ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

EVALUACIÓN DEL SISTEMA DE TELEFONÍA IP ASTERISK MEDIANTE LA IMPLEMENTACIÓN DE UN PROTOTIPO DE RED EN AMBIENTES IPV4 E IPV6

PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES

SOLIS HERRERA FRANCISCO JAVIER

panchin_ups@hotmail.com

VACA ARAUJO XAVIER DAVID

dav1ne@hotmail.com

DIRECTOR: EGAS ACOSTA CARLOS ROBERTO MSc.

cegas@ieee.org

Quito, Enero 2014

DECLARACIÓN

Nosotros, Solis Herrera Francisco Javier y Vaca Araujo Xavier David, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedemos nuestros derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

_____ _____

Solis Herrera Francisco Javier **Vaca Araujo Xavier David**

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Solis Herrera Francisco Javier y Vaca Araujo Xavier David, bajo mi supervisión.

Ing. Carlos Egas

DIRECTOR DEL PROYECTO

AGRADECIMIENTO

Agradezco a Dios por todas las bendiciones recibidas y por haberme permitido concluir mi más grande sueño.

A mi madre Yolanda Herrera por su infinito cariño, amor, cuidado, paciencia y apoyo a lo largo de toda mi vida; gracias mamita linda por haberme enseñado a levantarme con más fuerza luego de cada caída y seguir siempre hacia adelante con más optimismo.

A mi padre Alonso Solis por ser un ejemplo de responsabilidad, honestidad, esfuerzo, sacrificio y perseverancia; gracias viejito por tu entrega diaria y lucha constante para que nunca nos falte lo necesario y por hacernos felices.

A todos mis Hermanos por ser un ejemplo de lucha, esfuerzo y perseverancia; en especial a Eduardo y Jackelinne por haber confiado en mí y apoyarme cuando lo necesité. Estaré eternamente agradecido.

A la mujer que amo, la madre de mi hijo. Ody gracias por todo tu amor, entrega, empuje, consejos y paciencia durante estos maravillosos años junto a ti.

Al Ing. Carlos Egas por su apoyo constante durante la realización del presente proyecto de titulación.

A mi compañero de Tesis, mi gran amigo David Vaca por su apoyo, sacrificio, responsabilidad y compañía para la culminación de este proyecto.

A todos mis amigos por ser un parte importante de mi vida y permitirme compartir experiencias inolvidables.

A todos los profesores de la Facultad por haber sido la luz del conocimiento y saber impartir sus enseñanzas con entrega y dedicación.

Francisco Solis H.

AGRADECIMIENTO

Las palabras no serían suficientes para agradecer el inmenso apoyo y retribuir la ayuda en el presente proyecto de titulación; es menester primero agradecer a Dios por concluir no solo el proyecto sino una ardua trayectoria académica.

Un gran agradecimiento a nuestro director el Ing. Carlos Egas por la conducción, asistencia y la atención brindada a lo largo de todo el proceso de desarrollo del proyecto.

Un reconocimiento especial a mi familia, a quien estaré eternamente agradecido por su incondicional afecto abnegación y apoyo desde el primero hasta el último momento.

De igual manera quiero agradecer a mi compañero de tesis Francisco por la dedicación y entrega en el presente proyecto además de ser un gran amigo fuera y dentro de las aulas.

A mis amigos quiero darles las gracias por acompañarme durante todo el camino y compartir toda esta experiencia en la vida universitaria.

Finalmente quiero agradecer a mis maestros que supieron instruirme y compartirme sus enseñanzas durante toda mi vida estudiantil.

David Vaca

DEDICATORIA

Al culminar una etapa más de mi vida quiero dedicar este logro al primer amor de mi vida YOLANDA HERRERA, mamita linda lo que soy te lo debo a ti, a tu amor, a tu dulzura, a tu ejemplo y a tus enseñanzas.

Una dedicatoria muy especial a ti viejito lindo por haberme enseñado el significado de lucha constante y perseverancia. Me enseñaste que los sueños se pueden hacer realidad pero que para conseguirlos tenemos que luchar por ellos, que nada es producto de la suerte, que todo se gana en base a trabajo y sacrificio y que nada es imposible para quien quiere hacerlo y se esfuerza por conseguirlo.

A mi Hermano Eduardo por confiar en mí y por ser un apoyo constante en una de las etapas más importantes para poder alcanzar un título profesional y contar con más oportunidades en la vida.

A ti Matías te dedico este logro, espero poder ser un buen padre para ti y transmitirte todo lo que tu abuelo me enseñó, te has convertido en mi motor de inspiración y en mi amor infinito, te amo hijo mío.

Francisco Solis H.

DEDICATORIA

A Dios y mi familia, sin quienes no podría haber conquistado este reto.

A mis padres por sentar en mi las bases y la convicción de alcanzar mis sueños, por mostrarme su ejemplo de superación, entrega y comprensión en cada día, al amor que destellan le dedico esta consecución.

A mis hermanos por compartir un lazo más allá de la sangre y permitirme ser parte de sus alegrías de sus vivencias en el día a día, les dedico este objetivo y con todo cariño que puedan superar y cumplir todas sus metas.

A mis abuelitos por mostrarme su cariño y amor así como el ejemplo de su templanza y constancia, a su sabio concejo dedico este proyecto.

Una dedicación especial para Dios que me permite compartir este logro con mi familia, tenerlos a mi lado en esta y todas las etapas de mi vida es para mí lo más importante.

David Vaca

TABLA DE CONTENIDO

DECLARACIÓN ..... *I*

CERTIFICACIÓN ..... *II*

AGRADECIMIENTO ..... *III*

DEDICATORIA ..... *V*

ÍNDICE DE FIGURAS ..... *XII*

ÍNDICE DE TABLAS ..... *XVII*

RESUMEN ..... *XIX*

PRESENTACIÓN ..... *XXI*

CAPÍTULO 1

CONCEPTOS FUNDAMENTALES DE VoIP *1*

1.1 VOIP (VOICE OVER INTERNET PROTOCOL) **1**

1.1.1 VENTAJAS DE VOIP 1

1.2 FUNCIONAMIENTO DE VOIP **2**

1.3 PROTOCOLOS, ESTÁNDARES Y CODECS EN LA TECNOLOGÍA DE VOIP **2**

1.3.1 PROTOCOLOS DE SEÑALIZACIÓN 4

1.3.2 PROTOCOLOS DE TRANSPORTE 11

1.3.3 CODECS DE AUDIO 12

1.4 PROBLEMAS A RESOLVER EN VOIP **15**

1.4.1 RETARDO O LATENCIA 15

1.4.2 FLUCTUACIÓN DE FASE (JITTER) 16

1.5 TELEFONÍA IP **16**

1.5.1 SISTEMA IP PURO COMO SOLUCIÓN DE TELEFONÍA 17

1.6 ASTERISK **17**

1.6.1 ARQUITECTURA DE ASTERISK 18

1.6.2 CONCEPTOS DE ASTERISK 19

1.6.3 ARQUITECTURA DEL DIALPLAN 20

1.7 HARDWARE Y SOFTWARE PARA TELEFONÍA IP 23

1.7.1 INTERFACES PARA DISPOSITIVOS DE VOZ 23

1.7.2 TELÉFONOS IP 23

1.7.3 SOFTPHONES 23

CAPÍTULO 2

CARACTERÍSTICAS GENERALES DE LOS PROTOCOLOS IPV4 E IPV6 24

2.1 INTRODUCCIÓN 24

2.2 LIMITACIONES DE IPV4 24

2.2.1 AGOTAMIENTO DE LAS DIRECCIONES IP 25

2.2.2 NECESIDAD DE UNA SIMPLE CONFIGURACIÓN 25

2.2.3 REQUERIMIENTOS DE SEGURIDAD A NIVEL DE IP 25

2.2.4 NECESIDAD DE UN MEJOR SOPORTE EN LA ENTREGA EN TIEMPO REAL DE LOS DATOS 25

2.3 DESCRIPCIÓN DE IPV6 25

2.3.1 CARACTERÍSTICAS DE IPV6 26

2.3.2 ESTRUCTURA DEL ENCABEZADO DE IPV6 27

2.3.3 ENCABEZADOS DE EXTENSIÓN 29

2.3.4 DIRECCIONAMIENTO 29

2.4 FUNCIONALIDADES DE IPV6 33

2.4.1 ICMPV6 33

2.4.2 DESCUBRIMIENTO DE VECINOS 34

2.4.3 AUTOCONFIGURACIÓN DE DIRECCIONES STATELESS 34

2.4.4 AUTOCONFIGURACIÓN DE DIRECCIONES STATEFUL 34

2.5 ENRUTAMIENTO EN IPV6 35

2.5.1 TABLA DE ENRUTAMIENTO 35

2.5.2 ENRUTAMIENTO ESTÁTICO 35

2.5.3 ENRUTAMIENTO DINÁMICO 36

2.6 COEXISTENCIA Y TRANSICIÓN DE IPV4 A IPV6 38

2.6.1 DOBLE PILA 39

2.6.2 TUNELIZACIÓN 39

2.6.3 TRADUCCIÓN 40

2.7 COMPARACIÓN IPV4 FRENTE A IPV6 40

2.7.1 CAMBIOS EN EL DIRECCIONAMIENTO 40

2.7.2 CAMBIOS A NIVEL DE CABECERAS 41

2.7.3 FRAGMENTACIÓN DE PAQUETES 42

2.7.4 IMPLEMENTACIÓN DE LA SEGURIDAD. 42

2.7.5 COMPATIBILIDAD EN MECANISMOS DE ENRUTAMIENTO 42

CAPÍTULO 3

ANÁLISIS Y DETERMINACIÓN DE LOS REQUERIMIENTOS PARA LA IMPLEMENTACIÓN DEl PROTOTIPO 43

3.1 REQUERIMIENTOS 43

3.1.1 REQUERIMIENTOS A NIVEL DE RED 43

3.1.2 REQUERIMIENTOS A NIVEL DE SERVICIOS Y APLICACIONES 45

3.1.3 CONSIDERACIONES DE SEGURIDAD 47

3.2 DESCRIPCIÓN DE La topología del PROTOTIPO 49

3.2.1 SEGMENTO DE RED LOCAL 49

3.2.2 SEGMENTO DE RED IPV6 PURA 50

3.2.3 SEGMENTO DE RED MIXTO 50

3.3 DISEÑO DEL PROTOTIPO 50

3.3.1 DISEÑO DE LA RED ACTIVA 50

3.3.2 DISEÑO DE LA CENTRAL IP PBX 67

3.4 IMPLEMENTACIÓN DEL PROTOTIPO 85

3.4.1 CONFIGURACIÓN DE LOS EQUIPOS DE RED 85

3.4.2 REGISTRO DE USUARIOS EN EL SERVIDOR ASTERISK 90

3.4.3 CONFIGURACIÓN DE SERVIDORES 92

3.4.4 SERVICIOS DE TELEFONÍA IP IMPLEMENTADOS 94

3.4.5 IMPLEMENTACIÓN DEL ESCENARIO PARA EL ANALISIS DE RENDIMIENTO Y DENEGACIÓN DE SERVICIO 109

CAPÍTULO 4

PRUEBAS Y RESULTADOS 112

4.1 PRUEBAS DE CONECTIVIDAD 112

4.1.1 PRUEBAS DE RED 112

4.2 ESTABLECIMIENTO DE LLAMADAS 118

4.2.1 EN AMBIENTES IPV4 118

4.2.2 EN AMBIENTES IPV6 128

4.2.3 EN AMBIENTES MIXTOS 129

4.3 PRUEBAS DE SEGURIDAD 135

4.4 PRUEBAS DE RETARDO 137

4.4.1 FUENTES DE RETARDO 137

4.4.2 RETARDO DE PAQUETES DE VOIP 141

4.5 MEDICIÓN DE JITTER 147

4.5.1 ANÁLISIS DE RESULTADOS 149

4.6 ANCHO DE BANDA 150

4.6.1 ANÁLISIS DE RESULTADOS 155

4.7 PRUEBAS DE RENDIMIENTO 156

4.7.1 EJECUCIÓN DE LA PRUEBA 156

4.7.2 RESULTADOS 157

4.8 PRUEBAS DE PÉRDIDA DE PAQUETES 160

4.9 SOLUCIÓN DE PROBLEMAS 165

4.9.1 PROBLEMAS DE NAT Y VOIP 165

4.9.2 RETARDOS EXCESIVOS 166

4.9.3 REDUCCIÓN DEL JITTER 167

4.9.4 PORCENTAJE DE PÉRDIDA DE PAQUETES 168

4.9.5 ALTERNATIVAS DE SEGURIDAD 169

CAPÍTULO 5

CONCLUSIONES Y RECOMENDACIONES 170

5.1 CONCLUSIONES 170

5.2 RECOMENDACIONES 174

BIBLIOGRAFÍA..... 176

ANEXO A:

INSTALACIONES..... A-1

A.1 INSTALACIÓN DE LINUX A-2

A.2 INSTALACIÓN DE ASTERISK A-14

A.3 INSTALACIÓN DE SOFTPHONES A-21

A.4 INSTALACIÓN DE CLIENTE DE CORREO SQUIRRELMAIL A-34

ANEXO B:

ARCHIVOS DE CONFIGURACIÓN DE ASTERISK B-1

B.1 ARCHIVO SIP.CONF B-2

B.1.1 Archivo sip_trunks.conf B-5

B.1.2 Archivo sip_custom.conf B-5

B.2 ARCHIVO IAX.CONF B-6

B.3 ARCHIVO EXTENSIONS.CONF B-7

B.3.1 Archivo extensions_database.conf B-11

B.4 ARCHIVOS DE CONFIGURACIÓN DE DAHDI B-12

B.4.1 chan_dahdi.conf B-12

B.4.2 dahdi-channels.conf B-13

B.5 FEATURES.CONF B-14

B.6 MEETME.CONF B-17

B.7 VOICEMAIL.CONF B-17

ANEXO C:

ARCHIVOS DE CONFIGURACIÓN DE LOS ROUTERS C-1

C.1 ROUTER 1 ..... C-2

C.2 ROUTER 2 ..... C-4

C.3 ROUTER 3 ..... C-5

C.4 ROUTER 4 ..... C-6

C.5 ROUTER 5 ..... C-8

C.6 ROUTER 6 ..... C-10

ANEXO D:

HOW TO..... D-1

ÍNDICE DE FIGURAS

CAPÍTULO 1

Figura 1.1 Protocolos de la tecnología VoIP 3
Figura 1.2 Componentes del sistema SIP 5
Figura 1.3 Establecimiento de una llamada SIP 8
Figura 1.4 SIP en el stack de protocolos 9
Figura 1.5 Fases de una llamada IAX 10
Figura 1.6 Retardos en VoIP 16
Figura 1.7 Esquema de telefonía IP puro 17
Figura 1.8 Interconectividad de Asterisk 18
Figura 1.9 Arquitectura de Asterisk 19
Figura 1.10 Componentes de un *Dialplan* 21

CAPÍTULO 2

Figura 2.1 Estructura de un paquete IPv6 27
Figura 2.2 Cabecera IPv6 27
Figura 2.3 Cadena de encabezados de Extensión 29
Figura 2.4 Formato de una dirección IPv6 30
Figura 2.5 Ámbitos de una dirección IPv6 32
Figura 2.6 Paquete ICMPv6 33
Figura 2.7 Clases de Protocolos de Enrutamiento Dinámico 37
Figura 2.8 Arquitectura de la Capa Dual IP 39
Figura 2.9 Encapsulado de IPv6 en IPv4 39
Figura 2.10 Modificaciones en los encabezados 41

CAPÍTULO 3

Figura 3.1 Diagrama de Topología del Segmento de Red Local 51
Figura 3.2 Diagrama de Topología del Segmento de Red IPv6 puro 53
Figura 3.3 Esquema de conectividad del segmento de red mixto 56
Figura 3.4 Diagrama de topología del segmento de red Mixto 59
Figura 3.5 Diagrama del prototipo implementado..... 61
Figura 3.6 Configuraciones de sip.conf para protocolos de Internet 70
Figura 3.7 Tarjetas FXO Y FXS utilizadas en el prototipo 79
Figura 3.8 Teléfono IP D-LINK DPH-150S 80
Figura 3.9 Softphone Zoiper..... 81
Figura 3.10 Softphone Linphone 81
Figura 3.11 Interfaz gráfica del generador de tráfico Ostinato 83
Figura 3.12 Interfaz gráfica del generador de tráfico de VoIP SIPP 84
Figura 3.13 Topología para configuración de equipos 85
Figura 3.14 Topología para configuración de RIPng 87
Figura 3.15 Topología para configuración de Enrutamiento dinámico, NAT y Tunelización 88
Figura 3.16 Modelo de Túnel IPv6 sobre IPv4 90
Figura 3.17 Resultado del comando ***sip show peers*** 91
Figura 3.18 Archivo de configuración de las zonas 92
Figura 3.19 Archivo de configuración de resolución inversa 93
Figura 3.20 Página de inicio de cliente Squirrelmail 94
Figura 3.21 Archivo de configuración de ***meetme.conf*** 98
Figura 3.22 Archivo de configuración ***voicemail.conf*** 100
Figura 3.23 Pantalla de autenticación para cliente de correo..... 101
Figura 3.23 Interfaz gráfica de un cliente de correo (squirrelmail)..... 101
Figura 3.25 Resultado del comando ***database show*** 105
Figura 3.26 Diagrama de flujo del IVR 106
Figura 3.27 Archivo de configuración ***dahdi_channels.conf***..... 108
Figura 3.28 Entorno para la realización de la Denegación de Servicio 110
Figura 3.29 Configuración del canal SIPP (archivo sip.conf)..... 110
Figura 3.30 Configuración de la extensión para el canal SIPP..... 111

CAPÍTULO 4

Figura 4.1 Esquema del ambiente de pruebas 113
Figura 4.2 Ejecución del comando ping hacia la dirección pública del servidor 114
Figura 4.3 Ejecución del comando tracert hacia la dirección del servidor 115
Figura 4.4 Ejecución del comando ping a un *host* en una red remota IPv6 115
Figura 4.5 Ejecución del comando tracert desde un *host* remoto IPv6, hacia un *host* en la red del servidor 116
Figura 4.6 Ejecución de ping desde un *host* IPv6 que atraviesa una red IPv4 (mediante un túnel) hacia un *host* en la red del servidor 117
Figura 4.7 Ejecución comando tracert desde un *host* que atraviesa una red IPv4 mediante un túnel 117
Figura 4.8 Ejecución del comando ping desde el servidor hacia un *host* IPv6 detrás de un túnel 118
Figura 4.9 Ambiente de pruebas IPv4 119
Figura 4.10 Intercambio de mensajes en una llamada SIP-SIP 119
Figura 4.11 Captura de paquetes llamada SIP-SIP en un ambiente IPv4 121
Figura 4.12 Detalle de Captura de paquetes llamada SIP-SIP en un ambiente IPv4 121
Figura 4.13 Intercambio de mensajes en una llamada IAX-IAX 122
Figura 4.14 Intercambio de mensajes en una llamada SIP-IAX 124
Figura 4.15 Intercambio de mensajes en una llamada SIP-SIP a través de NAT 125
Figura 4.16 Llamada establecida entre un softphone con dirección IP pública y el servidor de telefonía 126
Figura 4.17 Intercambio de mensajes en una llamada entre teléfonos analógicos y teléfonos IP 127
Figura 4.18 Resultado en la consola de Asterisk para una llamada hacia la PSTN 127
Figura 4.19 Intercambio de mensajes en una llamada hacia la PSTN 128
Figura 4.20 Ambiente de pruebas IPv6 128
Figura 4.21 Intercambio de mensajes en una llamada SIP en ambientes IPv6 . 129
Figura 4.22 Ambiente de pruebas IPv4 e IPv6 130

Figura 4.23 Intercambio de mensajes en una llamada IPv4-IPv6 131
Figura 4.24 Intercambio de mensajes en una llamada IPv6-IPv4 132
Figura 4.25 Intercambio de mensajes llamada a través de un túnel 134
Figura 4.26 Autenticación de un usuario para registrarse 135
Figura 4.27 Registro de un usuario con lista de acceso 135
Figura 4.28 Denegación del registro con listas de acceso 136
Figura 4.29 Denegación del registro con listas de acceso 136
Figura 4.30 Diagrama de flujos de tráfico de paquetes de VoIP 142
Figura 4.31 Valores máximos de Retardo en un *Stream* RTP..... 143
Figura 4.32 Indicador de MOS de Linphone..... 144
Figura 4.33 Procedimiento para obtención de datos de jitter en wireshark 148
Figura 4.34 Ancho de banda para un canal con códec GSM 150
Figura 4.35 Ancho de banda para un canal con códec G711 151
Figura 4.36 Ejemplo de obtención de parámetros de ancho de banda en wireshark 152
Figura 4.37 Ancho de banda teórico para llamada con códec GSM y G711 154
Figura 4.38 Ancho de banda del canal en una llamada SIP IPv4-Ipv6..... 154
Figura 4.39 Resultados de Gnome-System-Monitor al realizar la prueba 158
Figura 4.40 Indicador de estadísticas de SIPP..... 158
Figura 4.41 Monitor del Sistema antes de colapsar 161
Figura 4.42 Monitor del Sistema durante la congestión 161
Figura 4.43 Captura de *stream*s RTP generados con SIPP 162
Figura 4.44 Análisis de *stream* RTP..... 163
Figura 4.45 Mensajes de llamadas exitosas en la consola de Asterisk..... 164
Figura 4.46 Mensajes de llamadas no completadas en la consola de Asterisk.. 165
Figura 4.47 Configuración del jitter buffer en el fichero sip.conf..... 167
Figura 4.48 Mensajes de llamadas no completadas en la consola de Asterisk.. 168

ÍNDICE DE TABLAS

CAPÍTULO 1

Tabla 1.1 Principales Códecs para VoIP 14
Tabla 1.2 Caracteres usados para establecer un patrón de numeración 22

CAPÍTULO 3

Tabla 3.1 Direccionamiento IPv4 de la red Local del Servidor 52
Tabla 3.2 Direccionamiento IPv6 de la red Local del Servidor 52
Tabla 3.3 Direccionamiento IPv6 del segmento de Red IPv6 Puro 54
Tabla 3.4 Características del protocolo de enrutamiento a utilizar 55
Tabla 3.5 Direccionamiento IPv4 en el segmento de red mixto..... 57
Tabla 3.6 Direccionamiento IPv4 e IPv6 del Segmento de Red Mixto 59
Tabla 3.7 Direccionamiento IP para el protocolo IPv4..... 59
Tabla 3.8 Direccionamiento IP para el protocolo IPv6..... 60
Tabla 3.9 Requerimientos y especificaciones para *router*s 63
Tabla 3.10 Nomenclatura y serie de los *router*s del prototipo 64
Tabla 3.11 Características del *switch*..... 65
Tabla 3.12 Resumen de Características del IOS 12.3(10) 66
Tabla 3.13 Versiones de Asterisk..... 68
Tabla 3.14 Archivos de configuración incluidos en el diseño de la central..... 69
Tabla 3.15 Contextos y extensiones del Plan de Numeración 72
Tabla 3.16 Archivos de configuración adicionales..... 95
Tabla 3.17 Datos a ingresar en Asterisk 103

CAPÍTULO 4

Tabla 4.1Retardo Algorítmico 137
Tabla 4.2 Retardo de Paquetización 138
Tabla 4.3 Resolución de tiempo de lectura según el tipo de Sistema Operativo 139
Tabla 4.4Retardos de Red según el tipo de conexión 140
Tabla 4.5 Cálculo de Retardo Total en Paquetes de VoIP 144
Tabla 4.6 Retardos en paquetes VoIP según el tipo de llamada 145
Tabla 4.7 Datos de Jitter obtenidos del prototipo 149
Tabla 4.8 Anchos de banda obtenidos de acuerdo a cada tipo de llamada 153
Tabla 4.9 Anchos de banda obtenidos de acuerdo a cada tipo de llamada 153
Tabla 4.10 Tasas de error porcentuales de ancho de banda 155
Tabla 4.11 Resumen de Pruebas de rendimiento del Servidor 159

RESUMEN

El presente proyecto de titulación propone el análisis de los requerimientos para el diseño de una central de telefonía IP implementada mediante Asterisk, la misma que es evaluada en ambientes IPv4, en ambientes IPv6 y en ambientes mixtos. Se analizan las características más importantes tales como son retardos, ancho de banda, jitter, pérdida de paquetes y calidad de la comunicación; para así determinar la funcionalidad de Asterisk al trabajar en ambientes mixtos, así como también para destacar las características relevantes de los protocolos de Internet involucrados (IPv4 e IPv6).

En el capítulo 1 se hace un estudio de la tecnología de VoIP, protocolos de señalización y transporte, codecs, servicios y aplicaciones de la misma. Se compara la telefonía IP y la telefonía tradicional (analógica) para establecer las ventajas y desventajas entre las dos tecnologías. Se realiza también un estudio del *software* Asterisk, el mismo que permite utilizar los servicios de telefonía IP y probar la funcionalidad del mismo en ambientes de red mixtos.

En el capítulo 2 se realiza un análisis de los Protocolos de Internet IPv4 e IPv6, detallando las características más importantes de los mismos tales como: direccionamiento y enrutamiento. En este capítulo se resalta principalmente las limitaciones de IPv4 y las ventajas de IPv6, para así hacer un análisis de las implicaciones de la migración de IPv4 a IPv6; así como también la convivencia de ambos protocolos y se analiza la compatibilidad entre los mismos. Al final de este capítulo se hace una comparación entre ambos protocolos, para de esta manera poder entender el porqué del surgimiento de un nuevo protocolo que va a remplazar al actual.

En el Capítulo 3 se realiza el análisis de los requerimientos (de red y de telefonía) para el diseño del prototipo. Se incorporan varios servicios de telefonía tales como: contestadora automática, buzón de voz, música en espera, voicemail, integración con bases de datos, directorio, IVR, integración de telefonía IP con la PSTN, transferencia de llamadas, grabación de llamadas, etc.

Se realiza además la descripción de los archivos de configuración de Asterisk más importantes, para el levantamiento de la central permitiendo que la misma trabaje con los dos protocolos de Internet estudiados en el capítulo 2.

En el Capítulo 4 se hace una descripción de las pruebas realizadas en el prototipo para examinar el estado en cuanto a conectividad y rendimiento del mismo. La pruebas se realizan mediante un *software* analizador de tráfico, que permita recopilar la información requerida para tal efecto.

Finalmente en el capítulo 5 se presentan las conclusiones y recomendaciones, generadas al realizar el estudio y la implementación del prototipo, con las características mencionadas.

PRESENTACIÓN

Hoy en día las comunicaciones han evolucionado en forma rápida creando la necesidad de disminuir costos y sobre todo el hecho de poseer redes convergentes. Estos antecedentes sumados al creciente número de usuarios conectados a Internet y la deficiente manera actual en que las direcciones IPv4 son asignadas, han provocado una pronta escasez en el espacio de direcciones. Por estas razones IPv6 se presenta como la solución más viable a dicho problema, adicionalmente incluye entre otras mejoras como: mayor espacio de direccionamiento, seguridad y calidad de servicio.

La industria telefónica en los últimos tiempos, ha desarrollado equipos y *software* que nos permiten tener un mayor número de servicios y aplicaciones muy versátiles. Este avance se produce gracias al desarrollo de las aplicaciones de código abierto y la tecnología VoIP. La telefonía logró un extraordinario avance en su implementación, que tradicionalmente se mostraba costosa, complicada de llevar a cabo y además carecía de facilidad al momento de incorporar una solución basada en *software*.

El mayor espacio de direccionamiento y mejores características del protocolo IPv6, así como las mayores prestaciones y facilidades de la telefonía IP revelan un nuevo escenario para la implementación de sistemas de telefonía. Por esta razón es evidente la necesidad de analizar los fundamentos teóricos y el modo de implementar servicios de telefonía sobre IPv6 e IPv4. Este análisis se completa mediante la implementación de un proyecto que analice el enrutamiento y protocolos necesarios que interconecten ambientes netamente IPv4 con IPv6.

El diseño del proyecto permite obtener una solución que incorpore interoperabilidad y además posibilite una comunicación de voz y datos con mejores prestaciones que la telefonía tradicional. Entre las características del proyecto debe considerarse un diseño económico y efectivo, compatible con las redes actuales IPv4 y con funcionamiento a largo plazo.

En este contexto, el presente proyecto proporciona un prototipo que brinda experiencias de las ventajas y desventajas que implica la coexistencia y compatibilidad en ambientes mixtos IPv4 e IPv6. Estas experiencias servirán como base y guía gracias a los resultados y depuración de las distintas pruebas. Adicionalmente se sientan los precedentes para una futura implementación de proyectos con aplicaciones de telefonía IP. Los entornos abarcados incluyen los ambientes que actualmente operen en IPv4, ambientes en migración a IPv6 y ambientes netamente de IPv6. Es decir que en general podría ser implementado, sobre cualquier tipo de red actual o futura en cuanto a direccionamiento.

CAPÍTULO 1

CONCEPTOS FUNDAMENTALES DE VOIP

1.1 VOIP (*VOICE OVER INTERNET PROTOCOL*)

La VoIP es la tecnología utilizada para la transmisión de voz sobre una red de datos (internet, intranet, ISP[1]) utilizando para este propósito el protocolo IP[2]. La VoIP convierte las señales de voz en paquetes de datos los mismos que son transportados a través de la red mediante el protocolo IP en lugar de líneas telefónicas. Puesto que esta tecnología convierte la voz en paquetes de datos para transportarlos a través de la red se utiliza conmutación de paquetes[3].

Esta tecnología posee características tales como: recepción de mensajes de voz en el correo electrónico, identificación de llamadas, bloqueo de llamadas, transferencia de llamadas, interacción con bases de datos, etc.

1.1.1 VENTAJAS DE VOIP

La VoIP puede ser usada para remplazar la telefonía tradicional en un entorno empresarial, en un pequeño negocio o en casa, o simplemente para añadir ventajas a un sistema de telefonía tradicional. [**1**]

Existen diversas ventajas, entre las cuales mencionamos las principales:

- Comunicación efectiva entre usuarios de distintos lugares a un bajo costo.
- Tráfico de voz y datos integrado sobre el mismo acceso.
- Independencia de tecnología de transporte (Ethernet, Token Ring, etc.) y del dispositivo de acceso, soportando desde teléfonos convencionales,

[1] *Internet Service Provider* (ISP): Organización que brinda conexión de Internet a sus clientes.
[2] Internet Protocol (IP): Método o protocolo por el cual los datos son enviados de un dispositivo a otro en internet.
[3] Conmutación de paquetes: Envió de datos a través de una red en la cual el mensaje se divide en varios paquetes los mismos que pueden tomar varios caminos para alcanzar el destino final, por lo tanto no llegan necesariamente en un orden establecido.

teléfonos IP, teléfonos inalámbricos (*wireless*), hasta computadoras personales, portátiles y otros dispositivos móviles. [**2**]

- Escalabilidad de usuarios.
- Transmisión en forma confiable y segura de voz, datos y video.
- Disponibilidad de varios servicios de manera universal y económica.
- Interconexión con otras redes (la PSTN por ejemplo).
- Interoperabilidad de diversos proveedores: Al usar un protocolo universal es compatible con otras redes.
- Servicios de valor añadido como correo de voz, identificador de llamadas, bloqueo de llamadas, desvío de llamadas, interacción con base de datos, etc.
- Gracias a la existencia de teléfonos *software* (*softphones*[4]) un usuario podrá disponer de su propia extensión sin la necesidad de tener un terminal físico.

1.2 FUNCIONAMIENTO DE VOIP

La VoIP digitaliza la voz y luego la convierte en paquetes de datos comprimidos, los mismos que en el destino son nuevamente convertidos en voz estándar.

Los paquetes de voz son transportados a través de redes de datos en lugar de líneas telefónicas dedicadas; estas señales de voz son encapsuladas en paquetes IP y transportadas a través de redes IP por medio de enlaces tipo *Ethernet, Frame Relay, ATM*[5], entre otros.

1.3 PROTOCOLOS, ESTÁNDARES Y CODECS EN LA TECNOLOGÍA DE VOIP

Para poder realizar una llamada es necesario que existan acuerdos entre ambas partes para poder establecer y mantener la misma; a estas reglas se las conoce

[4]Softphone: Es un software que hace la simulación de teléfono convencional por computadora
[5]Asynchronous Transfer Mode (ATM): Tecnología de telecomunicación en la cual la información es transmitida en forma de paquetes de tamaño fijo (celdas ATM) que pueden ser enrutadas individualmente mediante el uso de los canales virtuales.

como protocolos[6]; los cuales son los encargados, entre otras cosas de regular y permitir la interconexión de equipos de distintos fabricantes.

En VoIP existen protocolos de señalización y transporte. La señalización está encargada de mantener y administrar una llamada entre dos dispositivos.

Los protocolos de señalización son los encargados del establecimiento de la llamada; un protocolo de señalización es aquel que se encarga de gestionar los mensajes y procedimientos utilizados para poder establecer una comunicación. [**3**]

Los protocolos de transporte son los encargados de asegurar la comunicación de voz. Los protocolos más extendidos e importantes para VoIP se presentan en la figura 1.1. Sin embargo existen otros protocolos que no se incluyen en el desarrollo del presente prototipo, pero que por su relevancia merecen ser mencionados y se presentan en la siguiente figura.

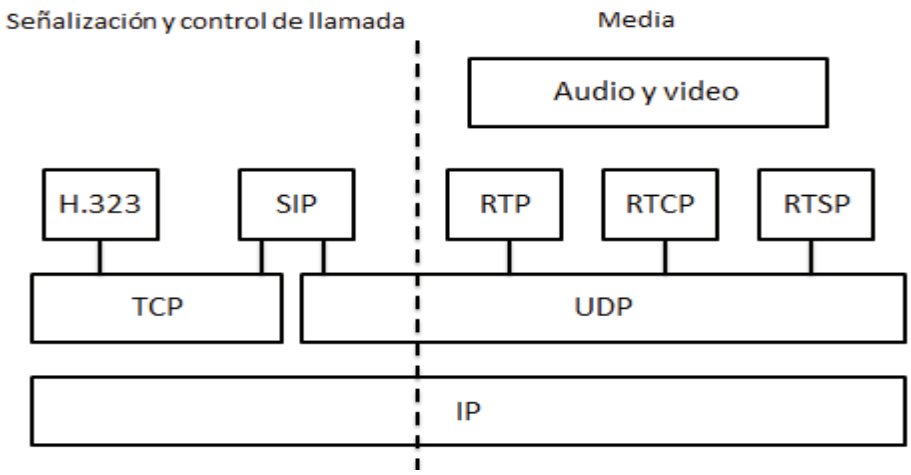


Figura 1.1 Protocolos de la tecnología VoIP [**3**]

Los datos se encapsulan sobre los protocolos de transporte multimedia que utilizan los protocolos de señalización para establecer las llamadas.

Los protocolos de señalización pertenecen a la capa sesión, mientras los protocolos de transporte multimedia RTP están en la capa de transporte pero sobre UDP.

[6] Protocolo: Conjunto de reglas y acuerdos que los dispositivos deben seguir para poder comunicarse entre sí.

1.3.1 PROTOCOLOS DE SEÑALIZACIÓN [4]

Los protocolos de señalización realizan los siguientes procesos:

- **Localización de usuarios:** En cualquier parte de la red.
- **Establecimiento de sesión:** Permite al usuario aceptar o no la llamada.
- **Negociación de sesión:** Para negociar entre las partes el flujo de información, algoritmos de compresión, puertos, etc.
- **Gestión de los participantes en una llamada:** Añadir o eliminar los miembros de una sesión ya establecida.

1.3.1.1 SIP [5] [6] [7] [8] [9] [10]

SIP (*Sesion Initial Protocol*) se encuentra especificado en el RFC 3261, es un protocolo de señalización cliente-servidor encargado del inicio, mantenimiento y término de una sesión multimedia (voz y video) a través de una red de paquetes. Por su naturaleza punto a punto incorpora beneficios de la arquitectura web a la telefonía IP; hace posible el desarrollo de nuevos servicios y aplicaciones que no eran posibles en la telefonía tradicional. Adicionalmente es un protocolo fundamentado en texto altamente extensible basado en otros protocolos como HTTP[7].

En SIP la señalización y los datos viajan de manera separada teniendo debido a esto problemas con NAT [8] en el flujo de audio cuando este flujo debe superar los *router*s y *firewall*s. SIP utiliza el puerto 5060 para señalización y 2 puertos RTP por cada conexión de audio.

SIP es un protocolo de propósito general ya que puede transmitir sin dificultad alguna cualquier información y no sólo audio y video.

1.3.1.1.1 Características de SIP

Las características más importantes de *SIP* que lo hacen apto para el desarrollo de aplicaciones web y que incorporan funcionalidades de telefonía IP son:

---

[7] HTTP (*Hyper Text Transfer Protocol*): Protocolo de Red utilizado para transferencia de información de texto en la WEB.

[8] NAT (*Network Address Translation*): Mecanismo que convierte las direcciones IP privadas en públicas.

- ***Localización de usuarios:*** Busca (mediante dirección IP) y encuentra al usuario llamado en el dispositivo correspondiente y establece la conexión.
- ***Intercambio y negociación de capacidades de los terminales:*** SIP es capaz de negociar el tipo de códec y aplicaciones disponibles durante la sesión multimedia.
- ***Disponibilidad de usuarios:*** Se determina si el destinatario de la llamada está disponible, si así lo es se acepta o no la llamada.
- ***Gestión de Participantes:*** Se pueden incorporar así como también eliminar participantes a la comunicación durante la llamada.
- ***Direccionamiento estándar de Internet:*** Mismo formato de direccionamiento que internet tanto para nombres como para direcciones IP (ejemplo. nombre_usuario@nombre_dominio.com)
- ***Protocolo encapsulado en texto*:** Gracias a esto es posible una integración de aplicaciones web más simple con control de errores.
- ***Terminales multifuncionales inteligentes:*** Tanto en terminales telefónicas, computadores personales u otros dispositivos de comunicación como por ejemplo teléfonos 3G.

1.3.1.1.2 Componentes del Sistema SIP [3] [5] [7]

En SIP se especifican como elementos básicos a los servidores y a los agentes de usuario; dichos componentes se ilustran en la figura 1.2.

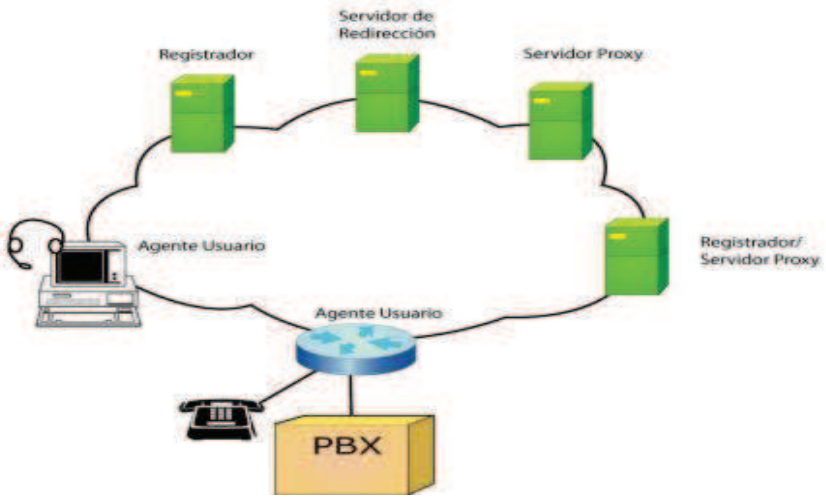


Figura 1.2 Componentes del sistema SIP **[7]**

Dentro de los servidores se puede definir la siguiente clasificación:

- ***Registradores:*** Toman datos acerca de la ubicación de usuarios nuevos que se conectan a la red; actualizando su base de datos si cambia su localización.
- ***Intermediarios:*** También conocidos como proxy, se encargan de orientar las peticiones y/o respuestas a su destino.
- **Redirección:** Procesan solicitudes SIP y retornan la dirección (IP o URL) de la parte llamada.

Por otro lado un agente de usuario es una entidad terminal que inicia y termina sesiones por intercambio de solicitudes y respuestas.

Los agentes de usuario pueden ser:

- ***Agente de usuario Cliente (UAC):*** Funciona como cliente iniciando peticiones SIP.
- ***Agente de usuario Servidor (UAS):*** Funcionando como servidor cuando una petición SIP es recibida y retornando una respuesta al usuario.

Los agentes de usuario realizan las siguientes tareas:

- Localizan a un usuario mediante la redirección de la llamada
- Implementan servicios de redirección si no hay respuesta
- Implementan filtrado de llamadas en función del origen o destino
- Almacenan información de administración de llamadas.

1.3.1.1.3 Mensajes SIP [3] [10]

El protocolo SIP define solicitudes y respuestas. Teniendo seis clases de solicitudes:

- ***INVITE:*** Para iniciar sesión.
- ***BYE:*** Enviado para finalizar una sesión.
- ***ACK:*** Confirma una solicitud *INVITE*.
- ***REGISTER:*** Transmite información de localización de usuario.
- ***CANCEL:*** Cancela el establecimiento de una sesión.
- ***OPTIONS:*** Informa las capacidades de envío y recepción de entidades SIP.
- ***STATUS:*** Informa al servidor acerca del estado de señalización de la sesión.

De la misma manera se cuenta con seis clases de respuesta:

- ***1XX:*** Respuestas informativas.
- ***2XX:*** Respuestas de éxito.
- ***3XX:*** Respuestas de redirección.
- ***4XX:*** Errores de solicitud.
- ***5XX:*** Errores de servidor, mediante un requerimiento aparentemente válido.
- ***6XX:*** Errores globales. La solicitud no se puede procesar por ningún servidor.

1.3.1.1.4 Establecimiento de una Sesión SIP [8]

Se realiza primero, el Registro de los usuarios enviando solicitudes *REGISTER*. El servidor Proxy que actúa como *Register*, verifica si el usuario puede ser autenticado y envía un mensaje de OK en caso positivo.

A continuación se realiza el establecimiento de la sesión, mediante una petición *INVITE* (usuario A) del usuario al proxy. El proxy responde con un TRYING 100 para parar las retransmisiones y reenvía un *INVITE* al usuario del otro lado de la llamada (usuario B). El usuario B envía un *RINGING* 180 cuando el teléfono suena y éste es a su vez es enviado por el proxy al usuario A; cuando el usuario B acepta la llamada (descuelga) se envía un *OK* 200.

A partir de este momento la llamada queda establecida e inicia el intercambio de los paquetes de voz en forma de tráfico RTP[9] durante toda la conversación.

Finalmente, corresponde realizar el cierre de sesión mediante una única solicitud *BYE*, enviada al proxy por cualquiera de los dos usuarios.

Un resumen detallado de lo antes descrito se lo muestra en la figura 1.3

[9] RTP (Real-Time Transport Protocol): Protocolo de capa sesión en el modelo OSI y de capa aplicación en TCP/IP que permite la transmisión de información en tiempo real tal como audio y video.

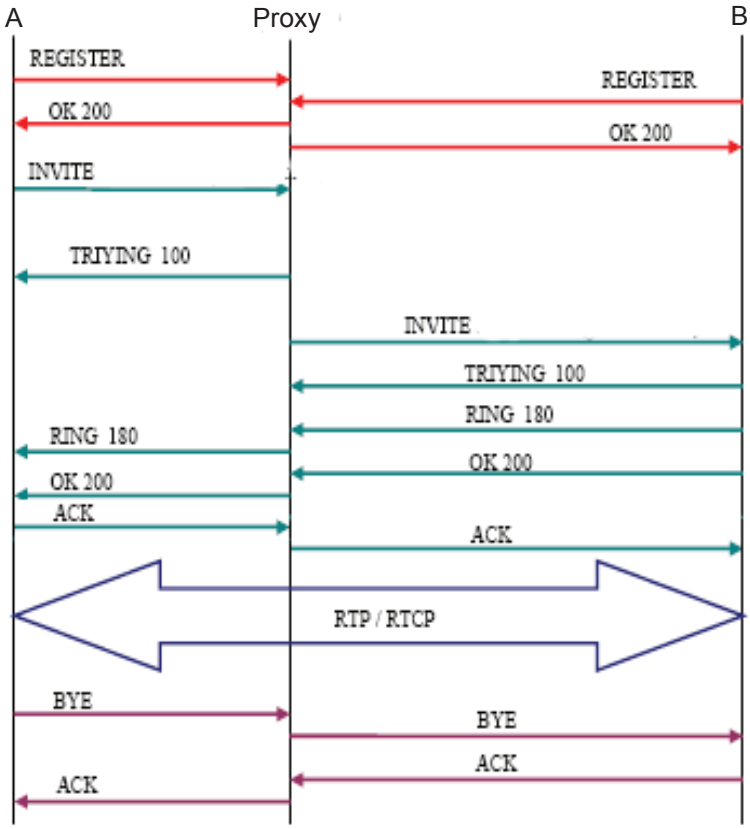


Figura 1.3 Establecimiento de una llamada SIP **[8]**

1.3.1.1.5 SIP y el Modelo OSI [10]

Algunos de los protocolos y mecanismos utilizados por SIP se listan a continuación y se ilustran en la figura 1.4:

- TCP/UDP: Para transportar información de señalización.
- DNS: Para resolver nombres de dominio.
- RTP: Para transportar las comunicaciones de voz, datos y video.
- RTSP: Para controlar el envío de *stream*ing media.
- XML [10](*eXtensible Markup Language*): Para transmitir información de eventos.
- MIME (*Multipurpose Internet Mail Extension*): Para el intercambio transparente de cualquier tipo de archivo en Internet. intercambio a través de Internet de todo tipo de archivos (texto, audio, vídeo, etc.)

[10] XML (*eXtensible Markup Language*): Lenguaje de marcas o etiquetas que permite la organización y etiquetado de documentos de la web.

- SAP[11] (*Session Advertisement Protocol*): Para publicar sesiones multimedia vía *multicast*.

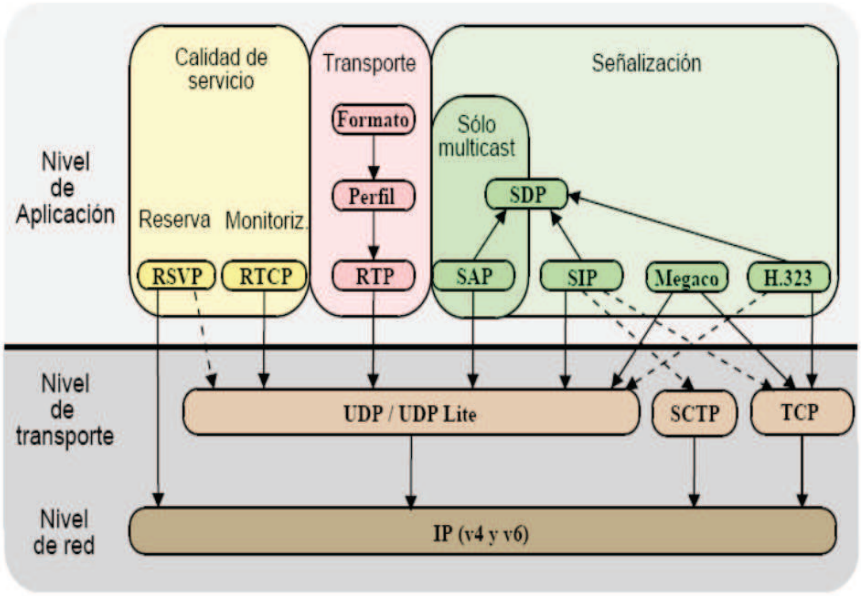


Figura 1.4 SIP en el stack de protocolos **[**10**]**

1.3.1.2 IAX (*Inter-Asterisk eXchange Protocol*) [3] [9] [10] [11]

IAX es un protocolo propietario, desarrollado por la empresa *DIGIUM,* está dedicado a la comunicación entre servidores y clientes Asterisk.

