: Comandos para compilar Asterisk

Al final, se inicia el servicio de asterisk:

```
service asterisk start
/etc/init.d/asterisk start
```

Con `rasterisk` o `asterisk -r` se puede ya entrar al *CLI*⁵ de Asterisk. (Figura 3.16)

```
root@pbx-asterisk: # rasterisk
Asterisk 11.10.0, Copyright (C) 1999 - 2013 Digium, Inc. and others.
Created by Mark Spencer <markster@digium.com>
Asterisk comes with ABSOLUTELY NO WARRANTY; type 'core show warranty' for details.
This is free software, with components licensed under the GNU General Public
License version 2 and other licenses; you are welcome to redistribute it under
certain conditions. Type 'core show license' for details.
=====
Connected to Asterisk 1.8.15 currently running on pbx-asterisk (pid = 1003)
pbx-asterisk*CLI
```

Figura 3.16: rasterisk

⁵Command Line Interface

3.4.0.1. Simbología Asterisk.

Antes de poder entrar a la configuración de asterisk, es necesario entender su gramática o simbología, ya que con esto podemos interpretar cada una de las configuraciones de los archivos. (Cuadro 3.5)

Cuadro 3.5: Simbología Asterisk

Símbolo	Aplicación	Ejemplo
=>	Sirve para declarar objetos o valores como extensiones	exten =>200
corchetes	Sirve para declarar contextos	[asistente-voip]
`\${EXTEN}`	Sirve para la creación de algunas variables	Dial(SIP/\${EXTEN})

3.4.0.2. Configuración de la plataforma.

A continuación se mostrará la configuración de la plataforma para su funcionamiento, sin embargo no se mostrará cada paso a seguir, porque esto no pretende ser un manual, pero si dar a conocer su funcionamiento.

- **Creación de extensiones.**

Archivo *sip.conf*

Una de las configuraciones más importantes son las extensiones (sean SIP o IAX), ya que son necesarias para poder recibir y generar llamadas, entre otras cosas, puesto que pueden ser utilizadas para otros servicios.

Primero se explicará de manera concreta algunas de las opciones de configuración. (Cuadro 3.6 y 3.7)

Cuadro 3.6: Opciones de configuración SIP

Aplicación	Descripción
host	Tiene dos opciones, la primera es declarar la IP para que de esta manera solo tome el registro del dispositivo que tenga esa IP. La segunda es <i>dynamic</i> , de esta manera cualquiera que cumpla con el usuario y contraseña puede registrarse
disallow	Desabilita todos los códecs de audio.
allow	Habilita únicamente los codecs seleccionados (ulaw, alaw, g729, g711, etc.)
bindport	Número de puerto udp a utilizar, por defecto usa el 5060
context	Es el contexto a donde se dirigirán las llamadas entrantes
accountcode	Puede ser meramente descriptivo, indicando de quien es la línea
secret	Contraseña que utilizará para poder registrar la extensión o troncal.
language	El idioma que va a utilizar puede ser "en", "es", "mx" u otro.
nat	Le dice a asterisk si la extensión a usar estará haciendo NAT o no.
type	Puede tener tres opciones: <i>friend</i> el cual puede realizar y recibir llamadas, <i>peer</i> que solo recibe llamadas, <i>user</i> que solo las genera.

Cuadro 3.7: Opciones de configuración SIP (Continuación)

dtmfmode	Es la frecuencia con la que funcionan los dígitos de los telefonos puede ser <i>rfc2833</i> , <i>auto</i> e <i>info</i> .
allowguest	Parámetro de seguridad. Permite que usuarios sin autenticar realicen llamadas Preferentemente ponerlo en NO
username	Nombre de usuario

A continuación se muestra el archivo de configuración para crear extensiones SIP, de la misma manera se pueden crear las IAX solo que en su respectivo archivo (*iax.conf*).

El contexto *[Template]* es un plantilla, en el cual se pueden poner las opciones generales para todas las extensiones, con la finalidad de no tener que repetirlas en cada una de las que se creen (Figura 3.17). Después se generaran las extensiones, en este caso la 201 y 202 las que de manera básica deben contener un *username* (201 o 202) y un *secret* para que se puedan registrar en Asterisk.(Figura 3.18)

```
[Template](!)
dtmfmode=rfc2833
host=dynamic
type=friend
nat=yes
language=es
call-limit=50
disallow=all
allow=g722
allow=g729
allow =alaw
+++++
```

Figura 3.17: Configuración de Extensión SIP (Template)

Como una nota de seguridad, el *secret* se recomienda contenga símbolos, núme-

```
[201](template)
secret=4s1s+3n+3v01p
username=201
callerid=5512007887
context=asistente-voip

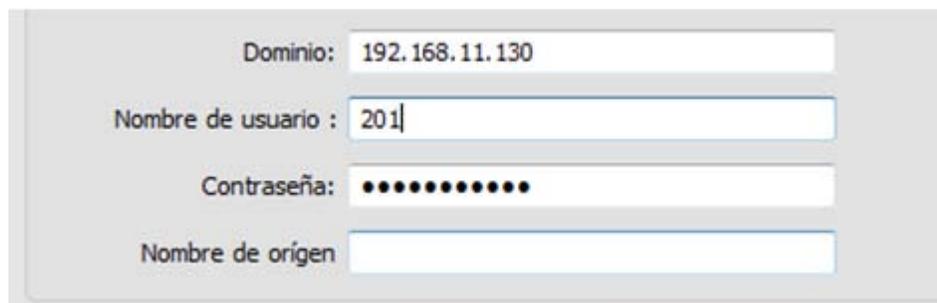
[202](template)
secret=L304s1s+3n+3v01p
username=202
callerid=5512090451
context=asistente-voip

+-----+
```

Figura 3.18: Configuración de Extensiones SIP

ros y letras, con la finalidad de que sea difícil de descifrar.

Ya que se tiene la extensión creada en el archivo *sip.conf*, se registrará un softphone de la marca Zoiper⁶. Para esto, se requieren de por lo menos tres datos importantes: la extensión, la contraseña y la IP a donde se registrará. (Figura 3.19)



Dominio: 192.168.11.130

Nombre de usuario : 201

Contraseña: ●●●●●●●●●●

Nombre de origen

Figura 3.19: Configurando el softphone

■ Creación de un Plan de marcación.

Archivo *extensions.conf*

En el archivo de configuración *sip.conf* se crearon las extensiones, o también llamados usuarios, para poder recibir o realizar las llamadas; sin embargo las exten-

⁶<http://www.zoiper.com/en>

siones no sirven de nada si no existe un plan de marcado.