IAX utiliza un solo puerto (4569) para enviar conjuntamente información de señalización y datos, esto mediante un mecanismo de multiplexación o "*trunking*". El *trunking* (troncalización) permite que varias comunicaciones puedan ser mostradas en un solo canal; de tal forma que se disminuye la latencia y el ancho de banda requerido para la transmisión.

En IAX todo el tráfico de audio debe pasar obligatoriamente por el servidor IAX; lo que se resume en un aumento del ancho de banda en los servidores IAX, sobre todo cuando existen muchas llamadas simultáneas.

El protocolo IAX se refiere generalmente a la segunda versión del protocolo IAX2.

[11] SAP (*Sesion Advertisement Protocol*): Protocolo de aviso de sesión, tiene como fin las sesiones multimedia multicast.

1.3.1.2.1 Fases de una llamada IAX

Una llamada IAX consta de tres fases:

1) *Establecimiento de la llamada*

El terminal o estación que desea iniciar la llamada (terminal A) envía un mensaje *NEW* al otro terminal (terminal B), el cual le responde con un mensaje *ACCEPT*. Posteriormente el terminal destino empezará a timbrar en espera de que el usuario conteste. Una vez que el usuario conteste, el terminal B enviará un mensaje *ANSWER* al terminal A para notificar que el usuario contestó.

2) *Flujo de Audio*

Se inicia el intercambio de audio mediante tramas. Se envían tramas M y F en ambos sentidos con información vocal. Las tramas M son mini tramas que transportan el audio en paquetes de datos; estas tramas contienen solo una cabecera de 4 bytes para reducir el uso del ancho de banda. Las tramas F son tramas completas que incluyen información de sincronización.

3) *Liberación de la llamada*

Para finalizar la llamada (desconexión), cualquiera de los dos terminales que participan en la llamada, debe enviar un mensaje de *HANGUP*.

En la Figura 1.5 se muestra un ejemplo de una llamada IAX.

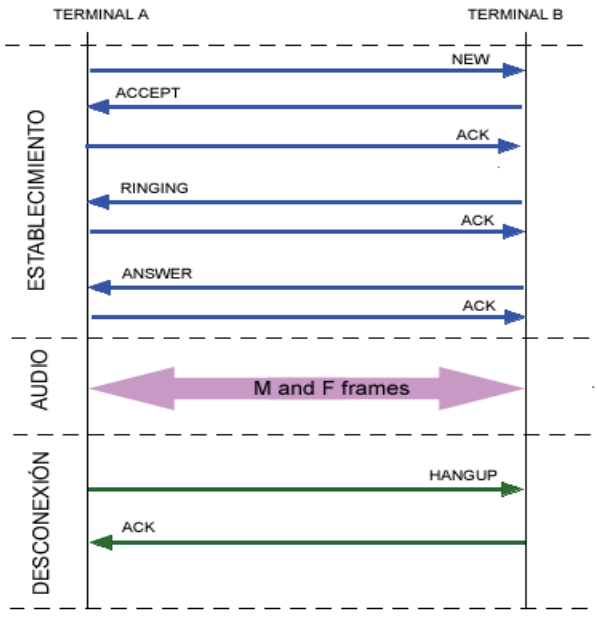


Figura 1.5 Fases de una llamada IAX **[11]**

1.3.2 PROTOCOLOS DE TRANSPORTE

Los protocolos de transporte tienen como objetivo principal, trasladar la información útil del origen al destino, cumpliendo los requerimientos que demandan las aplicaciones multimedia, tales como mayores requerimientos de ancho de banda y transmisión de tráfico en tiempo real.

Los protocolos de transporte, más empleados en una red integrada de voz y de datos, son RTP y su protocolo de control RTCP y RTSP.

1.3.2.1 RTP [12]

RTP (*Real Time Transport Protocol*) es un protocolo de trasporte, que tiene como objeto proporcionar servicios de difusión de audio y video, videoconferencia y simulaciones en tiempo real de extremo a extremo en redes de paquetes. Este protocolo no dispone de mecanismos para asegurar la calidad de servicio.

El proceso de transporte implica dividir en paquetes el flujo de bits que proporciona el codificador de señal, enviar dichos paquetes por la red y re ensamblar el flujo de bits original en el destino.

El protocolo RTP trabaja en conjunto con el protocolo auxiliar RTCP, para obtener información sobre la calidad de la transmisión y participantes de la sesión.

RTP generalmente se ejecuta sobre UDP, para hacer uso de sus funciones de multiplexación y detección de errores.

1.3.2.1.1 Características

Las principales características de este protocolo son:

- No realiza ninguna reserva de recursos, con el único objetivo de evitar la pérdida de paquetes y jitter.
- Contiene un identificador del tipo de carga en cada paquete, el cual describe el tipo de codificación que se ha empleado en su generación.
- RTP se implementa sobre UDP.

1.3.2.2 RTCP [12]

El protocolo RTCP (*Real Time Control Protocol*) es una parte del RTP y trabaja en conjunto con el mismo.

Cada flujo RTP cuenta con información adicional proporcionada por RTCP, utilizada para realizar un seguimiento de la calidad de la transmisión.

La información adicional que proporciona RTCP a RTP se lista a continuación:

- Incluye detalles sobre participantes, estadísticas de rendimiento y pérdidas, que permiten realizar cierto control de flujo y congestión.
- Permite ver si la congestión es local o generalizada.
- Puede utilizarse para hacer codificación adaptativa.

1.3.2.2.1 RTSP

El protocolo RTSP (*Real Time Streaming Protocol*) especificado en el RFC 2326, tiene como objeto establecer y controlar *stream*s multimedia de audio y video, actuando como control remoto de servidores multimedia.

1.3.3 CODECS DE AUDIO [4] [13]

La señal de audio que se emite por parte de un usuario es analógica, razón por la cual necesita ser digitalizada para ser transmitida a través de la red de datos en un sistema de VoIP. Para este propósito se utilizan algoritmos matemáticos, implementados en *software* más conocidos como codecs, los cuales permiten codificar y decodificar la señal de audio.

El proceso de conversión que realiza un códec es complejo, pero la mayoría se basan en la codificación modulada mediante pulsos (PCM) o variaciones.

Existen diferentes modelos de codecs de audio utilizados en VoIP, y dependiendo del algoritmo escogido en la transmisión, variará la calidad de la voz, el ancho de banda requerido para la comunicación y la carga computacional.

Para seleccionar un codec se debe tomar en cuenta los siguientes criterios:

- ***Complejidad:*** Cantidad de CPU necesaria para procesar el algoritmo de compresión.

- ***Compresión de voz:*** Permite a la señal de voz utilizar un ancho de banda menor para transmitirse por la red de paquetes.
- ***Calidad de la voz:*** Depende del usuario; sin embargo para calificar este parámetro se utiliza la técnica de las escalas MOS.

1.3.3.1 G.711

Estandarizado por la ITU[12] en 1972 y es el principal algoritmo utilizado por la PSTN por ser el más simple y de menor carga computacional. Muestrea a una frecuencia de 8 KHz, utilizando modulación de pulsos codificados (PCM) para comprimir, descomprimir y decodificar. Existen dos tipos de este códec:

- ***Ley A:*** Codifica cada 13 muestras en palabras de 8 bits. Esta ley es utilizada en Europa. Requiere un ancho de banda de 64 Kbps
- ***Ley u:*** Codifica cada 14 muestras en palabras de 8 bits. Utilizada en E.E.U.U. y Japón. Requiere un ancho de banda de 64 Kbps.

1.3.3.2 G.726

Estandarizado por la ITU en 1991 y sustituyó al estándar G.721, este estándar es conocido como Diferencial de Adaptación de la Modulación por pulsos Codificados (ADPCM). Puede trabajar a velocidades de 16, 24 y 32 Kbps. La ventaja más importante de este códec, es la disminución del ancho de banda requerido sin aumentar en mayor grado la carga computacional.

1.3.3.3 G.728

Codifica con un ancho de banda de 3.4 KHz con transmisiones a 16 Kbps. Comúnmente utilizado en sistemas de video, conferencias a 56 ó 64 Kbps.

1.3.3.4 G.729

Codifica señales de audio con ancho de banda de 3.4 KHz y transmisiones a 8 Kbps.

---

[12] Unión Internacional de las telecomunicaciones (ITU): organismo regulador de las telecomunicaciones a nivel mundial.

1.3.3.5 G.729 A

Desarrollado por diferentes empresas privadas. Necesita menor carga computacional que G.729 y transmisiones a 32 Kbps. Genera una buena calidad de voz comparable con ADPCM. No puede transportar tonos como DTMF o fax pero es el que menor tasa de bits proporciona (8 Kbps).

1.3.3.6 G.723.1

Estandarizado por la ITU en 1995. Trabaja a 6.3 Kbps (basada en la técnica de compresión *MultiPulse MultiLevel Quantization*) o 5.3 Kbps con un ancho de banda de 3.4 KHz. Este codec debe ser licenciado para poder ser usado.

1.3.3.7 GSM (GLOBAL SYSTEM MOBILE)

Es libre, no requiere el pago de una licencia para utilizarlo. Trabaja a 13 Kbps con una carga de CPU aceptable.

1.3.3.8 MP3

Códec estandarizado por la ISO y optimizado para música, es utilizado por los teléfonos IP principalmente para ofrecer servicios de música en espera.

En la tabla 1.2 se presenta un resumen de los codecs más importantes:

Codec	**Estandarizado**	**Codificación**	**Bit rate (Kbps)**	**Sampling Rate (KHz)**	**Tamaño de muestra (ms)**
G.711	ITU-T	PCM	64	8	Muestreada
G.726	ITU-T	ADPCM	16/24/32/40	8	Muestreada
G.728	ITU-T	CELP	16	8	2.5
G.729	ITU-T	CELP	8	8	10
G.723.1	ITU-T	CELP Y *MultiPulse MultiLevel Quantization*	5.6/6.3	8	30
GSM	ETSI	RPE-LPT	13	8	22.5

Tabla 1.1 Principales Códecs para VoIP **[**14**]**

1.4 PROBLEMAS A RESOLVER EN VOIP [2] [4] [13] [15]

Los principales problemas que debe enfrentar la VoIP son:

1.4.1 RETARDO O LATENCIA

El retardo se define como el tiempo que tarda la voz en salir del usuario que habla y llegar al usuario que escucha. Es el efecto más notorio y perjudicial para la transmisión de voz en una red. De acuerdo al documento G.114 de la ITU-T el retardo máximo permisible en una comunicación extremo a extremo, en una llamada de voz debe ser menor a 150 ms.

A continuación se mencionan algunas fuentes del retardo en VoIP.

1.4.1.1 Retardo de Propagación

Es el tiempo necesario para que las señales electromagnéticas u ópticas viajen de un punto a otro.

1.4.1.2 Retardo de Serialización

Es la cantidad de tiempo que se tarda en sacar un bit o un byte a la línea de transmisión y depende de la velocidad del medio de transmisión.

1.4.1.3 Retardo de manejo

Es el tiempo que tarda el procesamiento de los paquetes de voz en dispositivos intermedios en una red.

Son muchos los factores que influyen para causar retardo (codec, serialización, retardo en las colas, retardo de propagación) por lo tanto el retardo total equivale a la suma de todas las fuentes.

En la figura 1.6 se puede apreciar los tipos de retardos presentes en una red de comunicaciones.

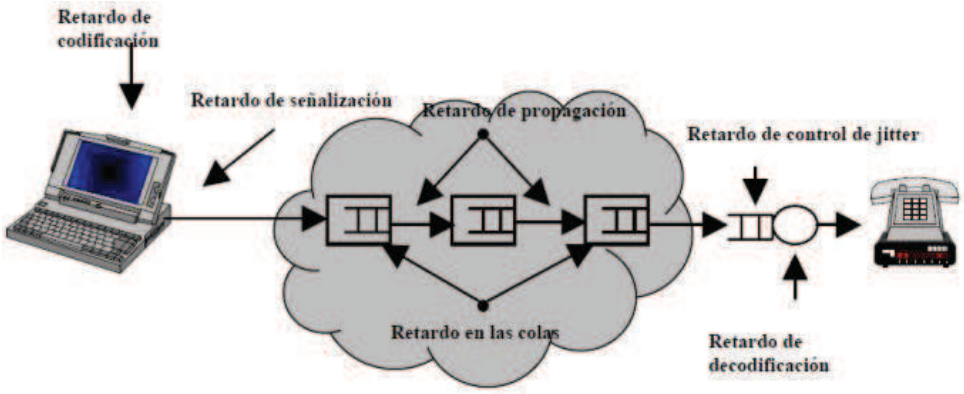


Figura 1.6 Retardos en VoIP **[4]**

1.4.2 FLUCTUACIÓN DE FASE (JITTER)

Es la variación del tiempo de llegada de un paquete. La diferencia entre cuando se espera recibir el paquete y cuando se recibe.

Para reducir este problema se debe almacenar y retener los paquetes el tiempo suficiente para ordenarlos y reproducirlos en su secuencia correcta.

Los umbrales de jitter aceptados en telefonía IP para una buena calidad de voz son menores a 20 ms y para una calidad de voz aceptable menores a 50 ms.

1.5 TELEFONÍA IP [4]

Telefonía IP se refiere a un sistema totalmente organizado y controlado de comunicaciones telefónicas que usan VoIP. La telefonía IP incorpora algunas consideraciones: latencia y retraso, jitter, codificación y compresión de voz, eco, entre otros.

La telefonía IP no utiliza circuitos físicos para la conversación, sino que envía múltiples conversaciones a través del mismo canal (circuito virtual) codificadas en paquetes y en flujos independientes. La telefonía IP difiere de la tradicional porque no usa conmutación de circuitos, sino conmutación de paquetes. La voz es enviada en paquetes a través de redes IP, pero si es necesaria la comunicación con teléfonos analógicos existen algunas soluciones (tarjetas para telefonía analógica, ATAs [13]).

[13] ATA: Dispositivo para conectar teléfonos analógicos a un sistema de telefonía digital.

1.5.1 SISTEMA IP PURO COMO SOLUCIÓN DE TELEFONÌA

En este sistema las centrales PBX son IP puras, la voz se digitaliza, se comprime y se encapsula sobre el protocolo IP.

Para esta solución se distinguen opciones basadas en *software* y en *hardware*.

En la figura 1.7 se incluye un ejemplo de sistema de telefonía IP puro:

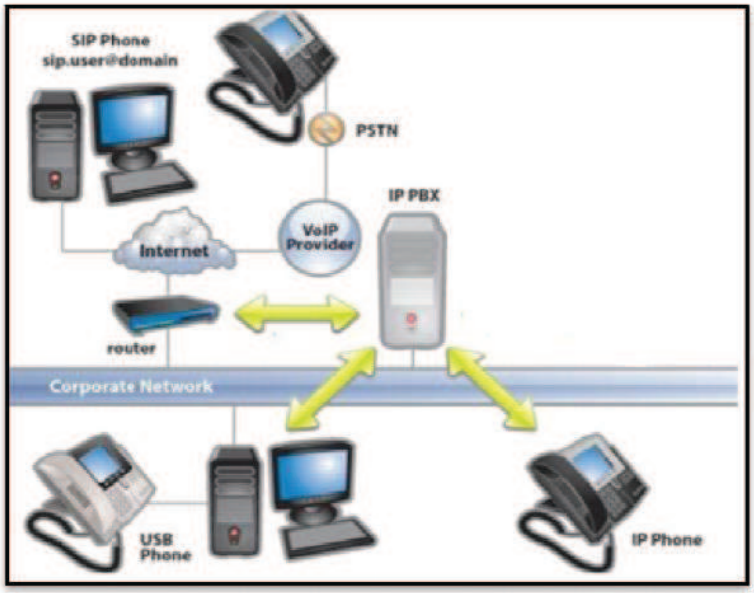


Figura 1.7 Esquema de telefonía IP puro **[**16**]**

1.6 ASTERISK [10] [17] [18] [19] [20]

Asterisk es un *software* PBX[14] bajo el concepto de *software* libre; está además definido como un servidor de telefonía altamente integrado que ofrece servicios de telefonía como registro de llamadas, buzón de voz, grabación de llamadas, desvío y transferencia de llamadas, salas de conferencias, gestión de colas (*call center*), IVR (*Interactive Voice Response*), integración con sistemas de síntesis de voz y reconocimiento del habla. [**10**]

[14] PBX (Private Branch Exchange): Es una central telefónica privada encargada de gestionar las llamadas internas, salientes y entrantes.

Asterisk es una plataforma que tiene la gran ventaja de ser compatible con varios protocolos de VoIP como SIP, H.323, IAX2, MGCP (*Media Gateway Control Protocol*), SCCP (*Skinny Client Control Protocol*).

El *software* permite además conectividad a redes tradicionales de telefonía mediante tarjetas de telefonía tradicional (analógica); así como conectividad a redes de telefonía digital mediante técnicas E1 y T1.

Asterisk hoy en día es uno de los más populares *software* libres basados en VoIP, que opera sobre múltiples sistemas operativos. [**21**]

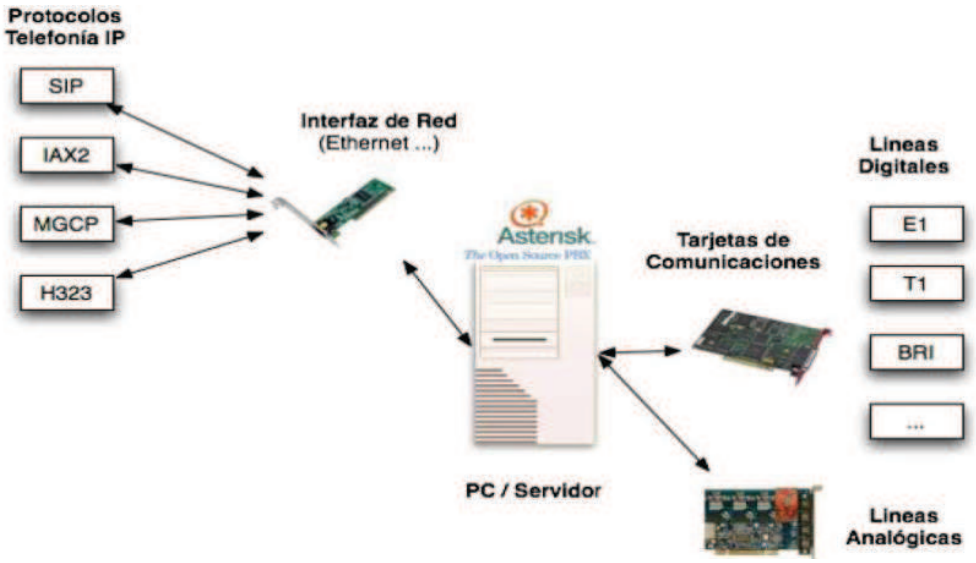


Figura 1.8 Interconectividad de Asterisk [**21**]

1.6.1 ARQUITECTURA DE ASTERISK

La Arquitectura básica de Asterisk es la que se muestra en la Figura 1.9, está constituida por APIs (*Aplication Programming Interface*), las cuales están definidas alrededor de un núcleo central avanzado. Las conexiones internas de la PBX son manejadas por el núcleo.

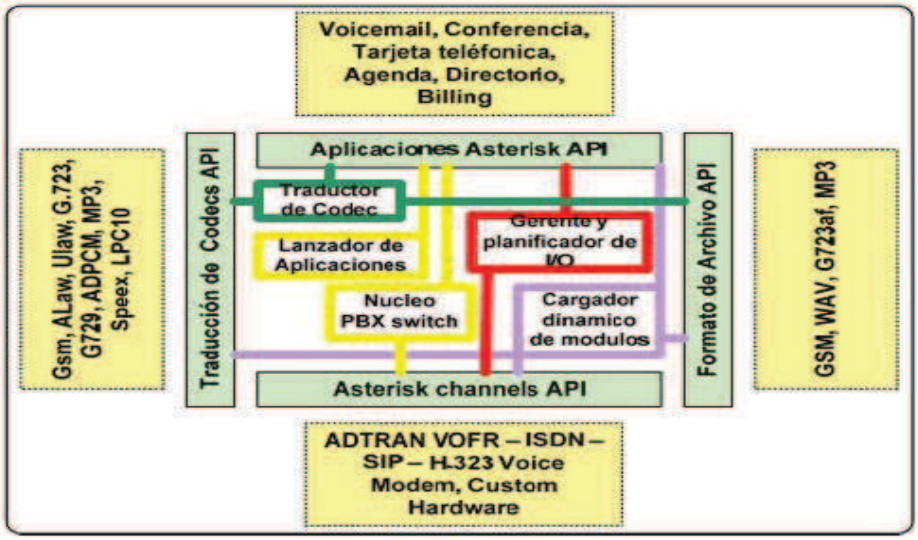


Figura 1.9 Arquitectura de Asterisk [**18**]

La capa de núcleo es transparente y abstracta para protocolos, codecs e interfaces específicas de *hardware* para aplicaciones telefónicas, lo cual permite a Asterisk conectar *hardware* y aplicaciones mediante las siguientes APIs:

API de canal, controla todas las llamadas del sistema, ya sea analógica o digital.

API de aplicaciones, establece la interfaz entre aplicaciones de usuario y funcionalidades de Asterisk, permite crear aplicaciones que se comunican con recursos externos.

API de traducción de codec, encargada de controlar la traducción del codificador-decodificador entre los participantes en la comunicación.

API de formato de archivos, permite la reproducción y grabación de archivos de audio, con distintos formatos principalmente los ***.wav*** y los ***.mp3.***

1.6.2 CONCEPTOS DE ASTERISK [17]

1.6.2.1 Canal

Es una conexión que direcciona la llamada, ya sea entrante o saliente en el Sistema Asterisk. Asterisk soporta una serie de canales, los más importantes son:

- SIP, IAX2, H.323, MGCP como protocolos de VoIP.
- Zap/Dahdi: líneas analógicas y digitales.

1.6.2.2 *DialPlan* (Plan de Marcación)

Es la configuración en la cual se indica las acciones a realizar, para las llamadas entrantes y salientes; es decir define el comportamiento lógico de la Central.

1.6.2.3 Extensión

Es una lista de comandos a ejecutar, se accede a las extensiones cuando se recibe una llamada por un canal dado o el usuario llamante marca la extensión.

Se accede a las extensiones cuando:

- Se recibe una llamada por un canal dado
- El usuario llamante marca la extensión

Cada extensión se compone de: Nombre, prioridad y aplicación.

- ***Nombre:*** Agrupa una lista de acciones o pasos
- ***Prioridad:*** Define el orden
- ***Aplicación:*** Define la ejecución

1.6.2.4 Contexto

Es una colección de extensiones. El dial plan de Asterisk se divide en uno o varios contextos.

1.6.2.5 Aplicaciones

Son comandos asociados que se ejecutan secuencialmente, controlan el comportamiento de la llamada y del sistema en sí. Su sintaxis es la siguiente:

```
NOMBRE_APLICACIÓN(argumentos)
```

Ejemplo: ***Dial*** *(*`SIP/telefono-${EXTEN},20`*;* `La aplicación Dial realiza una llamada en base a los argumentos SIP(Canal Sip), Exten(Número destino) y 20(Tiempo de intento)`

1.6.3 ARQUITECTURA DEL *DIALPLAN*

Cuando una llamada cursa el sistema, está pertenece a un *dialplan* el cual a su vez pertenece a un determinado contexto; y dentro de este contexto la llamada pertenece a una extensión en particular.

Esta asociación del *dialplan* con el contexto y con la extensión se puede apreciar en la figura 1.10:

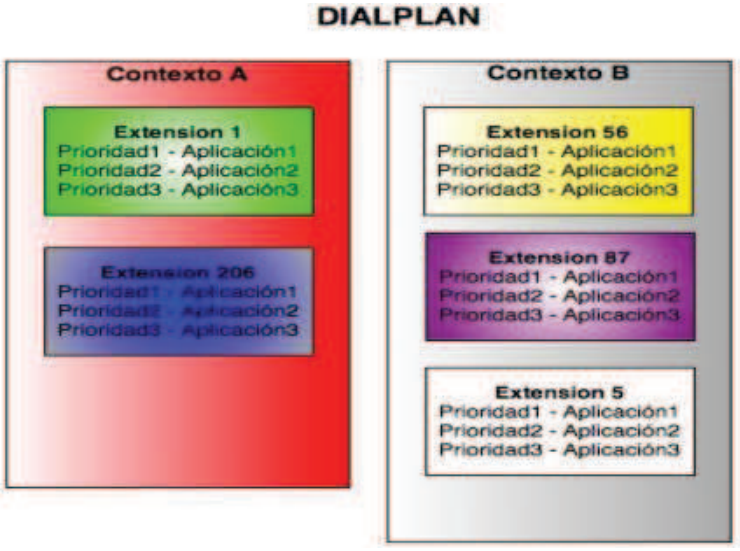


Figura 1.10 Componentes de un *Dialplan* [**20**]

El archivo de configuración ***extensions.conf*** es el más importante para la puesta en marcha de una central Asterisk. En él se define todo lo relacionado con el plan de llamadas; cualquier número marcado desde una extensión será procesado dentro de este archivo.

Este archivo está dividido en tres bloques:

- La parte ***general*** donde se configuran algunos parámetros generales
- La parte ***globals*** donde se definen las variables globales que se van a utilizar en la central.
- Una última parte donde queda toda la configuración de las llamadas.

1.6.3.1 Patrones de Numeración

El *dialplan* no está limitado sólo a números fijos, se pueden utilizar patrones de números para controlar el flujo de las llamadas.

Para esto el identificador de extensión debe iniciar con el símbolo "_"

Los caracteres que pueden usarse son los mostrados en la tabla 1.3:

Carácter	Asignación
X	Acepta un número del 0 al 9
Z	Acepta un número del 1 al 9
N	Acepta un número del 2 al 9
[1237-9]	Acepta cualquier número dentro de los corchetes
[a-z]	Acepta cualquier letra minúscula
[A-Z]	Acepta cualquier letra mayúscula
.	Acepta uno o más caracteres
!	Acepta cero o más caracteres

Tabla 1.2 Caracteres usados para establecer un patrón de numeración [**10**]

1.6.3.2 Extensiones Estándares

Son extensiones que existen por defecto en la PBX y se tiene las siguientes:

- **i:** extensión inválida; usada cuando al pedir el ingreso de una extensión, esta no existe dentro del contexto.
- **t:** extensión tiempo fuera; usada cuando al pedir el ingreso de una extensión se supera el tiempo de espera al no haber ingresado ninguna.
- **s:** extensión de inicio (*start*); usada para entrar en un contexto con cualquier extensión.

1.6.3.3 Variables

Asterisk puede usar variables globales, compartidas o variables de canal, como argumentos para los comandos dentro del *dialplan*. Para referenciarlas dentro del *dialplan* se usa la siguiente sintaxis:`${NOMBRE:offset:longitud}`

Si se usan los indicadores *offset* y *longitud,* la variable resultante es una cadena de la variable original, que comienza en la posición indicada por el *offset* y contiene un número de caracteres indicado por la *longitud,* contados hacia la derecha. El primer carácter se cuenta en la posición cero.

- Si el valor *offset* es negativo, la posición se toma desde la izquierda.

- Si el valor *longitud* es omitido o es negativo, se toma todos los caracteres indicados después de la posición *offset.*

1.7 HARDWARE Y SOFTWARE PARA TELEFONÍA IP

1.7.1 INTERFACES PARA DISPOSITIVOS DE VOZ [18] [19]

Son dispositivos que permiten conectarse con la red de telefonía tradicional típica para lo cual realizan la conversión de la señal analógica a digital y viceversa.

1.7.1.1 FXO (*Foreign eXchange Office*)

También denominado gateway, esta tarjeta recibe señalización y se debe conectar a la línea telefónica. Generalmente es de color rojo.

1.7.1.2 FXS (*Foreign eXchange Station*)

Esta tarjeta emite señalización, permite conectar teléfonos analógicos a un computador. Generalmente es de color verde.

1.7.2 TELÉFONOS IP

Teléfono digital que utiliza una conexión de red de datos en lugar de una conexión de red telefónica, debe soportar el protocolo IP y al menos un protocolo de VoIP.

1.7.3 SOFTPHONES

Software que simula el funcionamiento de un teléfono convencional; permitiendo usar la computadora para realizar llamadas a otros softphones y a otros teléfonos ya sean analógicos (adaptados a la red mediante ATAs) o teléfonos IP.

CAPÍTULO 2

CARACTERÍSTICAS GENERALES DE LOS PROTOCOLOS IPV4 E IPV6

2.1 INTRODUCCIÓN [22] [23]

IP es un protocolo de capa 3 en el modelo OSI y de capa Red en la arquitectura TCP/IP, que permite la interconexión de redes, para el intercambio de paquetes de datos individuales entre dispositivos finales identificados por una dirección de origen y destino; proporcionando de esta manera un mecanismo de transporte sin importar que el destino se encuentre o no en la misma red.

La PDU (Unidad de datos de protocolo) que maneja el protocolo IP se denomina paquete y se utiliza para encapsular los segmentos TCP[15] o datagramas UDP[16], para su entrega al *host* destino.

Actualmente existen dos versiones del protocolo IP que están siendo utilizadas; la versión 4 (IPv4) y la versión 6 (IPv6). Las versiones previas de IP (de la 1 a la 3) se definieron sucesivamente y fueron sustituidas hasta alcanzar IPv4; razón por la cual llegamos hasta la etiqueta de versión 6 en el desarrollo del IP.

Esta última versión surge como respuesta al crecimiento del Internet, ya que brinda la posibilidad de expandir las redes para las exigencias futuras, en base a un mayor suministro de direcciones y funcionalidades.

2.2 LIMITACIONES DE IPV4 [21] [22] [23]

El protocolo IPv4 definido en la RFC[17] 791 demostró ser muy robusto, de fácil implementación e interoperable, sin embargo no se anticiparon algunos problemas, los mismos que se describen a continuación.

---

[15] Protocolo de control de transmisión (TCP): protocolo de capa transporte orientado a conexión, entrega confiable y control de flujo.
[16] Protocolo de datagramas de usuario (UDP): protocolo de capa transporte no orientado a conexión y entrega no confiable.
[17] *Request for coment*: Solicitud de comentarios del protocolo Internet.

2.2.1 AGOTAMIENTO DE LAS DIRECCIONES IP

El protocolo IPv4 permite generar 4.294.967.296 direcciones, sin embargo el crecimiento exponencial del Internet ha hecho que cada vez las direcciones sean más escasas.

2.2.2 NECESIDAD DE UNA SIMPLE CONFIGURACIÓN

La mayoría de implementaciones IPv4 se deben configurar manualmente, además posee una jerarquía más compleja; por lo que se busca en IPv6 una configuración más sencilla que simplifique el proceso.

2.2.3 REQUERIMIENTOS DE SEGURIDAD A NIVEL DE IP [21]

Es necesario hacer seguro el tráfico entre usuarios finales para proteger su información, así como para limitar el acceso no autorizado a recursos propios de una red. Por lo tanto se requiere de un mecanismo de autenticación y cifrado, que se consiguió mediante la implementación del protocolo IPSec.

2.2.4 NECESIDAD DE UN MEJOR SOPORTE EN LA ENTREGA EN TIEMPO REAL DE LOS DATOS [24]

IPv4 provee mecanismos para dar prioridad a determinado tipo de tráfico a través del campo ToS[18] de su cabecera, sin embargo constituye un modelo fijo y limitado para la diferenciación del tráfico. Con lo cual no se asegura la clasificación del tráfico y no se asigna la calidad requerida para el envío de tráfico en tiempo real.

2.3 DESCRIPCIÓN DE IPV6 [23] [25] [26]

IPv6 es considerado un protocolo más robusto y eficiente, ya que combina el direccionamiento ampliado con una cabecera más eficiente y rica en características, para satisfacer demandas tales como escalabilidad, seguridad, configuración y administración de redes, soporte para GoS[19] , movilidad, políticas de enrutamiento, mecanismos de transición y coexistencia con IPv4.

---

[18] *Type of Service*: campo que permite aplicar calidad de servicio en paquetes de alta prioridad.
[19] *Grade of Service*: indicador de calidad de servicio en base a parámetros percibidos por el usuario

2.3.1 CARACTERÍSTICAS DE IPV6 [21] [26] [27] [28] [29]

2.3.1.1 Mayor Capacidad de direccionamiento

IPv6 incrementa el tamaño de la dirección IP de 32 a 128 bits, permitiendo utilizar direcciones de origen y destino de 16 bytes, que se componen por 8 segmentos de 2 bytes cada uno, lo que permite tener 3.4 x 10^{38} posibles direcciones.

2.3.1.2 Nuevo Formato de la cabecera

Para reducir el procesamiento de los paquetes en los *router*s, algunos campos del encabezado IPv4 se eliminaron, IPv6 simplifica el encabezado del paquete de 12 campos que se utiliza en IPv4, a solo 8 campos ofreciendo así varias ventajas.

2.3.1.3 Soporte para encabezados de extensión

Se incorpora opciones adicionales a las del encabezado base, mediante encabezados de extensión; permitiendo de esta manera mayor flexibilidad en cuanto al número y funcionalidad de las opciones.

2.3.1.4 Autoconfiguración

Los *hosts* IPv6 utilizan los mensajes de anuncio recibidos desde un enrutador para configurar de forma automática direcciones locales de capa red.

2.3.1.5 Seguridad

El protocolo IPv6 incorpora seguridad mediante el protocolo IPSec, el cual proporciona dos facilidades principales a través de sus cabeceras específicas. La cabecera de autenticación abreviada como AH, que proporciona una función de solo autenticación; y una función combinada de autenticación y cifrado, llamada encapsulado de seguridad de la carga útil conocida como ESP. IPsec actualmente es un mecanismo opcional para IPv4, mientras que está incluido en IPv6.

2.3.1.6 Calidad de Servicio

IPv6 identifica paquetes por clases de servicios o prioridad, gracias al campo Clase de tráfico en su encabezado; además permite la identificación del tráfico, mediante un campo *Flow Label* en su encabezado.

2.3.1.7 Interacción con nodos vecinos

Consiste en un conjunto de mensajes de control y procesos, que determinan las relaciones entre nodos vecinos y permiten a los *host* descubrir direcciones, prefijos de direcciones y más parámetros de configuración.

2.3.2 ESTRUCTURA DEL ENCABEZADO DE IPV6 [21] [25] [26]

Un paquete IPv6, también llamado datagrama IPv6, consta de un encabezado IPv6 y una carga IPv6, como se muestra en la figura 2.1; se adiciona un encabezado a cada uno de los paquetes para permitir el manejo de los datos de extremo a extremo.

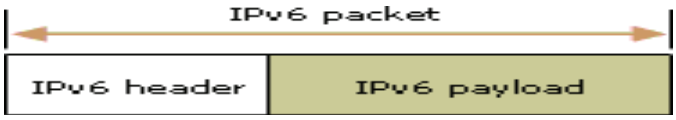


Figura 2.1 Estructura de un paquete IPv6 [**27**]

Se realizaron algunos cambios en el formato del encabezado base de IPv6 con respecto a IPv4, algunos campos IPv4 se han eliminado, cambiado de posición, modificado e incluso se han añadido nuevos. La cabecera IPv6 es más simple (solo 8 campos y un tamaño fijo de 40 bytes), más flexible y más eficiente; se prevé su extensión por medio de encabezados adicionales que facilitan el procesamiento para su enrutamiento. En la figura 2.2 se presenta el tamaño y formato de dicho encabezado.

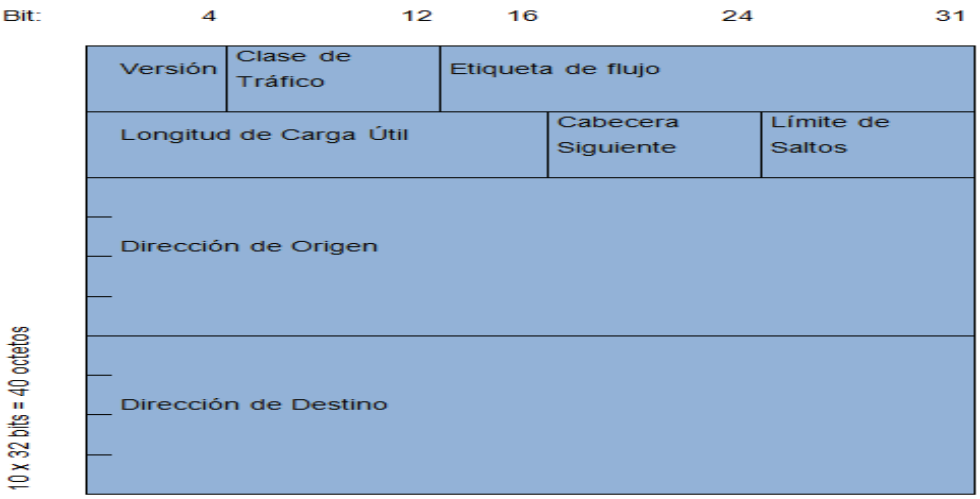


Figura 2.2 Cabecera IPv6 [**21**]

2.3.2.1 Versión (4 bits)

Identifica la versión del protocolo IP utilizado. En el caso de IPv6 el valor para este campo es igual a 6.

2.3.2.2 Clase de tráfico (8 bits)

Permite identificar y distinguir los paquetes de acuerdo a la clase de servicio o prioridad.

2.3.2.3 Etiqueta de flujo (20 bits)

Identifica y diferencia paquetes del mismo flujo en la capa de Red, de modo que proporcionen un control especial de los paquetes que pertenecen a un flujo dado.

2.3.2.4 Longitud de la carga útil (16 bits)

Indica el tamaño en bytes de los datos enviados junto al encabezado IPv6. Es decir representa la longitud total de la PDU de capa transporte, más todas las cabeceras de extensión.

2.3.2.5 Cabecera siguiente (8 bits)

Identifica al encabezado de extensión, que sigue inmediatamente al encabezado básico de IPv6.

2.3.2.6 Límite de saltos (8 bits)

Indica el número máximo de saltos o *router*s que el paquete IPv6 puede pasar, antes de ser descartado; se decrementa en una unidad en cada salto, el paquete se descarta si el límite de saltos llega a cero.

2.3.2.7 Dirección Origen (128 bits)

Contiene la dirección IPv6 del *host* origen.

2.3.2.8 Dirección Destino (128 bits)

Contiene la dirección IPv6 de destino.

2.3.3 ENCABEZADOS DE EXTENSIÓN [21] [25] [30] [31]

Para reemplazar algunas de las funcionalidades de los campos faltantes en comparación al encabezado IPv4, se introdujo el concepto de encabezado de extensión; que proporcionan información adicional que se incluyen en un encabezado. Dichos encabezados no tienen una cantidad, ni un tamaño fijo y están relacionados a una opción en particular.

Las cabeceras de extensión cuentan típicamente con campos que identifican al siguiente encabezado, el tamaño del encabezado y las opciones que proporciona dicho encabezado; cuando en un mismo paquete existen varios encabezados de extensión, estos se colocan en serie formando una cadena de encabezados. A continuación se presenta algunos ejemplos del uso de encabezados de extensión.

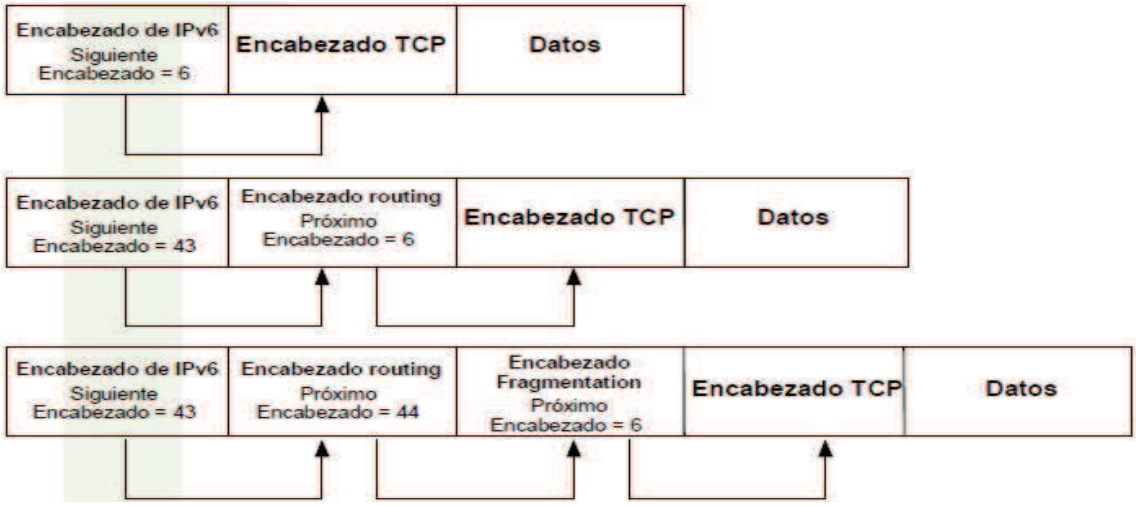


Figura 2.3 Cadena de encabezados de Extensión [**21**]

2.3.4 DIRECCIONAMIENTO [21] [25] [32]

Para una mejor descripción de las direcciones IPv6, se adoptó una notación en la cual los 128 bits se han dividido en ocho bloques de 16 bits, separados mediante el símbolo "**:**", en donde cada bloque se representa por 4 dígitos hexadecimales.

Las direcciones IPv6 no se asignan a los nodos, sino a interfaces individuales en los nodos; de esta manera una única interfaz puede presentar múltiples direcciones únicas y cualquiera de las direcciones asociadas a las interfaces de un nodo, pueden ser utilizadas para identificar de manera única a dicho nodo.

2.3.4.1 Arquitectura de Direccionamiento [33]

En la RFC 3513 se define que los 128 bits de una dirección se dividen en dos partes bien diferenciadas, tal como se muestra en la figura 2.4:

Figura 2.4 Formato de una dirección IPv6

2.3.4.1.1 Prefijo de subred

Identifica la subred (o "*link*" en terminología IPv6) a la que está conectado el sistema y que es común a todos los sistemas en ella.

Los prefijos en IPv6 se pueden considerar equivalentes al significado de la máscara de subred utilizada en IPv4, indicando bits que se mantienen fijos o corresponden a la parte de red.

2.3.4.1.2 Identificador de interfaz

Identifica a una interfaz dentro de la subred y debe ser distinto para cada sistema conectado a la subred; generalmente los últimos 64 bits de una dirección IPv6 comprenden el identificador de interfaz. Se puede determinar un identificador de interfaz en base a las alternativas siguientes:

- Utilizando el formato EUI-64[20], en el caso de direcciones *unicast*.
- Generando aleatoriamente un identificador que cambie al cabo de cierto intervalo de tiempo.
- Asignar un identificador durante la configuración automática de direcciones IPv6 con estado, este es el caso del DHCPv6.

2.3.4.2 Tipos de direcciones ipv6 [34] [35]

De acuerdo al tipo de direccionamiento que se maneje, los 128 bits de una dirección IPv6 pueden clasificarse en tres tipos de direcciones:

- *Unicast*, identificación individual
- *Anycast*, identificación selectiva
- *Multicast*, identificación en grupo

[20]Identificador Único Extendido: formato para obtener direcciones IPv6 en base a direcciones MAC

2.3.4.3 Direccionamiento *unicast*

Identifica a una única interfaz, de modo que los paquetes enviados a una dirección *Unicast*, son entregados en una única interfaz. Se utilizan para comunicaciones entre dos nodos, en un esquema a manera punto a punto.

2.3.4.3.1 Direcciones unicast globales

Una dirección de unidifusión global, es una dirección IPv6 a partir del prefijo *unicast* público mundial (2001 :: / 16). Dichas direcciones son equivalentes a las direcciones públicas en IPv4, además son globalmente ruteables y accesibles en la Internet. Actualmente para la asignación de direcciones, se tiene reservado el rango 2000::3 que comprende direcciones desde 2000:: a 3fff:ffff:ffff:ffff:ffff:ffff:ffff.

2.3.4.3.2 Direcciones link-local

Este tipo de dirección puede utilizarse solamente a nivel local, es decir solo en el enlace (subred) especifico, en el cual la interfaz se encuentra conectada.

Se crean dinámicamente utilizando un prefijo de enlace local de FE80 y un identificador de interfaz de 64 bits típicamente basado en el formato IEEE EUI-64.

2.3.4.3.3 Direcciones site-local

Diseñadas para ser utilizadas en el direccionamiento dentro de un sitio, sin la necesidad de un prefijo global, son similares a las direcciones privadas en el entorno IPv4, estas direcciones no son accesibles desde otros sitios y se pueden utilizar para *hosts* que no tienen una conexión directa al Internet, a través de IPv6.

2.3.4.3.4 Direcciones especiales [23] [25]

Al igual que en el IPv4 existen direcciones especiales asignadas a propósitos específicos, dentro de esta clasificación cabe mencionar la dirección no especificada y la dirección de *loopback*.

- ***Dirección No especificada***

Indica la ausencia de una dirección, no debe ser asignada a un nodo o interfaz, ya que solo representa que una interfaz está iniciándose y no le ha sido asignada una dirección; nunca será utilizada como una dirección de destino. Se representa como la dirección 0:0:0:0:0:0:0:0 o de forma abreviada como ::0.

- ***Dirección de Loopback***

Se utiliza para referenciar a la propia máquina, no debe ser asignada a una interfaz física pues representa una interfaz virtual que permite realizar pruebas y comunicaciones dentro de un mismo nodo. Se representa con la dirección 0:0:0:0:0:0:0:1 o en forma abreviada como ::1.

2.3.4.4 Direcciones Anycast

Es una dirección que es asignada a más de una interface con la propiedad que un paquete enviado a una dirección *anycast* es enrutado a la interface más cercana.

2.3.4.5 Direcciones *Multicast*

La multidifusión es la base de muchas funciones de IPv6 y es un sustituto para la dirección de difusión. Permite identificar grupos de interfaces, ya que cada interfaz puede pertenecer a cualquier número de grupos de multidifusión. Una dirección de *multicast* está identificada mediante el prefijo FF; adicionalmente destaca un nuevo grupo de 4 bits conocidos como el ámbito (alcance hasta el cual se puede utilizar la dirección), las direcciones continúan siendo asignadas por el proveedor, pero pueden pertenecer a diferente ámbito. Los diferentes tipos de ámbitos se presentan en la figura 2.5.

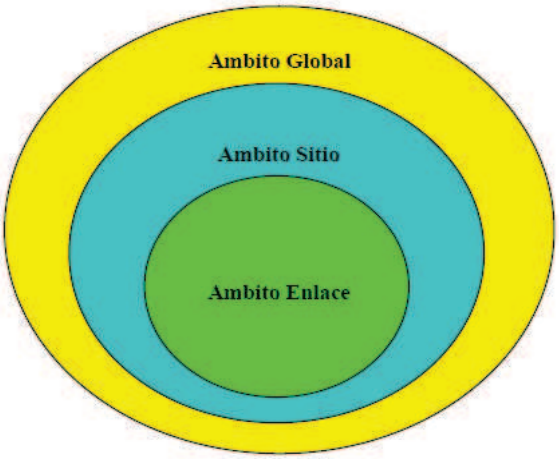


Figura 2.5 Ámbitos de una dirección IPv6 [**36**]

2.4 FUNCIONALIDADES DE IPV6

2.4.1 ICMPV6 [25] [36]

El Protocolo de Mensajes de Control de Internet, es un protocolo utilizado por IPv6 para reportar errores encontrados en el procesamiento de paquetes y mejorar otras funciones entre capas, como el diagnóstico y el descubrimiento de vecinos. Su valor en el campo siguiente cabecera es de 58; es parte integral de Ipv6 y debe ser incorporado en cualquier implementación de servicios con IPv6.

Un mensaje ICMPv6 es precedido por la cabecera IPv6 y las correspondientes cabeceras de extensión en el caso de que estas fueran necesarias de acuerdo al esquema de la figura 2.6.

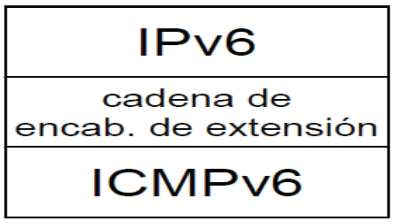


Figura 2.6 Paquete ICMPv6 [**25**]

Los mensajes ICMPv6 se dividen en dos clases: mensajes de error y mensajes informativos.

Los mensajes de error permiten detectar los siguientes inconvenientes en la red tales como destino inalcanzable, paquete demasiado grande, tiempo agotado, problema de parámetro o error de cabeceras.

Los mensajes informativos permiten realizar un diagnóstico y búsqueda de problemas; se detallan a continuación los diferentes mensajes informativos:

- *Echo request*, se envía a un dispositivo destino solicitando inmediatamente una respuesta; permite verificar el estado de una determinada conexión.
- *Echo reply*, es enviado como repuesta a un mensaje Echo Request.

2.4.2 DESCUBRIMIENTO DE VECINOS [25]

El protocolo de *Neighbor Discovery*, se define en el RFC 4681 y constituye un mecanismo a través del cual un nodo se incorpora a una red. Ha combinado las funcionalidades de otros protocolos IPv4 como ARP, ICMP *Router Discovery e ICMP Redirect*.

El protocolo de Descubrimiento de vecinos es utilizado por *routers* y *hosts* para los siguientes propósitos:

- Determinar la dirección MAC de los nodos de la red: El *host* envía un mensaje NS (Solicitud de vecino), informando su dirección MAC y solicitando la de su vecino; el vecino responde con un mensaje NA (Anunciación de vecino), informando su dirección MAC.
- Encontrar *router*s vecinos.
- Determinar Prefijos y otros datos de configuración de la red: los *router*s envían mensajes a la dirección *multicast all nodes*.
- Detectar direcciones duplicadas.
- Redireccionamiento de paquetes.
- Autoconfiguración de direcciones.

2.4.3 AUTOCONFIGURACIÓN DE DIRECCIONES STATELESS [37]

Permite atribuir direcciones IPv6 de tipo *unicast* a las interfaces de los diferentes *host*, evitando la necesidad de una configuración manual de los *host*s y tan solo una configuración mínima de ser necesaria en los *router*s. Permite que un *host* genere su dirección IP, mediante una combinación de información disponible localmente e información anunciada por los *router*s.

2.4.4 AUTOCONFIGURACIÓN DE DIRECCIONES STATEFUL

Se utiliza al protocolo DHCPv6 *Dynamic Host Configuration Protocol* para que los *hosts* obtengan direcciones de interfaz e información de configuración y los parámetros a partir de un servidor. Esta configuración permite tener un mayor control en la asignación de direcciones, ya que no solo proporciona las direcciones necesarias; sino también información de configuración como por ejemplo otros parámetros para servidores DNS, FTP, entre otros.

2.5 ENRUTAMIENTO EN IPV6 [38]

El proceso de enrutamiento consiste en determinar la mejor ruta para que un paquete llegue a su destino, ya sea en una red local o remota; este proceso al igual que en IPv4 es desarrollado por los *router*s. Para determinar la mejor ruta y reenviar un paquete, cada *router* utiliza información contenida en la tabla de enrutamiento tal como direcciones y valores de métrica de acuerdo al protocolo que se utilice en particular, así como su correspondiente interfaz de salida.

2.5.1 TABLA DE ENRUTAMIENTO

Es un archivo de datos que se almacena en la memoria RAM del *router*, posee la información de rutas para redes conectadas directamente y para redes conectadas remotamente, incluye detalladamente los siguientes parámetros:

- Dispositivo que origina la información
- Dirección destino y máscara de subred
- Dirección IP del siguiente salto
- La interfaz de salida a través de la cual se envía el paquete
- Un valor que se utiliza para seleccionar la mejor ruta entre varias rutas, en las que existe coincidencia para la dirección destino.
- Redes conectadas directamente a la interfaz del *router*.

2.5.2 ENRUTAMIENTO ESTÁTICO

Permite conectarse con redes remotas y está basado en entradas de la tabla de enrutamiento que se configuran manualmente, llamadas rutas estáticas que incluyen:

- Dirección IP y mascara de subred de la red remota
- Dirección IP del siguiente salto o la interfaz de salida

Dichas rutas no se alteran con un cambio en la topología de la red, ya que fueron asignadas manualmente y principalmente se utilizan en los siguientes casos:

- Una red está conformada por pocos *router*s
- Una red se conecta mediante un solo ISP hacia el Internet
- Una red extensa configurada como topología *Hub and spoke*

2.5.3 ENRUTAMIENTO DINÁMICO

Este tipo de enrutamiento, permite que los *router*s aprendan de manera dinámica acerca de los cambios en la topología de la red, sin necesidad de la intervención de un administrador. Los *router*s utilizan determinados protocolos de enrutamiento dinámico, que intercambian información entre *router*s acerca del estado y la accesibilidad a redes conectadas remotamente, de esta manera se consigue que la información de las rutas en la tabla de enrutamiento se genere, se mantenga y se actualice automáticamente.

2.5.3.1 Protocolos de Enrutamiento Dinámico

Se constituyen como un conjunto de procesos, algoritmos y mensajes que permiten el intercambio de la información de enrutamiento, para determinar la mejor ruta hacia una red determinada e integrar dichas rutas a la tabla de enrutamiento. Entre las actividades que realiza un protocolo de enrutamiento dinámico podemos mencionar:

- Descubrimiento de redes.
- Intercambio de la información de enrutamiento actualizada.
- Escoger el mejor camino hacia una red destino.
- Actualizar y mantener la tabla de enrutamiento.

Los protocolos de enrutamiento, utilizan un valor conocido como métrica para asignar costos a las distintas rutas aprendidas con un mismo protocolo, a fin de determinar cuál de ellas es la mejor.

Los protocolos de enrutamiento dinámicos, se clasifican en base a sus características específicas; en forma general se clasifican de acuerdo al concepto de sistemas autónomos (conjunto de *router*s administrados por una misma organización) en los siguientes tipos:

- *Protocolos de Gateway Interior* (IGP): se utilizan para el enrutamiento de paquetes dentro de un sistema autónomo.
- *Protocolos de Gateway Exterior* (EGP): se utilizan para el enrutamiento entre sistemas autónomos.

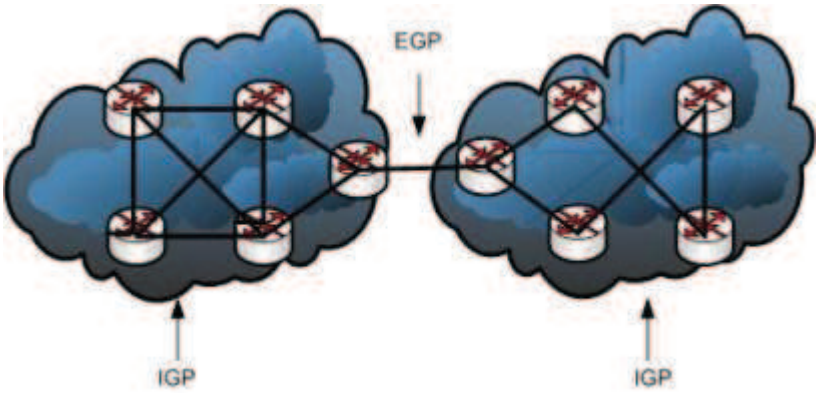


Figura 2.7 Clases de Protocolos de Enrutamiento Dinámico [**38**]

Dentro de los protocolos IGP encontramos los protocolos vector distancia y de *link state*. En los protocolos de vector distancia, las rutas se publican como vectores de distancia, la cual se define en términos del valor de métrica y de dirección, definida como el *router* del siguiente salto o la interfaz de salida.

Por otro lado los protocolos de *link state*, permiten obtener un mapa completo de la topología de la red y seleccionar el mejor camino hacia todas las redes destino.

Dentro del entorno IPv6 se definen los siguientes protocolos de enrutamiento dinámicos: RIPng para IPv6, OSPFv3 IPv6, IS-IS para IPv6, BGPv4

Se explican a continuación, los protocolos de interés para el desarrollo del prototipo.

2.5.3.1.1 RIPng para IPv6 [38] [39]

Es un protocolo de enrutamiento vector distancia, que se muestra como una adaptación del RIPv2 para IPv6; al igual que las versiones en IPv4 utiliza el conteo de saltos como métrica, es un protocolo de tipo IGP que se utiliza en redes simples y planas de tamaño moderado.

Una vez que se inicializo RIPng el *router* procede a enviar periódicamente cada 30 segundos actualizaciones, que contienen las rutas en su tabla de enrutamiento a través de cada una de sus interfaces. Para evitar problemas tales como *loops* de enrutamiento, RIPng establece un número máximo de 15 saltos para alcanzar su destino. Los mensajes de RIPng son encapsulados en un segmento UDP con números de puerto origen y destino 521.

2.5.3.1.2 OSPFv3 [40]

El *Open Shortest Path First* es un protocolo de enrutamiento de tipo *link state* que introduce el concepto de áreas para realizar la escalabilidad, un área OSPF es un conjunto de routers que comparte información de *link state*. Es una adaptación de OSPF para IPv6 y se utiliza dentro de un sistema autónomo.

Define la métrica como un valor arbitrario llamado costo, el cual puede ser asignado por el administrador de la red; en el caso del IOS[21] de los *routers* CISCO se utiliza el ancho de banda como métrica.

OSPFv3 trabaja con direcciones IPv6, distribuyendo por la red solamente el prefijo de estas direcciones, razón por la cual no es compatible con direcciones IPv4.

2.6 COEXISTENCIA Y TRANSICIÓN DE IPV4 A IPV6

La adopción de IPv6 se realizará de forma paulatina, por lo que la prioridad es la compatibilidad entre las dos versiones del IP. Se permite de esta forma implementar servicios IPv6 utilizando la infraestructura actual del IPv4; de modo que las redes IPv4 puedan comunicarse con redes IPv6 y viceversa. Con el objetivo de facilitar la coexistencia se han diseñado técnicas o mecanismos que permitan que los *host* y *routers* que utilizan IPv6 puedan interoperar con *host* IPv4 y utilizar la actual infraestructura de enrutamiento IPv4.

De esta manera progresivamente se conseguirá llegar al objetivo final de la migración, el cual consiste en convertir todos los nodos IPv4 en nodos IPv6.

Dentro de los mecanismos para coexistencia entre los protocolos e incluso una migración eventual y gradual al IPv6, tenemos las siguientes opciones:

- Doble Pila
- Tunelización
- Traducción

[21] *Internetworking Operating System* (IOS): software utilizado en routers y switches CISCO que permiten funciones de enrutamiento, conmutación, trabajo de internet y telecomunicaciones.

2.6.1 DOBLE PILA [25] [41]

Es uno de los mecanismos que permite compatibilidad entre nodos con IPv6 e IPv4 de la manera más directa, debido a que con la utilización de este método los nodos están equipados con pilas para los dos protocolos y se vuelven capaces de enviar y recibir paquetes tanto para IPv4 e IPv6, la ilustración de esta arquitectura se muestra en la figura 2.8.

Un nodo con soporte para ambos protocolos, puede ser configurado con los dos tipos de direcciones, una IPv4 y una IPv6; si bien es cierto estas direcciones pueden estar relacionadas, esto no es obligatorio ya que el nodo puede estar configurado con direcciones IPv4/IPv6 independientes entre sí.

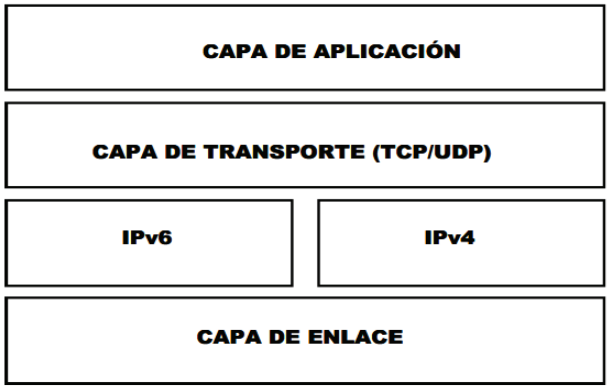


Figura 2.8 Arquitectura de la Capa Dual IP [**23**]

2.6.2 TUNELIZACIÓN [32] [41]

Este mecanismo permite utilizar la infraestructura de enrutamiento IPv4 existente, para transportar el tráfico IPv6. En esta técnica el contenido de un paquete IPv6 se encapsula en un paquete IPv4, de modo que se envíen por la infraestructura IPv4 y los dispositivos de borde sean los únicos que necesitan estar configurados con doble pila, el encapsulamiento se muestra en la figura 2.9.

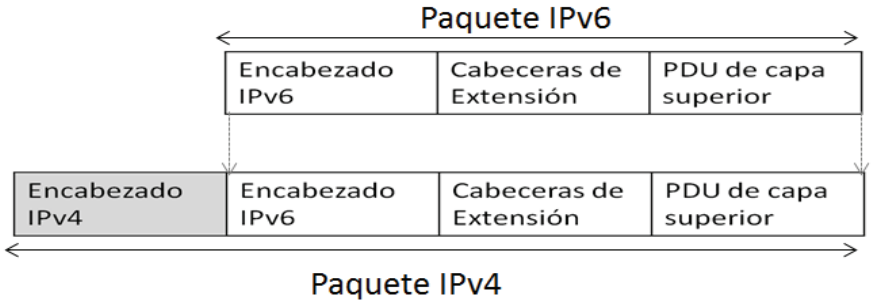


Figura 2.9 Encapsulado de IPv6 en IPv4 [**32**]

2.6.2.1 Túneles Estáticos

Requieren una configuración manual en los extremos del túnel, además las configuraciones de la interfaz del túnel deben ser especificadas a lo largo del túnel utilizando rutas estáticas.

2.6.2.2 Túneles Automáticos

No necesitan de una configuración manual, los túneles se crean de manera dinámica. En un extremo, el túnel utiliza una dirección IPv6 compatible con IPv4; mientras que la dirección del nodo destino, está incluida en el paquete que se encapsulo en el túnel.

2.6.3 TRADUCCIÓN

Este mecanismo permite compatibilidad entre nodos IPv4 e IPv6, con la ventaja de que el proceso de enrutamiento es transparente, en la comunicación entre nodos con diferentes versiones del protocolo IP. Los mecanismos de traducción permiten interoperabilidad entre nodos solo IPv4, solo IPv6 o nodos doble pila; ya que se produce una conversión directa de protocolos, la cual se consigue de distintas maneras e incluso en capas distintas.

2.7 COMPARACIÓN IPV4 FRENTE A IPV6

Las dos versiones del protocolo IP cumplen con la función de capa red, de interconectar redes e intercambiar paquetes, entre un origen y un destino que se identifican claramente con una dirección; la diferencia fundamental está en el tamaño de dichas direcciones y el formato de cabeceras que se añade a la carga útil, recibida desde la capa transporte. A continuación se describe las diferencias, similitudes, ventajas y desventajas que presenta un protocolo en comparación al otro.

2.7.1 CAMBIOS EN EL DIRECCIONAMIENTO

El campo de dirección origen y dirección destino se amplió de 32 bits a 128 bits, permitiendo un aumento del número de direcciones.

En cuanto al tipo de direcciones, de acuerdo al método de entrega en IPv4 como en IPv6 existen direcciones de tipo *Unicast* y *Anycast*, mientras que en IPv4 se permiten las direcciones de *broadcast*, en el IPv6 este tipo de dirección no se encuentra definida. De igual manera al referirnos al tipo de direcciones según su acceso a Internet, en IPv4 se diferencian claramente direcciones públicas y privadas; en IPv6 en analogía existen direcciones globales similares a las públicas y direcciones de enlace o sitio análogas a las privadas; la ventaja en IPv6 radica en que a un interfaz se le puede asignar varias direcciones a la vez, y de esta manera aprovechar los beneficios de cada tipo de dirección.

2.7.2 CAMBIOS A NIVEL DE CABECERAS

Se eliminó, modifico y se añadió nuevos campos en el encabezado IPv4 volviéndolo más simple, flexible y eficiente. El número de campos se redujo de 12 a solamente 8 para IPv6; mientras que el tamaño variable del encabezado IPv4 (que podía variar entre 20 y 60 bytes) pasa a convertirse en un encabezado de tamaño fijo en IPv6, con una longitud de 40 bytes. Además se añade una gran flexibilidad al utilizar las cabeceras de extensión para opciones adicionales. Los cambios en los campos de las cabeceras de cada versión se presentan en la figura 2.15.

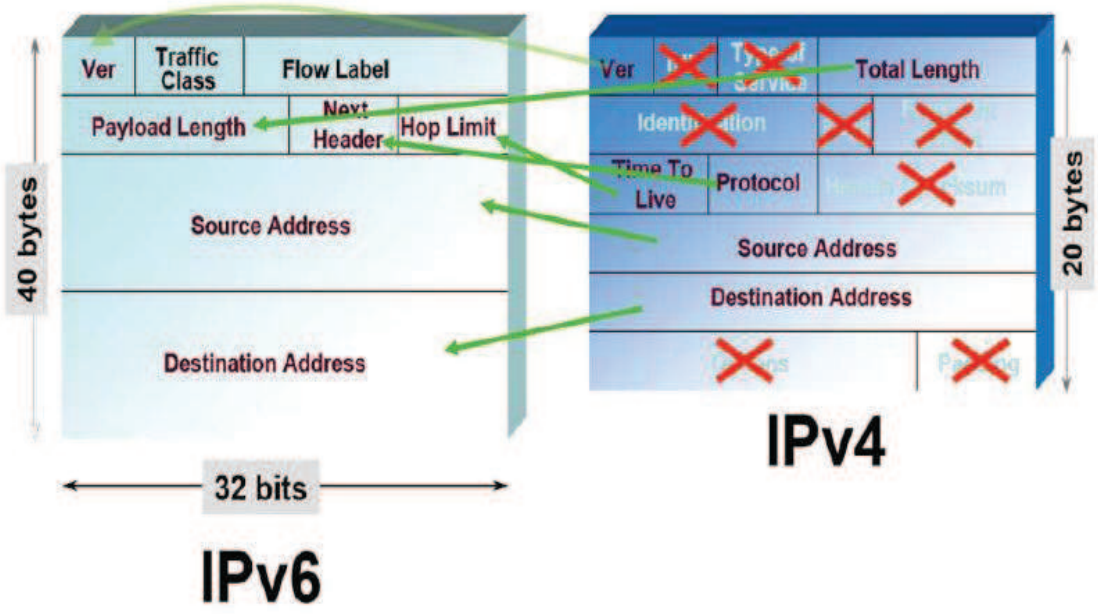


Figura 2.10 Modificaciones en los encabezados [**23**]

2.7.3 FRAGMENTACIÓN DE PAQUETES

La fragmentación de los paquetes en IPv6 se realiza solamente en los dispositivos finales; los nodos intermedios o *routers* no realizan este proceso, mejorando su tiempo de procesamiento.

2.7.4 IMPLEMENTACIÓN DE LA SEGURIDAD.

La implementación de la seguridad en las dos versiones del protocolo IP, se realiza en base al protocolo IPSec. Sin embargo dicha implementación es de carácter opcional para IPv4; pero de carácter obligatorio para IPv6, a pesar de ello se cuenta con la ventaja de que estas capacidades de seguridad, fueron diseñadas para que fueran utilizadas con IPv4 e IPv6.

2.7.5 COMPATIBILIDAD EN MECANISMOS DE ENRUTAMIENTO

La diferencia en cuanto al enrutamiento estático, se encuentra básicamente en el modo de configurar una interfaz, ya que para IPv4 utilizara la dirección de 32 bits; mientras que para IPv6 la dirección de 128 bits.

Mientras tanto en los mecanismos de enrutamiento dinámicos, la forma de implementación depende del tipo de protocolos; ya que algunos protocolos tales como IS-IS, poseen la cualidad de soportar múltiples protocolos de capa red, por lo que permite que las interfaces se puedan configurar tanto con la versión 6 como con IPv4. Por otro lado en protocolos tales como RIP y OSPF es necesario que la versión adecuada del protocolo de enrutamiento se configure, ya que las versiones del protocolo de enrutamiento no son compatibles entre sí.

CAPÍTULO 3

ANÁLISIS Y DETERMINACIÓN DE LOS REQUERIMIENTOS PARA LA IMPLEMENTACIÓN DEL PROTOTIPO

En este capítulo se analiza y se determina los requerimientos para constituir el escenario de pruebas adecuado, el cual permita determinar los parámetros de retardo, jitter, ancho de banda y pérdida de paquetes utilizando al sistema de telefonía IP Asterisk en conexiones de tipo IPv4 e IPv6. Posteriormente se establecen los criterios de diseño apropiados a nivel de red, que una vez establecidos permitan generar un esquema de topología y direccionamiento que incluya direcciones utilizando las dos versiones del protocolo IP. Finalmente se describe el diseño e implementación de la central telefónica IP PBX; para ello se analiza minuciosamente la versión de Asterisk a utilizar y se estructura la edición de los archivos de configuración necesarios para poner en funcionamiento los servicios de VoIP en el prototipo.

Para complementar la fase de diseño e implementación del prototipo se detalla el *hardware* a utilizar como teléfonos IP, *router*s, tarjetas de telefonía y estaciones de trabajo. Además se realiza una descripción de los recursos de *software* que se incluyen en el diseño, tanto del lado de los *hosts* como del servidor; incluyendo las funcionalidades de las aplicaciones y los servicios relacionados

3.1 REQUERIMIENTOS

3.1.1 REQUERIMIENTOS A NIVEL DE RED

3.1.1.1 Conectividad IPv4

Con el objeto de comprobar la conectividad a nivel de IPv4 se incluye en el prototipo *hosts* conectados directamente a la central IP PBX.

Este tipo de *hosts* permiten que se pueda apreciar los efectos a nivel de capa red y de capa aplicación dentro de un ámbito de red local.

Adicionalmente se incluye a nivel de IPv4 redes remotas conectadas a la PBX; dichas redes acceden a la central a través de *router*s unidos por medio de enlaces seriales, de modo que permitan apreciar el funcionamiento del prototipo en un ámbito de redes interconectadas. Dentro del entorno de pruebas para redes remotas se evalúa la compatibilidad de distintos tipos de direcciones; por lo cual es necesario incluir la configuración de un mecanismo de traducción. Al implementar este mecanismo se puede comprobar la conectividad IP en un entorno de conversión de direcciones públicas y privadas.

3.1.1.2 Conectividad IPv6

Para verificar la conectividad bajo el protocolo IPv6 se incluye en el prototipo *hosts* con direcciones IPv6 conectados directamente a la PBX, este tipo de *hosts* permiten comprobar las características de conectividad a nivel local y los efectos sobre las aplicaciones de Telefonía IP que se relacionen con las características propias de IPv6.

Existen además *host*s con direcciones IPv6 conectados con la central de manera remota a través de *router*s, para establecer dicha conectividad se utiliza un mecanismo de enrutamiento que permite además comprobar el efecto del procesamiento del enrutador sobre las aplicaciones de telefonía IP.

Por lo tanto se incluye la configuración de un protocolo de enrutamiento compatible con IPv6 y que permita que los paquetes de voz sean entregados de manera adecuada.

3.1.1.3 Ancho de banda y Rendimiento

Una de las finalidades del prototipo consiste en analizar el efecto del procesamiento, cantidad de nodos y dispositivos intermedios sobre el ancho de banda; así como el rendimiento en los enlaces y las comunicaciones de voz a implementarse.

Por tal motivo es necesario incluir mecanismos que permitan controlar características como velocidad de transmisión en relación con el número de usuarios y el modo de transmisión; así como el dimensionar y configurar de manera adecuada los enlaces para permitir dichas comunicaciones.

3.1.1.4 Retardos

Con el propósito de medir y monitorear el retardo dentro del prototipo, es necesario generar un entorno de red en donde se presenten retardos debido a factores como:

- NIC de las estaciones de trabajo.
- Número y tipo de dispositivos de red presentes.

Además es necesario considerar que el retardo no depende únicamente de la distancia o el número de dispositivos; sino que guarda una estrecha relación con la función que desempeñan los dispositivos de red, que se encuentren en la ruta que sigue un paquete entre su nodo origen y destino.

Por lo tanto se requiere distinguir los efectos de retardo al utilizar un dispositivo que realice procesamiento a nivel de capa 2 en un entorno de red local; así como un dispositivo de capa 3 que procese los paquetes a nivel de red para un entorno de redes y conexiones remotas.

Finalmente se debe considerar el retardo que se genera como producto del procesamiento en la central IP PBX, en donde se alojan además servicios adicionales para cumplir con las funciones típicas de telefonía IP en el prototipo.

3.1.2 REQUERIMIENTOS A NIVEL DE SERVICIOS Y APLICACIONES

Dentro de los requerimientos del protocolo se incluye el comprobar algunas funciones típicas de una PBX en un ambiente empresarial tales como:

- Dial plan
- Transferencia de llamadas
- Grabación de llamadas
- Conferencia de voz

- Directorio
- Correo y buzón de voz
- Conexión a Base de datos
- Respuesta interactiva de voz

Para que dichas aplicaciones puedan efectuarse es necesario que la red del prototipo cuente con algunos servicios fundamentales, dichos servicios permiten al usuario interactuar con las diferentes funciones de la central; por lo tanto es necesario incluir servicios tales como:

- ***Servicio de nombres de dominios***, que permitan al usuario acceder con facilidad a un nombre equivalente a las direcciones IP, esto posibilita a los usuarios disponer y manejar sin conflictos los diferentes servicios. El servidor permitirá acceder a una determinada aplicación utilizando un nombre de dominio, de modo que la dirección IP versión 4 o versión 6 detrás de dicho dominio sea transparente para el usuario.
- ***Servicio de Correo***, que permite al usuario enviar, recibir, almacenar y filtrar mensajes de correo electrónico, independientemente de la red que el usuario esté utilizando. Es necesario que el servicio de correo se encuentre entrelazado con la aplicación de correo de voz, esto permite que los usuarios accedan a los mensajes de voz grabados a partir de una cuenta de correo electrónico.
- ***Servicio de páginas Web***, que permite al usuario acceder a páginas web o cualquier tipo de información, para lo cual procesa aplicaciones desde el lado del servidor y genera respuestas a diversas peticiones del lado de los clientes. Por lo tanto este servicio es necesario para complementar el acceso y manipulación del resto de servicios y aplicaciones que utiliza el prototipo.

3.1.3 CONSIDERACIONES DE SEGURIDAD [42]

Debido a la sencillez y rapidez en el diseño y funcionamiento de un sistema de telefonía IP utilizando Asterisk es recomendable incorporar mecanismos de seguridad inmersos en el sistema y que deben considerarse en el mismo.

Un sistema de telefonía IP implementado en base a *software* requiere de tres niveles básicos de seguridad:

1. Seguridad externa (acceso al sistema)
2. Seguridad de autenticación
3. Seguridad de operación

Dentro de las consideraciones de seguridad externa se restringe el acceso al sistema desde fuentes externas como el Internet; dentro de las opciones más comunes se encuentra el uso de un *firewall* que puede implementarse por *hardware* o *software*. Dentro del presente prototipo no se incluye la configuración de este mecanismo; pero se mencionan algunas alternativas que se analizan y permiten proporcionar una guía para proyectos futuros.

El *Session Border Controler* es un dispositivo típico empleado como solución de seguridad en la telefonía IP y comunicaciones multimedia sobre IP en general. Este dispositivo desarrolla la función de gestionar el flujo de datos de sesiones multimedia incluyendo datos de señalización e información pura.

Dentro del prototipo se requiere implementar un nivel de seguridad a nivel de aplicación; es decir incluir mecanismos de protección contra atacantes externos considerando que ya penetraron el sistema y tienen acceso a la central a nivel de red. Con este propósito se consideran dos aspectos que otorguen al sistema un nivel de seguridad en contra de accesos a los recursos de telefonía IP sin autorización del administrador. Los mecanismos de seguridad por *software* se describen a continuación.

3.1.3.1 Seguridad por autenticación

Este nivel de seguridad impide el registro y acceso de un intruso al sistema de telefonía IP. Se presenta una vez que el intruso puede ver la central, es decir tiene acceso a ella a nivel de red.

Para ello se implementan dos mecanismos fundamentados en la autenticación:

- Listas de acceso: permite configurar un padrón en donde se revise la correspondencia de los dispositivos que deben o pueden registrarse en el sistema. En base a esta configuración se limita el acceso a ciertas extensiones desde determinadas direcciones IP entrantes. Por lo tanto no se aceptan pedidos de autenticación SIP desde cualquier dirección IP.
- Contraseñas de usuarios: se incrementa el nivel de seguridad incluyendo la autenticación para el registro de usuarios SIP; con este objeto se conforma claves seguras para las entidades SIP.

3.1.3.2 Seguridad de Operación

Consiste en limitar el acceso a extensiones de acuerdo a las funciones que desempeñan dentro de todo el sistema. Al configurar los usuarios en el sistema se controla sus accesos mediante el uso de contextos en el plan de marcado.

De esta manera determinados usuarios creados en los canales SIP o IAX solo pueden marcar estrictamente lo que el administrador les permite al habilitar un contexto en el momento de definir al usuario.

La configuración de los mecanismos de seguridad, autenticación y operación se describen a detalle en la **sección 3.3.2.2.3.**

3.2 DESCRIPCIÓN DE LA TOPOLOGÍA DEL PROTOTIPO

El escenario que responde a los requerimientos descritos para completar las pruebas está basado en interconectar la central IP PBX instalada en un servidor principal, con un conjunto de *host*s IPv4 e IPv6, los cuales comprenden estaciones de trabajo y teléfonos IP.

De esta manera el prototipo incluye una red local en donde se encuentra el servidor principal en el cual se instaló la central PBX Asterisk, un servidor de páginas web, correo y nombres de dominio; así como todos los complementos necesarios para el funcionamiento de las aplicaciones de telefonía IP.

3.2.1 SEGMENTO DE RED LOCAL

Para cumplir con los requerimientos de pruebas y mediciones es necesario incluir un segmento de red local, el mismo que simule el comportamiento de los parámetros a medir a nivel local. Con este objetivo es necesario que se interconecte al servidor con *host*s de tipo IPv4 e IPv6 dentro de la misma subred.

Se distinguen dentro de este segmento dos subredes una IPv4 y una IPv6 que funcionan simultáneamente gracias a la doble pila del servidor Asterisk; por consiguiente se cuenta con una sola topología física, pero con dos topologías lógicas.

Para este efecto se determina como *host* de tipo IPv4 a teléfonos IP y softphones, instalados en estaciones de trabajo configuradas con direcciones IPv4; mientras que por otro lado se utiliza como *host* tipo IPv6 softphones alojados en estaciones de trabajo configuradas con direcciones IPv6.

Esta estructura de red local permite comprobar los efectos de retardo, ancho de banda, jitter y pérdida de paquetes al realizar llamadas y correr las aplicaciones de telefonía IP en un entorno de red LAN. En donde todos los dispositivos están en el mismo segmento de red permitiendo verificar las ventajas en un entorno en donde el throughput aumenta notablemente, por motivos tales como ausencia de colisiones, ancho de banda dedicado a cada puerto y una operación full dúplex.

3.2.2 SEGMENTO DE RED IPV6 PURA

Se requiere de un segmento de red IPv6 puro con la finalidad de simular un ambiente propicio para satisfacer los requerimientos de pruebas y mediciones en aplicaciones de VoIP utilizando el protocolo IPv6; en donde todos los dispositivos intermedios y finales sean compatibles con direcciones de tipo IPv6. Este segmento de red debe estar conformado por un conjunto de *router*s, los cuales permiten verificar el efecto de procesamiento en cada uno y sus correspondientes enlaces sobre una conversación telefónica IP utilizando direcciones IPv6.

3.2.3 SEGMENTO DE RED MIXTO

Con el objetivo de analizar el comportamiento de los parámetros de telefonía IP en ambientes propios del Internet tales como traducción de direcciones y mecanismos de coexistencia; es necesario establecer un segmento de red en donde coexistan dichos mecanismos de conectividad y permitan analizar el efecto que producen sobre una comunicación telefónica.

Este segmento de red está formado por dispositivos intermedios que permiten interconectar *host*s de tipo IPv4 e IPv6 mediante una infraestructura basada en un mecanismo de traducción de direcciones de tipo NAT y un mecanismo de coexistencia de tipo tunelización, estos mecanismos son configurados respectivamente en el correspondiente nodo intermedio.

3.3 DISEÑO DEL PROTOTIPO

3.3.1 DISEÑO DE LA RED ACTIVA

3.3.1.1 Esquema de Topología y Direccionamiento

Para el análisis de los parámetros de retardos, *jitter*, ancho de banda y pérdida de paquetes dentro del prototipo es necesario dividir el prototipo de acuerdo al tipo de red y el protocolo de red que se utiliza. Por esta razón se reparte a la red, en tres entornos formados cada uno por diferentes segmentos de red.

3.3.1.1.1 Segmento de Red Local

La red local del servidor está conectada hacia dos redes remotas que permiten respectivamente comprobar la conectividad y la funcionalidad de la telefonía IP.

Existe conectividad hacia un escenario puramente IPv6 mediante uno de los segmentos de red y hacia un escenario IPv4 en el que está inmerso un mecanismo de traducción de direcciones a través de un segundo segmento de red. Por lo cual se distingue como el segmento de red local a la porción en donde inter operan las dos versiones del protocolo IP; ya que se presentan tanto *host*s configurados con direcciones IPv4 y *host*s con direcciones IPv6.

El segmento está formado por el servidor Asterisk y un conjunto de *host*s conectados mediante un *switch* central, el cual permite establecer la conexión entre los dispositivos finales dentro de una misma subred. Es necesario distinguir las dos subredes simultáneas que coexisten físicamente dentro del mismo segmento de red, cada una de las cuales se detalla a continuación:

- ***Subred IPv4***

Con las consideraciones anteriores se escoge a nivel de IPv4 un espacio de direccionamiento de tipo privado y de clase B, se incluye como dirección de red la 172.31.0.0 con una máscara de red 255.255.0.0; puesto que en este segmento de red no se necesita acceso a Internet y el número de *host*s disponibles es suficiente para los clientes de telefonía IP que existen.

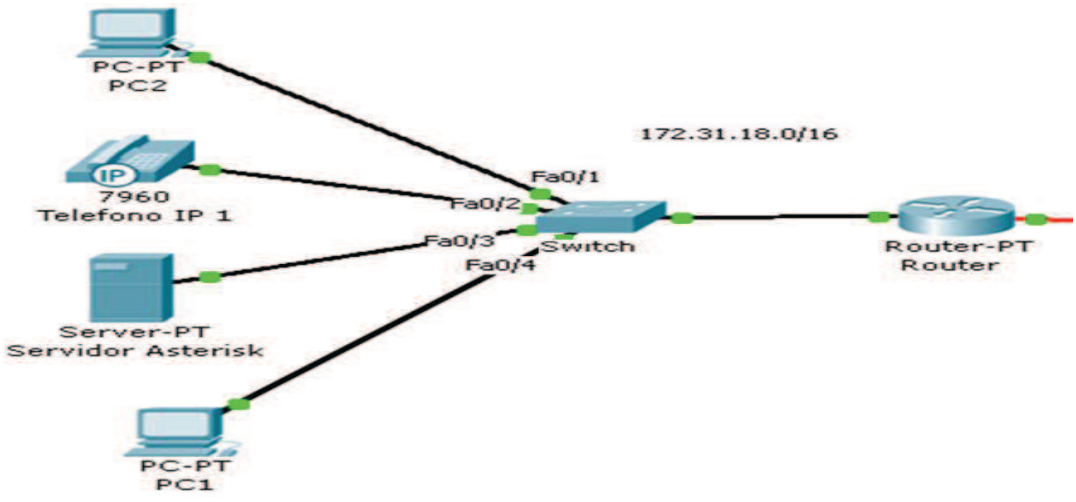


Figura 3.1 Diagrama de Topología del Segmento de Red Local

Se configura el siguiente esquema de direccionamiento en base a los dispositivos inmersos en el segmento de red LAN:

Dispositivo	**Interfaz**	**Dirección IP**	**Máscara de subred**	**Gateway por defecto**
Servidor Asterisk	NIC	172.31.18.41	255.255.0.0	172.31.18.2
Teléfono IP 1	LAN	172.31.18.58	255.255.0.0	172.31.18.2
PC1 Softphone Zoiper	NIC	172.31.18.60	255.255.0.0	172.31.18.2

Tabla 3.1 Direccionamiento IPv4 de la red Local del Servidor

- ***Subred IPv6***

Es necesario considerar el esquema de direccionamiento para el mismo segmento de red utilizando el protocolo IPv6, para lo cual se utiliza direcciones de tipo global *unicast* por su característica de ser enrutable y accesible globalmente; se permite de este modo la interconexión de todas las subredes presentes en las interfaces de los ruteadores, incluso dentro de otros segmentos del prototipo.

Se utiliza una estructura de direccionamiento basada en el espacio 2000::/112; permitiendo un número de hasta 65 535 *host*s, valor más que suficiente para cubrir las direcciones necesarias para cada cliente de telefonía IP dentro de la red LAN, incluyendo al servidor de Asterisk. Adicionalmente al emplear un espacio de direccionamiento de tipo global se proporciona a dicha red local la posibilidad de conectarse con otras redes remotas IPv6.

Dispositivo	**Interfaz**	**Dirección IP**	**Máscara de Subred**	**Gateway por defecto**
Servidor Asterisk	NIC	2000::AA:3	/112	2000::AA:1
PC2 Softphone Linphone	NIC	2000::AA:10	/112	2000::AA:1

Tabla 3.2 Direccionamiento IPv6 de la red Local del Servidor

3.3.1.1.2 Segmento de Red IPv6 Puro

Este segmento de red permite comprobar las funcionalidades que se presentan en un ambiente puramente IPv6; el segmento está formado por cinco subredes conectadas a través de 4 *router*s unidos entre sí por enlaces seriales, de modo que permitan establecer una conexión extremo a extremo entre el servidor Asterisk y un cliente IPv6.

El entorno de este segmento de red hace necesario que se utilicen direcciones del tipo global *unicast* para interconectar las subredes utilizando un espacio de direccionamiento dentro del rango 2000::0/112.

Se utiliza una convención en base a los *router*s para identificar las subredes en cada uno de los enlaces. Para una Subred entre los *routers* RX y RY asignamos la dirección de red 2000::XY:0/112 .

Figura 3.2 Diagrama de Topología del Segmento de Red IPv6 puro

Se asigna las dos primeras direcciones disponibles a los interfaces del enlace serial, siendo la primera dirección la interfaz de salida y la segunda dirección la interfaz del próximo salto.

El esquema de direccionamiento para este segmento de red es el de la tabla 3.3.

Dispositivo	Interfaz	Dirección IP	Máscara de Subred	Gateway por defecto
R1	Eth0/0	2000::AA:1	/112	No aplicable
	S1/0	2000::12:1	/112	No aplicable
R2	S0/0	2000::12:2	/112	No aplicable
	S0/1	2000::23:1	/112	No aplicable
R3	S0/1	2000::23:2	/112	No aplicable
	S0/2	2000::34:1	/112	No aplicable
R4	S0/1	2000::34:2	/112	No aplicable
	Eth0/0	2000::4C:1	/112	No aplicable
PC3 Linphone	NIC	2000::4C:30	/112	2000::4C:1

Tabla 3.3 Direccionamiento IPv6 del segmento de Red IPv6 Puro

Dentro de este segmento de red se incorpora la configuración de un protocolo de enrutamiento dinámico; esto permite incluir los efectos debidos al intercambio de información acerca de rutas, cambios en la topología y actualizaciones de las tablas de enrutamiento.

El canje de información requiere parte de los recursos del *router*, tiempo de la CPU e incluso ancho de banda del enlace de la red. Este procedimiento permite comprobar el efecto del procesamiento sobre cada aplicación de telefonía IP a través de pruebas de retardo, *jitter* y pérdida de paquetes.

Entre las ventajas de incluir un protocolo de enrutamiento dinámico para permitir el encaminamiento de los paquetes en el prototipo, se incluyen las siguientes:

- La complejidad de la configuración es independiente del tamaño de la red.
- Permite que la red se adapte a cambios en la topología.
- El escalamiento es adecuado para topologías simples y complejas.

De acuerdo con los requerimientos del prototipo y las características como protocolo se configura RIPng como mecanismo de enrutamiento. A continuación se describe los detalles en cuanto al protocolo y el diseño de la red en la tabla 3.4.

Requerimiento	RIPng
Soporte de IPv6	Si
Velocidad de Convergencia	Lenta
Escalabilidad-Tamaño de Red	Pequeña
Uso de recursos	Bajo
Implementación y Mantenimiento	Simple
Encapsulamiento	UDP

Tabla 3.4 Características del protocolo de enrutamiento a utilizar

RIPng cumple con las necesidades del prototipo en base a la evaluación de los siguientes parámetros:

- Es ideal para una red de tamaño pequeño
- El consumo de los recursos de CPU del *router* y de ancho de banda en los enlaces no es alto;
- El número de *router*s en el prototipo no supera el número de saltos establecidos como métrica, por lo que no se descartan paquetes
- El protocolo está estandarizado y es fácil de configurar
- Permite prevenir *loops* de enrutamiento
- Este protocolo se encapsula en UDP al igual que la mayoría de paquetes inmersos en las aplicaciones de telefonía IP.

3.3.1.1.3 Segmento de Red Mixto

Este segmento de red está formado por dos *router*s unidos por un enlace de tipo IPv4 que interconectan remotamente la red local del servidor con dos subredes una de tipo IPv4 y una de tipo IPv6.

Para cumplir con los requerimientos de pruebas y alcanzar conectividad hasta cada una de las subredes remotas, es necesario configurar un mecanismo de

traducción de direcciones NAT para la subred IPv4 y un mecanismo de tunelización para la red IPv6, tal y como se muestra en la figura 3.3.

Figura 3.3 Esquema de conectividad del segmento de red mixto

- ***Infraestructura de conexión entre Routers***

Es necesario emplear como dispositivos intermedios dos *router*s unidos por un enlace serial de tipo IPv4, que sirve como infraestructura de conexión.

Adicionalmente se requiere configurar las interfaces de este enlace con direcciones públicas para implementar el mecanismo de NAT; ya que además este tipo de direcciones no presentan ningún impedimento para la implementación del mecanismo de tunelización. Por esta razón se escogió el espacio de direccionamiento 200.200.200.56.0/24, que incluye un número de *host*s suficientes para la asignación de las interfaces de los *router*s que intervienen en el enlace.

Finalmente es necesario configurar los dos *router*s de acuerdo a su requerimiento específico; para lo cual denominamos al *router* 5 como el conectado directamente al servidor y *router* 6 al que se encuentra conectado hacia las subredes remotas.

Se describe a continuación a detalle los mecanismos de conectividad utilizados dentro del segmento de red mixto:

- ***NAT***

Este mecanismo se configura de modo que permita verificar el efecto de incluir un esquema de traducción de direcciones públicas y privadas sobre un *router*; así como el resultado de esta implementación sobre las llamadas y demás funciones de telefonía IP.

Adicionalmente la configuración del mecanismo NAT en uno de los *router*s de este segmento de red permite apreciar las ventajas y desventajas de esta traducción de direcciones en comparación a la configuración del protocolo IPv6.

Al implementar la traducción NAT se utiliza un rango de direcciones IP públicas para un conjunto de direcciones IP privadas; las IP privadas son mapeadas de forma estática para cada uno de los diferentes *host* a utilizarse.

Para la respectiva asignación se emplea el espacio de direcciones privadas pertenecientes a la red 172.31.0.0/16; esta red proporciona la cantidad de *host*s adecuadas para comprobar la incorporación de NAT; dentro de la **sección 3.4.1.** se explica la configuración de los equipos activos de red en donde se incluye el rango de direcciones públicas para realizar la traducción.

El dispositivo sobre el cual se debe configurar este NAT es el *router* R5, ya que comprende el límite entre el direccionamiento privado de la red local del servidor y el direccionamiento público de la subred remota IPv4 conectada al *Router* R6.

Es necesario asignar un espacio de direccionamiento público en una de las interfaces de R5; se escoge por lo tanto asignar a la interfaz Ethernet 0/0 de R6 el espacio 200.200.6.2/24 para compatibilidad con NAT y cubrir con una cantidad de *host*s suficientes en un entorno corporativo. Se presenta a continuación el esquema de direccionamiento relacionado al mecanismo de NAT y la subred IPv4.

Dispositivo	**Interfaz**	**Dirección IP**	**Máscara de Subred**	**Gateway por defecto**
R5	Eth0/0	172.31.18.2	255.255.0.0	No aplicable
	S0/0	200.200.56.1	255.255.255.0	No aplicable
R6	Eth0/0	200.200.6.1	255.255.255.0	No aplicable
	S0/0	200.200.56.2	255.255.255.0	No aplicable
PC4 Softphone Linphone	NIC	200.200.6.2	255.255.255.0	200.200.6.1

Tabla 3.5 Direccionamiento IPv4 en el segmento de red mixto

La configuración del *router* R5 se explica en la sección de Implementación y se complementa con el correspondiente archivo de configuración presentado en el **anexo C.**

- ***Tunelización***

Dentro del mismo segmento de red se requiere configurar un mecanismo que permita la interoperabilidad entre IPv4 e IPv6 en los *router*s que interconectan las redes. Por este motivo se incluye la implementación de un mecanismo de coexistencia, que permita conectar un *host* con dirección IPv6 del lado de red remota, con los *host* de tipo IPv4 en la red local del servidor.