El plan de marcado es el que indicará a las extensiones qué hacer, es decir, dirá si pueden realizar llamadas ya sean internas (entre extensiones), locales, celulares, LDN, etc.. Además de que se pueden anexar las configuraciones correspondientes para el *voicemail*, las colas de atención o ACD, IVR, salas de conferencia, etc..

La estructura que lleva es la siguiente.

```
[contexto]  
extensión, prioridad, aplicación.
```

Se debe tomar en cuenta que existen diferentes tipos de extensiones como son las *numéricas(1234)*, las *alfanuméricas(recep12)* y las *especiales(s,t,i)*, la prioridad trata del número de secuencia que llevará las extensiones, inicia con 1 y puede ir en orden secuencial o sustituirse con "n", y la aplicación es la acción a ejecutar.

Algunos ejemplos son:

```
exten =>2345,1,Dial(SIP/${EXTEN})  
exten =>Laboratorio,1,Queue(lab)  
exten =>s,1,Answer()  
exten =>t,1,Goto(contexto, prioridad)  
exten =>i,1,Playback(error)
```

Como hemos estado analizando la estructura del *dialplan*, la aplicación es muy importante para poder ejecutar ciertas acciones o tareas, ya que estas aplicaciones controlan el comportamiento del *dialplan*.

Las aplicaciones mas comunes se muestran en el cuadro 3.8:

Cuadro 3.8: Aplicaciones en Asterisk

Aplicación	Descripción
Answer	Contesta el canal utilizado.
Hangup()	Cuelga el canal al terminar una llamada.
Busy	Da estatus de ocupado en el canal.
wait()	Espera los segundos marcados dentro del paréntesis.
WaitExten()	En el IVR, espera a que el llamante marque un dígito, la cantidad de segundos que está en el paréntesis.
Dial	Esta aplicación sirve básicamente para llamar, sin embargo es de suma importancia de la que más adelante se hablará. Su sintaxis básica es <code>Dial(type/identifier, timeout, options, url)</code> .
Goto	Sirve para hacer un brinco entre prioridades y contextos del <i>dialplan</i> . Su sintaxis es: <code>Goto(context,exten,priority)</code> .
GotoIf	Es un brinco condicional, si se cumple la condición ejecuta una acción, si no, se va a otra. Su sintaxis es: <code>Gotolf(Condicion?accion1:accion2)</code>
GotoIfTime	Es un brinco condicionado por el tiempo, ya que se ejecuta dependiendo la fecha y hora configurada. Su sintaxis es: <code>GotolfTime(<times>/<weekdays>/<mdays>/<months>?[[context/]exten/]priority)</code>
Background	Ejecuta una grabación la cual puede ser interrumpida por el usuario al apretar algún dígito.
Playback	Igual que Background ejecuta una grabación pero esta no puede ser interrumpida por el usuario.
Queue	Manda a una cola de atención.

- **Aplicación *Dial*.**

Como se comentó anteriormente la aplicación *Dial* es de suma importancia, ya que ejecuta la llamada hacia cualquier destino y tecnología, (como tecnología nos referimos a SIP, IAX o DAHDI).

Esta es su sintaxis *Dial(type/identifíer, timeout, opciones, url)* como ejemplo tenemos:

```
Dial(SIP/200, 15, tTw)
```

donde sugiere que se marcará con una tecnología SIP a la extensión 200, durante 15 segundos y con opción a transferir llamadas.

Otro ejemplo puede ser

```
Dial(SIP/12345678@192.168.1.2, 15, ttw)
```

donde la salida se ejecutará el número 12345678, redireccionándolo por la IP 192.168.1.2, durante 15 segundos con opción a transferir.

Sin embargo como se aprecia son extensiones o números bien definidos, pero qué pasa cuando no tenemos una definición de números, por ejemplo, supongamos que necesitamos hacer llamadas a muchos números locales o celulares, es imposible que se defina todos, por lo que existe una manera de poder crear una salida con dígitos variables, para eso es necesario entender lo del cuadro 3.9 y 3.10.

Cuadro 3.9: Aplicación DIAL

Aplicación	Acción
123	Es una extensión numérica la cual pueden ser aterrizadas llamadas o aplicaciones.
_XXX	Toma valores de 0 al 9, ejemplo 000 - 999 o podría obedecer este patrón 2XX que quiere decir 200-299
_ZXX	El patrón Z toma un rango de 1-9 es decir por la forma ZXX usaría este rango 100-999

Cuadro 3.10: Aplicación DIAL (Continuación)

.	Este punto es un comodín, el cual significa que puede tomar cualquier valor sin importar la cantidad de dígitos, se debe tener mucho cuidado al usarlo.
s	Es una extensión especial que significa <i>start</i> normalmente es usada en IVR's, para iniciar una secuencia ya que no obedece a algún patrón especial
t	Timeout, este tipo de extensión sirve para cuando termina un tiempo de espera determinado ejecuta dicha acción
i	Invalid, esta extensión se usa para cuando se digita de manera incorrecta una opción, ejecuta alguna acción.
_NXX	El patrón N toma valores de 2-9 es decir por la forma NXX usaria este rango 200-299 si estuviera de esta manera seria 210-999
[1-7]	Este patrón puede tomar valores 1,5,6 o 7
Laboratorio101	Es una extensión alfanumérica la cual pueden ser aterrizadas llamadas o aplicaciones

Una vez que se ha entendido esto se mostrarán ejemplos de *Dialplan* de salida.

```
exten =>_XXXXXXXX,1,Dial(SIP/${EXTEN},30,tTw)
```

Como se aprecia en este primer ejemplo, se puede marcar un numero de 8 dígitos comprendido entre 0 y 9, por ejemplo 12097887, el cual viajará por SIP, el `/${EXTEN}`, es una variable que toma el valor digitado y lo inserta, será marcado durante 30 segundos.

```
exten =>_04455XXXXXXXX,1,Dial(DAHD/${EXTEN},30,tTw)
```

En esta segunda opción se marcará un número que comienza con 04455 y esta acompañado de 8 dígitos de entre 0 y 9, por ejemplo 0445541313992 y lo manda a través de DAHDI, sonará durante 30 segundos.

```
exten =>_ZN3XX,1,Dial(IAX/${EXTEN},30,tTw)
```

La tercera opción es un número de 5 dígitos el cual el primero es del 1 al 9, el segundo del 2 al 9, el tercero es 3 y los últimos 2 son del rango 0 al 9, por ejemplo, el 12345 que viajara a través de IAX, sonará durante 30 segundos.

```
exten =>s,1,Dial(SIP/201,30,tTw)
```

Y en la última opción es una llamada de entrada a la extensión "s", la que hace que caiga en la extensión SIP 201, que durará 30 segundos el timbrado.