El segmento mixto debe incluir un espacio de direccionamiento del tipo IPv6 en la interfaz LAN en el *router* 6, R6 está situado en el extremo que atraviesa la infraestructura IPv4 presente en el enlace serial inter *router*s.

Adicionalmente en este segmento de red se incluye la comunicación desde el segmento de red IPv6 puro hacia un *host* IPv6 en el extremo remoto del *router* R6 en su interfaz Ethernet 0/1. Por lo tanto se asigna el espacio de direccionamiento 2000::CC:0/112 de tipo global *unicast*, dicha asignación permite interconexión con las diferentes subredes IPv6 fuera del segmento de red mixto.

Para que la conectividad hacia el *host* IPv6 sea posible es necesario implementar un mecanismo de coexistencia que sea simple, de fácil implementación y que aproveche la estructura de red IPv4.

Por lo cual en la configuración de los *router*s se debe incluir la configuración de un túnel que cumpla con los requerimientos de conectividad de extremo a extremo.

La configuración del túnel debe llevarse a cabo en cada uno de los *router*s; esta se describe en la sección de implementación y se detalla en los archivos de configuración de los *Router*s 5 y 6 respectivamente presentados en el **anexo C.**

Una vez conocidas las especificaciones y requerimientos del segmento de red, se presenta el siguiente esquema de direccionamiento y su correspondiente diagrama de topología:

Dispositivo	Interfaz	Dirección IPv6	Dirección IP	Gateway
R5	Eth0/0	No asignada	172.31.18.2/16	No aplicable
	S0/0	No asignada	200.200.56.1/24	No aplicable
R6	Eth0/0	2000::cc:1/112	No asignada	No aplicable
	S0/0	No asignada	200.200.56.2/24	No aplicable
PC5 Softphone Linphone	NIC	2000::cc:2/112	No asignada	2000::cc:1/112

Tabla 3.6 Direccionamiento IPv4 e IPv6 del Segmento de Red Mixto

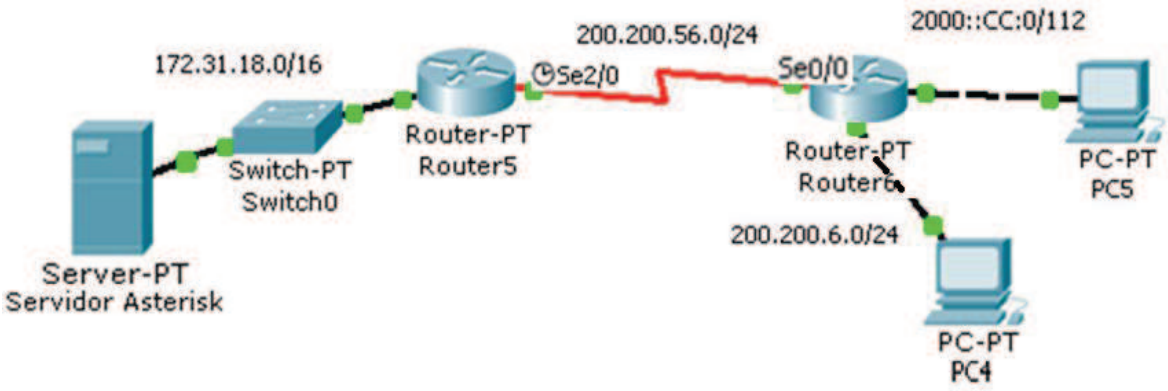


Figura 3.4 Diagrama de topología del segmento de red Mixto

3.3.1.2 Resumen de Direccionamiento

El resumen de la estructura de direccionamiento que se implementa en cada segmento de red se muestra por separado dependiendo del tipo de direcciones. Se describe el esquema de direccionamiento, la especificación de *host*s y puerta de enlace predeterminado de los segmentos de red interconectados; así como los enlaces entre ellos utilizando el protocolo IPv4.

Segmento	Dirección de subred	Máscara	Default Gateway	Primer Host	Último Host	Broadcast
LAN Servidor	172.31.18.0	255.255.255.0	172.31.18.1	172.31.18.2	172.31.18.254	172.31.18.255
Enlace R5-R6	200.200.56.0	255.255.255.0	No aplicable	200.200.56.1	200.200.56.254	200.200.56.254
LAN R6	200.200.6.0	255.255.255.0	200.200.6.1	200.200.6.2	200.200.6.254	200.200.6.255

Tabla 3.7 Direccionamiento IP para el protocolo IPv4

Es necesario además identificar las diferentes subredes, *host*s y dispositivos intermedios que utilizan el protocolo IPv6. Para este efecto se detalla el esquema de direccionamiento de los segmentos de red que incorporan IPv6, así como los enlaces que los interconectan.

Segmento	**Dirección de subred**	**Máscara**	**Default Gateway**	**Primer Host**	**Último Host**
Red LAN IPv6	2000::AA:0	/112	2000::AA:1	2000::AA:2	2000::AA:FFFF
Enlace R1-R2	2000::12:0	/112	No aplicable	2000::12:1	2000::12:FFFF
Enlace R2-R3	2000::23:0	/112	No aplicable	2000::23:1	2000::23:FFFF
Enlace R3-R4	2000::34:0	/112	No aplicable	2000::34:1	2000::34:FFFF
Red LAN R4	2000::4C:0	/112	2000::4C:1	2000::4C:2	2000::4C:FFFF
Red LAN Túnel R6	2000::CC:0	/112	2000::CC:1	2000::CC:2	2000::CC:FFFF

Tabla 3.8 Direccionamiento IP para el protocolo IPv6

Finalmente se presenta el diagrama de topología completo del prototipo de red, en el que se interconectan todos los segmentos de red que conforman el prototipo; tal como se detalla en la figura 3.5.

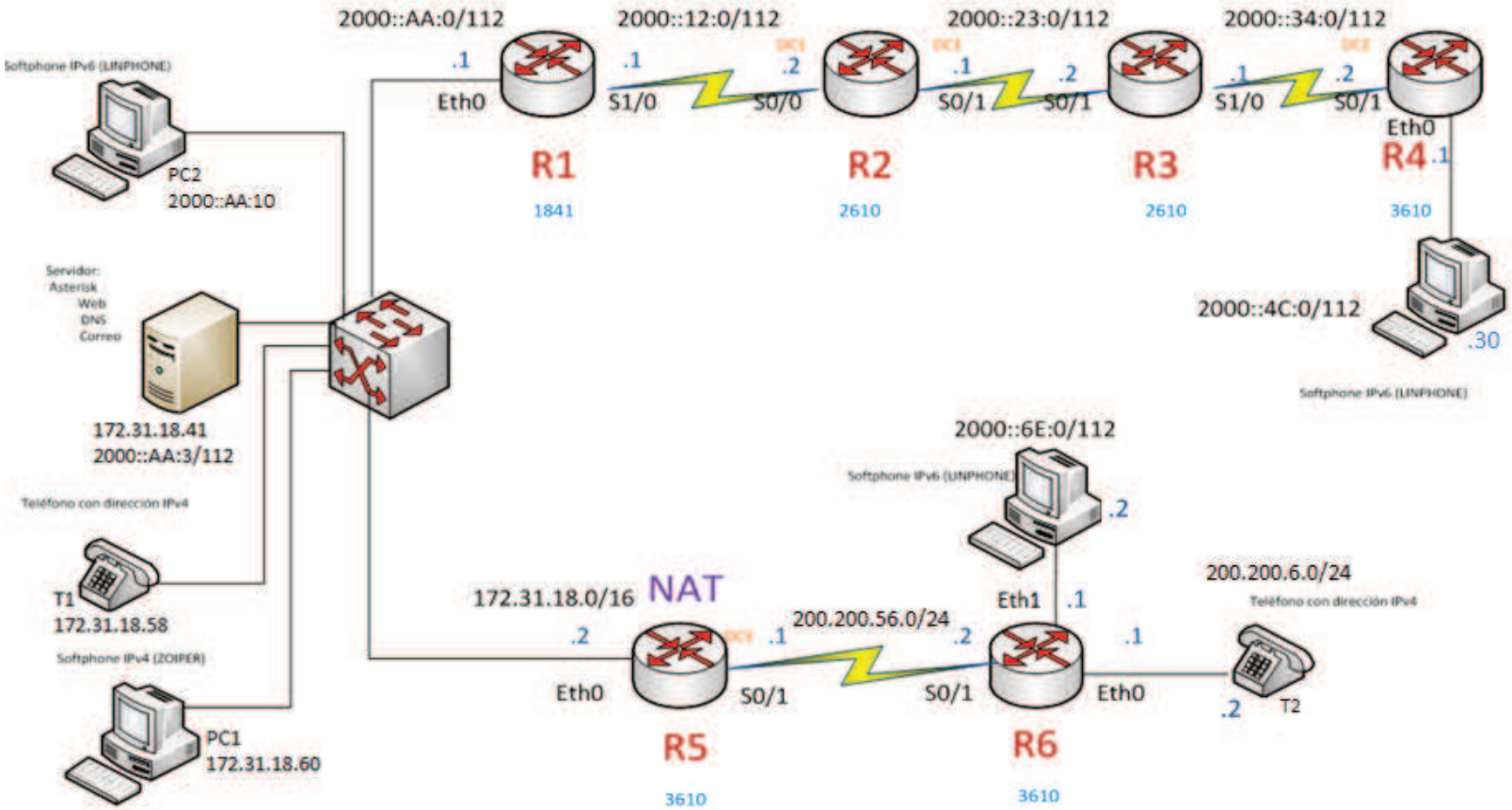


Figura 3.5 Diagrama del prototipo implementado

3.3.1.3 Especificación de los equipos de red y sus requerimientos de software

En esta sección se describen los equipos activos de red que se utilizan en el prototipo. Se definen sus principales características en comparación a las necesidades a nivel de *software* y *hardware* de nuestra red.

3.3.1.3.1 Equipos activos de red

- ***Routers***

Los *router*s desempeñan la función de encaminar los paquetes IPv4 e IPv6 entre los diferentes segmentos de red. Entre los requerimientos de la red se encuentran la implementación de diferentes protocolos de enrutamiento, mecanismos de traducción de direcciones y principalmente la capacidad de soporte para IPv6 tanto en mecanismos de enrutamiento como de interoperabilidad con IPv4.

En el entorno del segmento de red IPv6 puro es necesario un dispositivo con soporte para IPv6. Este tipo de *router* debe implementar el protocolo RIPng como mecanismo de enrutamiento y además incorporar las especificaciones de *hardware* necesarias tales como ranuras de expansión e interfaces seriales y de administración adecuadas.

Para el segmento de red mixto es necesario que los *router*s incorporen en su sistema operativo compatibilidad con IPv6; los *router*s deben configurar el mecanismo de NAT y además incluir el soporte para implementar túneles como interfaces virtuales.

Adicionalmente se considera la facilidad de manejo y la familiarización del administrador con la interfaz de configuración disponible en el *software* del dispositivo.

Todos los requerimientos descritos anteriormente se deben considerar en la selección de la marca y modelo de *router*. En base a un análisis e investigación de los *router*s disponibles se reduce las posibles soluciones a 3 marcas que cumplen con los requerimientos dentro del prototipo. Sin embargo es importante destacar que no son la única solución posible para desarrollar el presente prototipo.

El resumen de requerimientos, marcas y modelos se detallan en la tabla 3.9.

Especificaciones	**Requerimiento Mínimo**	**CISCO Serie 1900**	**HP Serie 6600**	**H3C Serie MSR 20-1x**
Características Generales	Administrable	Si	Si	Si
	Soporte para FTP y TFTP	Si	Si	Si
	Memoria RAM	512 MB - 2GB DRAM	4 GB SDRAM	256 MB DRAM
	Memoria FLASH	256 MB - 8 GB Flash	1 GB Flash	32 MB Flash
Protocolos y soporte de aplicaciones	Soporte de IPv6	Si	Si	Si
	Enrutamiento Estático	Si	Si	Si
	Enrutamiento Dinámico	RIP, OSPF, EIGRP, RIPng, OSPFv3, IS-ISv6 entre otros.	RIP, OSPF, RIPng, OSPFv3, BGP, IS-ISv6, entre otros.	RIP, OSPF, RIPng, OSPFv3 ,BGP, IS-ISv6, entre otros
	Soporte de NAT	Si	Si	Si
	Mecanismos de coexistencia IPv4-IPv6	Doble pila Túneles 6to4	Doble Pila Túneles 6to4	Doble Pila Túneles IPv6-IPv4
Características Funcionales	Interfaces de interconexión de datos	Ethernet/Fast Ethernet/Gigabit Ethernet.	Fast Ethernet/Gigabit Ethernet/10 GbE, entre otros	Ethernet/Fast ethernet
	Interfaces de administración	1 puerto serial de consola 1 puerto serial auxiliar 1 puerto usb de consola	Puerto de módem e interfaz de terminal Ethernet	1 puerto serial de consola 1 puerto serial auxiliar 1 puerto usb

Tabla 3.9 Requerimientos y especificaciones para *router*s [**43**] [**44**] [**45**]

Para seleccionar una de las tres alternativas consideramos adicionalmente los siguientes factores:

- La compatibilidad, facilidad de manejo, experiencia previa y familiarización con el manejo de la interfaz de administración.
- La disponibilidad de los equipos para la implementación y ejecución de pruebas de laboratorio.

En respuesta a dichos requerimientos se utiliza dentro de la red *router*s de marca CISCO. Esta marca se presenta como una solución ideal tanto a nivel de *hardware* como *software*.

Finalmente se presenta un resumen en la tabla 3.10 con el tipo de dispositivos y su nomenclatura de acuerdo a la implementación en el prototipo.

Nombre del Equipo	Marca	Serie
R1	CISCO	1900
R2	CISCO	1900
R3	CISCO	1900
R4	CISCO	1900
R5	CISCO	1900
R6	CISCO	1900

Tabla 3.10 Nomenclatura y serie de los *router*s del prototipo

- ***Switch***

En el entorno de red local es necesario un dispositivo de conmutación que permita a los dispositivos comunicarse evitando las colisiones. Dentro de los requerimientos para el *switch* únicamente se presentan los que se relacionan con respecto a aspectos de capa enlace y física; específicamente en cuanto a la velocidad de los puertos ya que las interfaces de los *host* y el servidor son de tipo *Fast ethernet* con una velocidad de 100 Mbps.

Dentro del prototipo no se manejan parámetros de configuración típicos en un *switch* tales como enlaces troncales o VLANS. Por esta razón no es necesario que el *switch* sea administrable. Se requiere únicamente cumplir con requisitos de velocidad de puertos y el número de puertos dentro de la red LAN.

El detalle de requerimientos de *switch*es para el prototipo se presenta en la siguiente tabla:

Especificación	**Requerimiento Mínimo**	**D-LINK DES-1008A**
Características Generales	Administrable	No
	Factor de Forma	Externo - No expansivo
	Memoria RAM	57 KB
Características Funcionales	Control de enlace de datos	CSMA/CD
	Número de puertos	8
	Puertos Ethernet	10 Mbps (half duplex) 20 Mbps (full duplex)
	Puertos Fast Ethernet	100 Mbps (half duplex) 200 Mbps (full duplex)
Características Adicionales	Ajuste de puertos	Auto ajuste MDI/MDIX para todos los puertos
	Instalación	Plug and play

Tabla 3.11 Características del *switch* [**46**]

Se incluye por lo tanto dentro del prototipo la utilización del *switch* D-LINK DES-1008A cuyas características son suficientes para cubrir las necesidades dentro de la red local. El *switch* proporciona 8 puertos suficientes para los *host*s dentro de la LAN y cada puerto es de tipo Ethernet / Fast Ethernet que proporciona la velocidad de transmisión adecuada para el diseño.

3.3.1.3.2 Características del sistema Operativo Inter Networking

En base a la selección de equipos de enrutamiento es necesario describir los requerimientos de *software* dentro del dispositivo. Entre los requerimientos del sistema operativo *internetworking* del *router* Cisco se encuentran:

- Soporte de IPv6.
- Soporte de NAT.
- Soporte para tunelización de IPv6 sobre IPv4.

La versión 12.3 de Cisco cumple con los requerimientos mínimos necesarios. Adicionalmente entre los requerimientos que cumple este sistema operativo se destacan la compatibilidad con versiones anteriores y la adición de nuevas soluciones que se muestran en la tabla 3.12.

Parámetro	Características y Beneficios
Enrutamiento	Mejoras a protocolos de enrutamiento como EIGRP, BGP y OSPF.
Servicios IP	Capacidades adicionales para servicios de Traducción de direcciones de Red (NAT)
IPv6	Soporte para *Simple Network Management Protocol* (SNMP) y traducción de direcciones IPv4 e IPv6

Tabla 3.12 Resumen de Características del IOS 12.3(10)

Se detalla a continuación las ventajas de la versión del IOS que son de interés y mantienen relación con algunos de los requerimientos del prototipo.

- ***NAT***

Incorpora un subsistema NAT que provee una forma alternativa de configurar la traducción. Se encuentran disponibles más opciones de despliegue para proveedores de servicio que ofrecen servicios basados en arquitecturas modernas como MPLS. Además se reduce la complejidad en las configuraciones en donde se requiere NAT.

- ***IPv6***

Se incluye el *Netwok Address Translation-Protocol Translations* (NAT-PT) que permite la traducción de paquetes entre redes de tipo IPv4 puro e IPv6 puro. El protocolo NAT-PT traduce los encabezados IP de las direcciones de red involucradas en el proceso así como los puertos origen y destino de ser necesario.

Permite mayor rendimiento en entornos en donde la traducción entre IPv4 e IPv6 es necesaria. La versión 12.3(10) del IOS de Cisco incorpora el protocolo SNMP usando transporte IPv6; lo que permite que la administración de la red pueda realizarse desde estaciones de trabajo puramente IPv6, independizando la administración respecto de los entornos IPv4.

3.3.2 DISEÑO DE LA CENTRAL IP PBX

La central PBX de telefonía IP diseñada consta tanto de elementos de *hardware* como de *software* para poder juntar las necesidades planteadas en el prototipo.

Puesto que el prototipo de red debe brindar varios servicios de telefonía IP y a la vez trabajar con dos protocolos de Internet de versiones distintas, es necesaria una descripción de la versión de Asterisk a escoger para cumplir dichos parámetros; así como una descripción de las configuraciones necesarias para cumplir con los requerimientos del prototipo descrito.

3.3.2.1 Solución de Software para Telefonía IP [17]

En el *software* Asterisk se presentan dos tipos de versiones: estándar y LTS (*Long Term Support*).

Las versiones estándar son aquellas que se mantienen por un corto periodo de tiempo (1 año de soporte completo y un año adicional para actualizaciones de seguridad); mientras que las versiones LTS tienen un soporte completo por 4 años más 1 año para actualizaciones de seguridad.

La tabla 3.13 muestra las versiones de Asterisk en la actualidad:

Versión	Tipo	Fecha de Liberación	Actualizaciones de Seguridad	End Of Life EOL
1.2.X		21/11/2005	07/08/2007	21/11/2010
1.4.X	LTS	23/12/2006	21/04/2011	21/04/2012
1.6.0.X	Standard	01/10/2008	01/05/2010	01/10/2010
1.6.1.X	Standard	27/04/2009	01/05/2010	27/04/2011
1.6.2.X	Standard	18/12/2009	21/04/2011	21/04/2012
1.8.X	LTS	21/10/2010	21/10/2010	21/10/2015
10.X	Standard	15/12/2011	15/12/2012	15/12/2013

Tabla 3.13 Versiones de Asterisk [**17**]

Es importante destacar que las versiones de la 1.2x a la 1.6.x soportan IPv6 únicamente en ámbito experimental (es decir han sido experimentadas con este protocolo pero aún no se garantiza su funcionamiento adecuado); por otro lado las versiones de la 1.6.2.x a la 10.x soportan IPv6 de manera completa, esto permite trabajar funcionalmente con este protocolo.

En el prototipo a implementar se escoge como servidor de telefonía IP la versión 1.8.12.2 de Asterisk, esta versión se muestra como la más estable al momento de la implementación y además con la característica de soporte para IPv6 de manera completa, por lo tanto se adapta a los requerimientos para el prototipo de red.

3.3.2.2 Estructura de la Central

Asterisk es una aplicación que simula una central telefónica por *software*, el *software* permite conectar un número establecido de extensiones (representación de los diferentes teléfonos y softphones) para realizar llamadas a nivel interno e interconectarse con redes externas como la PSTN.

A continuación se describe la forma de configurar Asterisk en base a sus archivos o ficheros de configuración, además se explica el procedimiento básico y los primeros pasos para completar dicha configuración.

Se resume en la tabla 3.14 los archivos de configuración y los parámetros que se permiten integrar en la central con la edición de dichos archivos:

Fichero	Ubicación	Descripción
sip.conf	*/etc/asterisk/*	Define variables generales, servidores y clientes bajo el protocolo SIP.
iax.conf	*/etc/asterisk/*	Define canales para servidores y clientes bajo el protocolo IAX.
extensions.conf	*/etc/asterisk/*	Estructura el plan de marcado y gestiona las conexiones de la PBX.

Tabla 3.14 Archivos de configuración incluidos en el diseño de la central

3.3.2.2.1 Configuración de Asterisk para trabajar en IPv6 [47]

La versión 1.8 de Asterisk soporta el protocolo IPv6 para tráfico SIP y RTP. El soporte se consigue mediante la incorporación de una nueva API, ésta interfaz fue creada para aplicaciones que llevan direcciones IPv6; las características y estructura de ésta API para conectividad IPv4 e IPv6 son especificadas en el RFC 3493 y el RFC 3542.

La nueva API hace que la aplicación sea independiente de la versión, puesto que el *stack* elige que versión de IP usará para la conexión; en el proceso un puerto de alguna aplicación llega a obtener una IP ignorando la versión de la misma.

La API trabaja con un *socket* que contiene una lista enlazada de direcciones, de esta manera la API se integra en las aplicaciones de *software* para traducir IPv4 a IPv6 y viceversa mediante la aplicación de un mecanismo de conversión IP; la central efectúa la conversión de estas direcciones mediante el casting implementado en el *socket*.

Asterisk realiza un *transcoding* y conversión de protocolo mediante sockets para interpretar los dos protocolos de Internet. Por lo tanto Asterisk se introduce en el medio de la comunicación; pone su IP en el SDP (*Session Discovery Protocol*) que se utiliza para describir sesiones *multicast* en tiempo real, así como para invitaciones, anuncios y cualquier otra forma de inicio de sesiones.

Para que Asterisk 1.8.12 pueda trabajar conjuntamente con los protocolos IPv4 e IPv6 es necesario configurar el archivo ***sip.conf*** (en el cual se configura los canales a utilizar para trabajar con el protocolo de VoIP SIP).

La configuración se ejecuta en la ruta ***/etc/asterisk/sip.conf*** configurando en sus primeras líneas aspectos relacionados al tipo de protocolo que puede manejar.
En la figura 3.6, se observa un ejemplo de la configuración a aplicar para aceptar uno de los protocolos independientemente o los dos protocolos conjuntamente.

udpbindaddr value	Description
192.168.100.50	Bind to a specific IPv4 address.
2001:db8::1	Bind to a specific IPv6 address
0.0.0.0	Bind to all IPv4 addresses on the system.
::	Bind to all IPv4 *and* IPv6 addresses.

Figura 3.6 Configuraciones de sip.conf para protocolos de Internet [**47**]

Los parámetros más importantes a configurar en el archivo *sip.conf* en la sección General son los que se muestran a continuación:

```
alloguest=no              ;Para que no se permitan llamadas de
                          usuarios no autenticados

udpbinaddr= ::            ;Colocamos esta configuración para
                          que se acepten tanto direcciones
                          ipv4 como direcciones IPv6

nat= yes                  ;Impone siempre el uso del parámetro
                          rport y envía el flujo audio/video
                          por el mismo puerto utilizado por el
                          dispositivo remoto.

allow=gsm,alaw,ulaw       ;Establecemos los codecs de audio
                          que asterisk puede usar.
```

Existen otros parámetros a configurar en el archivo ***sip.conf*** sin embargo los más importantes para el prototipo implementado son los antes mencionados.

En las extensiones de los canales, se configura los siguientes parámetros:

```
type=friend          ;  Para que la extensión se
                        autentique al servidor asterisk como
                        user (usando el campo From)o como
                        peer (mediante la IP y el puerto)


secret=asterisk2012  ;Establecemos la contraseña de dicha
                        extensión.
host=dynamic         ;Para que se puedan hacer conexiones
                        remotas al servidor con una
                        extensión con IP dinámica.
```

El archivo de configuración completo aplicado al prototipo se presenta de una manera detallada en el ***anexo B.***

3.3.2.2.2 Estructuración del Plan de Marcación [47]

En el fichero de configuración extensions.conf se configura el plan de marcación y el comportamiento de todas las conexiones a través de la PBX; en el plan de marcación se controla como se gestionan y encaminan las llamadas entrantes y salientes del sistema Asterisk

Para una mejor administración se diseñó el plan de marcación separado en varios contextos, los cuales se dividen de acuerdo al protocolo, la función o la tecnología con la cual trabajan. De esta manera se tiene un plan de numeración más organizado tanto para el administrador como para el usuario.

En base a esta configuración se destina a los usuarios que están detrás de una red IPV4 las extensiones de la 200 a la 299, ubicadas en el contexto llamado **redsip**; los usuarios que están detrás de una red IPv6 se encuentran asignados en las extensiones de la 400 a la 499 en el contexto **redipv6**. Mientras tanto los

usuarios detrás de una red IPv4 pero que utilizan el protocolo IAX2 ocupan las extensiones de la 300 a la 399 en el contexto **iaxred.**

Para incluir el uso de teléfonos analógicos se asigna el contexto **analógicos**, en el cual se incluye la extensión 2511. Se tiene además otros contextos no menos importantes como son los de **conferencias, IVR, buzón, directorio** que hacen referencia a algunos de los servicios implementados en la PBX.

Por consiguiente dicha organización permite tanto llamadas internas en la PBX así como salida hacia otras redes o a la PSTN.

En la tabla 3.15 se muestra un resumen de los contextos implementados en la central.

CONTEXTO	EXTENSIONES
Pruebas	500,600,700
Redsip	200-299
Iaxred	300-399
redipv6	400-499
Buzón	98,99
Directorio	97
Analógicos	2511
Conferencias	3500,3501,3502
IVR	Llamadas entrantes de la PSTN
consulta-notas	511,512
Locales	[2-6]xxxxxx
Nacionales	0[2-6]xxxxxx
Celulares	09[7-9]xxxxxxx
internacionales	00x.

Tabla 3.15 Contextos y extensiones del Plan de Numeración

El archivo de configuración ***extensions.conf*** en base al presente diseño se puede observar en el ***anexo B.***

- ***Aplicaciones utilizadas en el plan de marcación***

En el diseño del plan de marcación se utiliza algunas aplicaciones de Asterisk que permiten realizar determinadas funciones como marcar, colgar, entre otras.

Las aplicaciones usadas en el prototipo se configuran dentro del fichero **extensions.conf**; las aplicaciones y el modelo de contexto para su utilización en el prototipo se describen a continuación:

Primero se detalla el nombre del contexto, posteriormente se define qué operación realizar con la extensión marcada.

```
Ejemplo:    [CONTEXTO]
            exten => extensión, prioridad, aplicación
```

La función principal de la central es permitir ejecutar llamadas por lo cual se incluye la aplicación **Dial**. Esto permite realizar una llamada a un canal o dispositivo concreto; se incluyen parámetros como *tipo y nombre del canal así como el tiempo de intento de la llamada.*

```
Ejemplo:  exten => 201,1,Dial (SIP/telefono-${EXTEN},20)
```

Es necesario que al marcar una extensión Asterisk conteste la llamada y espere determinados segundos para ejecutar la siguiente línea de programación, por lo cual se incluye la aplicación **Answer**.

```
Ejemplo:  exten => 201,1,Answer(3)
```

Para informar al usuario determinados mensajes o menús interactivos, se incorpora la aplicación **Playback**. Se utiliza para reproducir archivos de audio y no devolver el control de marcado hasta que se termine de reproducir el archivo.

```
Ejemplo:  exten => 201,1,Playback(demo-congrats)
```

En algunos servicios es necesario reproducir el audio y retornar el control al plan de marcado, por esta razón se introduce el uso de la aplicación *Background*.

Con el propósito de finalizar la utilización de un canal se incluye la aplicación **HangUp**, la cual cuelga el canal y no necesita ningún otro parámetro.

*Ejemplo: exten => 201,1,**HangUp**()*

Para incluir interactividad en el plan de marcado es necesario conocer el estado de las variables de un canal o una llamada; con este propósito se incluye la aplicación **NoOp** o *No Operation* que no realiza ninguna acción pero es de utilidad para realizar pausas y permite saber el estado de una variable.

Los servicios de la central dependen de la extensión ingresada, por ello es necesario redirigir la ejecución de las aplicaciones a un lugar diferente en el plan de marcado; con este objeto se incluye las aplicaciones **Goto** que redirige la ejecución hacia otro contexto, extensión y prioridad.

*Ejemplo: exten => 201,1,**Goto**(redsip,203,1)*

Adicionalmente para acceder a algunos servicios es necesario cumplir con alguna determinada condición, por lo cual se incorpora la aplicación **Gotoif** que tiene un formato condicional para que se produzca la redirección.

Su sintaxis es:

***Gotoif**(<expresion_regular_condicional>?<redirección_si_verdadero:<redireccion_si_falso>)*

Finalmente es de utilidad incluir una aplicación que permita realizar pausas o tiempos de espera para el usuario; por esta razón se agrega la aplicación **WaitExtern** que incorpora pausas al plan de marcado.

*Ejemplo: exten => 201,n,**Waitexten**(5)*

3.3.2.2.3 Configuraciones de seguridad

Se describe el modelo de seguridad dentro del prototipo en base a los requerimientos de seguridad que se describen en la **sección 3.1.3**. Los dos mecanismos de seguridad requieren especificar parámetros propios de un usuario de telefonía IP que se complementan entre si y se explican a continuación:

Autenticación: limita el acceso y registro de entidades dentro del sistema. Se configura dentro del archivo **sip.conf** mediante la edición de los parámetros *permit, deny y secret*.

Se utiliza las líneas "*permit"* y "*deny*" para limitar las direcciones IP que pueden acceder a un canal sip. Este modelo se propone sólo permitir a un conjunto razonable de direcciones IP entrantes alcanzar cada usuario/extensión listado en el archivo sip.conf. La estructura básica de este mecanismo se detalla a continuación mediante un ejemplo:

```
[200]
username=200
context=telefonos
secret=asterisk2012
deny=0.0.0.0/0.0.0.0
permit=172.31.18.0/255.255.0.0
```

En este ejemplo se configura la sección específica de un canal sip asignado dentro del segmento de red local del servidor. El parámetro **deny=0.0.0.0/0.0.0.0** impide el acceso cualquier dirección; posteriormente el parámetro **permit=192.168.1.1/255.255.255.0** concede acceso solamente a aquellos dispositivos que se encuentren dentro de la red local del servidor 173.31.18.0/16.

Adicionalmente se incrementa el nivel de seguridad incluyendo la autenticación para el registro de usuarios SIP; con este objeto se conforma claves seguras para las entidades SIP. Por lo tanto es necesario configurar el parámetro **secret** en el archivo SIP.conf. Para los usuarios SIP del sistema se utiliza la contraseña **"asterisk2012"** para su registro.

Operación: limita el acceso a determinadas extensiones y funciones de la IP PBX para usuarios que ya se han registrado dentro del sistema. Este nivel de seguridad se consigue mediante un control en la habilitación de los contextos al definir al usuario.

En el siguiente párrafo se ilustra un ejemplo de restricción de accesos a extensiones presentes en el dial plan del prototipo mostrado en el **anexo B.3**

```
[PSTN]
exten => _xxxxxxx,1,Dial(DAHDI/1/${EXTEN},20)

[celulares]
exten => _0xxxxxxxxx,1,Dial(DAHDI/1/${EXTEN},20)
```

De esta manera solamente los usuarios SIP e IAX que incluyan los contextos PSTN o celulares podrán tener acceso a dichas extensiones, es decir solamente dichos usuarios pueden establecer llamadas con la PSTN y con teléfonos celulares.

3.3.2.3 Servidores Requeridos para la Central

En el prototipo planteado es necesaria la instalación de cuatro servidores, todos ellos implementados en el Sistema Operativo **Centos 6**. Se requieren los siguientes servidores: servidor de telefonía IP implementado con Asterisk, servidor DNS, servidor de correo electrónico y un servidor de páginas web para poder ingresar mediante otros computadores de la red a la cuenta de correo.

Para garantizar el funcionamiento y la eficiencia cada servidor es recomendable instalar en máquinas independientes, esto permite optimizar los recursos de cada servidor, mejorar la disponibilidad y brindar mayor capacidad de almacenamiento.

Dentro de un entorno empresarial se enfrenta una gran densidad de *hosts* y un alto grado de uso de los recursos, por lo tanto es indispensable dimensionar cada servidor en base a los requerimientos específicos de *hardware*, *software* para satisfacer la demanda de clientes.

Sin embargo el prototipo pertenece a un entorno de pruebas, con condiciones controladas y un ambiente simulado, por lo cual se pueden incorporan todos los servidores dentro un solo computador. Esto se justifica debido a que entre los servidores del prototipo se destacan requerimientos comunes de *software* como el sistema operativo Centos 6, adicionalmente la densidad de *hosts* es pequeña y el

grado de uso de los recursos como servidores, canales y dispositivos finales es reducido puesto que se ajusta a un modelo de pruebas.

Por lo tanto las características de memoria, almacenamiento y procesamiento de un solo computador son suficientes para desempeñar las funciones de central IP PBX y de servidor para cada uno de los servicios detallados a continuación:

3.3.2.3.1 Servidor Asterisk

Este servidor permite a los usuarios realizar llamadas telefónicas internamente, hacia otras redes y comunicarse con la PSTN; permite también acceder a servicios de Telefonía IP como llamadas en espera, transferencia de llamadas, acceso a base de datos de Asterisk, etc.

Además este servidor debido a su arquitectura, características y funcionamiento brinda una estructura funcional y jerárquica. Dicha estructura permite el control de acceso a ciertas aplicaciones y establece permisos a determinados usuarios para alguna actividad en especial (por ejemplo controlar las llamadas celulares así como las internacionales).

La instalación del servidor Asterisk en Centos se detalla en el ***anexo A.2.***

3.3.2.3.2 Servidor de nombres de dominio (DNS) [48]

Un servidor de nombres de dominio DNS traduce una dirección IP en un domino o viceversa.

Un nombre de dominio es más sencillo de recordar que una dirección IP y permite que los clientes puedan acceder fácilmente a su cuenta de correo (para la aplicación de correo de voz).

Para el diseño de este servidor se tienen las siguientes consideraciones:

- El número de dominios, para el prototipo solo se requiere de un dominio para el acceso a la aplicación de correo.
- El nombre de dominio, el cual se denomina como “tesis.telefoniaip.” para guardar relación con la aplicación de correo y sea de fácil manejo para los usuarios.
- Las zonas de autoridad, que definen la jerarquía de los dominios.

La estructuración completa del servidor se muestra en la **sección 3.4.3.1.** junto con los respectivos ficheros de configuración.

3.3.2.3.3 Servidor de páginas web [49]

El servidor de páginas web permite a los *host*s acceder al contenido de páginas web utilizando el protocolo HTTP al cuál se asigna generalmente el puerto TCP 80. El acceso a las páginas web se solicita a través de la URL (*Uniform Resourse Locator*) al servidor web en base a peticiones por parte de los *hosts*.

Es necesario implementar este servidor para poder acceder al contenido de los mensajes de voz, mediante un navegador editando la dirección IP del servidor o el dominio asignado a éste.

En la ruta ***/var/www/html/*** se copia el contenido a visualizarse. En este caso es una carpeta completa que corresponde al servidor de correo.

3.3.2.3.4 Servidor de correo [50]

El servidor de correo permite enviar y manejar mensajes entre usuarios independientemente de la red en que se encuentren conectados.

El servicio de correo es necesario para incorporar funcionalidades de correo de voz y buzón de voz en nuestra central IP PBX; para lograrlo se definen una serie de protocolos, cada uno con una finalidad concreta:

- ***SMTP, Simple Mail Transfer Protocol:*** Es el protocolo que se utiliza para que dos servidores de correo intercambien mensajes.
- ***POP, Post Office Protocol:*** Se utiliza para obtener los mensajes guardados en el servidor y pasárselos al usuario.
- ***IMAP, Internet Message Access Protocol:*** Su finalidad es la misma que la de POP, pero el funcionamiento y las funcionalidades que ofrecen son diferentes.

Finalmente se presentan algunas de las características del servidor de correo para el presente prototipo:

- Gestión de carpetas
- Búsquedas de direcciones

- Servicio de búsqueda en e*mail*s
- Arquitectura de *plug-ins* (permite añadir características no imprescindibles)
- Interfaz de usuario sencilla y potente
- Gestión de *attachments (para archivos adjuntos)*
- Comprobación de correos entrantes cada cierto intervalo de tiempo

Para satisfacer las propiedades descritas e incluir la característica de *software* libre se selecciona el servidor Squirrelmail como servidor de correo para el prototipo. En el proyecto se implementa un servidor de correo en base a Squirrelmail el mismo que se detalla en la **sección de implementación 3.4.3.2**.

3.3.2.4 Recursos para telefonía utilizados en el Prototipo

3.3.2.4.1 Tarjetas para telefonía analógica

En el prototipo diseñado, se utiliza las tarjetas OpenVox FX0-100 REV1.1 y OpenVox FXS-100REV1.1; montadas sobre la placa HERMS DM400A, que permite cuatro puertos FXO o FXS. Las tarjetas analógicas OpenVox son compatibles para ambientes Opensource Asterisk, Elastix, Tribox; y con ranura PCI, por lo que se adaptan perfectamente al servidor que está implementado bajo el *software* Asterisk.

En la figura 3.7, se muestra las tarjetas utilizadas montadas sobre la placa mencionada, la tarjeta de color rojo es una tarjeta de tipo FXO; mientras que la de color verde es una de tipo FXS.

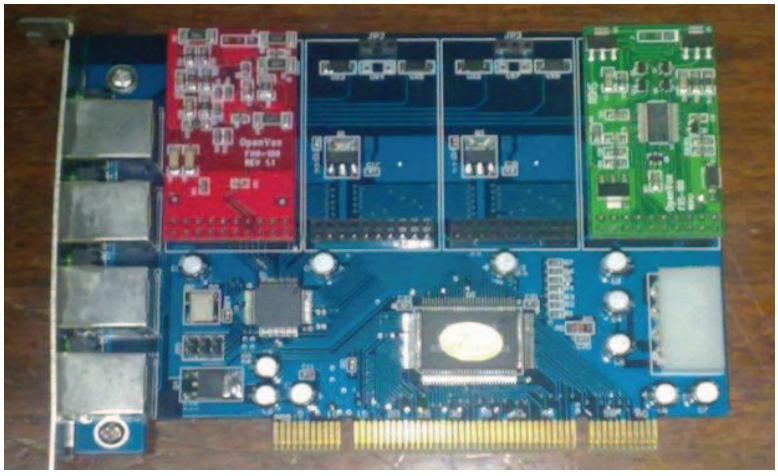


Figura 3.7 Tarjetas FXO Y FXS utilizadas en el prototipo

3.3.2.4.2 Teléfonos IP

Figura 3.8 Teléfono IP D-LINK DPH-150S

En el prototipo diseñado, se incorpora el uso del teléfono D-Link DPH-150S; el cual es un teléfono IP que posee entre sus características más importantes, las siguientes:

- Soporte *Power over Ethernet* (PoE) integrado.
- Comunicación a través de Internet o red LAN.
- 2 Puertos de conexión, permite hacer llamadas y navegar al mismo tiempo.
- Se configura mediante DHCP o mediante IP estática.
- Fácil administración y configuración.
- Soporta únicamente IPv4
- Administrable vía Web y manualmente.
- Soporta los códecs: G711a/u, G729a/b, G723.1
- Multiusuario, soporta cuentas SIP.
- Supresión de silencio.
- Cancelación de eco.
- Generación de ruido de confort.

3.3.2.4.3 Softphones

Dentro del prototipo, se incorpora la utilización de los siguientes softphones:

Zoiper

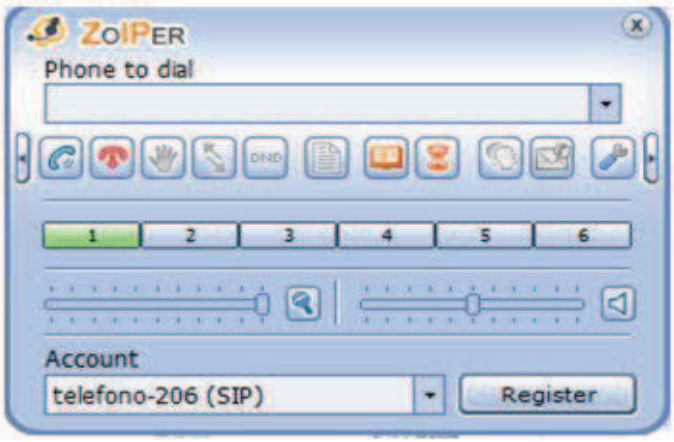


Figura 3.9 Softphone Zoiper

Las principales características de este softphone son:

- Gratuito, se distribuye bajo licencia Freeware
- Soporta los protocolos SIP e IAX2
- Funciona bajo el sistema operativo Windows
- Soporta el protocolo IPv4.
- Fácil administración

Linphone

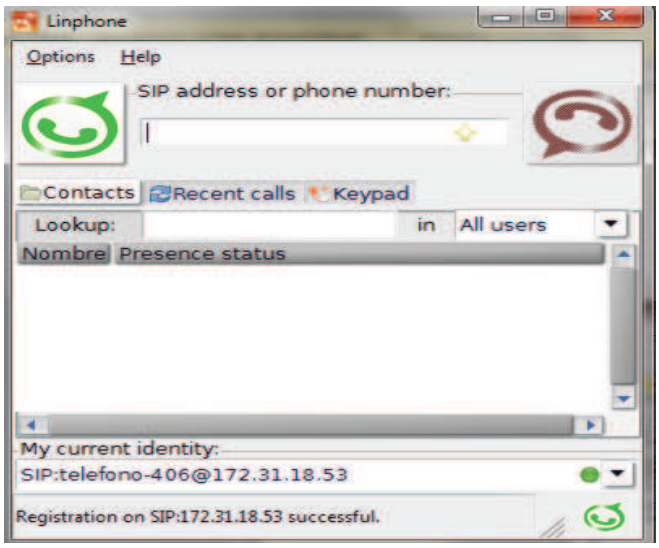


Figura 3.10 Softphone Linphone

Las principales características de Linphone son:

- Agente SIP compatible con RFC3261
- Soporta múltiples llamadas simultáneamente, con funciones de manejo de llamadas: cortar con música, resume, transferencia, etc.
- Audio con los siguientes códecs: speex, G711 (ulaw, alaw), GSM, G722.
- Soporta DTMF (tonos telefónicos) usando SIP INFO o RFC2833.
- Manejo eficiente del ancho de banda.
- Soporta IPv6.

La instalación de los softphones mencionados, se detalla en el ***anexo A.3***

3.3.2.5 Recursos para pruebas de rendimiento de la Central

Para realizar las pruebas de rendimiento de Asterisk es necesario generar una denegación de servicio a la central IP PBX mediante una herramienta que sea capaz de generar tráfico SIP. El manejo de esta herramienta debe permitir enviar paquetes de audio y además proporcionar información que resulte valiosa para evaluar su rendimiento. Entre esta información se incluye el número máximo de llamadas simultáneas que puede soportar el sistema.

Dentro de este escenario de pruebas se evalúan algunas de las alternativas para generar el estado de congestión del servidor, cada una de las cuales se describen a continuación:

- ***Software generador de tráfico***

En el mercado se encuentran excelentes programas que simulan en un ambiente de laboratorio un generador de tráfico de todo tipo de paquetes.

Para este efecto se incluye el generador de tráfico Ostinato que funciona como un generador y analizador de tráfico de paquetes de redes de comunicaciones. Ostinato permite la creación de paquetes personalizados con edición de cualquier campo para diversos protocolos como: Ethernet, 802.3, LLC, VLAN, ARP, IPv4, IPv6, IP Tunneling, TCP, UDP, ICMPv4, ICMPv6, etc. Dentro de las ventajas de este *software* están su libre distribución, pues es de licencia abierta.

Además su amigable interfaz gráfica permite manipular el tipo de paquetes que se pretende generar de acuerdo al protocolo de cada capa.

Sin embargo esta opción se descarta en el prototipo puesto que no cuenta con los protocolos específicos para VoIP tales como SIP e IAX. En la figura 4.35 se presenta la interfaz del *software* Ostinato y los diferentes protocolos que maneja.

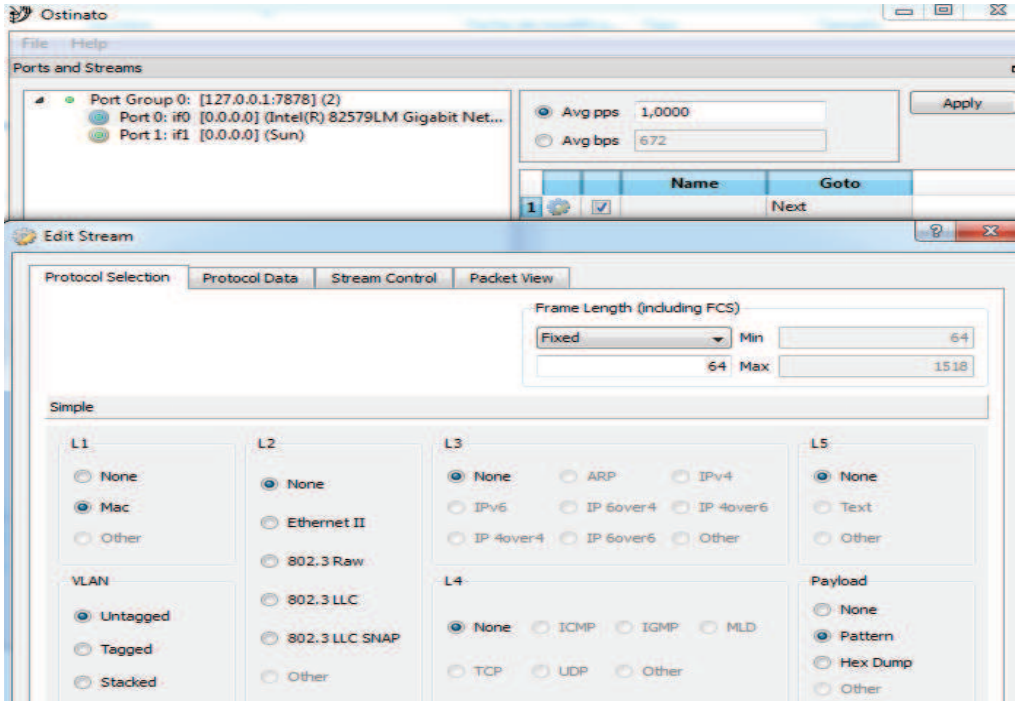


Figura 3.11 Interfaz gráfica del generador de tráfico Ostinato

- ***Generadores de Tráfico en hardware***

Comprenden equipos altamente especializados para generar paquetes de red en condiciones mucho más controladas tales como entornos corporativos y de pruebas. Las características de dichos dispositivos son superiores a los disponibles por *software*, pero con la desventaja de tener un alto costo y una disponibilidad muy limitada en el mercado. Por esta razón se descarta su uso en las pruebas de rendimiento.

Finalmente se menciona una herramienta para generar tráfico de VoIP bajo el protocolo SIP, la misma que se utiliza dentro del prototipo y se describe a detalle a continuación:

- ***SIPP***

SIPP es una aplicación gratuita y libre de testeo, generación de tráfico y de análisis de rendimiento del protocolo SIP. Es capaz de realizar múltiples llamadas de forma simultánea empleando el protocolo SIP. Incluye algunos escenarios de usuarios básicos para realizar pruebas, los mismos que se encargan de establecer y liberar múltiples llamadas SIP con los métodos INVITE y BYE.

Entre las principales características de este *software* están que soporta IPv6, autenticación SIP, retransmisiones UDP, puede enviar tráfico RTP (audio o audio con video). Posee además para su análisis un indicador de estadísticas de las pruebas que se realizan, como por ejemplo: promedio de llamadas, duración y número simultáneo de llamadas, etc.

La ejecución de esta aplicación se realiza mediante una interfaz de línea de comandos en Windows. La interfaz de SIPP se presenta en la siguiente figura:

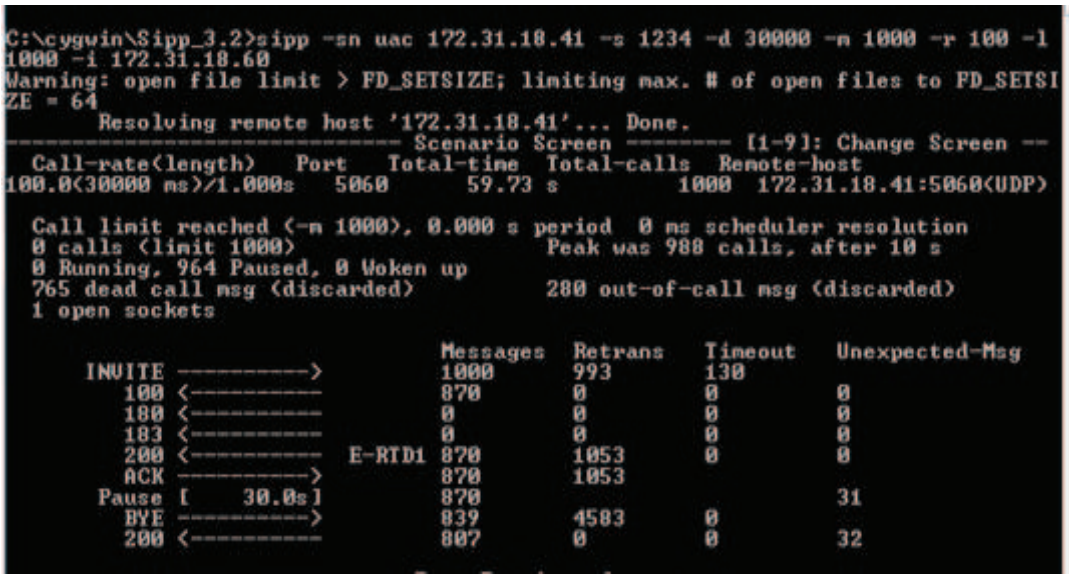


Figura 3.12 Interfaz gráfica del generador de tráfico de VoIP SIPP

3.4 IMPLEMENTACIÓN DEL PROTOTIPO

3.4.1 CONFIGURACIÓN DE LOS EQUIPOS DE RED

Es necesario explicar las modificaciones realizadas en cada uno de los *router*s utilizados en el prototipo; estas se detallan en el **anexo C** que incluye los archivos de configuración de cada dispositivo. La nomenclatura de los dispositivos se basa en la nomenclatura descrita en la topología que se indica en la siguiente figura:

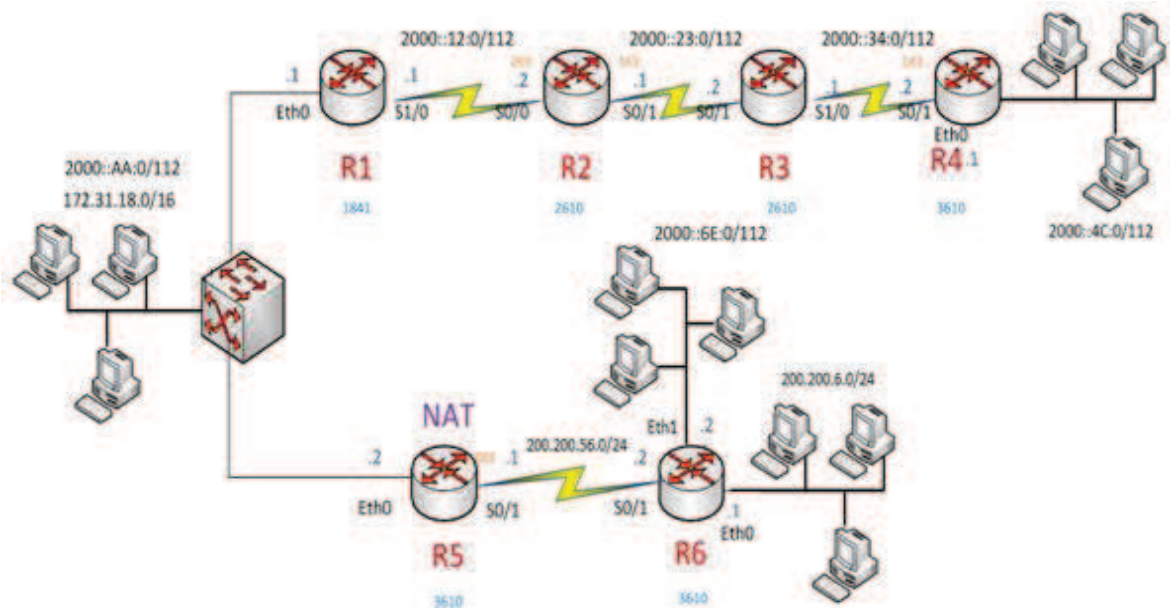


Figura 3.13 Topología para configuración de equipos

3.4.1.1 Configuración básica

Es necesario establecer los parámetros fundamentales para la operación de los dispositivos de encaminamiento tales como los identificadores, criterios básicos de red y de seguridad. La configuración de estos parámetros se detalla en el **anexo C** para cada uno de los *router*s inmersos en el prototipo y se mencionan a continuación.

- Configuración del nombre del dispositivo; se identifica con un nombre a cada uno de los *router*s inmersos en el prototipo.
- Búsqueda de DNS; el prototipo es un ambiente de laboratorio y pruebas en donde no es ideal que cada *router* intente buscar una entrada de DNS para un

nombre que en realidad puede tratarse solamente de un error de escritura. Por lo cual se desactiva la búsqueda de DNS.

- Seguridad; en base a métodos de autenticación y encriptación se añade un nivel de seguridad a la configuración de los equipos. Se incluye en el desarrollo del prototipo los siguientes mecanismos:
  - Establecer una contraseña para acceder del modo usuario al modo privilegiado y tener acceso así a una mayor variedad de configuraciones y comandos.
  - Configurar un mensaje del día que advierte a intrusos que el acceso para disponer del equipo es restringido.
  - Establecer contraseñas para las formas de acceso al equipo tanto con la línea de consola como con las líneas de terminal.
- Habilitar el direccionamiento IPv6; se ingresa al modo de configuración global para admitir el uso y procesamiento de direcciones de tipo IPv6 en el dispositivo.
- Configuración de interfaces; se asigna cada una de las direcciones IP según el esquema de direccionamiento diseñado y su respectiva máscara de subred. Adicionalmente se determina si el equipo necesita proporcionar o no una señal de temporización para finalmente levantar la interfaz mediante un comando de activación.

3.4.1.2 Configuración de enrutamiento

3.4.1.2.1 Enrutamiento Dinámico [38]

Conocido el esquema de direccionamiento asignado en el segmento de red IPv6 puro es necesario habilitar el protocolo de enrutamiento dinámico seleccionado en la **sección 3.3.1.1.2** entre los *routers*. De este modo se puede completar la entrega de paquetes entre las diferentes redes.

Se presenta los *routers* a configurar en el segmento de red IPv6 puro en la siguiente figura:

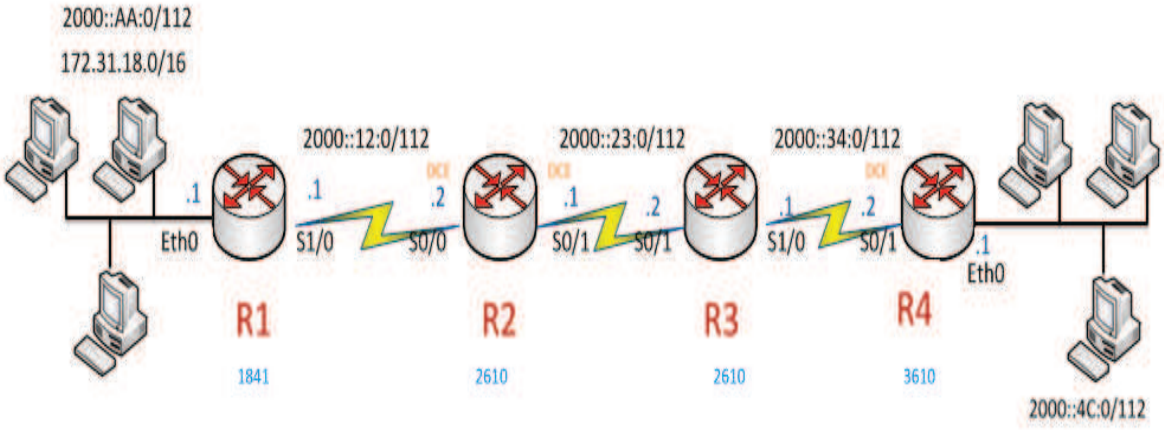


Figura 3.14 Topología para configuración de RIPng

El modelo de configuración para habilitar el protocolo RIPng se basa en el siguiente procedimiento:

- Crear e ingresar al modo de configuración del *router* RIPng. Es necesario identificar el proceso RIP al que se hace referencia durante toda la configuración del protocolo dentro del *router*.

```
Router(config)#ipv6 router rip [proceso]
```

- Habilitar el protocolo RIPng en cada una de las interfaces que intervienen en el proceso de enrutamiento. Se debe especificar el proceso RIP al que pertenece la interfaz.

```
Router(config-if)#ipv6 rip [proceso] enable
```

- Finalmente es necesario verificar que las rutas se incorporen en la tabla de enrutamiento de cada *router*.

La configuración del protocolo RIP en los *routers* R1, R2, R3 y R4; se detalla en el **anexo C.**

3.4.1.2.2 Enrutamiento estático

Conocido el esquema de direccionamiento del segmento de red mixto es necesario implementar un mecanismo de enrutamiento estático que permita encaminar los paquetes de datos. El intercambio de datos se presenta entre las redes conectadas a los dos *routers* a nivel de IPv4. Se presenta los *routers* a configurar en el segmento de red mixto en la figura 3.15.

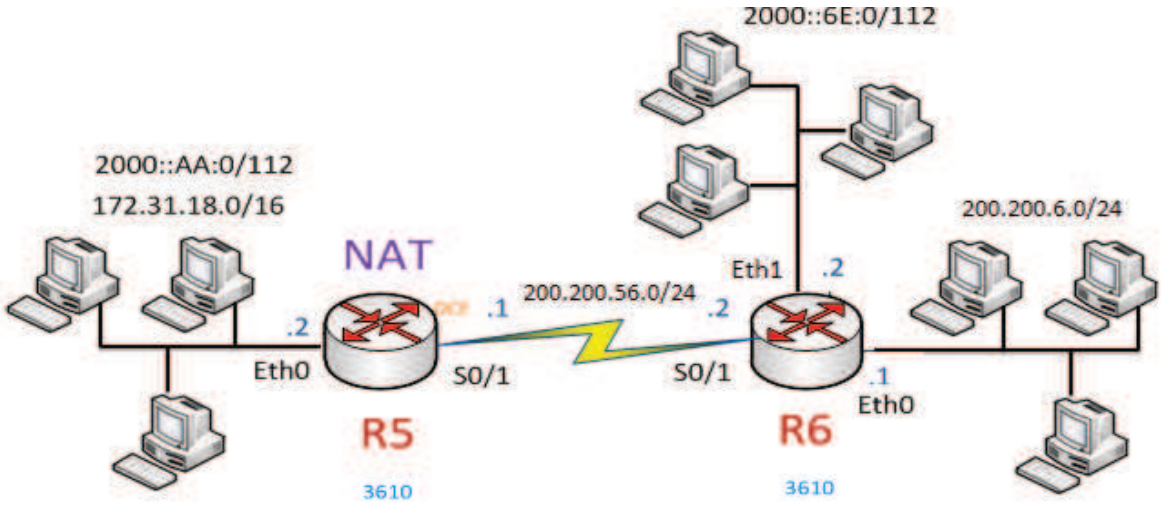


Figura 3.15 Topología para configuración de Enrutamiento dinámico, NAT y Tunelización

Con este propósito se configura un mecanismo de enrutamiento estático de modo que se puedan interconectar las diferentes subredes. Adicionalmente con la implementación de rutas estáticas se aprovechen algunas características de enrutamiento tales como:

- Facilitar el mantenimiento de la tabla de enrutamiento en redes de tamaño moderado.
- El enrutamiento se establece desde y hacia redes de conexión única, por lo que la ruta hacia un destino es siempre la misma y permite además proporcionar mayor seguridad.
- Permite implementar una ruta por defecto, como mecanismo para acceder a una red que no presente una coincidencia más específica con cualquier ruta en la tabla de enrutamiento.
- Se adecua a una topología simple, por lo que se adapta al modelo del segmento de red mixto.

La configuración de este mecanismo de enrutamiento implementado en los *routers* R5 y R6 se detallan en el **anexo C.**

3.4.1.3 Configuración de NAT

Este mecanismo se configura en el segmento de red mixto y tiene por objetivo simular una solución para resolver la escasez de direcciones además de su efecto sobre los servicios de telefonía IP. Los *routers* que se configuran con este mecanismo se presentan en la figura 3.15.

Se escoge el mecanismo de NAT estático debido a la poca densidad de dispositivos en el segmento de red. Este mecanismo es de simple configuración y permite asociar estáticamente una dirección pública a una dirección IP privada.

Para la sección de direccionamiento privado se utiliza direcciones del mismo rango asignado para la red local. Este rango consiste en un pool de direcciones privadas del tipo B con una dirección de red 172.31.0.0/16.

Para la asignación de direcciones públicas se emplea el rango 199.99.9.40 a 199.99.9.62 que permite hasta 23 direcciones públicas; cantidad suficiente para el número de *hosts* con direcciones privadas a ser traducidas.

Este mecanismo se ejecuta en el *router* R5 y su configuración se detalla en el **anexo C**.

3.4.1.4 Configuración de túneles

La tunelización IPv4-IPv6 que se incluye en el segmento de red mixto cumple con los requerimientos para que exista conectividad IPv6 a través de la estructura de red IPv4. Para ello los *routers* en los extremos de la red soportan la doble pila con las versiones 4 y 6 del protocolo IP. Se puede configurar el túnel manualmente debido a que el número de *host* a conectarse mediante el túnel es pequeño. La implementación de este tipo de túnel es simple en comparación a otros mecanismos de interoperabilidad.

Es necesario implementar en cada *router* una interfaz virtual de túnel; tanto en el sentido de R5 a R6 como en sentido R6 a R5. Esto permite definir a la interfaz serial de cada *router* como una interfaz virtual de IPv6. Es necesario especificar claramente la interfaz de salida como la fuente del túnel; y como destino la dirección del siguiente salto.

A continuación se ilustra un ejemplo del modelo de túnel a implementarse en el *router* 6, cuyo sentido va de R6 a R5:

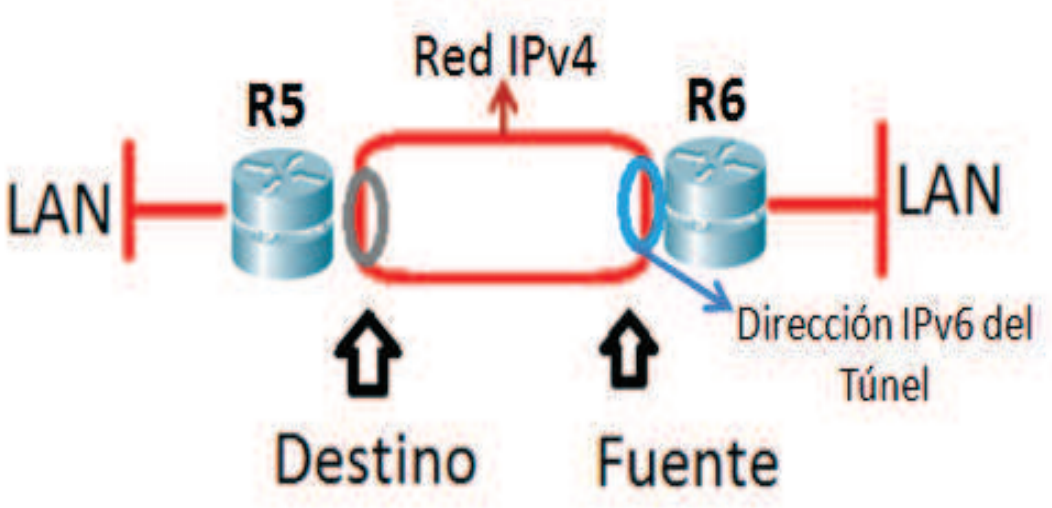


Figura 3.16 Modelo de Túnel IPv6 sobre IPv4

La configuración del mecanismo para tunelización implementada en los *routers* R5 y R6, se detalla en el **anexo C**.

3.4.2 REGISTRO DE USUARIOS EN EL SERVIDOR ASTERISK

Para lograr que los clientes tengan acceso al servidor de telefonía IP tanto en IPv4 como en IPV6 se procede a editar el archivo de configuración ***sip.conf , iax.conf y dahdi.conf.*** de acuerdo al protocolo y tecnología usada respectivamente en el servidor de telefonía IP; logrando así que los clientes se registren en el mismo, creando así una conexión de VoIP mediante los usuarios de este servicio.

Una vez creados los canales de comunicación se procede al registro de los usuarios en el servidor. Es necesario instalar los respectivos softphones y conectar los teléfonos IP así como los analógicos a la red.

La instalación y configuración de los diferentes clientes de telefonía se describen en el **anexo A.3.**

En cada uno de los terminales se debe configurar la dirección IP del servidor y su nombre de usuario para que se pueda registrar al servidor Asterisk. Posteriormente el servidor solicita una contraseña de autenticación para proceder con el registro.

Para comprobar el registro de los usuarios se ejecuta en la consola de configuración de asterisk cualquiera de los siguientes comandos:

```
CLI> sip show peers
CLI> iax2 show peers
CLI> dahdi show peers
```

Estos comandos indican los usuarios registrados así como el puerto y la dirección desde los cuales los clientes están accediendo al servidor.

```
tesis*CLI> sip show peers
Name/username              Host                                  Dyn Forcerpo
t ACL Port     Status
cnt                        172.31.18.41                              N
      5060     OK (1 ms)
telefono-201               (Unspecified)                          D   N
      0        UNKNOWN
telefono-202/telefono-202  200.200.6.10                           D   N
      5060     OK (242 ms)
telefono-203/telefono-203  172.31.18.60                           D   N
      5060     UNREACHABLE
telefono-204               (Unspecified)                          D   N
      0        UNKNOWN
telefono-205/telefono-205  172.31.18.58                           D   N
      5060     OK (35 ms)
telefono-206               (Unspecified)                          D   N
      0        UNKNOWN
telefono-401               (Unspecified)                          D   N
      0        UNKNOWN
telefono-402               (Unspecified)                          D   N
      0        UNKNOWN
telefono-403               (Unspecified)                          D   N
      0        UNKNOWN
telefono-404               (Unspecified)                          D   N
      0        UNKNOWN
telefono-405/telefono-405  2000::4c:30                            D   N
      5060     UNREACHABLE
12 sip peers [Monitored: 3 online, 9 offline Unmonitored: 0 online, 0 offline]
tesis*CLI>
```

Figura 3.17 Resultado del comando ***sip show peers***

En la figura 3.17 se muestra una captura de pantalla de los usuarios SIP registrados en el prototipo implementado.

Los archivos de configuración para el registro de usuarios tales como: sip.conf, iax.conf y dahdi.conf se incluyen en el ***anexo B.***

3.4.3 CONFIGURACIÓN DE SERVIDORES

3.4.3.1 Implementación del servidor DNS

Se implementa este servidor sobre Centos 6 como sistema operativo. Dentro del servidor se traduce la dirección ***172.31.18.41*** en el dominio ***tesis.telefoniaip***.

Los archivos de configuración que se editan para levantar este servicio son:

- `/var/named/chroot/etc/named.conf :`
- `/var/named/chroot/var/named/data/franciscosolis.zone`
- `/var/named/chroot/var/named/data/18.31.172.in-addr.arpa.zone`

En el archivo ***named.conf*** se realiza la configuración para utilizar el servicio.

En el segundo archivo (franciscosolis.zone) se editan las zonas de envío en modo de direcciones IP, registros MX para el servidor de correo.

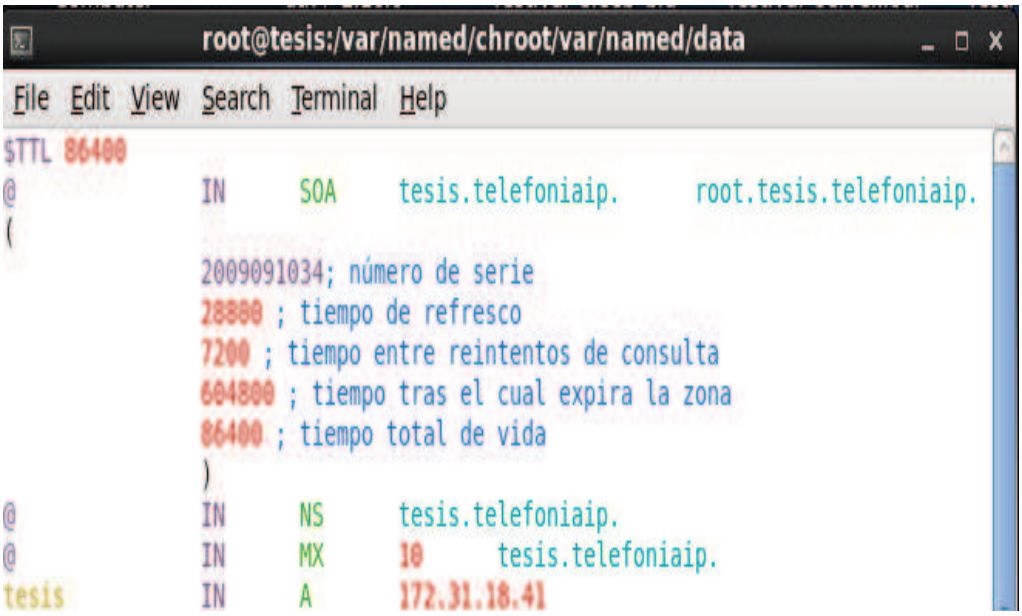


Figura 3.18 Archivo de configuración de las zonas

En el tercer archivo (`18.31.172.in-addr.arpa.zone`), se configura la resolución inversa del dominio.

Figura 3.19 Archivo de configuración de resolución inversa

3.4.3.2 Implementación del servidor de correo Squirrelmail [51] [52]

Para la implementación de este servicio se requiere editar previamente algunos archivos de configuración de Linux que habilitan algunos de los requerimientos de Squirrelmail:

- En el archivo ***/var/named/chroot/var/named/data/franciscosolis.zone*** se edita el registro MX (*Mail eXchange record*). Este registro indica que servidor se encarga del procesamiento del correo electrónico de ese dominio.
- Puesto que Centos 6 utiliza Postfix es necesario también configurar *Postfix* para aceptar correo para nuestro dominio. Esta configuración se la realiza en el archivo ***/etc/postfix/main.cf***.
- Finalmente se tiene que habilitar POP3 en el archivo de configuración ***/etc/dovecot.conf***.

SquirrelMail incluye toda la funcionalidad deseada para un cliente de correo como un robusto soporte MIME, libreta de direcciones y administración de carpetas. Para la configurar Squirrelmail es necesario editar parámetros como preferencias de la organización y herramientas de Servidor. Estas opciones se editan en el

archivo el archivo conf.pl permitiendo configurar los dominios para los servidores SMTP e IMAP.

La configuración detallada de Squirrelmail y los parámetros de edición se incluyen en el **anexo A.4.**

Una vez concluida la instalación y configuración inicial de Squirrelmail se accede a la página de inicio que permite autenticación y entrada al sistema.

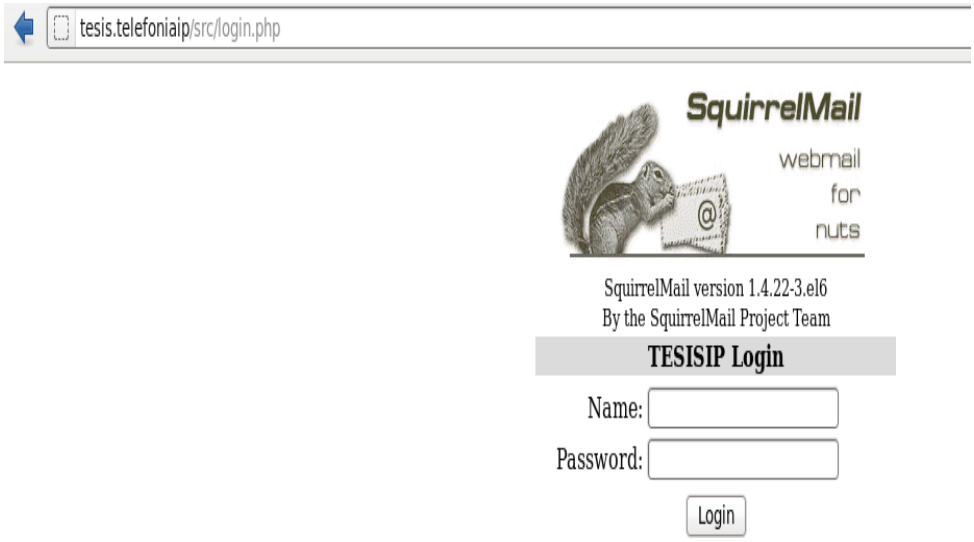


Figura 3.20 Página de inicio de cliente Squirrelmail

La instalación de Squirrelmail como cliente se detalla en el ***anexo A.4.***

3.4.4 SERVICIOS DE TELEFONÍA IP IMPLEMENTADOS [2] [53]

Una vez establecida la configuración básica de la central Asterisk IP PBX es necesario implementar los servicios característicos de una central telefónica. Posteriormente se requiere comprobar su funcionamiento así como el efecto de la inclusión de IPv6 en su desarrollo. El conjunto de servicios a implementar se menciona a continuación:

- Transferencia de llamadas
- Grabación de llamadas
- Conferencia de voz
- Directorio
- Correo y buzón de voz

- Conexión a bases de datos de Asterisk
- Respuesta interactiva de voz (IVR)
- Conectividad desde y hacia LA PSTN

Para la implementación de estos servicios es necesario editar algunos ficheros de configuración especiales de Asterisk. La ubicación, servicio relacionado y función de dichos ficheros se describen en la siguiente tabla:

Fichero	Servicio	Ubicación	Descripción
features.conf.	Transferencia de llamadas	*/etc/asterisk/*	Configura el parqueo de llamadas.
features.conf	Grabación de llamadas	*/etc/asterisk/*	Asigna la función de grabar una llamada a una determinada combinación de teclas (extensión).
meetme.conf	Conferencia de voz	*/etc/asterisk/*	Posibilita crear conferencias de audio.
extensions.conf	Directorio	*/etc/asterisk/*	Permite al usuario conocer una extensión en particular.
Voicemail.conf	Correo y buzón de voz	*/etc/asterisk/*	Configura el buzón de voz para llamadas no contestadas o líneas ocupadas.
extensions.conf	Conexión a bases de datos de Asterisk	*/etc/asterisk/*	Permite la interacción con la base de datos de Asterisk
dahdi_channels.conf	Respuesta interactiva de voz (IVR)	*/etc/asterisk/*	Proporciona la funcionalidad de una operadora automática.

Tabla 3.16 Archivos de configuración adicionales

3.4.4.1 Transferencia de llamadas

Es una función de PBX mediante la cual se puede destinar una llamada a otro usuario específico. Por ejemplo si la llamada es contestada por un usuario A, este mediante una combinación de teclas puede destinar la llamada a otro usuario B.

Este servicio se configura en el archivo: `/etc/asterisk/features.conf`

Para habilitar este servicio en el *dialplan* se debe incluir el siguiente comando en el archivo de configuración ***extensions.conf****:*

```
exten=> extensión,1,Dial(SIP/extensión,tiempo de marcado,tT)
```

dónde:

- ***t:*** Permite al llamado transferir la llamada digitando la secuencia de teclas definida en features.conf (*2 de acuerdo al anexo del archivo de configuración).
- ***T:*** Permite al llamante transferir la llamada digitando la secuencia de teclas definida en features.conf (*4 de acuerdo al anexo del archivo de configuración).

Las configuraciones de los archivos *features.conf* y *extensions.conf* para implementar este servicio se detallan en el **anexo B.**

Los parámetros más relevantes que se configuran en este archivo para la transferencia de llamadas dentro del presente prototipo son los detallados a continuación:

```
allowtransfer=> yes      ;Para permitir la transferencia de
                         llamadas

atxfer=> *2              ; Secuencia de dígitos con la cual
                         se realiza la transferencia asistida
```

3.4.4.2 Grabación de llamadas

Es una función de PBX mediante la cual se puede grabar el audio de una conversación en un archivo. Se puede implementar tanto del lado llamante como del llamado.

Este servicio se configura en el archivo:

```
/etc/asterisk/features.conf
```

Para habilitar este servicio en el *dialplan* se debe incluir en el archivo extensions.conf la siguiente sintaxis:

```
exten=> extensión,1,Dial(SIP/extensión,tiempo de
marcado,wWxX)
```

dónde:

- ***w:*** Permite al llamado empezar la grabación de la llamada digitando la secuencia de teclas definida en features.conf (*1 de acuerdo al anexo del archivo de configuración).
- ***W:*** Permite al llamante empezar la grabación de la llamada digitando la secuencia de teclas definida en features.conf (*1 de acuerdo al anexo del archivo de configuración). En este caso asterisk creará dos archivos de audio, uno por cada interlocutor. Estos archivos de audio se almacenan en ***/var/spool/asterisk/monitor.***
- ***x:*** Permite al llamado empezar la grabación de la llamada digitando la secuencia de teclas definida en features.conf (*3 de acuerdo al anexo del archivo de configuración).
- ***X:*** Permite al llamante empezar la grabación de la llamada digitando la secuencia de teclas definida en features.conf (*3 de acuerdo al anexo del archivo de configuración). A diferencia de las opciones w y W es que en este caso los dos canales (llamante y llamado) se graban en un único archivo de audio.

La implementación de este servicio se detalla en los archivos *features.conf* y *extensions.conf* incluidos en el **anexo B.**

3.4.4.3 Conferencia de voz

Es un servicio de Asterisk mediante el cual es posible que dos o más usuarios puedan interactuar entre sí mediante voz en diferentes extensiones cada uno. Este servicio se habilita editando el archivo:

```
/etc/asterisk/meetme.conf
```

Meetme se apoya en DADHI para generar la sincronización de los canales de audio presentes en la conferencia. Se crea un canal pseudo-DADHI para cada conferencia. Sin DADHI instalado Meetme no funciona.

Se modifica el *dialplan* para utilizar las conferencias. Es necesario editar en el fichero *extensions.conf* la siguiente sintaxis:

```
exten=> extensión,1,Meeteme,(${EXTEN},scM(default))
```

Donde:

- **s:** Con esta opción se activa el menú del cuarto de conferencias para usuarios y administradores. Marcando la tecla asterisco se escuchará el menú.
- **c:** Al entrar en un cuarto de conferencias se le anunciará al usuario el número de personas presentes.
- **M(default):** Si en la conferencia está solo un usuario; éste escuchará la música de espera de la clase *default*.

El archivo de configuración ***meetme.conf*** implementado en el prototipo se muestra en la figura 3.21.

```
File  Edit  View  Search  Terminal  Help
[general]
audiobuffers=32

[rooms]
conf => 3500
conf => 3501,1234
conf => 3502,1234,5678
```

Figura 3.21 Archivo de configuración de ***meetme.conf***

En el **anexo B** se presenta de una manera detallada la configuración del archivo ***meetme.conf.***

3.4.4.4 Directorio

Mediante este servicio se permite obtener la información de las extensiones de usuarios pertenecientes a la PBX. Se accede al servicio llamando a una extensión en particular y editando el nombre del usuario a llamar. Para incorporar este servicio es necesario editar el fichero extensions.conf.

Para hacer uso de esta aplicación en el dialplan simplemente se crea una extensión la cual contenga la aplicación Directorio.

Este directorio se crea automáticamente en la base de datos de Asterisk de acuerdo a los usuarios que se tengan.

Para acceder al directorio los usuarios marcarán la extensión 97. La configuración completa del directorio se muestra en el archivo extensions.conf incluido en el **anexo B.**

3.4.4.5 Correo y Buzón de voz

Este servicio permite escuchar mensajes dejados por usuarios cuyas llamadas que no pudieron ser atendidas. El servicio permite adicionalmente revisar el mensaje en una cuenta de correo. Los parámetros relacionados con el buzón de voz se configuran en el siguiente archivo:

```
/etc/asterisk/voicemail.conf
```

Para utilizar el buzón de voz es necesario editar el dial plan. En el archivo ***extensions.conf*** se direcciona la llamada con el comando **VoiceMail.**

El parámetro más importante en la sección general de este archivo es establecer de manera adecuada la dirección del servidor de correo electrónico (previamente configurado).

En la sección específica del archivo se edita:

- La clave de acceso al buzón de voz
- Nombre del usuario detrás de esa extensión
- Dirección de correo electrónico de quien se encuentra detrás de una determinada extensión.

El archivo de configuración para permitir este servicio en el prototipo implementado se presenta en la figura 3.22.

```
[general]
format=wav49
servermail=asterisk@tesis.telefoniaip
attach=yes
delete=no
maxmsg=100
maxsecs=180
minsecs=2
skipms=3000
maxsilence=10
silencethreshold=128
maxlogins=3
emailsubject=Nuevo mensaje de ${VM_CALLERID}
emailbody=Buenos dias ${VM_NAME}, \n\nHemos recibido un mensaje en su buzon de voz..
emaildateformat=%A, %B %d, %Y at %r

[tesis]
201 => 1234,Diego Garcia,diego@tesis.telefoniaip
202 => 2345,Hector Moyon,hector@tesis.telefoniaip
205 => 3456,David Vaca,david@tesis.telefoniaip
2514 => 5678,Camilo Calle,camilo@tesis.telefoniaip
402 => 6789,Cristian Tintin,tintin@asterisk.test
403 => 5678,Julian Jaramillo,julian@tesis.telefoniaip
```

Figura 3.22 Archivo de configuración ***voicemail.conf***

De acuerdo a este fichero se han designado las extensiones 98 y 99 para acceder al buzón de voz. Adicionalmente a cada *host* se le ha asignado un nombre de usuario y una contraseña para acceder al correo de voz.

Esta configuración se encuentra en el archivo ***voicemail.conf*** detallado en el **anexo B.**

3.4.4.5.1 Funcionamiento del Servicio Voicemail

Para comprobar el funcionamiento de este servicio se llama a un usuario específico que no estará disponible; por lo cual Asterisk solicitará dejar un mensaje al mismo. Cuando el usuario abra su cuenta de correo recibe un mail de notificación que le informa que tuvo una llamada y podrá revisar su mensaje de voz. El procedimiento de acceso al servicio se ilustra en la figura 3.23.

Figura 3.23 Pantalla de autenticación para cliente de correo

El proceso de acceso al mensaje de voz se detalla en la figura 3.23. En este caso se observa que un usuario ha recibido un mensaje del softphone 205 el día Jueves 16 de Mayo de 2013 a las 3:44 pm.

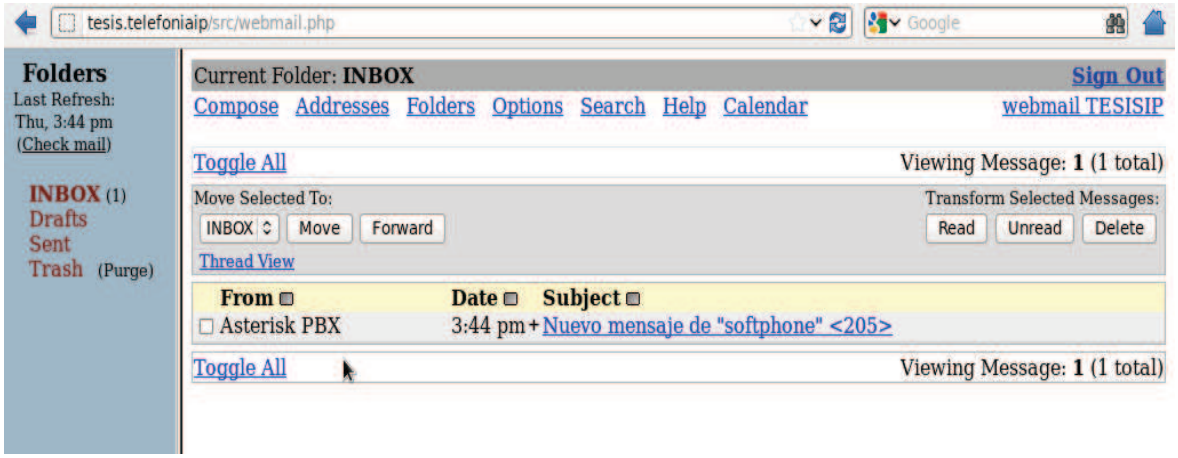


Figura 3.24 Interfaz gráfica de un cliente de correo (squirrelmail)

3.4.4.6 Conexión a base de datos de Asterisk

La base de datos de Asterisk usa la versión 1 de *Berkley* DB que es similar al *registry* de Windows. Este banco de datos puede ser usado por Asterisk para almacenar datos temporales y configuraciones.

Un ejemplo de uso es la transferencia con consulta; en donde si el teléfono está ocupado, este guarda la extensión en una base de datos y permanece reintentando hasta conseguir la conexión.

Los datos son agrupados en familias e identificados con una llave que es única dentro de la familia. Para probar esta funcionalidad de Asterisk se implementó una pequeña base de datos. Los datos se ingresan a la base de datos de Asterisk mediante consola.

Para el caso del prototipo se implementa como ejemplo de aplicación una base de datos para el registro de calificaciones de un profesor que permite además la consulta de las calificaciones por parte de los alumnos.

Un profesor ingresa una nota y el estudiante llama a una extensión en particular para escuchar la nota obtenida en el periodo.

Los datos que se ingresan a la base de datos de Asterisk ASTDB (*Asterisk Data Base*) se almacenan en el archivo astdb, el cual se encuentra en la siguiente ruta:

/var/lib/asterisk/astdb

Los datos empleados para este ejemplo que se ingresan mediante la consola de Asterisk se presentan en la Tabla 3.17:

Alumno	Cédula de Identidad
Érica Amanta	1708070605
Luis Amagua	1709080706
Carlos Cargua	1710090807
Darío Pilataxi	1711100908
Pablo Paguay	1712111009
Robert Navas	1713121110
Mario Gómez	1714131211
Alex Criollo	1715141312
Andrés Cárdenas	1716151413

Alumno	Nota
Érica Amanta	8
Luis Amagua	6,5
Carlos Cargua	4,8
Darío Pilataxi	5,5
Pablo Paguay	5
Robert Navas	7,1
Mario Gómez	7,3
Alex Criollo	5,5
Andrés Cárdenas	7,6

Profesores	Clave
Francisco Solis	54321
David Vaca	12345

Tabla 3.17 Datos a ingresar en Asterisk

La creación de una base de datos en Asterisk obedece a la siguiente estructura:

- La base de datos se organiza de acuerdo a una Familia (En este caso las familias serán Alumnos Profesores y Notas), a un Key (llave) y a un valor predeterminado a esta llave (para el caso de esta aplicación el key sería el nombre del alumno o profesor y su valor sería su número y clave respectivamente).

- Una vez bien identificados estos campos en la consola de Asterisk (CLI) se procede a editar uno a uno o de manera conjunta, los valores que estarán en la base de datos mediante el comando **database *put*** así:

```
CLI> database put familia llave valor
```

Un ejemplo de la forma de ingresar los datos de la tabla 3.24 se describe en base a la ejecución de la siguiente línea de comando:

```
CLI> database put profesores David-Vaca 12345
```

Para acceder a la información ingresada en la base de datos es necesario editar el archivo extensions.conf. Dentro del *dialplan* se establece una extensión para la utilización de los registros de la base de datos.

Las funciones utilizadas para tener acceso a la base de datos son:

- ***DB (Familia/llave):*** Mediante esta función Asterisk obtiene el valor de la llave de la familia indicada.
- ***DB_EXISTS (Familia/valor):*** Utilizada para verificar si existe un determinado valor en la base de datos.

Estas funciones son parte de Asterisk y se editan en el archivo de configuración ***extensions_database.conf*** o en alguna macro que se le asocie al número de extensión para gestionar la base de datos.

Dentro del archivo de configuración ***extensions_database.conf*** se asigna la extensión 511 para el ingreso de notas por parte de un profesor. Dentro del mismo archivo se asigna la extensión 512 para la consulta de notas por parte de los alumnos.

Para mostrar la base de datos de Asterisk desde consola se ejecuta el siguiente comando:

```
CLI> database show
```

En la figura 3.25 se tiene el resultado de este comando en el sistema:

```
tesis*CLI> database show
/Alumnos/1708070605                   : Erica-Amanta
/Alumnos/1709080706                   : Luis-Amagua
/Alumnos/1710090807                   : Carlos-Cargua
/Alumnos/1711100908                   : Dario-Pilataxi
/Alumnos/1712111009                   : Pablo-Paguay
/Alumnos/1713121110                   : Robert-Navas
/Alumnos/1714131211                   : Mario-Gomez
/Alumnos/1715141312                   : Alex-Criollo
/Alumnos/1716151413                   : Andres-Cardenas
/Alumnos/Alex-Criollo                 : 1715141312
/SIP/Registry/telefono-202            : 200.200.6.10:5060:3600:telefono-202:sip:telefono-202@200.200.6.10:5060
/SIP/Registry/telefono-203            : 172.31.18.60:5060:3600:telefono-203:SIP:telefono-203@172.31.18.60;line=ae
a944a5bc12ab5
/SIP/Registry/telefono-205            : 172.31.18.58:5060:3600:telefono-205:sip:telefono-205@172.31.18.58:5060
/SIP/Registry/telefono-405            : [2000::4c:30]:5060:3600:telefono-405:SIP:telefono-405@[2000::4c:30];line=
641fcf0bb2acb7c
/dundi/secret                         : KODTfCKlKV3BbY1kgTa2IA==
/dundi/secretexpiry                   : 1367527844
/notas/1708070605                     : 8
/notas/1709080706                     : 0
/notas/1710090807                     : 0
/notas/1711100908                     : 0
/notas/1712111009                     : 0
/notas/1713121110                     : 0
/notas/1714131211                     : 0
/notas/1715141312                     : 0
/notas/1716151413                     : 0
/profesores/David-Vaca                : 12345
26 results found.
tesis*CLI>
```

Figura 3.25 Resultado del comando ***database show***

3.4.4.7 Respuesta Interactiva de Voz

Consiste en un servicio de central telefónica que es capaz de recibir una llamada e interactuar con el usuario. La interactividad se consigue a través de grabaciones de voz y reconocimiento de respuestas por parte del usuario mediante teclas digitadas en el teléfono. Para configurar el servicio de IVR se debe configurar el fichero **extensions.conf** en donde se agregará las respectivas extensiones, la respuesta de la operadora automática del IVR y las funciones como resultado de las extensiones ingresadas por el usuario.

Para comprobar esta funcionalidad de Asterisk se ha diseñado un IVR en modo de contestadora automática. Este servicio de contestadora re direcciona las

llamadas de un usuario en particular, editando la extensión del mismo (para llamadas entrantes de la PSTN).

Asterisk acepta archivos de sonido en formato .mp3 que se pueden integrar en el desarrollo del IVR. Se procede a grabar un archivo con este formato para proporcionar al IVR un aspecto más real, similar a un entorno de oficina.

Para poder utilizar el archivo de audio debe ser guardado en el directorio de sonidos de Asterisk. El directorio se encuentra en la ruta:

/var/lib/asterisk/sounds.

En la figura 3.26 se presenta el diagrama de flujo que detalla la estructura del IVR

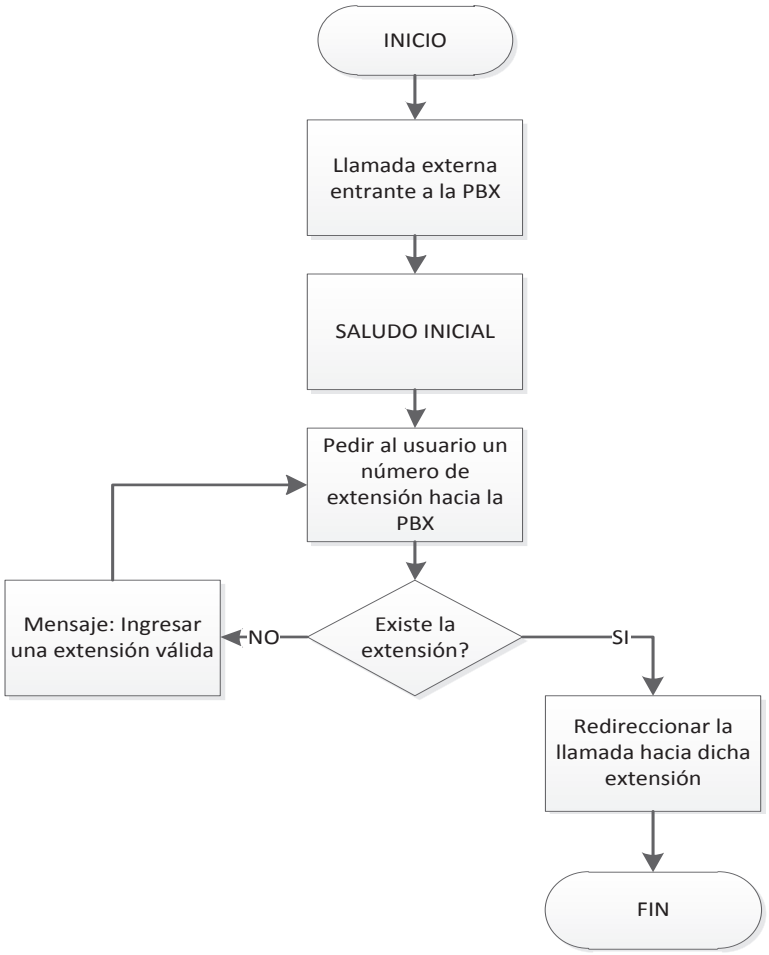


Figura 3.26 Diagrama de flujo del IVR

La configuración del IVR se detalla en el fichero **extensions.conf** incluido en el **anexo B.**

3.4.4.8 Conectividad desde y hacia la PSTN

Para poder conectar la central implementada desde y hacia la PSTN es necesario tener una línea telefónica. Adicionalmente se requiere una tarjeta de telefonía analógica con por lo menos un puerto FXO y un puerto FXS. Finalmente es necesario configurar la conectividad en el servidor Asterisk.