En la figura 3.20 se muestra un *Dialplan* funcional:

```
[ivr]
exten =>s,1,Answer()
same =>n,Dial(SIP/${EXTEN},60,tTw)
same =>n,VoiceMail(200@asistentevoip,b)
same =>n,Congestion()
exten =>1,1,Answer()
same =>n,Dial(SIP/${EXTEN},30,tTW)
exten =>2,1,Answer()
same =>n,Dial(SIP/${EXTEN},30,tTW)
same =>n,VoiceMail(${EXTEN}@asistentevoip,b)
same =>n,Congestion()
exten =>3,Dial(SIP/201,5,tTw)
exten =>4,Dial(SIP/i2next/0445530295776,20,tTw)
same =>n,VoiceMail(214@asistentevoip,b)
same =>n,Congestion()
+-----+
```

Figura 3.20: Dialplan

- **Creación de troncales y de rutas salientes a la PSTN**

En esta sección se hablará de las troncales, una definición de troncal es el medio que permite comunicar al conmutador Asterisk con la PSTN con otros conmutadores ya sea los tradicionales o IP, y permite la entrada y salida de las llamadas.

Los tipos de troncales dependen del proveedor que brinde el servicio, puede ser por medio físico, o sea, por líneas de fibra óptica o cobre, y la otra manera es IP, vía internet, enlaces dedicados, en la cual ya no es necesario un *hardware* como una tarjeta, y por lo tanto en el servicio de virtualización resulta la mejor opción.

Troncales SIP: Es la manera en que últimamente se está entregando el servicio de telefonía ya que se utiliza la red IP para el transporte de la voz. Prácticamente con esta troncal se tienen números telefónicos de cualquier parte del mundo. Se configura como si fuera una extensión SIP, anexando unos cuantos parámetros. La cantidad de canales dependerá del ancho de banda del enlace IP.

Troncales IAX2: Igual que en SIP, su medio de transporte para la voz es la red IP, y su fin es establecer enlaces entre 2 o más servidores Asterisk, no es un estándar como lo es SIP, por lo que son pocos los fabricantes de equipos que lo soportan. Se configura como una extensión IAX2 pero se anexan algunos parámetros. Ya se había mencionado que una de sus ventajas es la utilización de un solo puerto el UDP 4569 para la comunicación.

Troncales Digitales (Puertos E1, T1): En este caso se utiliza hardware especial ya que se entrega por medios físicos para el transporte de la voz como cable de cobre o fibra óptica. Su configuración varía dependiendo si es R2 o ISDN.

Troncales Análogas (Puertos FXO): En este caso, también se utiliza un hardware para recibir las líneas telefónicas. Por cada línea es necesario un puerto.

En la imagen 3.21 se muestra la configuración de una troncal SIP del proveedor Radian. El archivo donde se puede hacer es en *sip.conf* o dentro de éste se inclu-

ye uno que sea solo para troncales SIP, como puede ser: *sip-trunk.conf*

```
[radian]
type=friend
host=200.76.111.216
context=dialout
defaultuser=test
fromuser=test
username=test
secret=t3st77t3s1s
qualify=yes
disallow=all
allow=g722
allow=ulaw
allow=alaw
allow=gsm
dtfmode=rfc2833
srvlookup=yes

register=>test:t3st77t3s1s@200.76.111.216
```

Figura 3.21: Troncal SIP

Los parámetros para configurar una troncal son: el nombre de la troncal en este caso *[radian]*, del tipo *friend*, esto significa que realizará y recibirá llamadas; y el host será la dirección IP del proveedor en donde se vaya a registrar la troncal. El contexto por el cual se realizarán las llamadas es *context=dialout*.

El *username*, *fromuser*, *defaultuser* será como el proveedor llamará a su cliente, en este caso lo nombra: *test*, se usará una contraseña dada por el proveedor: *t3st77t3s1s*; la troncal usará los codec: *g722*, *ulaw*, *alaw*, *gsm*. El parámetro *sr-lookup=yes* significa que si el proveedor en lugar de entregar el servicio con una IP, da un nombre de dominio, este ayudará a que Asterisk busque a través de una consulta DNS el servidor SIP correspondiente al dominio solicitado.

Y al final llevará una cadena de registro:

```
register=>test:t3st77t3s1s@200.76.111.216
```

Esto significa que la troncal con el username *test* con la contraseña: *t3st77t3s1s*, se registrará a la IP: 200.76.111.216.

Ya se mencionaron algunos de los parámetros para la configuración de una troncal, aunque es importante resaltar que hay proveedores que no necesitan dar un usuario y contraseña, solo se manejan por medio de una *IP de confianza*, esto quiere decir que solo la IP que el proveedor reciba, será la que pueda usar su servicio.

Las rutas de salida que maneja el *context=dialout* en el *Dialplan*, en el caso de esta troncal, se muestra en la figura 3.22:

```
[dialout]
exten =>_XXXXXXXX,1,Dial(SIP/radian/${EXTEN})
exten =>_044XXXXXXXXXX,1,Dial(SIP/radian/${EXTEN})
```

Figura 3.22: Contexto dialout

Solo son llamadas hacia números locales y celulares locales.

■ Creación de IVR y rutas entrantes

El IVR o *Interactive Voice Response*, es quien se encarga de recibir y gestionar las llamadas de entrada al conmutador, mediante menús interactivos.

Se puede configurar de muchas maneras, existen IVR dentro de más IVR's, e incluso guardar u obtener datos directos de una base de datos conectada a Asterisk. El IVR consiste en la grabación que entrega un mensaje de entrada cuando se marca al número del contacto, el que caerá en una serie de opciones, las que pueden llevar a extensiones, a una cola de atención, al voicemail, redireccionarlo a un número celular, pedir un dato para que nos entregue otro o ir a otro menú con más opciones.

En la figura 3.23 el IVR se compone de la siguiente manera:

El IVR *Entradas*, recibe la llamada una extensión "s", esto significa que no es una extensión en específico; la extensión "s" será quien "conteste" y reproducirá una grabación de nombre: */bienvenida-soporte*, que se encuentra en la ruta:

```
[ivr-entradas]
exten =>s,1,Answer()
same =>n(inicio),Background(/var/spool/asterisk/dictate/bienvenida-soporte)
same =>n,WaitExten(5)
exten =>1,1,Dial(SIP/100,60,tTw)
exten =>2,1,Dial(SIP/101,60,tTw)
exten =>3,1,Dial(SIP/104&SIP/501,60,tTw)
exten =>t,1,Goto(inicio)
exten =>i,1,Playback(/var/spool/asterisk/dictate/pbx-invalid)
same =>i,n,Congestion()
exten =>5512090450,1,Answer()
same =>n,Dial(SIP/102,15,tTw)
same =>n,Dial(SIP/test/0445529474431,60,tTw)
+-----+
```

Figura 3.23: IVR

/var/spool/asterisk/dictate/, la grabación indica 3 opciones a elegir, de las que comprenden las extensiones 100, 101 y 104 junto con la 501 respectivamente.

Si el llamante tarda 5 segundos en elegir una de las opciones (esto se nota con la aplicación *WaitExten(5)*), le mandará con la aplicación *Goto* a la etiqueta de *INICIO*, que corresponde a irse al principio del menú para repetir la grabación. En cambio si la opción que digitó el llamante es errónea (identificada con la *i* de *INVALID*), entonces reproducirá la grabación de *pbx-invalid* y le mandará el tono de *Congestion*.

Al marcar el número 5512090450, se direccionará a la extensión 102, pero en caso de no contestar entonces se desvía hacía un celular 0445529474431, que sale por la troncal *SIP/test*.

3.4.1. Recursos Adicionales

Asterisk como tal, además de enviar o recibir llamadas, como ya se ha mencionado antes, también tiene algunos otros recursos interesantes, como los que a conti-

nuación se mencionarán y solo en algunos casos se tocará la forma de configuración.

■ **Estacionamiento de llamadas.**

Esta es una función de mucha ayuda cuando se recibe una llamada en una extensión que no nos corresponde o que simplemente queremos detenerla y contestarla otro lugar; la extensión 700 es usada para estacionar una llamada, por lo que se debe presionar # (más adelante se hablarán de las transferencias) para transferir y después marcar 700, con esto se mantendrá en espera a la persona llamante y se procederá a colgar, para que después desde una extensión cualquiera se marcará 701 o 702 para recuperar la llamada, se pueden alterar estos números 700, 701 o 702, desde el archivo *features.conf*.

■ **Captura de llamadas-Call Pickup**

Esta función permite poder tomar la llamada de otra persona que no se encuentre disponible para contestar, simplemente con marcar *8, la única condición es que las extensiones pertenezcan a un mismo grupo. Para que las extensiones pertenezcan al mismo grupo se debe de poner la opción *Callgroup* y *Pickupgroup* en el archivo de las extensiones, por ejemplo, las extensiones del área de ventas son las 102, 107, 103 y 105, en su archivo de configuración deben de tener *Callgroup=1* y *Pickupgroup=1* cada una, de esta forma, el sistema reconoce que están en el grupo y cuando se quiera jalar la llamada con *8 se dejará. Si se desea modificar el *8, se puede configurar desde *features.conf*

■ **Transferencia de llamadas.**

Esta función tiene dos alternativas:

1. Transferencia atendida.
2. Transferencia desatendida.

La diferencia entre una y otra radica básicamente en que la primera antes de transferir la llamada se puede preguntar a la persona a la cual se le va a transferir la llamada si desea recibirla, para esto solo se marca #2 + el número de extensión, y se preguntará si desea que le transfiera la llamada, en caso de ser afirmativa la respuesta, solo se colgará y de inmediato se enlazará las llamadas.

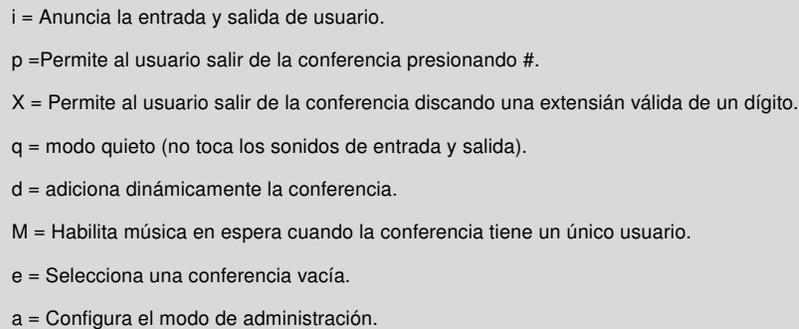
La segunda simplemente se marca #1 + el número de extensión y se colgará, ya que esta solo enlaza la llamada sin importar si el destino la quiere o no contestar.

■ Conferencias.

Las conferencias son salas que se crean para poder hacer audioconferencias entre varias extensiones, se edita desde el archivo *meetme.conf*, y en el *extensions.conf*, y tiene la siguiente sintaxis.

MeetMe(Confno[/options][[/password]])

En la figura 3.24 se mencionaran algunas de las opciones que maneja:



```
i = Anuncia la entrada y salida de usuario.  
p = Permite al usuario salir de la conferencia presionando #.  
X = Permite al usuario salir de la conferencia discando una extensión válida de un dígito.  
q = modo quieto (no toca los sonidos de entrada y salida).  
d = adiciona dinámicamente la conferencia.  
M = Habilita música en espera cuando la conferencia tiene un único usuario.  
e = Selecciona una conferencia vacía.  
a = Configura el modo de administración.
```

Figura 3.24: Parámetros de Conferencias

Ejemplo de una conferencia en archivo *meetme.conf*

conf =>350,2345,123

En la figura 3.25 se da el ejemplo de configuración desde una línea SIP, hacia una conferencia.

```
exten =>5512093435,1,Answer()  
same =>n,MeetMe(351,M)  
same =>n,Congestion()
```

Figura 3.25: Conferencia

■ Voicemail

El *voicemail* es una función que tiene como finalidad recuperar un mensaje de voz cuando una extensión se encuentra ocupada o no disponible, esta es contestada con una grabación, invitando a la persona llamante a dejar un mensaje de voz, el cual posteriormente puede ser revisado vía correo electrónico o ser escuchado desde la extensión.

Su configuración consta básicamente de dos archivos, *voicemail.conf* y *extensions.conf*, en el archivo *voicemail.conf* es donde se almacena toda la configuración del buzón, desde el mensaje que llegará adjunto, así como las extensiones habilitadas, etc..

Por otra parte en el *extensions.conf* se configura el *dialplan* para poder caer en determinado momento a la opción del *voicemail*.