La tarjeta de telefonía analógica utilizada para la conectividad, es la que se detalla anteriormente en la **sección 3.3.2.4.1.**

Para integrar la tarjeta de telefonía analógica y la línea troncal de la PSTN es necesario seguir el procedimiento que se detalla a continuación:

3.4.4.8.1 Instalación y configuración de la tarjeta OpenVox FX0-100 REV1.1 y OpenVox FXS-100REV1.1

Para la instalación de las tarjetas se procede de la siguiente manera:

- Conectar la tarjeta a un puerto PCI/PCIe disponible en el computador.
- Descargar de las fuentes de dahdi desde www.asterisk.org, el módulo y además las herramientas adicionales como libpri.
- Instalar los módulos.

3.4.4.8.2 Configuración del módulo Dahdi

Se edita el archivo de configuración ***chan_dahdi.conf*** y el archivo ***dahdi_channels.conf.***

En el archivo *chan_dahdi.conf* se incluye el archivo *dahdi_channels.conf* mediante el comando:

```
#include chan_dahdi.conf
```

En la figura 3.27 se puede observar los cambios que se realizan en el archivo ***dahdi_channels.conf*** ya que este archivo se crea con una configuración por defecto al instalar la tarjeta.

```
signalling=fxs_ks
callerid=asreceived
;group=0
group=5
;context=from-pstn
context=telefonos
channel => 1
callerid=
group=
context=default

;;; line="4 WCTDM/4/3 FXOKS"
signalling=fxo_ks
callerid="Channel 4" <2514>
mailbox=2514@tesis
group=5
context=telefonos
channel => 4
callerid=
mailbox=
group=
context=default
```

Figura 3.27 Archivo de configuración ***dahdi_channels.conf***

En la figura 3.25 se debe considerar que al referirse a **fxs_ks,** se trata de un módulo fxo y lo que indica como **fxo_ks** se refiera a un módulo fxs.

3.4.4.8.3 Configuración de llamadas hacia la PSTN

Para realizar llamadas salientes por la interfaz FXO que está configurada por el canal 1 en el archivo ***extensions.conf*** se especifica la extensión a llamar.

En este caso se puede realizar llamadas locales, llamadas nacionales, llamadas celulares y llamadas internacionales de acuerdo a los contextos que se tiene. Para tener salida por la interfaz simplemente se tiene que especificar la tecnología que se usa en este caso ***dahdi*** y el canal por el cual sale.

Para incorporar estas llamadas debe incluirse al dialplan en el fichero **extensions.conf** las siguientes líneas de programación:

```
exten => _xxxxxxx,1,Dial(Dahdi/1/${exten})

same => n,HangUp()
```

3.4.4.8.4 Configuración de llamadas desde la PSTN

El proceso para recibir las llamadas entrantes por la línea conectada a la interfaz FXO es algo complejo comparado a la interfaz FXS.

Por defecto todas las llamadas entrantes desde un canal FXO entran por la extensión “s”; por lo que se debe crear un contexto en el archivo **extensions.conf** para este canal.

Es necesario incluir en el *dialplan* del fichero extensions.conf la siguiente sintaxis:

```
[FXO]
exten => s,1,DIAL(DAHDI/1)
exten  => s,2,Hangup()
```

La configuración de los ficheros relacionados a la conectividad con la PSTN, se detallan en el **anexo B.**

3.4.5 IMPLEMENTACIÓN DEL ESCENARIO PARA EL ANALISIS DE RENDIMIENTO Y DENEGACIÓN DE SERVICIO

El escenario sobre el cual se desarrolla esta prueba consiste en un cliente SIP, el cual es una PC en la que se encuentra instalado el *software* generador de tráfico, el mismo que estará en red con el servidor de telefonía IP para realizar el ataque.

La prueba se realiza desde la IP origen 172.31.18.60 (*host* en la red del servidor de telefonía) hacia el *host* destino 172.31.18.41 (servidor de telefonía IP).

El ambiente de red empleado para estas pruebas se muestra en la figura 3.28.

Figura 3.28 Entorno para la realización de la Denegación de Servicio

Una vez que el servidor colapsa, los clientes reales del sistema, serán incapaces de generar siquiera una sola llamada.

Una vez instalado el *software* generador de tráfico SIP en el cliente, se tiene que configurar un canal SIP, así como una extensión en el servidor de telefonía IP; para que el mismo acepte la llegada de las peticiones que realizará el cliente SIPP. En la figura 3.29 se tiene la configuración aplicada para este propósito:

```
[sipp]
type=friend
context=sipp
host=dynamic
user=sipp
insecure=invite,port
canreinvite=no
disallow=all
allow=ulaw
allow=alaw
```

Figura 3.29 Configuración del canal SIPP (archivo sip.conf)

En este caso, al cliente no se le solicita autenticación para poder registrarse en el sistema y realizar las llamadas; además se ignora el puerto por el cual se generen las llamadas (insecure= invite, port).

La extensión designada para realizar esta prueba es la ***1234***, la cual reproduce un audio de llamada propio de Asterisk y luego cierra la llamada.

```
[sipp]
exten => 1234,1,Answer()
same => n,While(1)
same => n,Background(demo-instruct)
same => n,HangUp()
```

Figura 3.30 Configuración de la extensión para el canal SIPP (extensions.conf)

CAPÍTULO 4

PRUEBAS Y RESULTADOS

En este capítulo se describen las pruebas realizadas sobre el sistema implementado para verificar el funcionamiento de las aplicaciones de telefonía IP y los efectos característicos que se producen sobre estas comunicaciones tales como el retardo, jitter, ancho de banda y pérdida de paquetes.

En primera instancia se verifica la conectividad en todo el prototipo mediante pruebas de diagnóstico entre ambientes IPv4, ambientes IPv6 y en ambientes mixtos. Se verifica el establecimiento de llamadas mediante un análisis de los paquetes de voz sobre los protocolos SIP, IAX y RTP. Adicionalmente se realiza el análisis de los paquetes IPv4 e IPv6.

Las pruebas para la medición de los efectos sobre las comunicaciones se realizan mediante el *software* wireshark como analizador de protocolos. El *sniffer* permite analizar el contenido de paquetes, obtener estadísticas de ancho de banda retardos y de *jitter*; así como también monitorear el proceso de intercambio de mensajes durante el establecimiento y la ejecución de una llamada.

Con el objetivo de valorar la pérdida de paquetes se realizan pruebas de rendimiento de la PBX. Se realiza una denegación de servicio mediante llamadas simultáneas generadas a través de *software* para llegar a un estado de congestión y así poder medir los paquetes perdidos utilizando el *sniffer*.

4.1 PRUEBAS DE CONECTIVIDAD

4.1.1 PRUEBAS DE RED

Para diagnosticar el estado de la conectividad de red tanto a nivel local como a nivel remoto se hace uso de las herramientas ***ping y tracert***. Estos comandos se encuentran incluidos en los sistemas operativos Windows y Linux; además poseen la característica de ser compatibles con direcciones IPv4 e IPv6.

Las pruebas se llevan a cabo con *host*s incluidos en el siguiente diagrama de topología:

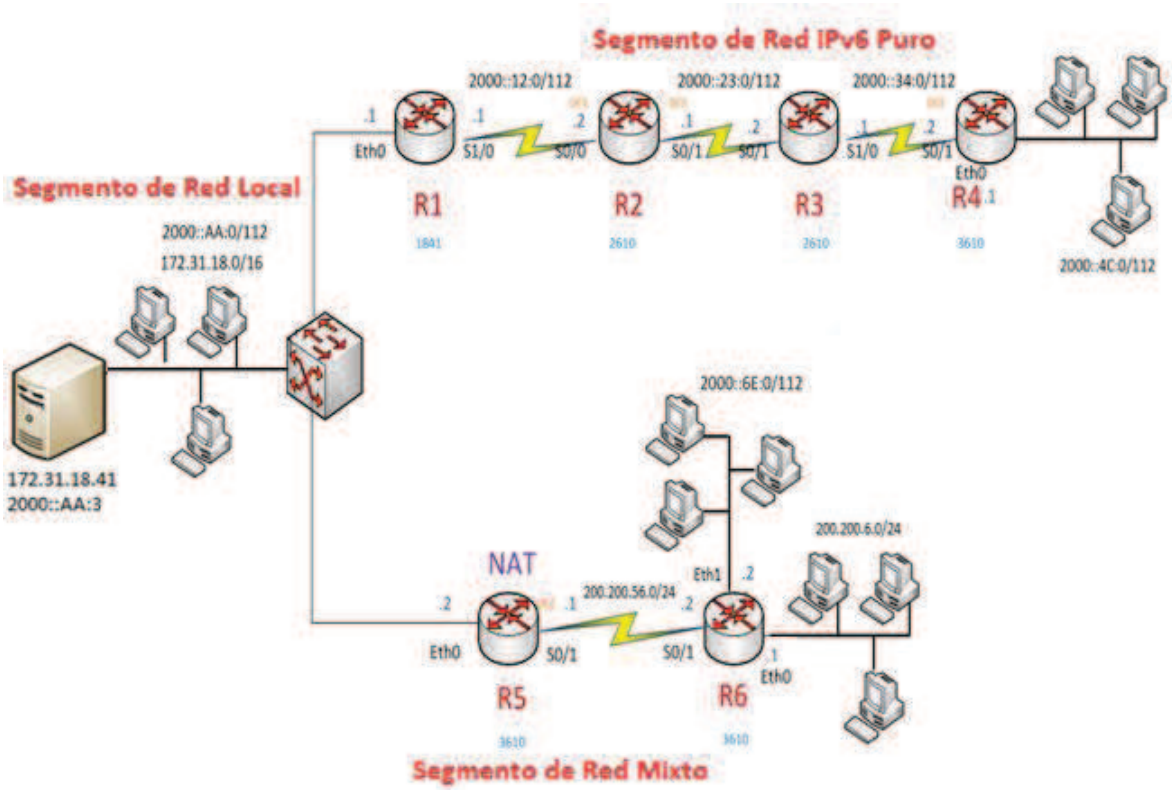


Figura 4.1 Esquema del ambiente de pruebas

4.1.1.1 Conectividad de red remota IPv4 pública

Se verifica la conectividad de la red pública 200.200.6.0 mediante el comando ***ping*** a la dirección del servidor 172.31.18.41. En este caso para que las redes externas a la del servidor local puedan acceder al mismo, deben atravesar la infraestructura en donde se incorporó NAT estático. Los *hosts* externos no pueden ver la dirección original del servidor, sino la dirección mapeada correspondiente a la dirección pública 199.99.9.33 que representa al servidor local.

Para la ejecución de la prueba el *host* origen posee la dirección 200.200.6.2 y el destino la dirección 199.99.9.33 (servidor). Los resultados de la prueba se muestran en la figura 4.2.

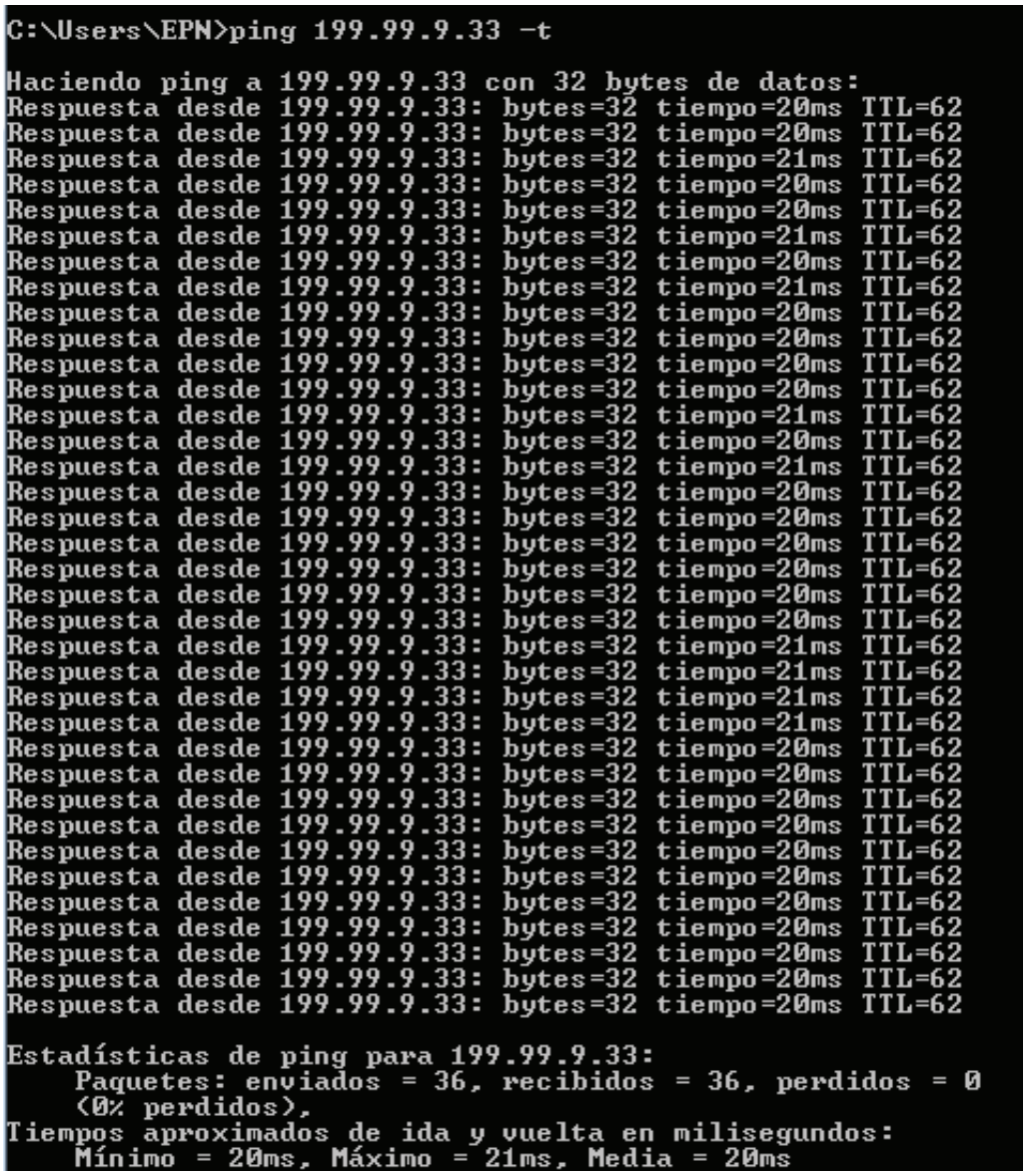


Figura 4.2 Ejecución del comando ping hacia la dirección pública del servidor

Luego de realizar la prueba se verifica que existe conectividad hacia el servidor y que el retardo promedio para alcanzar esta red es de 20 ms.

Además se verifica la conectividad entre estas redes con el comando ***tracert*** mediante el cual se puede tener información sobre los diferentes saltos que sigue el paquete IP de *router* en *router*. La ejecución de la prueba se muestra en la figura 4.3 y permite comparar el orden de los saltos con respecto a la topología de la figura 4.1.

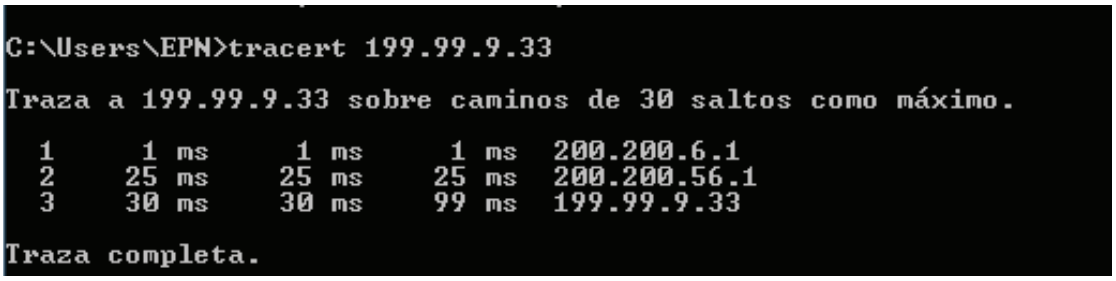


Figura 4.3 Ejecución del comando tracert hacia la dirección pública del servidor

4.1.1.2 Conectividad de red remota IPv6

Se verifica la conectividad de la red remota IPv6 ***(2000::4c:0/112)*** hacia el servidor de telefonía el cual tiene una dirección IPv6 estática ***2000::aa:3***.

Se utilizando como herramienta el comando ***ping*** de la consola de Windows. El *host* de origen posee la dirección 2000::4c:2; mientras que el destino es el servidor tal como se muestra en la ejecución de la prueba referida a la figura 4.4.

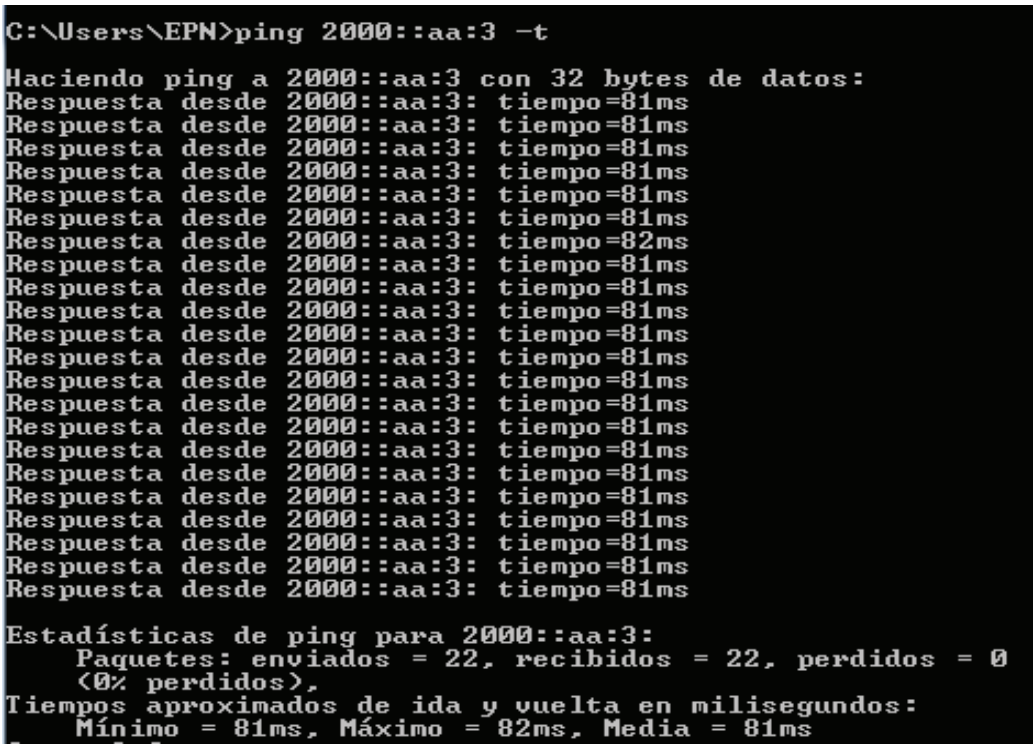


Figura 4.4 Ejecución del comando ping a un *host* en una red remota IPv6

Luego de realizar la prueba se verifica que existe conectividad de la red remota IPv6 hacia el servidor. Adicionalmente se observa que el retardo promedio para

alcanzar esta red es de 81 ms. Mediante este ping se comprueba además implícitamente que existe conectividad en la nube IPv6.

Adicionalmente se puede verificar la ruta que sigue el paquete *router* a *router* a través del comando tracert. Para la ejecución de este comando se utiliza el mismo *host* de origen y como *host* de destino la dirección IPV6 de un softphone ***(2000::aa:20)*** localizado en la red local del servidor .

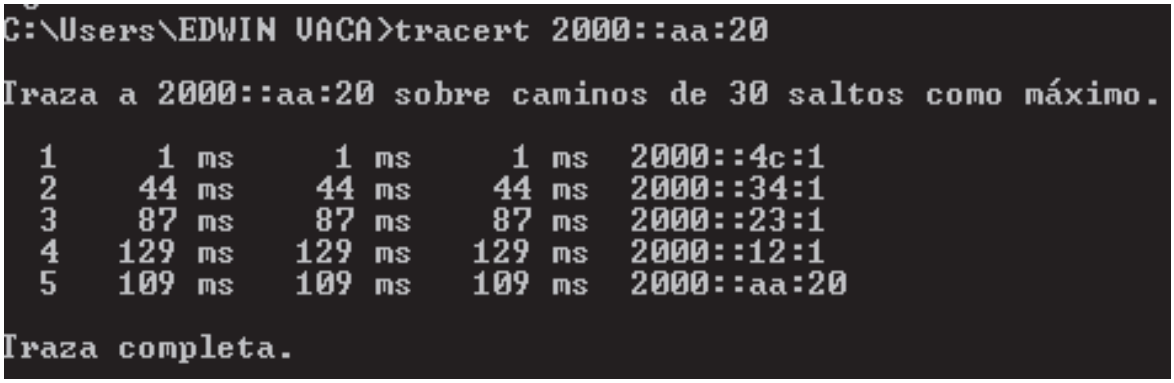


Figura 4.5 Ejecución del comando tracert desde un *host* remoto IPv6, hacia un *host* en la red del servidor

De acuerdo a la figura 4.5 se puede observar que la conectividad es total. Para alcanzar el destino el *host* origen realiza 5 saltos (4 *router*s y un salto hacia el *host* destino). En cada salto se tiene un retardo equivalente al tiempo de respuesta para los paquetes enviados; por lo que el retardo hacia el *host* destino es de 109 ms.

4.1.1.3 Conectividad de red remota IPv6 que atraviesa un túnel IPv4

Se dispone de un *host* remoto con dirección IPv6 que quiere comunicarse con el servidor de telefonía a través de una red IPv4. Por esta razón se implementa un túnel IPv6 sobre IPv4 en esta red.

Para comprobar la conectividad se emite el comando ***ping*** desde el *host* remoto hacia el servidor. El *host* origen posee la dirección 2000::6E:2 y el *host* destino la dirección 2000::AA:20, obteniendo los siguientes resultados:

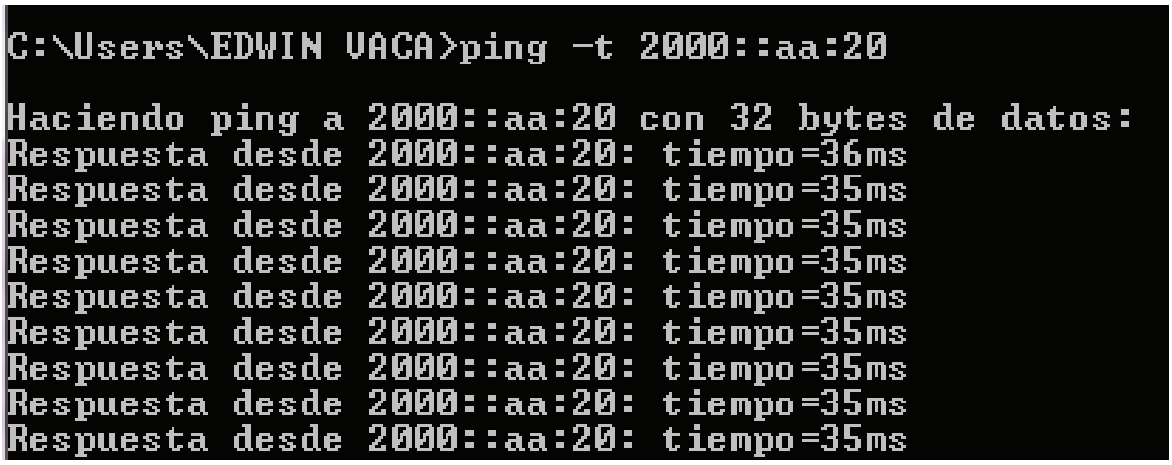


Figura 4.6 Ejecución del comando ping desde un *host* remoto IPv6 que atraviesa una red IPv4 (mediante un túnel) hacia un *host* en la red del servidor

Se comprueba que existe conectividad total y que el retardo promedio en alcanzar al destino es de 35 ms.

De acuerdo a los resultados obtenidos en la figura 4.6 se puede concluir que al realizar un túnel se obtienen más retardos (a nivel de red) con respecto a una red IPv4 pura. Se presenta en este caso un retardo adicional de 14 ms debido a que el *router* debe realizar un procesamiento extra; ya sea al encapsular IPv6 sobre IPv4 o desencapsular IPv4 para obtener IPv6.

Para verificar la conectividad a través del túnel se realiza la prueba del seguimiento del paquete *router* a *router*. La prueba se realiza hasta un softphone IPv6 (2000::aa:20) en la red local de servidor mediante el comando tracert.

Figura 4.7 Ejecución comando tracert desde un *host* que atraviesa una red IPv4 mediante un túnel

Para alcanzar su destino el paquete realiza 3 saltos (2 *routers* y un salto hacia el *host* destino). Se obtiene un tiempo de respuesta equivalente al retardo de 45 ms.

De igual manera se realiza el ping desde el servidor hacia el *host* con dirección IPv6 detrás de una red IPv4 a través de un túnel mediante el comando ***ping6.***

```
[root@tesis Desktop]# ping6 2000::6e:2
PING 2000::6e:2(2000::6e:2) 56 data bytes
64 bytes from 2000::6e:2: icmp_seq=1 ttl=62 time=43.0 ms
64 bytes from 2000::6e:2: icmp_seq=2 ttl=62 time=42.7 ms
64 bytes from 2000::6e:2: icmp_seq=3 ttl=62 time=42.7 ms
64 bytes from 2000::6e:2: icmp_seq=4 ttl=62 time=42.6 ms
64 bytes from 2000::6e:2: icmp_seq=5 ttl=62 time=42.8 ms
64 bytes from 2000::6e:2: icmp_seq=6 ttl=62 time=60.8 ms
64 bytes from 2000::6e:2: icmp_seq=7 ttl=62 time=42.9 ms
^C
--- 2000::6e:2 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6658ms
rtt min/avg/max/mdev = 42.685/45.415/60.871/6.314 ms
```

Figura 4.8 Ejecución del comando ping desde el servidor hacia un *host* IPv6 detrás de un túnel

4.2 ESTABLECIMIENTO DE LLAMADAS

Las pruebas de establecimiento de llamadas se las realiza con la herramienta de análisis de red Wireshark. Esta posee una función para observar el intercambio de mensajes entre llamadas de VoIP.

4.2.1 EN AMBIENTES IPV4

Se efectúan llamadas con *hosts* de tipo IPv4 en donde se comprueba el establecimiento de la comunicación. Se utiliza como origen y destino de las llamadas los *hosts* que pertenecen a las subredes que se incluyen en el escenario descrito por la imagen que se muestra a continuación:

Figura 4.9 Ambiente de pruebas IPv4

4.2.1.1 Llamada SIP-SIP en la misma LAN

Mediante la realización de esta prueba se verifica el establecimiento de una llamada entre dos teléfonos que utilizan el protocolo IPv4 y el protocolo de comunicación SIP.

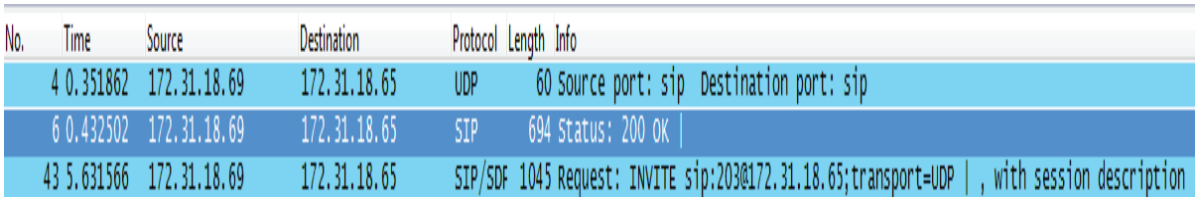
En la figura 4.10 se tiene los mensajes intercambiados en el establecimiento de esta llamada. Se define como *host* origen la dirección 172.31.18.69 y *host* destino la dirección 172.31.18.65:

Figura 4.10 Intercambio de mensajes en una llamada SIP-SIP

De acuerdo a la figura anterior en el proceso se realizan los siguientes intercambios de mensajes:

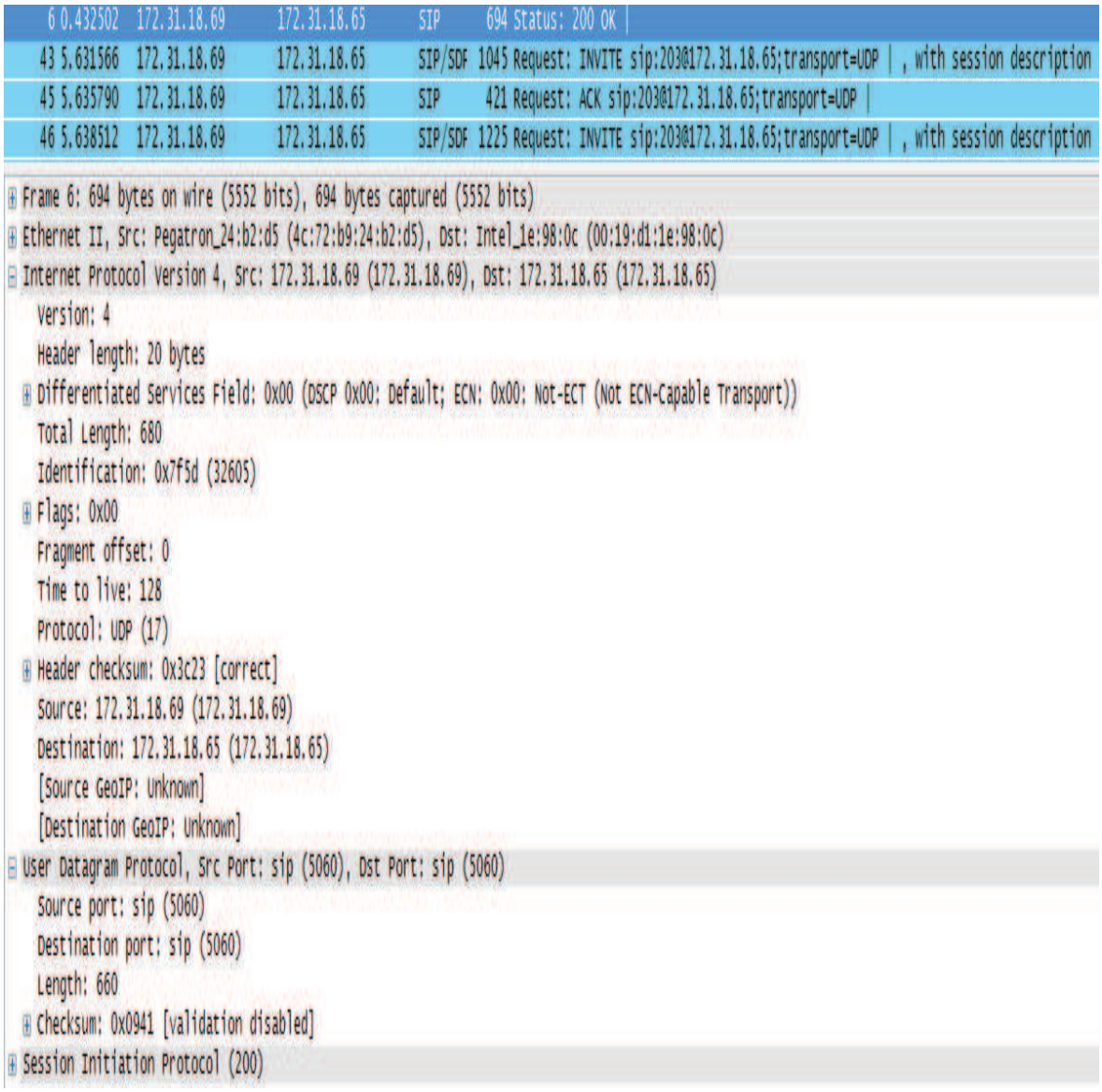
- El teléfono llamante envía una invitación a establecer una llamada, para lo cual pide utilizar cualquiera de los tres códecs: GSM, G711U ó G711A (negociación de códec).
- El servidor responde con un mensaje de error 401 que indica que en la invitación por parte del usuario no se incluyen las credenciales de autenticación. Por lo que se da paso al desafío para iniciar el proceso de autenticación.
- El usuario responde con un ACK.
- El usuario envía nuevamente una invitación para establecer la comunicación mediante cualquiera de los tres códec; pero esta vez incluye datos de autenticación.
- El servidor envía una respuesta informativa 100, la cual indica que está intentando autenticarlo.
- Una vez que el teléfono destino empieza a sonar, el servidor envía una respuesta informativa al usuario llamante (180 ringing).
- Si el usuario destino contesta, el servidor envía un mensaje de respuesta de éxito (200 OK). En este mensaje se envía los códec que el usuario destino puede utilizar.
- El teléfono llamante responde con un ACK confirmando la comunicación.
- Se establece la llamada utilizando el protocolo RTP para el intercambio de flujos de datos y utilizando el códec GSM.
- En este caso el usuario destino cuelga, por lo que el servidor envía un BYE al teléfono llamante.
- El servidor envía un mensaje de éxito (200 OK).
- El procedimiento de establecimiento y terminación de una llamada SIP es el mismo para todas las aplicaciones que utilizan este protocolo. Por ejemplo el buzón y correo de voz, el IVR, el directorio, las conferencias, entre otros. Por otro lado la negociación del códec dependerá del tipo de códec que puede utilizar el cliente (teléfono o softphone).

En este caso se puede apreciar claramente en la figura 4.12 que la llamada se realiza en un ambiente IPv4 puro, ya que la dirección origen y destino son de este tipo. Además se observa que SIP utiliza UDP para la conexión y que utiliza el puerto 5060 tal y como era de esperarse.



No.	Time	Source	Destination	Protocol	Length	Info
4	0.351862	172.31.18.69	172.31.18.65	UDP	60	Source port: sip Destination port: sip
6	0.432502	172.31.18.69	172.31.18.65	SIP	694	Status: 200 OK |
43	5.631566	172.31.18.69	172.31.18.65	SIP/SDP	1045	Request: INVITE sip:203@172.31.18.65;transport=UDP | , with session description

Figura 4.11 Captura de paquetes llamada SIP-SIP en un ambiente IPv4



6	0.432502	172.31.18.69	172.31.18.65	SIP	694	Status: 200 OK |
43	5.631566	172.31.18.69	172.31.18.65	SIP/SDP	1045	Request: INVITE sip:203@172.31.18.65;transport=UDP | , with session description
45	5.635790	172.31.18.69	172.31.18.65	SIP	421	Request: ACK sip:203@172.31.18.65;transport=UDP |
46	5.638512	172.31.18.69	172.31.18.65	SIP/SDP	1225	Request: INVITE sip:203@172.31.18.65;transport=UDP | , with session description

```
⊞ Frame 6: 694 bytes on wire (5552 bits), 694 bytes captured (5552 bits)
⊞ Ethernet II, Src: Pegatron_24:b2:d5 (4c:72:b9:24:b2:d5), Dst: Intel_1e:98:0c (00:19:d1:1e:98:0c)
⊟ Internet Protocol Version 4, Src: 172.31.18.69 (172.31.18.69), Dst: 172.31.18.65 (172.31.18.65)
    Version: 4
    Header length: 20 bytes
  ⊞ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))
    Total Length: 680
    Identification: 0x7f5d (32605)
  ⊞ Flags: 0x00
    Fragment offset: 0
    Time to live: 128
    Protocol: UDP (17)
  ⊞ Header checksum: 0x3c23 [correct]
    Source: 172.31.18.69 (172.31.18.69)
    Destination: 172.31.18.65 (172.31.18.65)
    [Source GeoIP: Unknown]
    [Destination GeoIP: Unknown]
⊟ User Datagram Protocol, Src Port: sip (5060), Dst Port: sip (5060)
    Source port: sip (5060)
    Destination port: sip (5060)
    Length: 660
  ⊞ Checksum: 0x0941 [validation disabled]
⊞ Session Initiation Protocol (200)
```

Figura 4.12 Detalle de Captura de paquetes llamada SIP-SIP en un ambiente IPv4

4.2.1.2 Llamadas IAX-IAX

Con la realización de esta prueba se pretende comprobar la operabilidad del sistema con el protocolo de comunicación IAX.

En una llamada entre dos teléfonos que utilizan el protocolo de comunicación IAX2 (como prueba se realiza una llamada entre dos softphones zoiper: dirección IP origen: 172.31.18.60, dirección IP destino: 172.31.18.58); en la captura de paquetes se despliega el siguiente intercambio de mensajes:

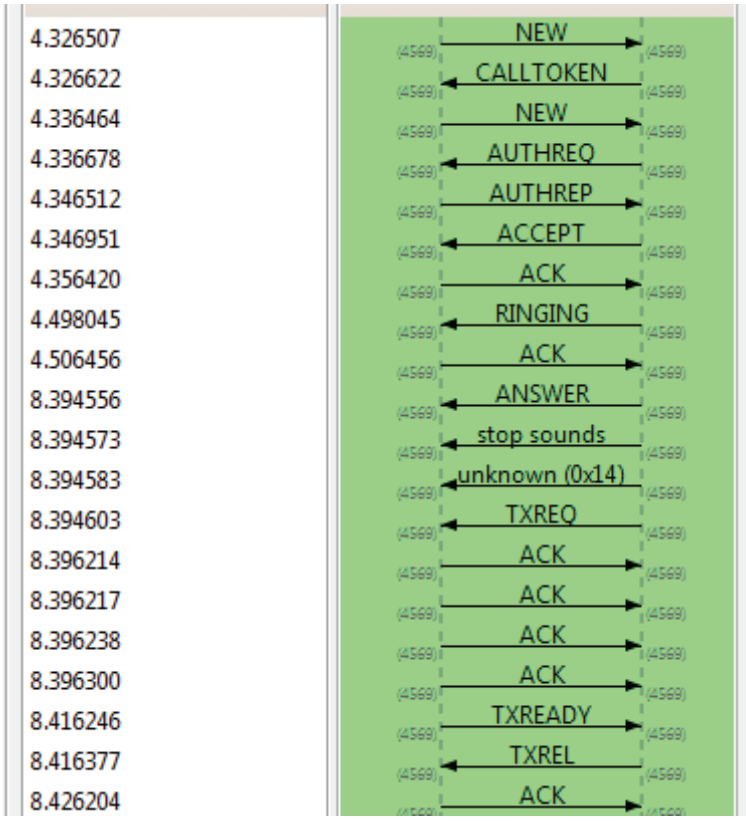


Figura 4.13 Intercambio de mensajes en una llamada IAX-IAX

- El teléfono llamante empieza la fase de establecimiento de la comunicación enviando un mensaje de **NEW al** servidor.
- El servidor responde con un mensaje de **CALLTOKEN** comunicándole que ahora él tiene el *token* para realizar la llamada.
- El teléfono llamante ahora en posesión del *token*; envía nuevamente un mensaje de **NEW** al servidor para establecer la llamada.
- El servidor pide autenticación al usuario mediante un mensaje de **AUTHREQ** .

- El cliente responde con sus datos de autenticación mediante un mensaje **AUTHREP.**
- Si los datos de autenticación del cliente son correctos el servidor envía un mensaje de respuesta **ACCEPT.**
- El usuario confirma este mensaje mediante un **ACK.**
- El servidor una vez que ha autenticado al cliente le informa a este que el teléfono destino está timbrando mediante un mensaje de **RINGING.**
- El teléfono llamante confirma este mensaje.
- Una vez que el teléfono destino ha contestado, el servidor le comunica esto al teléfono llamante mediante un mensaje de **ANSWER.**
- El servidor envía un mensaje de petición de transferencia **TXREQ.**
- El teléfono llamante acepta mediante un **ACK** y cuando está listo mediante un mensaje de transferencia preparada **TXREADY.**
- Para terminar la llamada cuando alguno de los dos cuelga, el servidor envía un mensaje de liberación de la transferencia mediante un **TXREL.**
- El cliente responde con un **ACK** y así se libera la comunicación.

4.2.1.3 Llamadas SIP-IAX

Se realiza esta prueba, para probar la compatibilidad entre dos protocolos de comunicación de VoIP distintos como los son SIP e IAX2. El intercambio de mensajes entre los *host* origen y destino se observa en la figura 4.14:

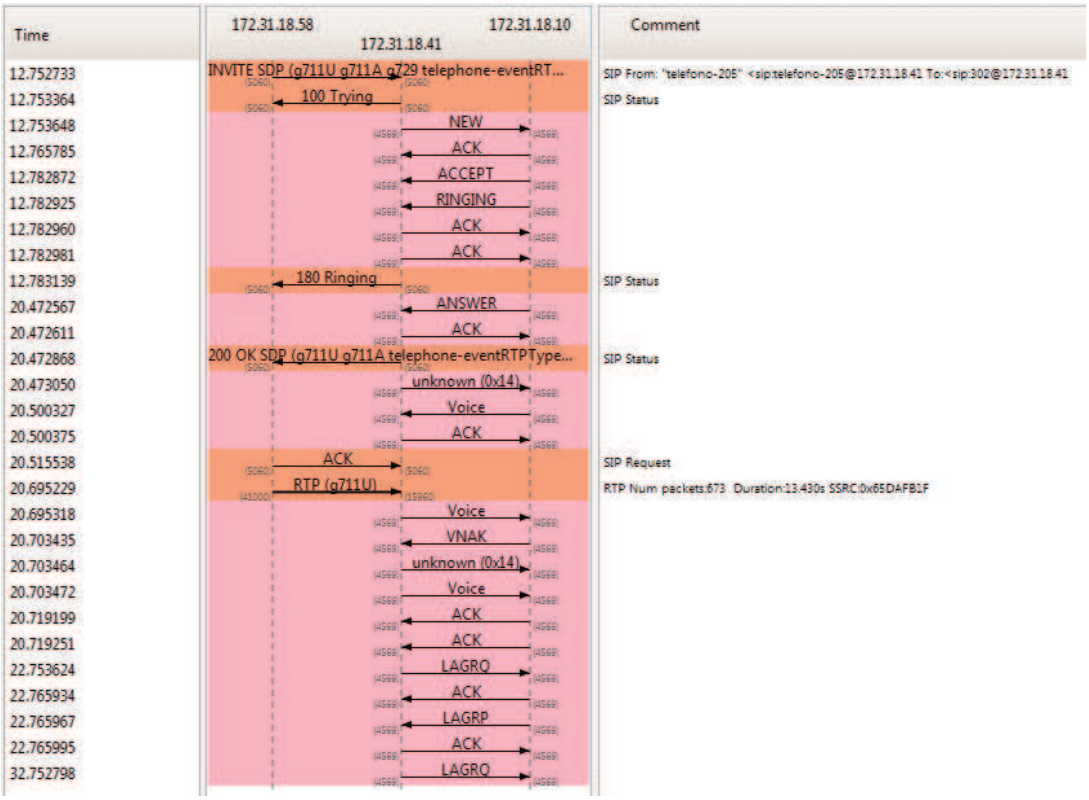


Figura 4.14 Intercambio de mensajes en una llamada SIP-IAX

En color anaranjado tenemos los mensajes del cliente SIP hacia el servidor. Se resalta en color rosado tenemos los mensajes entre el cliente IAX2 y el servidor.

En este caso los nuevos mensajes entre el servidor y el cliente IAX2 son una petición de ***VNAK*** (petición de retransmisión), una petición de LAG ***LAGRQ*** y una respuesta de LAG ***LAGRP*** (LAG es el retraso producido en una telecomunicación que dificulta el desarrollo normal de la misma; la sensación que se produce es que el interlocutor tarda en contestar, con lo que la conversación resulta molesta).

4.2.1.4 Llamadas SIP-SIP a través de NAT

Se efectúa una llamada desde una dirección pública hasta una privada. El origen es la dirección 200.200.6.20 mientras que el *host* destino 172.31.18.58, tal como se observa en la figura 4.15:

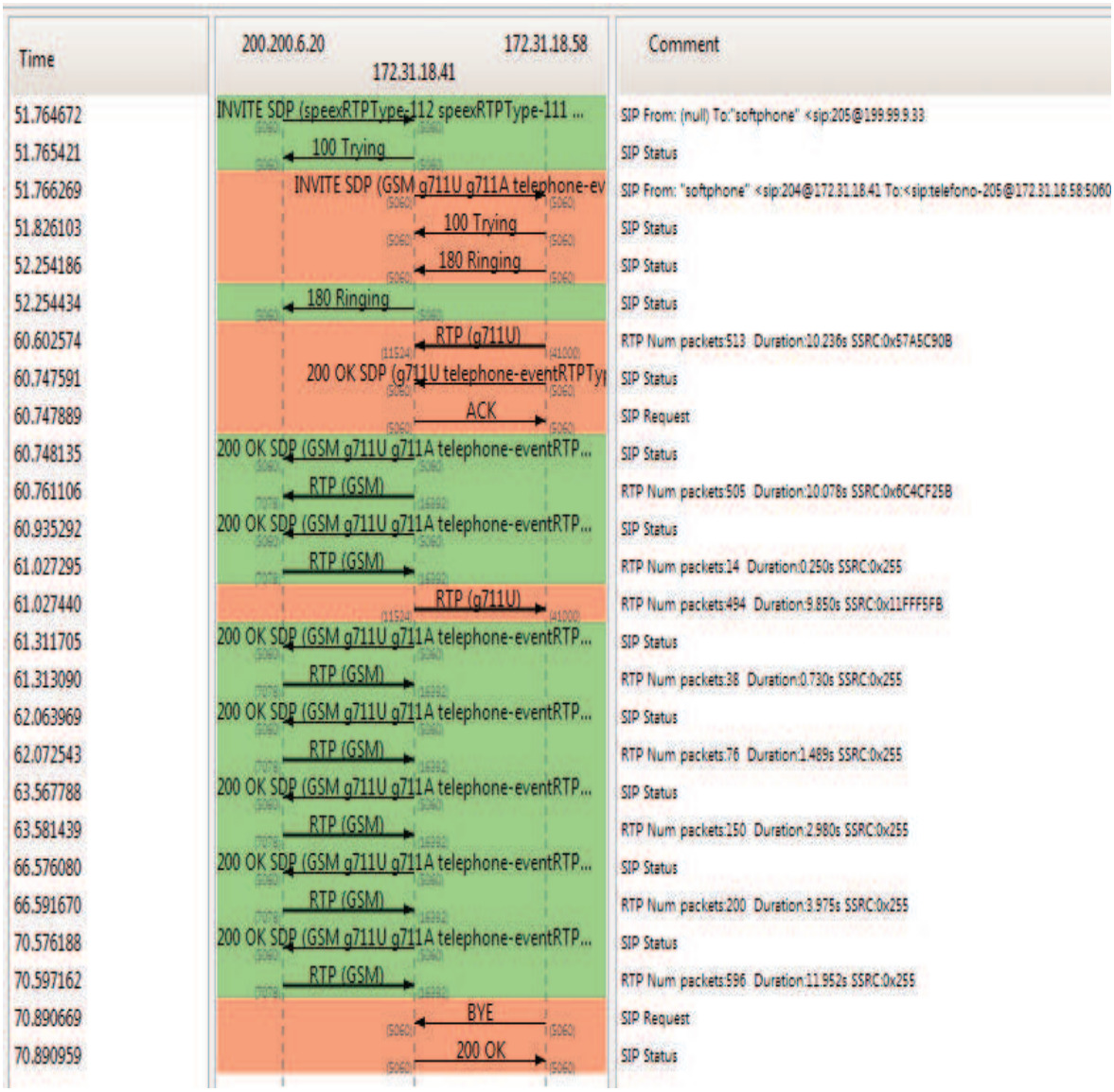


Figura 4.15 Intercambio de mensajes en una llamada SIP-SIP a través de NAT

Los mensajes intercambiados son los mismos que en una llamada SIP-SIP (negociación de sesión). Se debe destacar en esta prueba la simulación de una llamada entre una red local y otra independiente. Las redes se conectan gracias a que se implementa NAT en los *router*s de borde.