En el cuadro 3.11 se describen algunos parámetros que maneja el Voicemail para su configuración:

Cuadro 3.11: Parámetros de Voicemail

Opción	Significado
language	Especifica el idioma en que aparecen las grabaciones.
format	Especifica el formato en que se grabara el archivo de audio.
servermail	El servidor donde saldrán los correos.
attach	Esta opción le dice que el audio se mandará adjunto en el correo.
maxmsg	Cantidad máxima de mensajes de voz que puede almacenar por casilla
maxsecs	Cantidad máxima en segundos que puede almacenar por mensaje.
emailbody	Indica el cuerpo del mensaje
maxlogins	Número máximo de intentos de login.
sayduration	Anunciar o no la duración.
maxgreet	Cantidad máxima en segundos de mensaje personal para buzón.
maxsilence	Si mientras se graba el mensaje hay un silencio de 5seg. se acaba la llamada.
fromstring	Nombre de remitente del correo.
attachfmt	Formato de audio en que se enviara el correo
emailsubject	Indica el Título del mensaje
saycid	Con esta opción antes de escuchar el audio del mensaje se escucha quién y cuándo se dejo el mensaje.

A continuación se dara un pequeño ejemplo de configuración.

En el archivo *extensions.conf* se ponen los parámetros como lo muestra la figura 3.26:

```
exten =>5512090450,1,Answer()
same =>n,Dial(SIP/400,30,tTw)
same =>n,Voicemail(400@default,b)
same =>n,Congestion
```

Figura 3.26: Voice mail en el *extensions.conf*

La estructura es que al marcar el número 5512090450, sonará la extensión 400 durante 30 segundos, la que al no contestar se irá al buzón de voz de esa extensión. Los parámetros de configuración del buzón, se colocan en el archivo *voicemail.conf* como lo indica la figura 3.27:

```
[general]
language=mx
format=wav|gsm|wav49
servermail=pruebas@asistentevoip.com
attach=yes
maxmsg=200
maxsecs=3600
minsecs=3
maxgreet=1800
saycid=yes.
fromstring=Asistente VoIP
attachfmt=wav
sayduration=no
pagsubject= Buzon de Voz Asistente Voip
```

Figura 3.27: *Voicemail.conf*

3.5. Comparativo

Ya se mencionaron las partes fundamentales de Asterisk, sus componentes, sus configuraciones, etc.. En esta sección se hablará de las comparaciones que hay entre la telefonía IP y la tradicional.

Ventajas de la Telefonía IP de manera general:

- En un primer punto podemos mencionar que se reducen los costos en su instalación y mantenimiento ya que integra la infraestructura de la red de los datos y de la voz en una. Con esto hay un ahorro en su diseño.
- La telefonía IP facilita la movilidad, puesto que se puede disponer de una extensión telefónica en cualquier lugar. Esto gracias a que las extensiones se pueden registrar en los smartphone o en una lap y de esta manera se hace el enrutamiento de la llamada.
- Existe la seguridad y la privacidad en las llamadas por el hecho de que se requiere de la autenticación y puede haber cifrado de datos.
- Como ya se había mencionado en capítulos anteriores, la telefonía sobre IP posee una arquitectura que es escalable y muy flexible puesto que su configuración y reconfiguración se realiza conforme a las necesidades que vayan surgiendo.
- Es compatible con el hardware de diferentes fabricantes/proveedores al estar basada en estándares abiertos y reconocidos por organismos internacionales. Desde tarjetería, teléfonos, switches, etc..

Como también ya se ha mencionado, la telefonía sobre IP tiene funciones adicionales que se usan dependiendo de las necesidades de cada cliente. Por ejemplo, en la configuración de extensiones:

- Está la llamada entre extensiones
- La transferencia de llamadas, que pueden ser *No asistida (Directa)* y *asistida (Consultiva)*.
- El estacionamiento de llamadas.
- La llamada en espera (*Hold*).

- La captura de llamadas, que puede ser hacia una extensión o un grupo.
- La grabación de llamadas.
- El correo de voz, que pueden llegar al correo electrónico.

A continuación se hace un comparativo de costos entre un paquetes básicos de diferentes empresas.

Lo primero a mostrar es un cuadro referente con los servicios y costos de cada una de las empresas, ya que incluye no sólo el costo por minutos de llamadas, si no que también algunos servicios extras a considerar, es decir, el resultado de una factura puede variar en definitiva, pero se tiene que tomar en cuenta los servicios extras que nos ofrece dicho proveedor.

En la figura 3.28 se puede observar claramente la diferencia en cuanto a costos de llamadas, pero como ya se mencionó, se debe de tomar en cuenta los servicios que ofrecen, es decir, los servicios adicionales y sus costos.

	VIRTUALPHONE	TELMEX	AXTEL	CABLEVISIÓN
RENTA	\$100.00	\$187.50	\$269.00	\$159.00
LOCALES	\$1.00	\$1.48	ILIMITADAS	\$1.20
LDN	\$1.50	\$5.52	\$1.19	\$1.03
CEL 044	\$1.00	\$0.85	\$1.79	\$1.61
CEL 045	\$1.50	\$2.24	\$3.80	\$2.95
INTERNACIONAL	\$1.03	\$5.22	\$1.19	\$1.03
BUZÓN	\$0.00	\$0.00	\$30.00	\$5.20
FOLLOWME	\$0.00	SI	\$30.00	\$0.00
EXTENSIONES	HASTA 3	NO	NO	NO
IVR	\$0.00	NO	NO	NO
TRANSFER	\$0.00	NO	\$30.00	\$0.00
SMS	\$1.00	NO	NO	NO
MENSAJERIA INST.	\$0.00	NO	NO	NO
LINEAS FORÁNEAS	SI	NO	NO	NO
CONFERENCIA	\$15.00	NO	NO	NO
LLAMADA TRIPARTITA	\$0.00	\$0.00	\$30.00	\$0.00

Figura 3.28: Costos

Ahora se mandara un comparativo del paquete básico entre las cuatro compañías con la misma cantidad de llamadas, pero haciendo diferencia en los paquetes que cada una ofrecen.

TELMEX	Llamadas/mnts. Incluidas	Llamadas/mnts. totales realizadas	Total por cobrar	Precio Llamada/mnt.	Costo Total
LOCALES	100 llamadas	534 llamadas	434	\$1.48	\$642.32
LDN	100 minutos	200 minutos	100	\$1.48	\$148.00
CEL 044	100 minutos	608 minutos	508	\$0.85	\$431.80
CEL 045	0	50 minutos	50	\$2.24	\$112.00
Total consumo					\$1,334.12
Servicios especiales					\$33.62
Renta mensual					\$187.05
IEPS 3%					\$5.61
IVA 16%					\$213.46
Cargo por redondeo					\$0.46
Crédito por redondeo					-\$0.72
				Total a pagar	\$1,773.60

Figura 3.29: Telmex

La imagen 3.29 pertenece a Telmex. En un paquete básico ofrecen actualmente 100 llamadas locales, 100 minutos a celular 044 y 100 minutos a LDN, incluidos en el paquete. Como se ve, el costo total es de \$1773.60 pesos, pero si lo vemos en la figura 3.28 de Costos, éste no tiene muchos servicios adicionales.