Los mensajes en verde son los intercambiados entre el teléfono llamante y el servidor; los que están en anaranjado son los mensajes intercambiados entre el teléfono llamado y el servidor.

En este caso se observa que la llamada se realiza desde un softphone con una dirección pública hacia un teléfono IP con una dirección privada.

Al analizar el intercambio de mensajes se puede concluir que el proceso de NAT es transparente para Asterisk; a diferencia del *host* origen para el cual el destino es la dirección mapeada del servidor.

En la consola del softphone únicamente se muestra la dirección IP pública del servidor de telefonía. Esto se debe a que el mecanismo NAT impide ver la dirección real del servidor a los dispositivos externos. Al implementar NAT en el *router* de borde de la red local se simula una salida a Internet.

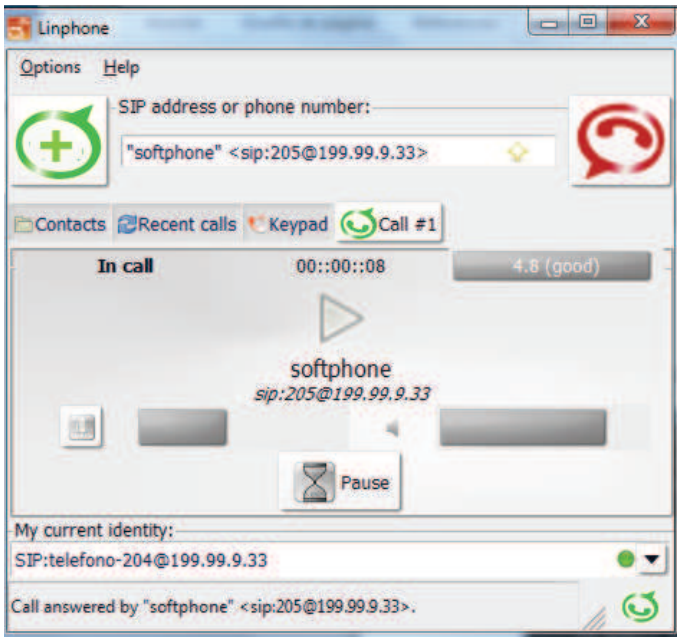


Figura 4.16 Llamada establecida entre un softphone con dirección IP pública y el servidor de telefonía

4.2.1.5 Llamadas entre teléfonos analógico y teléfonos IP

El intercambio de mensajes en una llamada entre canales DAHDI y SIP es el mismo que en una llamada SIP-SIP (negociación de sesión: códecs, puertos, etc.).

En el intercambio de mensajes de la Figura 4.17 se observa que Asterisk hace uso de la tarjeta FXS. Esto se puede apreciar en los comentarios que claramente detallan el uso del canal 4 que corresponde a una interfaz FXS. Asterisk hace uso de esta interfaz para el establecimiento de la comunicación entre los clientes (Dirección IP origen: 172.31.18.41, Dirección IP destino: 172.31.18.58).

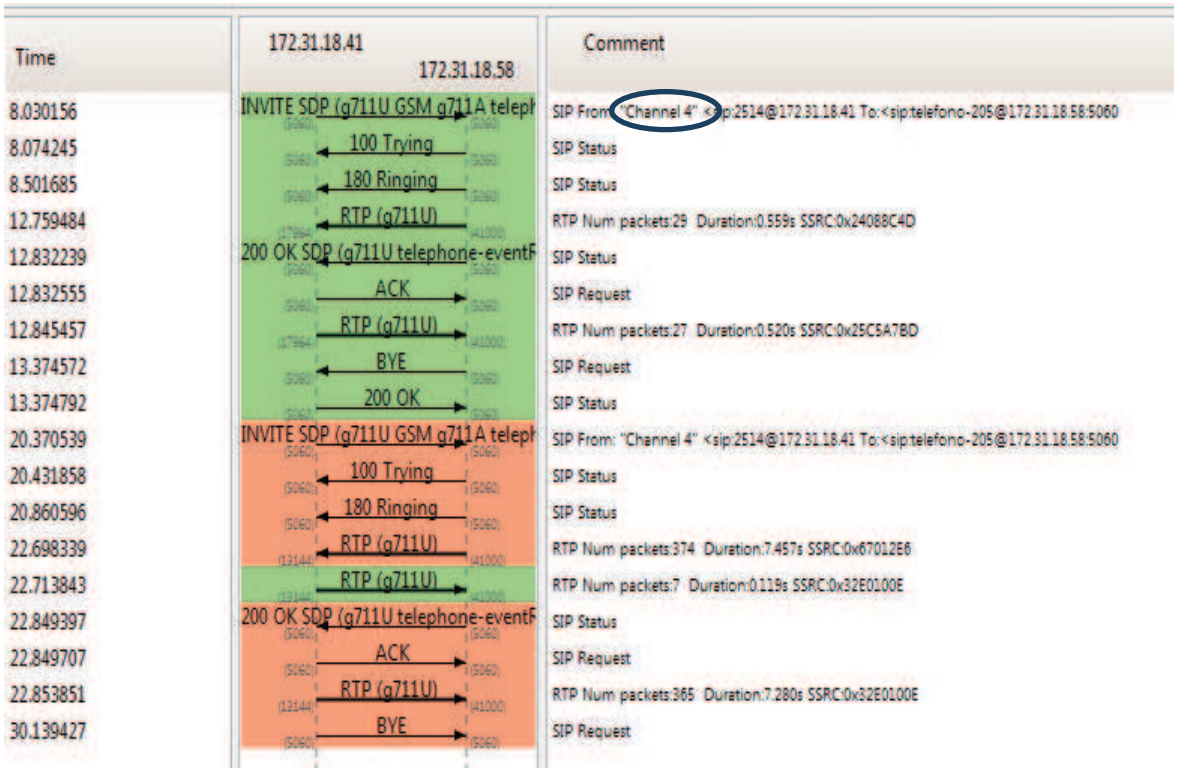


Figura 4.17 Intercambio de mensajes en una llamada entre teléfonos analógicos y teléfonos IP

4.2.1.6 Llamadas hacia y desde la PSTN

Para comprobar el establecimiento de una llamada desde y hacia la PSTN el sistema hace uso del canal 1. En dicho canal se encuentra una tarjeta FXO como se detalla en la consola de Asterisk, esto se ilustra en la figura 4.18.

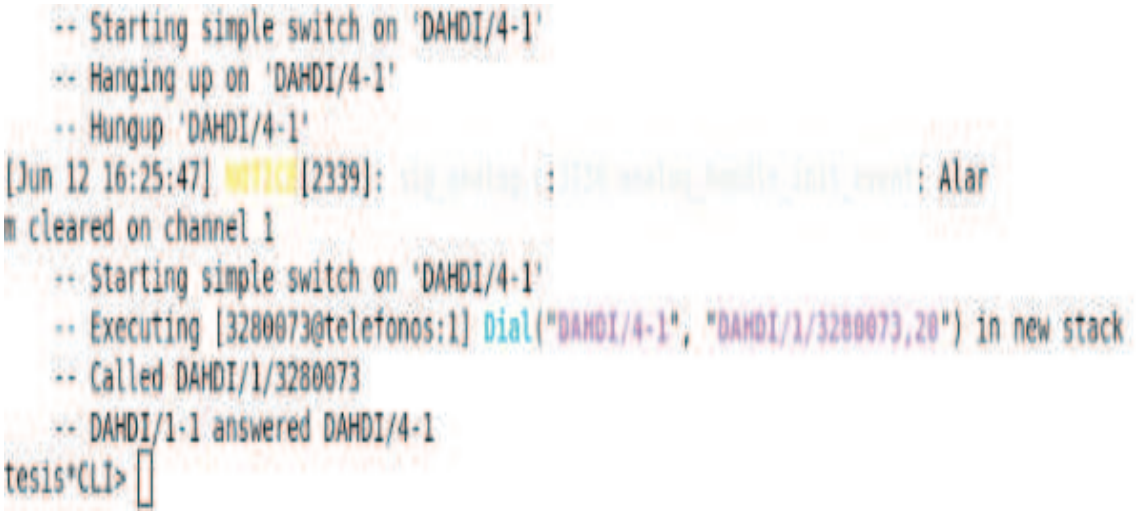


Figura 4.18 Resultado en la consola de Asterisk para una llamada hacia la PSTN

Los mensajes intercambiados en estas llamadas son similares a los de una llamada SIP-SIP. Se establece la llamada, se intercambian los paquetes de control ACK e informativo (180 y 200) y se negocia el tipo de códec a utilizar en la comunicación. Al finalizar el cliente envía un mensaje de BYE culminando la llamada.

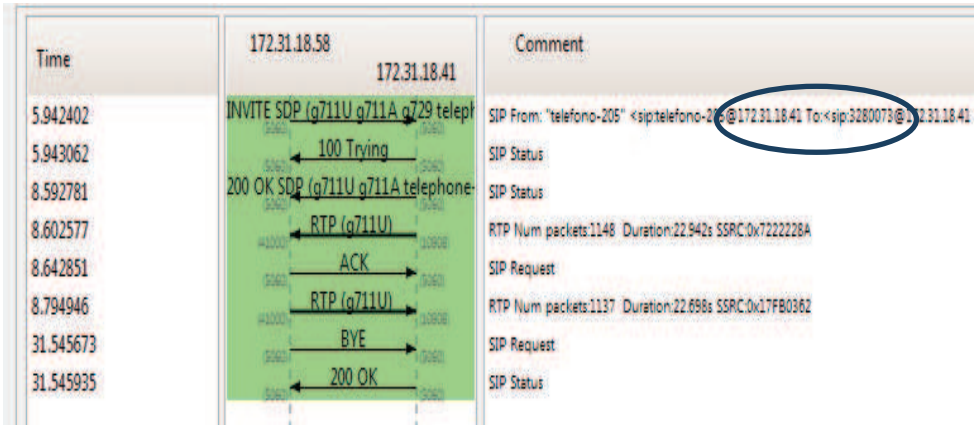


Figura 4.19 Intercambio de mensajes en una llamada hacia la PSTN

4.2.2 EN AMBIENTES IPV6

Para comprobar el establecimiento de llamadas entre *host* con direcciones IPv6 se utiliza como *hosts* de origen y destino aquellos que utilizan el protocolo SIP para registrarse en la central. Se emplea el protocolo SIP ya que Asterisk solo soporta este protocolo para llamadas IPv6. Los *hosts* que intervienen en este escenario de pruebas pertenecen al diagrama de topología que se muestra a continuación:

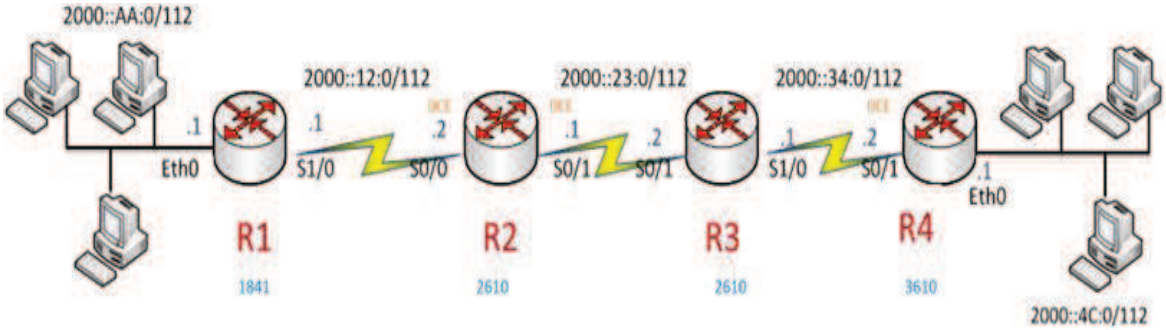


Figura 4.20 Ambiente de pruebas IPv6

4.2.2.1 Llamadas SIP-SIP

Se realiza una llamada entre dos usuarios con direcciones IPv6 con protocolo de comunicación SIP. La llamada se establece exitosamente.

Los mensajes intercambiados son similares a los que se explicó para una llamada SIP-SIP en IPv4. La diferencia en esta llamada radica en que el servidor hace uso de la doble pila para utilizar su dirección IPv6 en el establecimiento de la comunicación. Se ejecuta una llamada desde el *host* 2000::AA:20 hasta el *host* 2000::4C; el esquema de intercambio de los mensajes para la comunicación se presentan en la figura 4.21.

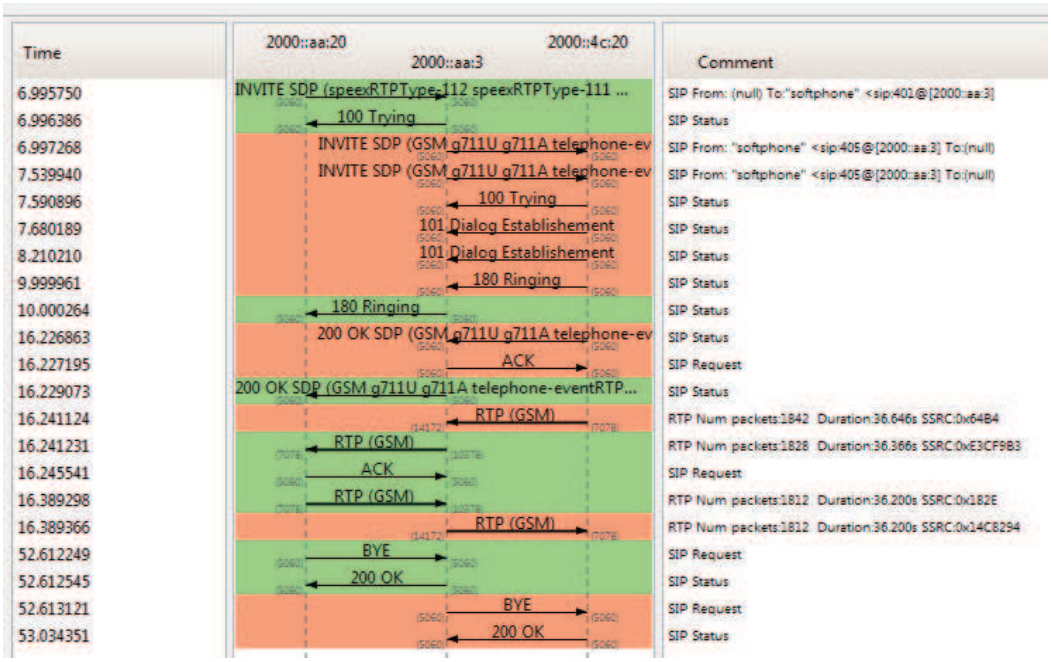


Figura 4.21 Intercambio de mensajes en una llamada SIP en ambientes IPv6

4.2.3 EN AMBIENTES MIXTOS

Se efectúan llamadas con *host*s de tipo IPv4 e IPv6 con el objetivo de comprobar el establecimiento de llamadas entre *host* conectados con mecanismos de NAT y tunelización. El escenario de pruebas está descrito por la topología que se muestra en la figura 4.22.

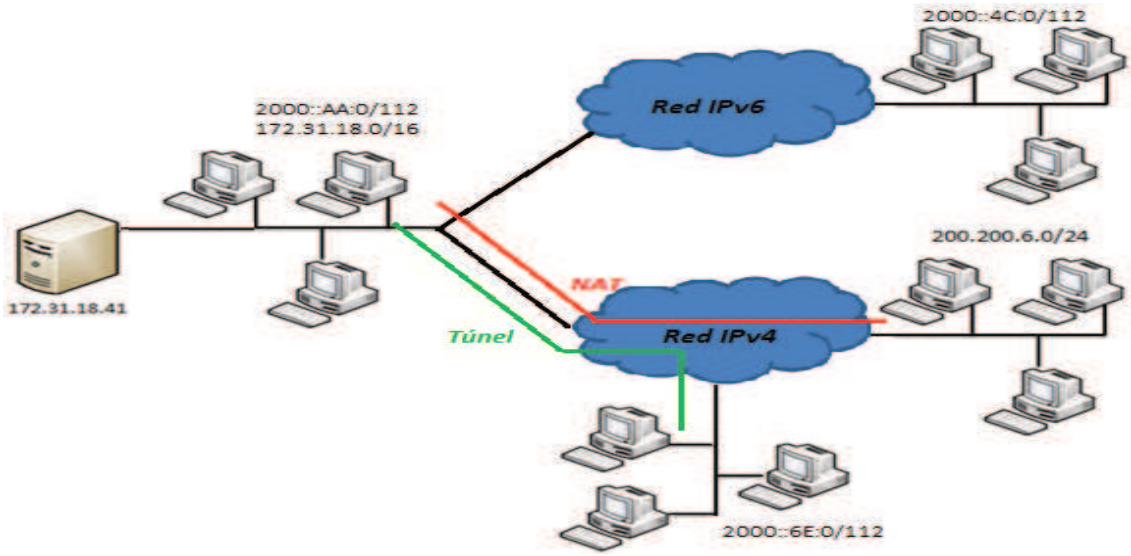


Figura 4.22 Ambiente de pruebas IPv4 e IPv6

4.2.3.1 Llamadas IPV4 a IPV6

Se realiza esta prueba con la finalidad de comprobar la convivencia de la aplicación de Telefonía IP en ambientes IPv4 e IPv6.

En este caso se realiza la prueba con una llamada originada desde un cliente de una red IPv4 (200.200.6.10) hacia un cliente en una red IPv6 (2000::4c:20).
La llamada es exitosa y los mensajes intercambiados entre los usuarios y el servidor son similares a los ya explicados en una llamada SIP-SIP.

En la figura 4.23 se puede observar que la llamada la inicia un cliente en una red IPv4 y se dirige hacia un cliente ubicado en la red remota IPv6. Por esta razón el servidor hace uso de la doble pila utilizando una dirección IPv4 y una dirección IPv6 (172.31.18.41 y 2000::aa:3) para establecer la comunicación con los usuarios respectivos.

Se observa que es Asterisk quién realiza la conversión de protocolo de Internet para el establecimiento de la comunicación. El servidor intercambia su dirección de una dirección IPv4 a una IPv6 y viceversa (en base a las direcciones propias asignadas al servidor). El servidor recibe un paquete IPv4 mediante su dirección IPv4 y lo reenvía mediante su dirección IPv6 cuando el *host* destino es de tipo IPv6. Este procesamiento se realiza internamente en el servidor gracias a su doble pila.

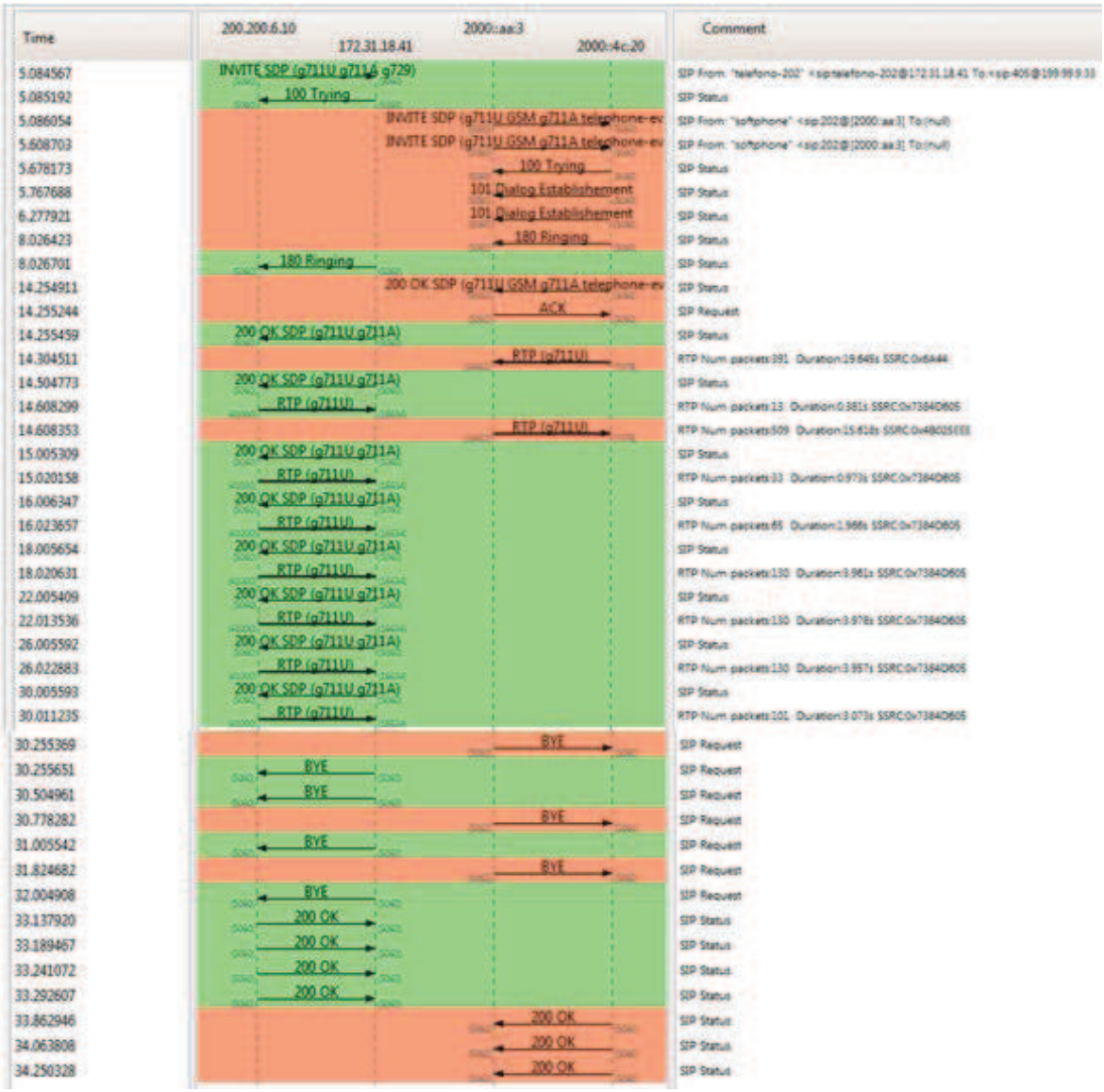


Figura 4.23 Intercambio de mensajes en una llamada IPv4-IPv6

Se concluye luego de la realización de esta prueba; que es posible la comunicación de un cliente de telefonía IP, de una red IPv4 hacia un cliente en una red IPV6.

4.2.3.2 Llamadas IPV6 a IPV4

Se pretende comprobar que la aplicación funciona cuando la llamada se origina de un cliente en una red IPv6 hacia un cliente en una red IPv4. Para este efecto se utiliza como dirección IP origen la 2000::4C:20 y como destino la 172.31.18.58.

Los mensajes intercambiados son mostrados en la figura 4.24.

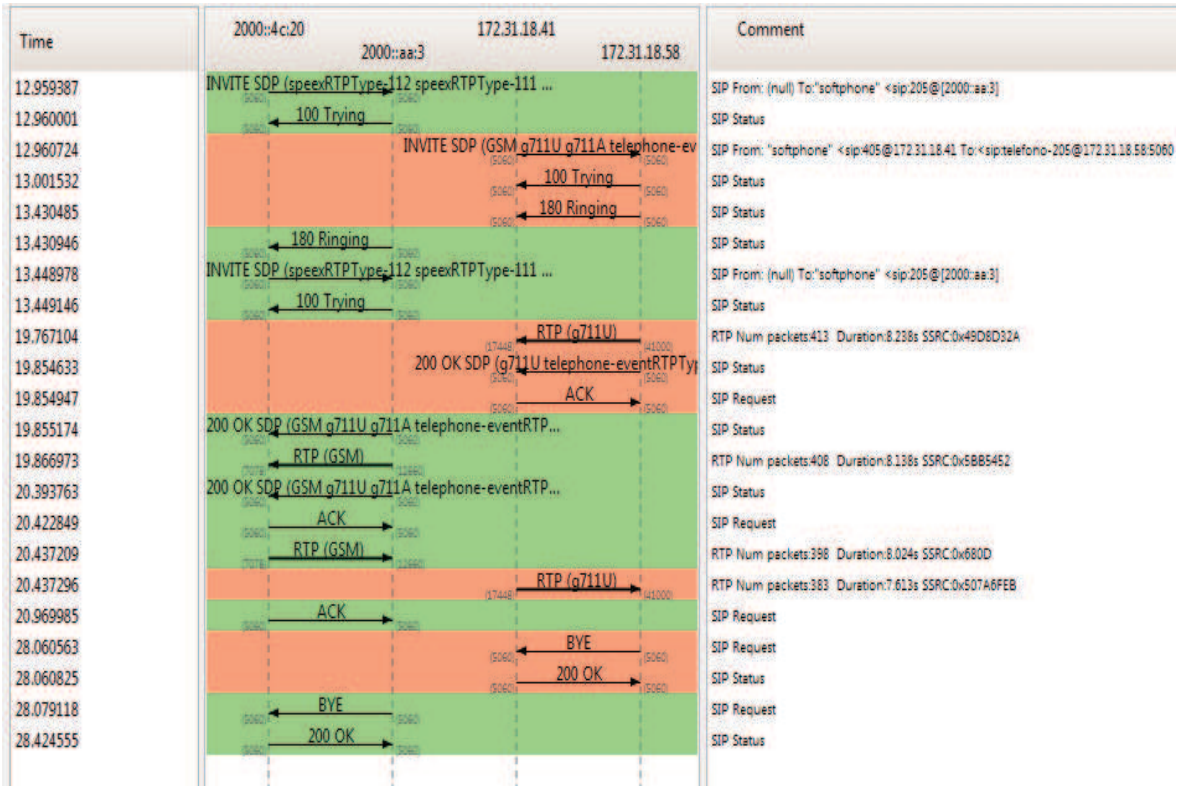


Figura 4.24 Intercambio de mensajes en una llamada IPv6-IPv4

Al igual que en el caso anterior se tiene éxito en el establecimiento de la comunicación y se observa en la figura que es Asterisk quién realiza el proceso de conversión de protocolo, el cual es transparente para el usuario.

Se puede concluir luego de realizar las dos pruebas (llamadas IPv4 a IPv6 y viceversa), que existe convivencia de la aplicación de telefonía IP en ambos protocolos de Internet.

4.2.3.2.1 Análisis de Resultados

La conectividad de la aplicación entre redes que manejan distintas versiones del protocolo de Internet (IP) se realiza gracias al servidor de telefonía IP. El servidor está encargado de realizar el proceso de conversión de direcciones (debido a la doble pila y a la funcionalidad de Asterisk para soportar ambos protocolos de Internet).

Es decir que para los dispositivos de conectividad intermedios el proceso de conversión es totalmente transparente. La comunicación no se realiza

directamente con el cliente destino sino que en este caso se conectan al servidor de telefonía que ejecuta las funciones de un *router* de borde para cada una de las redes a él conectadas. Por consiguiente es el servidor quién en su base de datos, busca a su cliente destino y enruta la llamada, sin importar el tipo de protocolo de Internet que maneje (IPv4 o IPv6). De esta manera se completa la conectividad de extremo a extremo (cliente origen – servidor – cliente destino).

4.2.3.3 Llamadas IPv6 (usando túneles 6to4) a IPv4

Se comprueba el funcionamiento de Asterisk en ambientes mixtos realizando una llamada a través de un túnel 6to4 implementado en la red. Para dicho efecto se utiliza como dirección IP origen a 2000::6e:2 y dirección IP destino a 172.31.18.58

En este caso para atravesar el segmento IPv4 el paquete IPv6 se encapsula en un paquete IPv4. El proceso ocurre mediante configuraciones en el *router* de borde que acoge al cliente IPv6; pero que a su vez maneja direccionamiento IPv4 de su otro lado.

Se debe tomar en cuenta que en este tipo de comunicación los retardos serán mayores. Esto se debe a que el *router* tendrá un mayor tiempo de procesamiento de los paquetes al realizar el proceso de encapsulamiento entre protocolos. En base a este proceso y su correspondiente retardo se puede transportar un paquete IPv6 a través de una red IPv4.

Luego de realizar una llamada entre dos teléfonos con direcciones IPv4 y dirección IPv6 respectivamente se tienen los mensajes de intercambio que se observan en la figura 4.25.

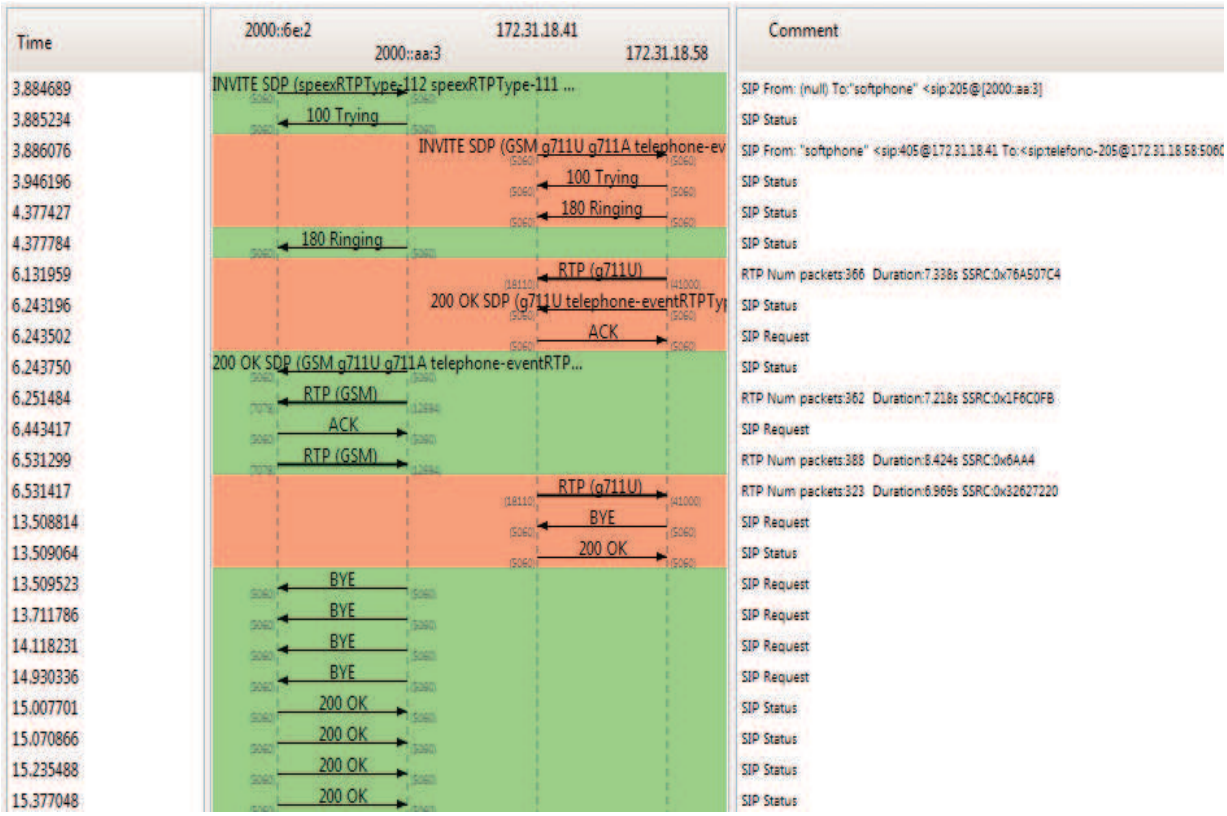


Figura 4.25 Intercambio de mensajes llamada a través de un túnel

4.2.3.3.1 Análisis de Resultados

Los resultados revelan que para el servidor Asterisk es transparente la tunelización realizada. El servidor toma únicamente la dirección origen y dirección destino sin saber que el paquete primero atravesó una red IPv4.

Los mensajes intercambiados corresponden al establecimiento, autenticación, negociación de códec y finalización de la llamada; similares a una llamada normal SIP-SIP.

Se puede comprobar mediante el intercambio de mensajes RTP entre ambos clientes que la aplicación de telefonía IP funciona de manera exitosa en una red en la cual se tiene ambientes mixtos y se tiene la necesidad de implementar un túnel para la convivencia de ambos protocolos de Internet.

El análisis evidencia además que las redes futuras que manejen direccionamiento IPv6 a nivel local podrán conectarse a través de la nube actual existente IPv4 (la cual es muy probable migre a IPv6 en un corto periodo de tiempo). Esta conexión hacia redes remotas e incluso con Internet mediante el

mecanismo de túnel se puede alcanzar implementando únicamente *router*s de borde que manejen ***túneles 6to4.***

4.3 PRUEBAS DE SEGURIDAD

De acuerdo a los mecanismos de seguridad descritos en la **sección 3.3.2.2.3** se verifica el nivel de seguridad a nivel de aplicación.

En base al mecanismo de autenticación se verifica el cumplimiento de la contraseña para el acceso y registro en la central. Como ejemplo se verifica la autenticación y registro de la extensión 406 correspondiente a un usuario Linphone con el protocolo SIP. El ingreso de la clave es cifrado y se puede observar en la siguiente figura:

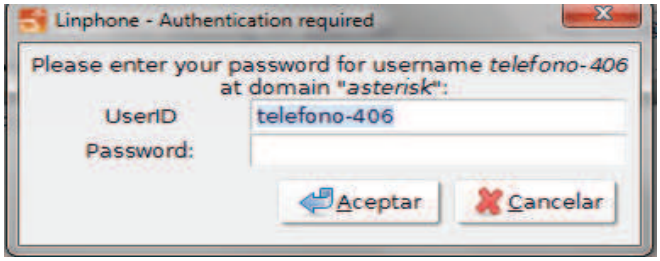


Figura 4.26 Autenticación de un usuario para registrarse

En base al mecanismo de listas de acceso se configura la extensión 205 con un valor del parámetro *permit* de modo que solo accedan usuarios dentro de la red 172.31.18.0/24. La configuración de ACL o listas de acceso se activa y permite el registro, el resultado de la configuración se presenta en la figura 4.27.

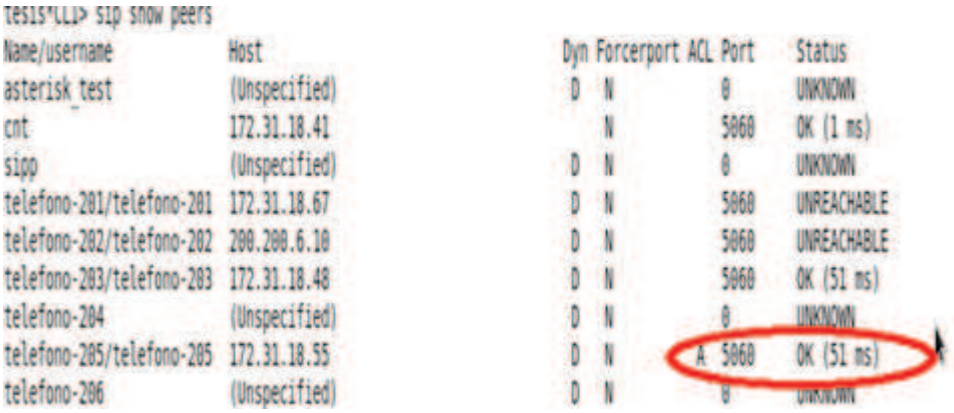


Figura 4.27 Registro de un usuario con lista de acceso

Posteriormente se configura el valor del parámetro *deny* con una dirección 0.0.0.0/0.0.0.0 con la cual se impide el acceso a cualquier usuario. Esta configuración impide el registro de la extensión 205 y se muestra en la siguiente figura:

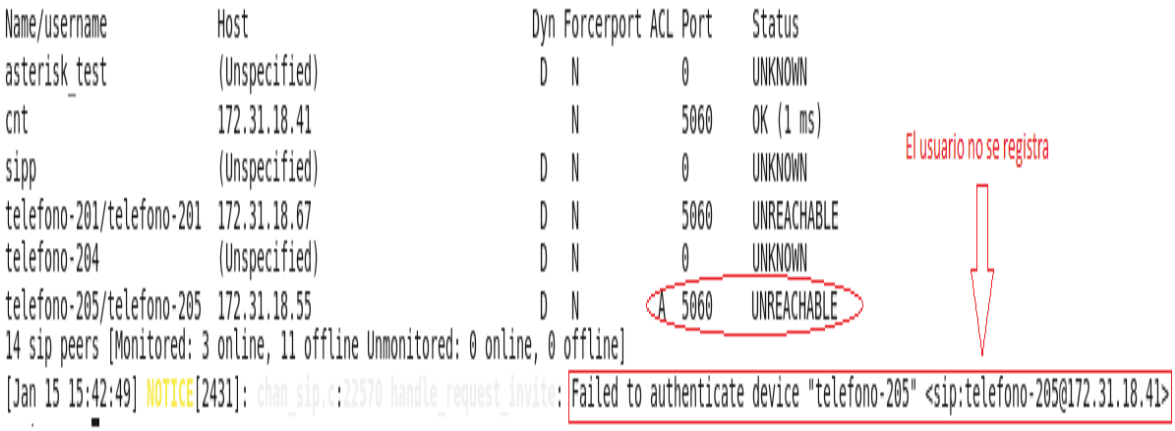


Figura 4.28 Denegación del registro con listas de acceso

Finalmente se comprueba la seguridad de operación en la extensión 205, verificando su acceso hacia teléfonos celulares. En primera instancia se permite el acceso al contexto que contiene las extensiones para llamar a celulares, la ejecución de esta llamada se realiza desde la extensión 205 hasta el número 0987314271.

Posteriormente se elimina de la extensión 205 el contexto “teléfonos” que habilita la llamada a celulares. El resultado de esta configuración se puede observar en la figura 4.29.

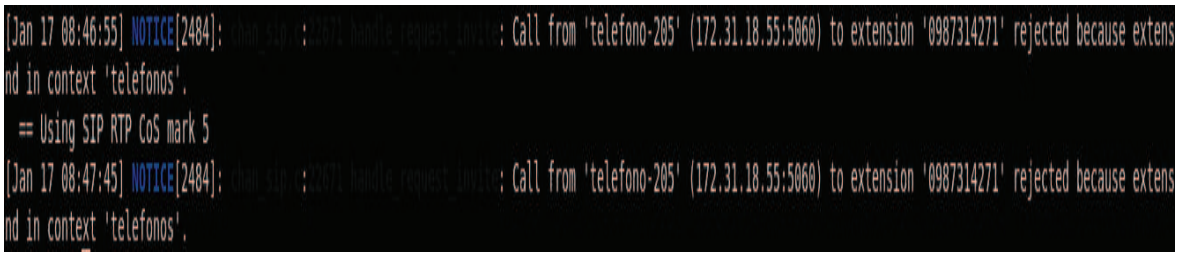


Figura 4.29 Denegación del registro con listas de acceso

La consola de Asterisk nos informa que se rechaza la llamada ya que la extensión 205 no tiene acceso al contexto “teléfonos” en donde se encuentran las extensiones de celulares; por esta razón no se concluye la llamada hasta el número 0987314271.

4.4 PRUEBAS DE RETARDO

El retardo dentro de las comunicaciones de tipo tiempo real es un factor de gran influencia en la calidad de las llamadas y percepción de los usuarios. Dentro de las pruebas efectuadas en el prototipo se consideran los retardos producidos por parámetros inherentes a los elementos de la telefonía IP y los producidos por la plataforma de interconexión que se utiliza. Por esta razón es necesario distinguir cada una de las fuentes de retardo y medir su valor mediante la realización de pruebas para finalmente analizar el efecto de retardo total sobre el prototipo y sus servicios.

4.4.1 FUENTES DE RETARDO

4.4.1.1 Retardo Algorítmico

Este retardo se debe al tipo de codificación empleada en el proceso de digitalización de la voz y es introducido por el CODEC. Se presenta en la tabla 4.1 los valores para los CODECS utilizados en el presente prototipo.

CODEC	Codificación	Retardo Algorítmico (ms)
G.711	PCM	0.125
GSM	RPE_LTP	20

Tabla 4.1Retardo Algorítmico

4.4.1.2 Retardo de Paquetización

Este retardo está relacionado con el tiempo que tarda el codificador de voz en recolectar dentro de un solo paquete o trama de información, una cantidad determinada de muestras de voz de la conversación, una vez que han sido

previamente codificadas y comprimidas. El valor de este retardo depende del tipo de CODEC y del tamaño de la carga útil del paquete.

CODEC	**Carga Útil (Bytes)**	**Retardo de Paquetización (ms)**
G.711	160	20
	240	30
GSM	32.5	20

Tabla 4.2 Retardo de Paquetización

4.4.1.3 Retardo de Red

Este tipo de retardo está relacionado con los dispositivos intermedios que se utilizan para interconectar los *host* origen y destino de los datos; es de interés analizar el comportamiento a nivel de red del tiempo para transmitir un paquete, en una red basada en el protocolo IP; el retardo consiste entonces en el intervalo de tiempo que tarda un paquete, para ser enviado y recibido entre un *host* emisor y uno receptor.

Con el propósito de conocer este retardo se utiliza el comando ping como herramienta de medición. Dicho comando se encuentra disponible en la suite de programas utilitarios del protocolo TCP/IP, instalado en los sistemas operativos Windows y Linux que gestionan los recursos en los diferentes *hosts* y el servidor. Este comando permite obtener estadísticas de red tales como el retardo y la pérdida de paquetes utilizando para ello el envío de paquetes ICMP con un tamaño promedio de 32 bytes.

Los resultados obtenidos mediante este comando son diferentes a las mediciones obtenidas al analizar el tráfico de telefonía IP. Esto se debe a que el retardo de red está basado en el comportamiento de los dispositivos de interconexión tales como *routers* y *switches*. Estos dispositivos utilizan paquetes de prueba cuyo

comportamiento en cuanto a prioridad y tamaño son diferentes a los de una aplicación de voz IP.

Los paquetes de prueba son de tipo ICMP y proporcionan información de retardos solo a nivel de red, sin incluir procesamiento de aplicación de telefonía IP.

En un ambiente empresarial los paquetes de VoIP poseen un nivel de prioridad más alto, por lo cual se da preferencia a su procesamiento que un paquete de prueba.

Sin embargo los paquetes de prueba permiten tener una estimación a nivel de capa red que posibilita la comparación frente a los retardos totales en cada una de las aplicaciones de telefonía IP dentro del prototipo.

Una de las limitaciones relacionadas al uso del comando ping es la precisión del tiempo de lectura y depende del tipo de sistema operativo con que funciona en el *host*. El valor de la resolución del reloj constituye el valor mínimo de retardo que se puede medir e influye en la obtención de una correcta evaluación del retardo. Por lo tanto se presenta a continuación los valores de precisión para los sistemas operativos involucrados en el prototipo en la tabla 4.3.

Sistema Operativo	**Precisión (ms)**
Windows	1
Linux	10

Tabla 4.3 Resolución de tiempo de lectura según el tipo de Sistema Operativo

Para explicar las mediciones se presenta a continuación un resumen tabulado de los retardos promedio, de acuerdo a los diferentes tipos de conexión posibles en el prototipo.

La ejecución de estas pruebas se detalla en la sección 4.1.1.

Tipo de Conexión	Host Origen	Host Destino	Retardo (ms)
IPv4 a IPv4 a nivel Local	172.31.18.60	172.31.18.41	<1
IPv6 a IPv6 a nivel Local	2000::AA:10	2000::AA:3	<1
IPv4 a IPv6 a nivel Local	172.31.18.60	2000::AA:3	<1
IPv6 a IPv4 a nivel Local	2000::AA:10	172.31.18.41	<1
IPv4 a IPv4 a nivel Remoto a través de la infraestructura NAT	200.200.6.10	199.99.9.33	20
IPv6 a IPv6 a nivel Remoto	2000::4C:30	2000::AA:3	82
IPv6 a IPv6 a nivel Remoto a través de un Túnel	2000::6E:2	2000::AA:20	35

Tabla 4.4Retardos de Red según el tipo de conexión

4.4.1.3.1 Análisis de resultados

Los resultados de la medición de retardos a nivel de red revelan que el mayor valor de retardo corresponde a 82 ms. Este valor se presenta en una conexión IPv6 a IPv6 remota debido a que un paquete atraviesa en esta ruta un total de 4 *router*s para llegar hasta su destino.

Por otro lado el retardo mínimo tiene un valor menor a 1 ms y se presenta en conexiones a nivel local. El valor de esta medición no es exacto puesto que es menor a la precisión del tiempo de lectura del comando ping en Windows y se presenta en todas las conexiones en las que no se atraviesan dispositivos de encaminamiento.

Adicionalmente se puede apreciar que el retardo a nivel de red para que un paquete viaje entre un *host* de tipo IPv4 a uno de tipo IPv6 o viceversa, a través de la misma ruta, poseen el mismo valor.

4.4.2 RETARDO DE PAQUETES DE VOIP

Este tipo de retardo está relacionado con el valor total de tiempo que tarda un paquete de VoIP en viajar desde el *host* emisor o llamante hasta llegar al *host* receptor o destino.

Incluye todos los retardos analizados en las secciones anteriores, adicionando el procesamiento del servidor y las características de cada una de las aplicaciones de telefonía IP. Es decir es el retardo total de extremo a extremo en una comunicación de tiempo real.