La figura 3.30 pertenece a Axtel, aquí las llamadas locales van incluidas en el paquete de manera ilimitada, sin embargo no pasa lo mismo para el resto de los conceptos. El costo total es de \$1936.60 pesos, pero en la figura 3.28 de Costos tampoco ofrece muchos servicios adicionales y los que llega a ofrecer tienen costo.

AXTEL	Llamadas/mnts. Incluidas	Llamadas/mnts. totales realizadas	Total por cobrar	Precio Llamada/mnt.	Costo Total
LOCALES	Ilimitadas	534 llamadas	0	\$1.48	\$0.00
LDN	0	200 llamadas	200	\$1.19	\$238.00
CEL 044	100 minutos	608 minutos	508	\$1.79	\$909.32
CEL 045	0	50 minutos	50	\$3.08	\$154.00
Total a pagar					\$1,301.32
Servicios especiales					\$150.00
Renta mensual					\$269.00
IEPS 3%					\$8.07
IVA 16%					\$208.21
Cargo por redondeo					
Crédito por redondeo					
Total a pagar					\$1,936.60

Figura 3.30: Axtel

En Cablevisión (Figura 3.31) resulta más difícil, ya que aunque la renta aparece como la más barata pero incluye solamente 100 llamadas locales, el resto de los conceptos se cobra por minutos.

El costo total es de \$2313.40 pesos, pero en la figura 3.28 de Costos tampoco ofrece muchos servicios adicionales.

La imagen 3.32 pertenece a un Proveedor SIP, la renta resulta más barata que en los 3 primeros, se observa que no incluye como tal minutos o llamadas en el plan, sin embargo para algunos conceptos son más baratos, además no cobran los servicios especiales.

Se concluye entonces que si se toman en cuenta los costos, pero también el plus de los servicios especiales, Telmex aparenta ser el más económico, el punto es que no proporciona varios servicios adicionales, en cambio se ve que Virtualphone es el que maneja todos los servicios adicionales por lo que el precio ya no resulta ser tan elevado.

CABLEVISIÓN	Llamadas/mnts. incluidas	Llamadas/mnts. totales realizadas	Total por cobrar	Precio Llamada/mnt.	Costo Total
LOCALES	100 llamadas	534 llamadas	434	\$1.20	\$520.00
LDN	0	200 llamadas	200	\$1.03	\$206.00
CEL 044	0	608 minutos	608	\$1.61	\$978.88
CEL 045	0	50 minutos	50	\$2.95	\$147.50
Total consumo					\$1,853.18
Servicios especiales					\$0.00
Renta mensual					\$159.00
IEPS 3%					\$4.77
IVA 16%					\$296.51
Cargo por redondeo					
Crédito por redondeo					
				Total a pagar	\$2,313.46

Figura 3.31: Cablevisión

VirtualPhone	Llamadas/mnts. incluidas	Llamadas/mnts. totales realizadas	Total por cobrar	Precio Llamada/mnt.	Costo Total
LOCALES	0	534 llamadas	534	\$1.00	\$534.00
LDN	0	200 llamadas	200	\$1.50	\$300.00
CEL 044	0	608 minutos	608	\$1.00	\$608.00
CEL 045	0	50 minutos	50	\$1.50	\$75.00
Total consumo					\$1,517.00
Servicios especiales					\$0.00
Renta mensual					\$100.00
IEPS 3%					\$3.00
IVA 16%					\$242.72
Cargo por redondeo					
Crédito por redondeo					
				Total a pagar	\$1,862.72

Figura 3.32: Virtualphone

Conclusiones

A través de este trabajo que ha sido puesto en marcha y de la experiencia en el campo laboral, se puede concluir que Asterisk es una alternativa económica y eficiente para ofrecerse en el mercado de las telecomunicaciones. Además de que se puede usar en distintas áreas, como la programación, redes, etc..

Es una tecnología que seguirá evolucionando y mejorando en su calidad de servicio y aplicación, teniendo a la mano un producto competitivo contra marcas ya existentes y más costosas por el licenciamiento al que esta sometido, como podría ser el caso de marcas ya mencionadas, y que aunque algunas de ellas también trabajen bajo el sistema UNIX, la implementación hace que el producto se eleve en costos. Asterisk está al alcance de todos por ser de código abierto, se puede descargar e instalar en máquinas virtuales y practicar sobre su funcionamiento casi en un 80 %.

Lo importante también es que además se están creando nuevas oportunidades de estudio y laborales dedicados a esta tecnología, porque han visto que si se encuentra a la altura de los demás.

Recomendaciones

Si el tema es del interés del lector, se recomienda adentrarse a detalle tanto en la teoría como en la práctica en el mundo de Asterisk, ya que al ser código abierto está en constante evolución por distintas comunidades libres y privativas, además de que por las necesidades en el mercado van creciendo junto con la creación e implementación de nuevos servicios.

Así también se recomienda conocer el Sistema Operativo Linux en cualquier distribución para instalar Asterisk y de esta manera navegar por sus archivos de configuración o realizar scrips para alguna función requerida e identificar probables fallas que se pudiera tener en la instalación o ejecución. Así como tener conocimientos en algún lenguaje de programación ya sea en PHP, C++, etc.. Manejar lo que son bases de datos, por ejemplo el MySQL. Todo esto porque algunas de sus aplicaciones se pueden desarrollar bajo estos elementos.

Existe el Elastix y FreePBX (entre otros), ambos son software que trabajan bajo Asterisk pero la diferencia es que tienen interfaz web, todo se maneja con ventanas y botones, es más fácil de manejar pero lo importante es conocer sus entrañas que como ya se dijo, tiene Asterisk de raíz, se recomienda saber sus comandos, conocer sus archivos y dejar de temer a la ventana de Shell.

Anexo

3.6. Plan de Marcación

En el año de 1996 se inició la apertura de las telecomunicaciones en México con la introducción de nuevos operadores de telefonía. Por ellos se creó la Comisión Federal de Telecomunicaciones (COFETEL), desligada de la Secretaría de Comunicaciones y Transportes, y es la que regula y promueve el desarrollo eficiente de las telecomunicaciones en México.

A la par de lo anterior, se elaboró el Plan Técnico Fundamental de Numeración, en donde se dieron a conocer los requerimientos para que la asignación de la numeración nacional fuera equitativa, eficiente, justa en cuanto a los recursos que se dispongan.

En dicho Plan de Numeración se definen las estructuras que deberán de adoptar los números geográficos y no geográficos, los códigos de servicios especiales y los prefijos de acceso al servicio de larga distancia, así como los procedimientos de marcación que deberán de utilizarse para originar llamadas dentro del territorio nacional.

La primera versión de la numeración consistía en el número local (número de abonado, Recomendación UIT-T E.164)⁷ a siete cifras en todo el país, excepto en las ciudades de México, Guadalajara y Monterrey, en donde se incrementó a ocho cifras, dicho incremento se llevó a cabo incorporando al inicio del número local los últimos dígitos de su identificador de Región (indicativo nacional de destino, Recomendación UIT-T E.164).

⁷<https://www.itu.int/rec/T-REC-E.164-201011-1/es>

A partir del 17 de noviembre 2001, la estructura de numeración en México cambió. El número nacional aumentó de 8 a 10 cifras, conformado por el número de identificación de región (indicativo interurbano) y el número local.

Para una mayor referencia en cuanto al Plan de numeración telefónica actual, dejamos dos ligas para su consulta:

http://www.cofetel.gob.mx/wb/Cofetel_2008/Cofe_plan_tecnico_fundamental_de_numeracion

http://www.anatel.org.mx/regulacion/Plan_Tecnico_Fundamental_Numeracion.pdf

3.7. Plan de Interconexión

La Unión Internacional de Telecomunicaciones define a la interconexión como: "los arreglos técnicos y comerciales bajo los cuales los proveedores conectan su equipo, redes y les dan a su clientes la posibilidad de tener acceso a los clientes, los servicios y las redes de otros proveedores de servicios"⁸

La interconexión es un punto importante que define la estructura de mercado, la viabilidad de los competidores y el éxito de los programas de desregulación. Desde que empezaron las telecomunicaciones, la interconexión ha sido vital para asegurar la conectividad de las redes, pero como se han desarrollado y evolucionado las tecnologías de comunicación (el Internet, la telefonía móvil, las redes inalámbricas, los sistemas satelitales), la interconexión ha tomado un papel importante ya que esas tecnologías deben ser capaces de interactuar o converger unas con otras y que al mismo tiempo deben de permitir la coexistencia de distintas empresas que compiten por el mismo mercado.

El Plan Técnico Fundamental de Interconexión e Interoperabilidad, se divide en 6 capítulos cuyo objetivo es establecer las disposiciones a que deberán sujetarse los concesionarios para proveer Servicios de Interconexión entre RPTs⁹.

Mencionaremos de manera breve dos puntos importantes a tomar en cuenta:

⁸ITU, 1995: 15

⁹Red(es) Pública(s) de Telecomunicaciones

1. *Señalización: de conformidad con el Plan de Señalización, la Interconexión de RPTs se realizará utilizando el sistema de señalización por canal común número 7 (SS7) y las Normas Oficiales Mexicanas que para tales efectos sean emitidas, así como otros protocolos que en el futuro sean incorporados a tal disposición y en su defecto las recomendaciones emitidas por la Unión Internacional de Telecomunicaciones. Los Concesionarios deberán observar al menos lo siguiente:*

- *Abstenerse de llevar a cabo acciones que deriven en el ocultamiento o interpretación errónea del origen o naturaleza del tráfico.*
- *Abstenerse de llevar a cabo acciones que deriven en una interpretación errónea con respecto al Concesionario que debe pagar las Tarifas de Interconexión correspondientes.*

En caso de que el Concesionario Solicitante no utilice en su red el sistema de señalización SS7 y la Interconexión no pueda realizarse bajo otra norma o estándar reconocido por ambas RPTs, será su responsabilidad instalar la compuerta respectiva que le permita llevar a cabo la Interconexión utilizando dicho protocolo. Sin perjuicio de lo anterior, los Concesionarios podrán acordar la utilización de protocolos de señalización alternativos que les permitan lograr una mayor eficiencia en la Interconexión de sus respectivas redes.

2. *Transmisión: los enlaces para la Interconexión entre redes deberán establecerse de manera digital utilizando el formato TDM (Multiplexión por División de Tiempo) con capacidad de nivel E1, o en múltiplos o submúltiplos de dicha capacidad, de acuerdo a la recomendación de la Unión Internacional de Telecomunicaciones UIT-T G.703 o la Norma Oficial Mexicana.*

Dichos enlaces podrán ser uni-direccionales o bidireccionales y se usará el total de la capacidad disponible en dichos enlaces. Para el caso de capacidades de nivel E1 se usarán los 31 canales disponibles. De igual forma, podrán establecerse enlaces para cursar tipos de tráfico específicos o mezclados, de conformidad con lo que acuerden las partes o en su caso, resuelva la Comisión.

En caso de que el Concesionario Solicitante no utilice en su red el protocolo TDM, será su responsabilidad instalar la compuerta respectiva que le permita llevar a cabo la

Interconexión utilizando dicho protocolo o alguno de los protocolos que posteriormente sean reconocidos. Sin perjuicio de lo anterior, los concesionarios podrán acordar la utilización de protocolos de transmisión alternativos que les permitan lograr una mayor eficiencia en la Interconexión de sus respectivas redes.

En caso de que exista interés en incorporar nuevos protocolos de señalización o estándares de transmisión, cualquier Concesionario podrá solicitar que la Comisión convoque a los demás Concesionarios a fin de analizar y recomendar sobre la adopción de los nuevos protocolos o estándares propuestos. Tomando en consideración las opiniones vertidas al respecto, la Comisión resolverá en definitiva.

Para mayor conocimiento de este Plan, se deja a continuación el link para consulta.

http://www.cft.gob.mx/work/models/Cofetel_2008/Resource/1863/1/Plan_Tecnico_17_abril.pdf

3.8. Plan de Señalización

El objetivo del Plan de Señalización es el establecimiento de las bases para un adecuado uso así como una buena administración de los recursos nacionales asociados a la señalización entre redes públicas de telecomunicaciones, todo esto para lograr que la interconexión e interoperabilidad de dichas redes sea eficiente y por lo tanto beneficie a los usuarios y operadores de telecomunicaciones de México.

En dicho Plan se dan a conocer que de manera administrativa la Secretaría tiene las siguientes facultades:

- *Interpretar el presente Plan para efectos administrativos.*
- *Asignar los códigos de puntos de señalización nacionales e internacionales a los operadores.*
- *Representar a México ante la U.I.T. en materia de señalización.*
- *Solicitar códigos de zona de señalización e identificación de red para los códigos de puntos de señalización internacional ante la U.I.T.*

- *Supervisar y controlar los recursos del Plan.*

entre otros.

En cuanto a los códigos de Punto de señalización, indica que la red mundial de señalización se encuentra estructurada en dos niveles funcionales, lo que hace que los planes de asignación de códigos para puntos de señalización nacionales o internacionales puedan ser independientes unos de los otros.

Para mayor conocimiento de este Plan, se deja a continuación el link para consulta.

http://www.sct.gob.mx/fileadmin/_migrated/content_uploads/118_Plan_Tecnico_Fundamental_de_Senalizacion.pdf

Bibliografía

Cabeza Galán, Antonio.(2000) *Fundamentos Básicos de las telecomunicaciones*. Madrid:CNSO

Escudero Pascual, Alberto y Berthilson, Louise. (2006) *VoIP para el desarrollo. Una guía para crear una infraestructura de voz en regiones en desarrollo*. Openlibra

Landívar, Edgar. (2008). *Comunicaciones Unificadas con Elastix*. Free Software Foundation, Inc.

Gómez López, Julio y Gil Montoya, Francisco. (2008) *Voip y asterisk. Redescubriendo la telefonía* Madrid. Alfaomega.

Joskowicz, José. (2011). *Conceptos básicos de la telefonía tradicional. Versión 6*.

Uruguay: Instituto de Ingeniería Eléctrica, Facultad de Ingeniería Universidad de la República.

Silva, Moisés & Alencar, Alexander (2009). *La Guía OpenR2*

Peters, James & Davidson, Jonathan (2000). *Voice over IP Fundamentals* USA: CISCO Press

Arena, Héctor Facundo (2003). *La biblia de Linux. Manual de uso, instalación y configuración*.

Buenos Aires Argentina: MP Ediciones

Van Meggelen, Jim & Madsen, Leif & Smith, Jared (2007) *Asterisk:The Future of Telephony*

USA: O'Reilly Media, Inc.

Goncalves, Flavio E. (2007).*Asterisk PBX "Guía de la Configuración"*. Janeiro

Cabrera Roselló, Christian. (2012). *Manual Curso de Asterisk*. México D.F.

Breve Historia de la telefonía

http://elastixbook.com/libros/cuce/vol1/es/capitulos/capitulo_1/1.1_Breve_historia_telefonia.html#_Toc168989278

26/11/2014

Funcionamiento del teléfono

www.grupoice.com/esp/cencon/pdf/educ/func_del_telefono.pdf

27/11/2013

Introducción a la Telefonía

http://www.naser.cl/sitio/Down_Papers/Introduccion%20a%20la%20telefonía.pdf

2/12/2014

DTMF

http://jupiter.utm.mx/tesis_dig/8076.pdf

2/12/2014

Señalización Ground Start

<http://what-when-how.com/cisco-voice-over-ip-cvoice/introducing-analog-voice-ports-on-cisco-ios-routers-routing-calls-over-analog-voice-ports-part-1/>

04/2014

Señalización R2

http://rosa_virgen_sm.geo.do/Comunicaciones/Tel_celular/R2_final.pdf

26/12/2013

Guía de referencia para instalaciones de Elastix IP PBX Versión 1.5.2-2 y E1 MFC/R2

<http://blogs.elastix.org/es/2009/11/30/guia-de-referencia-para-instalaciones-de-elastix%C2%AE-ip-pbx-version-1-5-2-2-y-e1-mfcr2/>

23/03/2014

Modelo OSI

http://www.info-ab.uclm.es/labelc/solar/Comunicacion/Redes/index_files/Modelos.htm

23/03/2015

Modelos OSI TCP/IP

<http://travelcomdata.blogspot.mx/2011/03/modelos-osi-tcpip.html> 23/03/2015

Protocolo UDP

[http://technet.microsoft.com/es-es/library/cc785220\(v=ws.10\).aspx](http://technet.microsoft.com/es-es/library/cc785220(v=ws.10).aspx)

23/03/2015

Datagrama IP

<http://www.tecnodelinglesalcastellano.com/2011/07/tamano-de-datagramas-ip-unidad-de.html>

20/03/2015

RFC0791. Protocolo IP

<http://www.rfc-es.org/rfc/rfc0791-es.txt>

23/03/2015

RFC0768. UDP

<http://www.rfc-es.org/rfc/rfc0768-es.txt>

23/03/2015

RTP-RTCP

http://www.efort.com/media_pdf/RTP_ES_EFORT.pdf

23/03/2015

RTP

<http://profesores.elo.utfsm.cl/agv/elo329/elo325/trabajos/temas1s01/Areas.html>

23/03/2015

RTP

<http://www.uco.es/i62gicaj/RTP.pdf>

23/03/2015

H323

<http://www.protocols.com/papers/voip2.htm>

26/11/2013

Asterisk

<http://www.asterisk.org/get-started>

26/11/2013

Versiones de Asterisk

http://www.quarea.com/es/versiones_de_asterisk

23/03/2015

VOIP

<http://itzamna.bnct.ipn.mx:8080/dspace/bitstream/123456789/5733/1/ice48.pdf>

2/12/2013

Wireshark

<http://voztovoice.org/?q=node/548>

2/12/2013

Asterisk

http://wikiasterisk.com/index.php/P%C3%A1gina_principal

2/12/2013

Wireshark

<http://www.unappel.ch/public/100119-wireshark-xmlite/>

2/12/2013

Comprendiendo mensajes SIP

<http://www.3cx.es/comprendiendo-mensajes-sip>

2/12/2013

Códec ILBC

http://www.wikiasterisk.com/index.php/Codecs_y_Formatos#ilbc

04/2014

RFC3951. ILBC

<http://www.ietf.org/rfc/rfc3951.txt>

07/2014

Códec

<http://dspace.espace.edu.ec/bitstream/123456789/2540/1/18T00516.pdf> 07/2014

Códec

<http://www.paratorpes.es/manuales/standares%20g.html>

07/2014

Codecs y formatos

http://www.wikiasterisk.com/index.php/Codecs_y_Formatos#ilbc

11/2014

Codecs

<https://www.sinologic.net/blog/2008-03/codec-g729-diferencias-entre-la-version-free-y-la-comercial.html>

12/2014

Normatividad de telecomunicaciones

<http://www.sct.gob.mx/acerca-de/normatividad/telecomunicaciones/disposiciones-relevantes/>

23/03/2015