Para realizar la medición de este retardo se utiliza al *software* analizador de tráfico Wireshark como un *sniffer*. El *software* permite realizar capturas de los paquetes de voz bajo el protocolo RTP y analizar en tiempo real los retardos presentes en la comunicación. Mediante el sniffer se generan estadísticas con parámetros de telefonía IP de interés para realizar el cálculo del retardo total en cada tipo de llamada.

Wireshark permite obtener el valor del parámetro Delta[22] mediante el análisis de un *stream* RTP. Dicho parámetro representa el valor de retardo del tráfico de VoIP y debe interpretarse correctamente, puesto que mide el retardo entre diferentes puntos de la conexión que son de interés.

Primero se realiza una medición entre puntos intermedios que los paquetes atraviesan para llegar a su destino. Posteriormente mediante la suma acumulativa de los retardos intermedios se puede obtener el retardo total extremo a extremo, entre el *host* origen y el *host* destino.

Se presenta a continuación un ejemplo de cálculo del retardo total en base a los valores de Delta obtenidos con el *sniffer* y un diagrama que muestra los respectivos puntos intermedios a lo largo de la conexión.

Los resultados finales según el tipo de llamada se resumen en la tabla 4.6.

[22] Delta: Diferencia entre el tiempo de sello de un paquete RTP y el tiempo de llegada.

Como ejemplo se analiza la llamada de tipo IPv6 a IPv4 entre el *host* 2000::4C:20 y el *host* 200.200.6.10, atravesando la infraestructura NAT que será transparente al *sniffer*. Los diferentes puntos intermedios se identifican en la figura 4.30

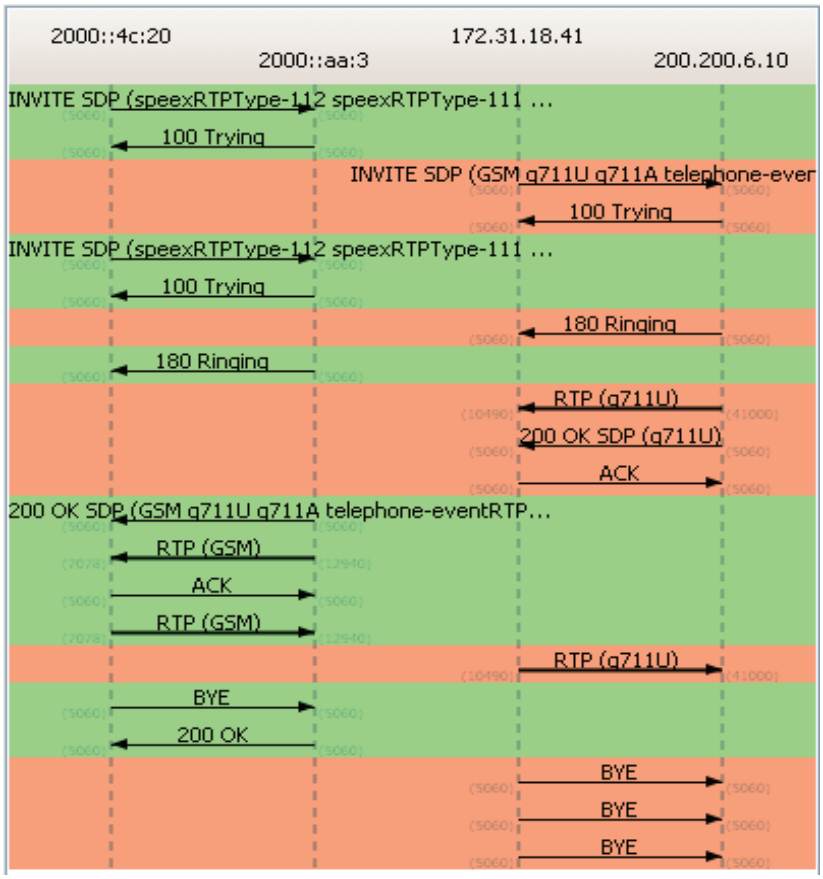


Figura 4.30 Diagrama de flujos de tráfico de paquetes de VoIP

La llamada se establece mediante el protocolo SIP para proceder a un intercambio de paquetes RTP utilizando diferentes tipos de codecs.

El gráfico de la figura 4.30 representa un análisis de flujos que permite identificar los diferentes puntos intermedios que en este caso corresponden a las direcciones IPv6 e IPv4 de la NIC del servidor.

Mediante el análisis del esquema de flujos se observa que los paquetes RTP que poseen las muestras de voz digitalizadas se envían desde uno de los *hosts* al servidor, estos paquetes utilizan el códec GSM en el caso de direcciones IPv6 y el códec G711u en el caso de IPv4.

Esto se debe a la configuración de la central telefónica que habilita el tipo de códec según el tipo de conexión; por lo cual los retardos de algoritmo y de paquetización inherentes a los códecs serán diferentes en cada extremo.

El segmento entre la dirección 2000::aa:3 y la dirección 173.31.18.41 corresponde al procesamiento de la central telefónica Asterisk. En este proceso se convierte la dirección IPv6 a IPv4 y se realiza un cambio en el tipo de códec utilizado.

Las dos direcciones corresponden a la misma interfaz física NIC del servidor por lo cual no existe intercambio de paquetes; sino un procesamiento realizado en base al *software* de Asterisk.

Es necesario a continuación analizar los *stream*s de RTP entre los diferentes segmentos. En cada *stream* se debe tomar los valores de retardo máximo puesto que se analiza la condición más crítica; posteriormente se debe sumar dichos valores respectivamente para encontrar el retardo extremo a extremo.

Se presenta en la figura 4.31 los valores obtenidos, mediante la herramienta *Stream* RTP del *sniffer* Wireshark.

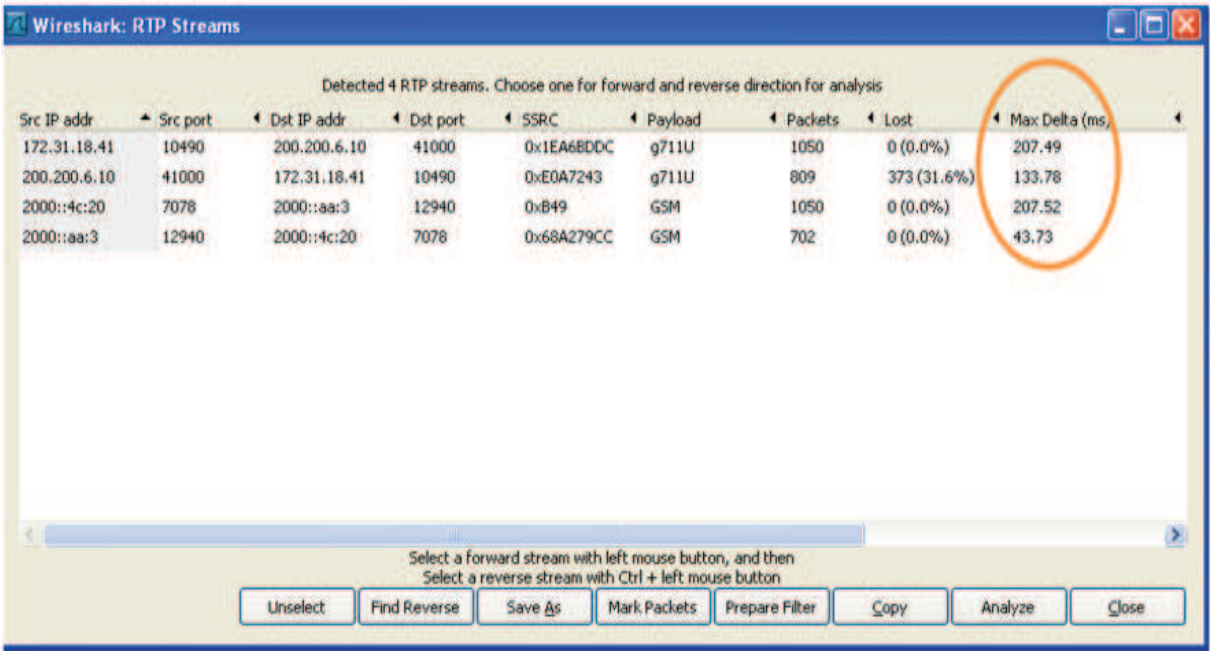


Figura 4.31 Valores máximos de Retardo en un *Stream* RTP

El cálculo del retardo total se realiza mediante la suma de los retardos intermedios y se presenta en la tabla 4.5.

Origen	Destino	Retardo (ms)
2000::4c:20	2000::aa:3	207,52
172.31.18.41	200.200.6.10	207,49
Retardo Total		**415,01**

Tabla 4.5 Cálculo de Retardo Total en Paquetes de VoIP

Dentro de las pruebas realizadas el softphone Linphone permite verificar como el retardo afecta en el MOS de cada una de las llamadas. El MOS permite la determinación del grado de satisfacción de los clientes en relación al retardo.

Se puede observar el valor del medidor de MOS en la interfaz gráfica de Linphone tal como se muestra en la figura 4.32.

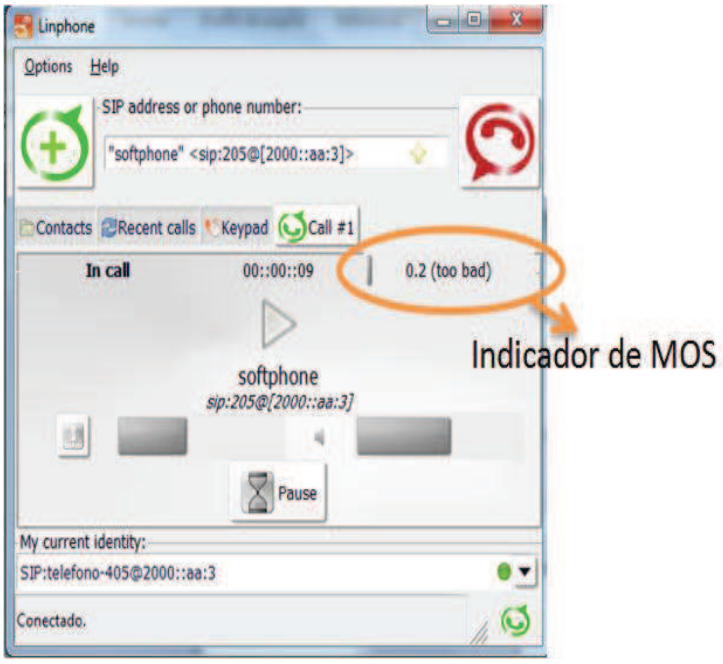


Figura 4.32 Indicador de MOS de Linphone

Finalmente se presenta en la tabla 4.6 los retardos totales que sufren los paquetes de VoIP en base a los cálculos mostrados en el ejemplo anterior.

Tipo de Llamada	Host Origen	Host Destino	Origen a Servidor (ms)	Servidor a Destino (ms)	Retardo Total (ms)
IPv4 a IPv4 a nivel Local	172.31.18.69	172.31.18.60	21,08	23,97	45,05
IPv6 a IPv6 a nivel Local	2000::AA:60	2000::AA:20	51,01	51	102,01
IPv4 a IPv6 a nivel Local	172.31.18.58	2000::AA:20	25,54	25,56	51,1
IPv6 a IPv4 a nivel Local	2000::AA:20	172.31.18.58	50,21	50,18	100,39
IPv4 a IPv4 a nivel Remoto a través de la infraestructura NAT	172.31.18.58	200.200.6.20	29,86	79,9	109,76
IPv6 a IPv4 a nivel Remoto a través de la infraestructura NAT	2000::4C:20	200.200.6.10	207,52	207,49	415,01
IPv4 a IPv6 a nivel Remoto a través de la infraestructura NAT	200.200.6.10	2000::4C:20	43,20	43,21	86,41
IPv6 a IPv6 a nivel Remoto	2000::AA:20	2000::4C:20	56,20	56,15	112,35
IPv4 a IPv6 a nivel Remoto	172.31.18.58	2000::4C:20	30,74	42,67	73,41
IPv6 a IPv4 a nivel Remoto	2000::4C:20	172.31.18.58	160,64	160,65	321,29
IPv4 a IPv6 a nivel Remoto a través de un Túnel	172.31.18.58	2000::6E:2	22,41	22,39	44,8
IPv6 a IPv6 a nivel Remoto a través de un Túnel	2000::6E:2	2000::4C:20	78,68	78,68	157,36

Tabla 4.6 Retardos en paquetes VoIP según el tipo de llamada

4.4.2.1 Análisis de resultados

Los resultados obtenidos mediante las capturas y análisis de cada tipo de llamada permiten apreciar los retardos que se presentan y están diferenciados por segmentos entre diferentes puntos de interés.

Al descomponer el retardo total en retardos intermedios se observa que cuando una llamada es iniciada por un *host* IPv6 la conexión hacia el servidor utiliza el códec GSM. Esto se debe a que es la configuración por defecto que Asterisk permite en su versión 1.8 para soportar IPv6.

Mientras tanto para una conexión entre el servidor y un *host* IPv4 se utiliza el códec G.711u, debido a que se habilitó dentro de la configuración realizada por el administrador. Además se observa que cuando la llamada es iniciada por un *host* IPv4 se utiliza el códec G.711u durante todas las conexiones para establecer la llamada hasta su destino, sin importar si el *host* destino es de tipo IPv4 o IPv6.

Los retardos por algoritmo y paquetización mostrados en las tablas 4.1 y 4.2 son diferentes para cada tipo de códec. Por esta razón el retardo desde un *host* IPv6 al servidor es mayor al retardo entre un *host* IPv4 y el servidor.

Adicionalmente se puede apreciar que la interfaz NIC del servidor posee dos direcciones IP, una para cada versión del protocolo. La central Asterisk introduce un retardo por el procesamiento que realiza su doble pila debido a la conversión de direcciones IPv6 a IPv4 y viceversa. Esta conversión se produce en base a sockets y se incluye en este procesamiento el cambio del tipo de códec de acuerdo al tipo de conexión.

En base a los resultados se observa que el retardo es mínimo al tratarse de llamadas entre *host* con el mismo tipo de direcciones. Se presenta un valor mayor en el caso de llamadas entre direcciones IPv6; ya que comparando el tipo de llamadas a nivel local el valor de tipo IPv6 corresponde a 102,01 ms, con una amplia diferencia frente al valor de tipo IPv4 de 45,05 ms. Esto se debe a que el retardo en los codecs GSM (usado en IPv6) es mucho mayor al retardo del G.711 (usado en IPv4); además el tiempo que tarda Asterisk en el procesamiento de direcciones IPv6 es mayor al de procesar IPv4.

Se puede además comprobar que el retardo posee un mayor valor cuando las llamadas se producen entre *host*s con diferente tipo de direcciones. El valor es mayor en el caso de una llamada de tipo IPv6 a IPv4. Esto se puede apreciar considerando el caso más crítico en la llamada a través de la infraestructura NAT; esta llamada tiene un valor de 415,01 ms debido al procesamiento de la central IP PBX para convertir el tipo de dirección, cambiar el tipo de códec y el procesamiento en los *router*s necesario para atravesar toda la ruta.

Mientras en el caso de una llamada de IPv4 a IPv6 el valor de retardo es menor; esto se puede apreciar considerando la condición más crítica en la llamada a través de la misma infraestructura NAT en donde el retardo tiene un valor de 86,41 ms.

Existe diferencia entre estos dos valores a pesar de que el tipo de direcciones entre los llamantes es similar. Esto se debe a que el tiempo de procesamiento dentro del servidor Asterisk para convertir direcciones de tipo IPv6 a IPv4 es mayor al necesario para convertir direcciones de IPv4 a IPv6.

A partir de los resultados anteriores se puede concluir que la calidad entre llamadas de IPv6 a IPv4 en el caso de la infraestructura NAT y las llamadas de IPv6 a IPv4 a nivel remoto son inapropiadas para una comunicación de tiempo real. Esto se debe a que los dos tipos de llamada superan el máximo tolerado de 150 ms establecido en la norma G.114 de la ITU.

4.5 MEDICIÓN DE JITTER

El contexto utilizado para realizar la medición del parámetro de *jitter* consiste en analizar cada una de las distintas clases de llamadas posibles de acuerdo al tipo de conexión. Se emplea la herramienta wireshark para examinar el valor de *jitter* que se produce en cada uno de los paquetes pertenecientes a una llamada.

Se presenta como resultado de la prueba los valores promedios para todo el conjunto de paquetes pertenecientes a una misma llamada; esta estadística es generada mediante el *sniffer* wireshark.

Con la finalidad de obtener los valores de *jitter* necesarios para realizar el análisis de este parámetro se elige en el sniffer la siguiente herramienta:

telephony/RTP/show all streams.

La herramienta del *sniffer* despliega una ventana en la cual se puede observar los dos canales que se establecen en la llamada (origen –servidor y servidor-destino). Para cada canal se presentan una serie de estadísticas, entre las cuales se selecciona el *jitter* promedio (para establecer el análisis de acuerdo a la condición más general).

En la figura 4.33 se tiene un ejemplo de la obtención de datos de *jitter* en una llamada SIP IPv4 a IPv6 desde el *host* 200.200.6.10 hasta el *host* 2000::4C:20.

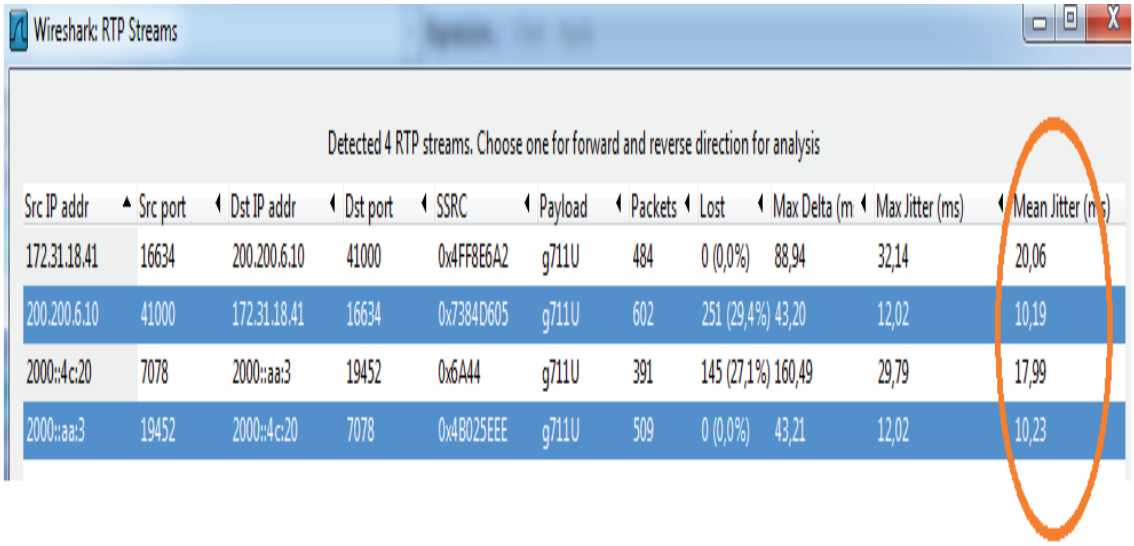


Figura 4.33 Procedimiento para obtención de datos de jitter en wireshark

Los resultados de los valores de jitter para los distintos tipos de llamadas se presentan en la tabla 4.7.

TIPO DE LLAMADA	Host Origen	Host Destino	Canal	JITTER [ms]
SIP-SIP misma LAN	172.31.18.69	172.31.18.60	Origen-Servidor	0.04
			Servidor - Destino	0.07
SIP NAT IPv4 – SIP IPv6 REMOTA	200.200.6.10	2000::4C:20	Origen-Servidor	10.19
			Servidor - Destino	10.23
SIP IPV6 REMOTA - SIP IPV4 NAT	2000::4C:20	200.200.6.10	Origen-Servidor	13.15
			Servidor - Destino	10.19
IPv6 REMOTA - IPv6 LOCAL	2000::4C:20	2000::AA:20	Origen-Servidor	6.10
			Servidor - Destino	6.10
IPv4 a IPv6 a través de túnel	172.31.18.58	2000::6E:2	Origen-Servidor	0.25
			Servidor - Destino	0.26

Tabla 4.7 Datos de Jitter obtenidos del prototipo

4.5.1 ANÁLISIS DE RESULTADOS

El *jitter* se describe como la variación en el tiempo de llegada de los paquetes y es un efecto de las redes de datos no orientadas a conexión y basadas en conmutación de paquetes. Esta razón justifica la existencia de jitter en las comunicaciones que se presentan en el prototipo debido a que la telefonía IP se incluye dentro de las comunicaciones por conmutación de paquetes.

Los resultados obtenidos con wireshark muestran el valor de la desviación media de la diferencia de retardo. Esta desviación se presenta en el espaciado entre dos paquetes en destino respecto al espaciado en origen; por esta razón se detallan los valores tanto en el sentido de canal de ida, como en el de vuelta.

Para tener una calidad de voz aceptable en un sistema de telefonía IP el *jitter* debe tener un valor menor a 50 ms. Por lo que se puede concluir de acuerdo a los datos de la tabla 4.7, que el prototipo implementado no tendrá inconveniente alguno en cuanto a calidad de voz en las llamadas.

4.6 ANCHO DE BANDA

El escenario de estas pruebas consiste en obtener datos sobre parámetros de ancho de banda mediante la herramienta de análisis de tráfico wireshark. Para lo cual se realizan las diferentes llamadas tanto en ambientes mixtos como en ambientes nativos.

Se asume como valores reales de ancho de banda los datos obtenidos mediante wireshark obtenidos en base a la implementación física del prototipo.

Los valores teóricos se obtienen mediante una herramienta de cálculo de ancho de banda para VoIP disponible en línea mediante Internet. Se utiliza como herramienta la calculadora de ancho de banda disponible en el foro de VoIP.

La herramienta en línea calcula el ancho de banda de un determinado número de llamadas simultáneas mediante la suma de los valores de ancho de banda de cada canal (entrada y salida) implicado en una llamada.

Se realiza este cálculo de acuerdo a la selección del tipo de códec a utilizar, el número de llamadas simultáneas, protocolo de comunicación (SIP, IAX2, MGCP, H323). La herramienta establece anchos de banda fijos para los protocolos implicados en cada llamada como lo son: UDP, RTP, IP.

Mediante esta herramienta se tiene que el ancho de banda para un canal que utiliza un códec GSM es de ***28.63 Kbps*** resumidos de acuerdo a la figura 4.34.

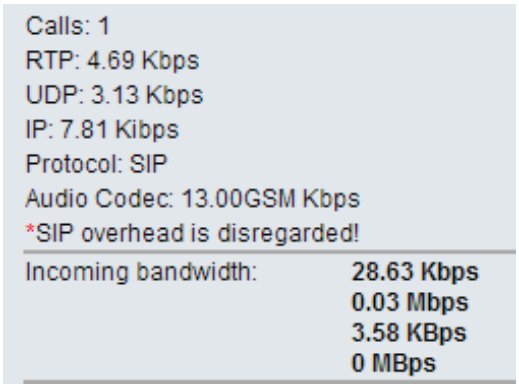


Figura 4.34 Ancho de banda para un canal con códec GSM

Mientras tanto el ancho de banda de un canal que utiliza un códec G711, es de ***79.63 Kbps***; resumidos de acuerdo a la figura 4.35.

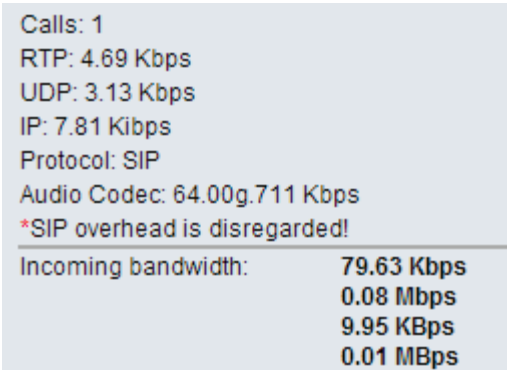


Figura 4.35 Ancho de banda para un canal con códec G711

Para obtener los valores reales en el prototipo es necesario determinar el ancho de banda con el sniffer; para lo cual se debe realizar la suma de dos valores de anchos de banda parciales. Estos valores se obtienen del análisis con wireshark, en donde el primer valor corresponde al canal de entrada y el segundo al canal de salida.

Dichos valores de ancho de banda dependen de los códecs que utiliza Asterisk para el prototipo diseñado incluyendo GSM y G711. Los valores de ancho de banda que están cercanos a 80 Kbps corresponden al códec G711 (64 kbps corresponden al códec, 4.69 kbps son ocupados por RTP, 3.13 kbps son ocupados por UDP y 7.81 kbps ocupados por IP)

Mientras tanto los valores que están alrededor de 30 Kbps (13 kbps correspondientes al códec, 4.69 kbps ocupa RTP, 3.13 kbps ocupados por UDP y 7.81 kbps ocupados por IP) corresponden al códec GSM.

Para la obtención de los parámetros de ancho de banda en wireshark se elige la siguiente herramienta: ***telephony/RTP/show all streams***. En dicha herramienta se presenta una pantalla en la cual se indican los dos canales de comunicación

(origen –servidor y servidor-destino); posteriormente se analiza cada canal y se escoge el valor más crítico de ancho de banda.

En la figura 4.36 se presenta un ejemplo de la obtención de los valores de ancho de banda en cada canal. Los valores corresponden a la ejecución de una llamada SIP IPv6 a IPv6 que utiliza un códec GSM.

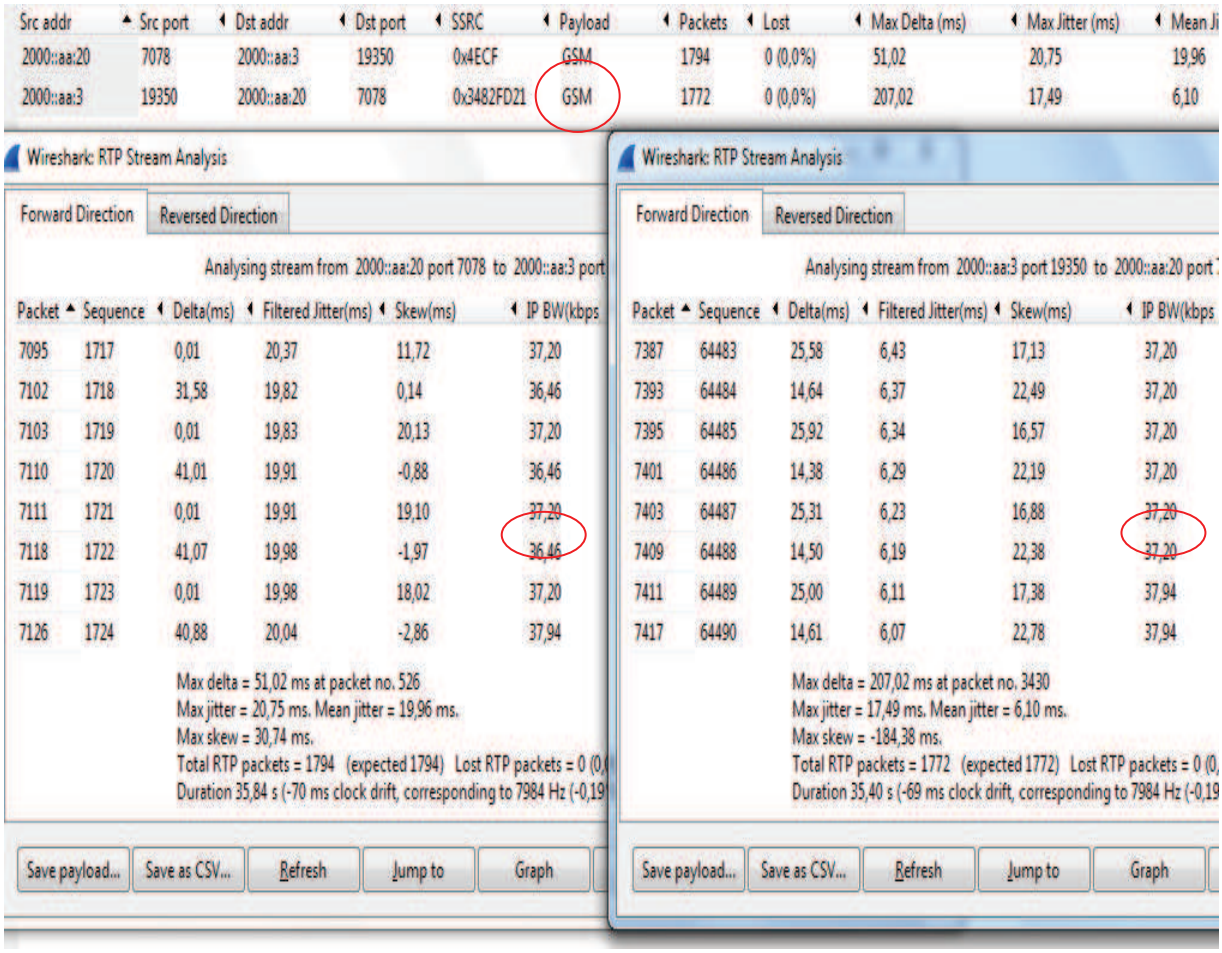


Figura 4.36 Ejemplo de obtención de parámetros de ancho de banda en wireshark

Para las pruebas de ancho de banda se emplea los siguientes dispositivos como origen y destino de llamadas:

TIPO DE LLAMADA	HOST ORIGEN	HOST DESTINO
SIP-SIP misma LAN	172.31.18.69	172.31.18.65
SIP NAT IPv4 – SIP IPv6	200.200.6.10	2000::4C:20
IPV6 A IPV4	2000::4C:20	172.31.18.58
IPv6 a IPv6	2000:AA:20	2000:AA:3
IPv6 a IPv4 a través de túnel	2000::6E:2	172.31.18.58

Tabla 4.8 Anchos de banda obtenidos de acuerdo a cada tipo de llamada

Los resultados para cada tipo de llamada se presentan en la tabla 4.9.

TIPO DE LLAMADA	CÓDEC CANAL DE ENTRADA	CÓDEC CANAL DE SALIDA	AB PARCIALES POR CANAL [Kbps]	AB TOTAL [Kbps]
SIP-SIP misma LAN	GSM	GSM	29.78 + 29.78	59.56
SIP NAT IPv4 – SIP IPv6	G711	G711	52.8 + 58.08	110.88
IPV6 A IPV4	GSM	G711	38.69 + 83.20	121.89
IPv6 a IPv6	GSM	GSM	37.94 + 37.94	75.88
IPv6 a IPv4 a través de túnel	GSM	G711	29.02 + 72	101.02

Tabla 4.9 Anchos de banda obtenidos de acuerdo a cada tipo de llamada

En la figura 4.37 se presenta el valor teórico de ancho de banda que ocuparía una llamada que utiliza un códec G.711 a la entrada y un códec GSM a la salida o viceversa (llamada IPv6 a IPv4 para este caso).

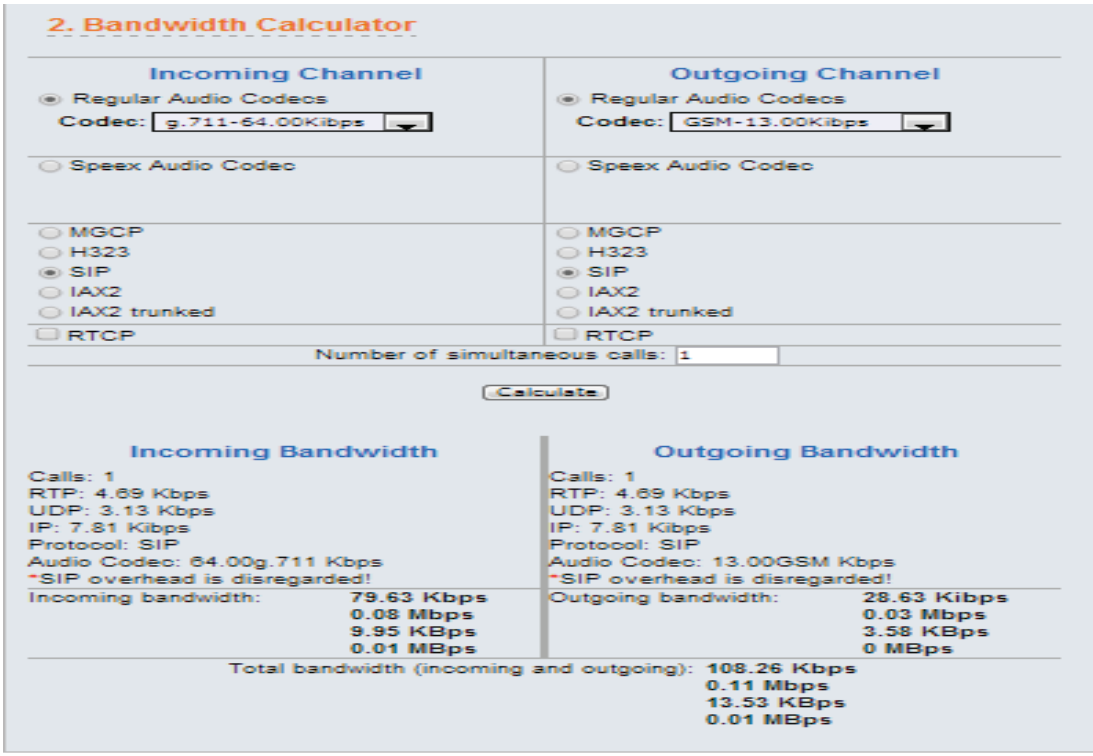


Figura 4.37 Ancho de banda teórico para una llamada con códec GSM y G711

En la figura 4.38 se presenta una gráfica del ancho de banda frente al tiempo en base a los resultados obtenidos en una llamada SIP.

Los valores de esta llamada IPv4 (detrás de NAT) a IPv6 corresponden al canal entre el servidor y el cliente IPv4.

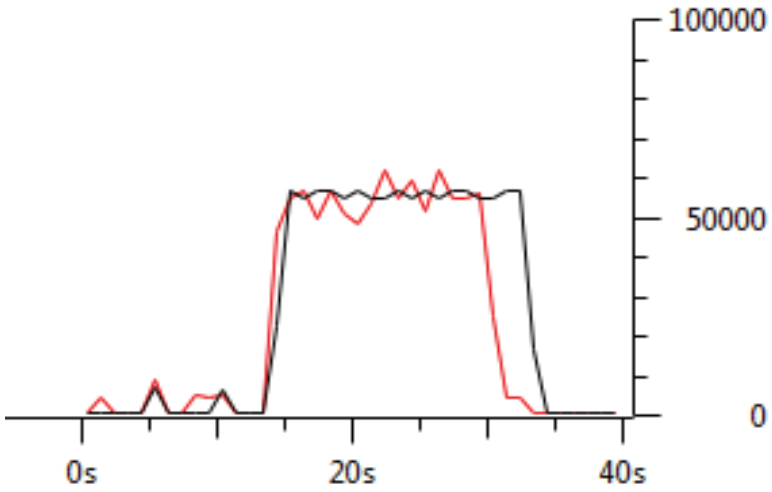


Figura 4.38 Ancho de banda del canal en una llamada SIP IPv4-Ipv6

Se observa en la figura 4.38 que el ancho de banda en el canal de entrada y en el canal de salida es de alrededor de 55 kbps (no se puede establecer el valor exacto debido a la escala). La llamada utiliza un códec G711 tanto en el canal de entrada como en el de salida, estos valores son semejantes a los establecidos en la tabla 4.9.

4.6.1 ANÁLISIS DE RESULTADOS

El ancho de banda total se calcula sumando los anchos de banda de los dos canales que intervienen en la llamada. Además se puede establecer que los parámetros que intervienen en el ancho de banda de cada canal son: tipo de códec a utilizar, número de llamadas simultáneas, protocolo de comunicación (SIP, IAX2, MGCP, H323); así como los anchos de banda de UDP, RTP e IP.

Se analiza los resultados obtenidos realizando una comparación con los valores de ancho de banda que se consideran teóricos.

En base a esta comparación se observa que los valores de ancho de banda obtenidos (reales) luego de realizar las pruebas superan con poco a los valores teóricos. Se presentan los errores calculados en la tabla 4.10.

TIPO DE LLAMADA	CÓDEC CANAL DE ENTRADA	CÓDEC CANAL DE SALIDA	Valor Teórico AB [Kbps]	Valor Real AB [Kbps]	Error Relativo [%]
SIP-SIP misma LAN	GSM	GSM	57.26	59.56	4
SIP NAT IPv4 – SIP IPv6	G711	G711	159.26	110.88	30.37
IPV6 A IPV4	GSM	G711	108.26	121.89	12.59
IPv6 a IPv6	GSM	GSM	57.26	75.88	32.51
IPv6 a IPv4 a través de túnel	GSM	G711	108.26	101.02	6.68

Tabla 4.10 Tasas de error porcentuales de ancho de banda

Se puede observar que los errores en cuanto a medición del ancho de banda no son críticos, excepto para los tipos de llamada en los cuales se tiene un error mayor al 30%. Estos altos valores se justifican debido a que en dichas llamadas intervienen IPv6 y NAT, los parámetros inmersos en los dos mecanismos no son considerados en la calculadora de ancho de banda con la cual se toman los valores teóricos.

Es decir que el ancho de banda de una llamada en la cual interviene IPv6 tendrá un mayor valor real que una llamada en la que solo se tengan redes IPv4.

Adicionalmente al tener implementado NAT en el sistema se tiene un mayor ancho de banda, el mismo que no es considerado por la herramienta de análisis utilizada para referirnos a los valores teóricos.

4.7 PRUEBAS DE RENDIMIENTO

Con esta prueba se verifica la robustez del servidor en cuanto a su capacidad de memoria para gestionar la demanda de solicitudes de clientes. Es decir así se establecerá el punto límite en el cual el servidor trabaja con buenas condiciones para ofrecer los servicios. El *software* generador de tráfico simula el incremento de la carga del sistema.

4.7.1 EJECUCIÓN DE LA PRUEBA

Para realizar la prueba de denegación de servicio al servidor Asterisk se ejecuta el ataque desde el cliente SIPP mediante los siguientes parámetros:

```
sipp -sn uac 172.31.18.41 –s 1234 –d 30000 –m 1000 –r 100 –l
                1000 –i 172.31.18.60 –trace_err
```

Dónde:

- El parámetro ***–sn*** permite usar uno de los escenarios predefinidos de sipp, los más usados son UAS (servidor) y UAC (cliente).

- El parámetro ***–s*** indica la extensión destino a llamar
- El parámetro ***–d*** se especifica la duración de la llamada en milisegundos.
- El parámetro ***–m*** indica el número total de llamadas que el cliente va a ejecutar.
- El parámetro ***–r*** establece el número de llamadas por segundo.
- El parámetro ***–l*** establece el número máximo de llamadas simultáneas que el cliente puede mantener.
- El parámetro ***–i*** permite especificar la dirección IP de origen de la llamada.
- El parámetro ***–trace_err*** guarda el log de estadísticas en pantalla de la ejecución de la prueba visualizando si existieran errores de la misma.

De esta manera en esta prueba se generarán 1000 llamadas con una duración de 30 segundos cada llamada, con una tasa de 100 llamadas cada segundo.

La duración de las llamadas generadas debe ser elevada, de tal manera que el número de llamadas que se crean por segundo sea mayor que las que se terminan. De otro modo será imposible encontrar la capacidad máxima del servidor.

4.7.2 RESULTADOS

Se corre la prueba tres veces con diferentes tasas de llamada obteniendo como resultado que el servidor soporta 600 llamadas simultáneas. Cada llamada tiene una duración de 30 segundos, luego de lo cual el servidor deja de responder a las solicitudes SIP generadas por el cliente.

En la figura 4.39 se puede observar mediante la herramienta gnome-system-monitor de Linux, que el servidor colapsa luego de 6 segundos con una tasa de 100 llamadas por segundo por lo que el ***servidor atiende a un máximo de 600 llamadas simultáneas.***

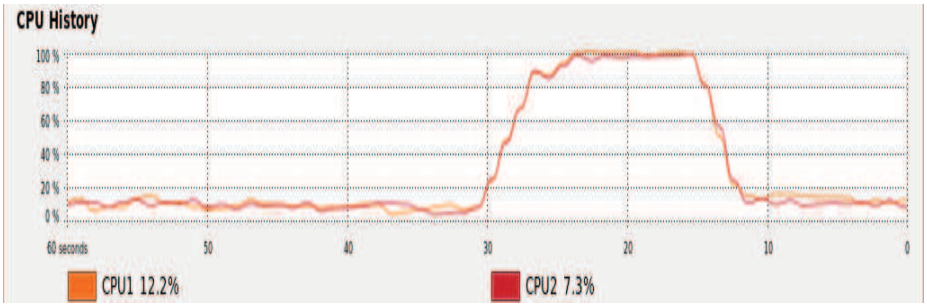


Figura 4.39 Resultados de Gnome-System-Monitor al realizar la prueba

Los resultados de la ejecución de la prueba, se presentan en la figura 4.40.

```
C:\cygwin\Sipp_3.2>sipp -sn uac 172.31.18.41 -s 1234 -d 30000 -m 1000 -r 100 -l
1000 -i 172.31.18.60
Warning: open file limit > FD_SETSIZE; limiting max. # of open files to FD_SETSI
ZE = 64
       Resolving remote host '172.31.18.41'... Done.
------------------------------ Scenario Screen --------- [1-9]: Change Screen --
  Call-rate(length)   Port   Total-time  Total-calls  Remote-host
100.0(30000 ms)/1.000s   5060      59.73 s         1000  172.31.18.41:5060(UDP)

  Call limit reached (-m 1000), 0.000 s period  0 ms scheduler resolution
  0 calls (limit 1000)                 Peak was 988 calls, after 10 s
  0 Running, 964 Paused, 0 Woken up
  765 dead call msg (discarded)        280 out-of-call msg (discarded)
  1 open sockets

                                 Messages  Retrans   Timeout   Unexpected-Msg
      INVITE ----------->        1000      993       130
         100 <-----------        870       0         0         0
         180 <-----------        0         0         0         0
         183 <-----------        0         0         0         0
         200 <-----------  E-RTD1 870       1053      0         0
         ACK ----------->        870       1053
       Pause [    30.0s]        870                           31
         BYE ----------->        839       4583      0
         200 <-----------        807       0         0         32

------------------------------ Test Terminated --------------------------------


------------------------------ Statistics Screen -------- [1-9]: Change Screen --
  Start Time                | 2013-06-19    16:21:17:980    1371676877.980180
  Last Reset Time           | 2013-06-19    16:22:17:733    1371676937.733180
  Current Time              | 2013-06-19    16:22:17:742    1371676937.742180
--------------------------+---------------------------+---------------------------
  Counter Name              | Periodic value            | Cumulative value
--------------------------+---------------------------+---------------------------
  Elapsed Time              | 00:00:00:009              | 00:00:59:762
  Call Rate                 |    0.000 cps              |   16.733 cps
--------------------------+---------------------------+---------------------------
  Incoming call created     |        0                  |        0
  OutGoing call created     |        0                  |     1000
  Total Call created        |                           |     1000
  Current Call              |        0                  |
--------------------------+---------------------------+---------------------------
  Successful call           |        0                  |      807
  Failed call               |        0                  |      193
--------------------------+---------------------------+---------------------------
  Response Time 1           | 00:00:00:000              | 00:00:00:726
  Call Length               | 00:00:00:000              | 00:00:42:245
------------------------------ Test Terminated --------------------------------
```

Figura 4.40 Indicador de estadísticas de SIPP

En la figura 4.40 se observa que el cliente generó 1000 llamadas de las cuales 807 fueron realizadas con éxito y se tiene 193 llamadas que no se concretaron. Se observa además que a los 30 segundos se corta la llamada y se termina mediante un mensaje de BYE. Los resultados de las tres pruebas realizadas se muestran en la tabla 4.11.

	Tasa de llamadas	**Número de llamadas generadas**	**Número de Llamadas simultáneas soportadas por el servidor**	**Número de llamadas exitosas**	**Número de llamadas fallidas**	**Tiempo luego del cual colapsa el servidor**
PRUEBA I	100 cps	1000	600	807	193	6 seg.
PRUEBA II	50 cps	1000	600	684	316	12 seg.
PRUEBA III	25 cps	1000	600	936	64	24 seg.

Tabla 4.11 Resumen de Pruebas de rendimiento del Servidor

4.7.2.1 Análisis de los Resultados

Los resultados tabulados anteriormente y la experiencia de la generación de la prueba hacia el servidor, indican claramente las condiciones para soportar un determinado tráfico y evitar que se degenere o se caiga un sistema de comunicación (de telefonía en este caso).

En un ambiente a nivel empresarial es necesario realizar un estudio que identifique y permita obtener los datos suficientes para dimensionar tanto la red de comunicación así como el servidor de aplicaciones y tener así una disponibilidad cercana al 100 %. Se debe plantear además un esquema de seguridad de red para que no se permitan ataques malintencionados que provoquen el colapso del sistema de comunicaciones.

En cuanto al prototipo implementado se debe tomar en cuenta que se trabaja a nivel de laboratorio y con usuarios “ficticios” (existen las terminales, pero no

personas que las ocupen). Es por esta razón que no se realiza un dimensionamiento en cuanto al tráfico a cursar por la red, pero se realiza esta prueba de denegación de servicio con el fin de advertir los peligros existentes.

4.8 PRUEBAS DE PÉRDIDA DE PAQUETES

El esquema de pruebas para el parámetro de pérdida de paquetes consiste en analizar el comportamiento del prototipo con la ayuda del sniffer Wireshark durante condiciones normales y en denegación de servicio.

Durante el normal funcionamiento idealmente no existe pérdida de paquetes por lo tanto es necesario analizar el funcionamiento del prototipo durante el estado de denegación de servicio. Para esto se utiliza la aplicación SIPP (que se describe en la sección 3.3.2.5.) se genera tráfico con paquetes SIP y RTP, de esta manera se origina llamadas hacia el servidor Asterisk con lo cual se consumen los recursos de procesamiento y canales dentro del servidor.

Al analizar cada tipo de llamada en condiciones normales se puede observar en la herramienta de análisis de tráfico RTP, que no existe pérdida de paquetes para ninguno de los tipos de llamadas que se pueden realizar en el prototipo.

Por otro lado al ejecutar la aplicación SIPP el servidor se congestiona ya que excede su capacidad para atender y procesar las llamadas. Para este efecto se genera un total de 1000 llamadas con una tasa de 100 llamadas por segundo y una duración de 30 segundos. Se considera que el valor para cada uno de estos parámetros permite realizar el análisis en una condición crítica de denegación del servicio. Bajo esta condición al realizar llamadas desde alguno de los *host*s de telefonía IP del prototipo la llamada no puede completarse.

Se muestra el resultado del monitor de procesamiento en el servidor con lo cual se puede constatar que se excedió su capacidad, el sistema está congestionado y existirán pérdidas de paquetes.

Este resultado del proceso de colapso del servidor es fácil de apreciar en las figuras 4.41 y 4.42, en las cuales se observa el proceso en que el uso del CPU y

de memoria llegan a su límite poco a poco luego de la generación de las diversas llamadas.

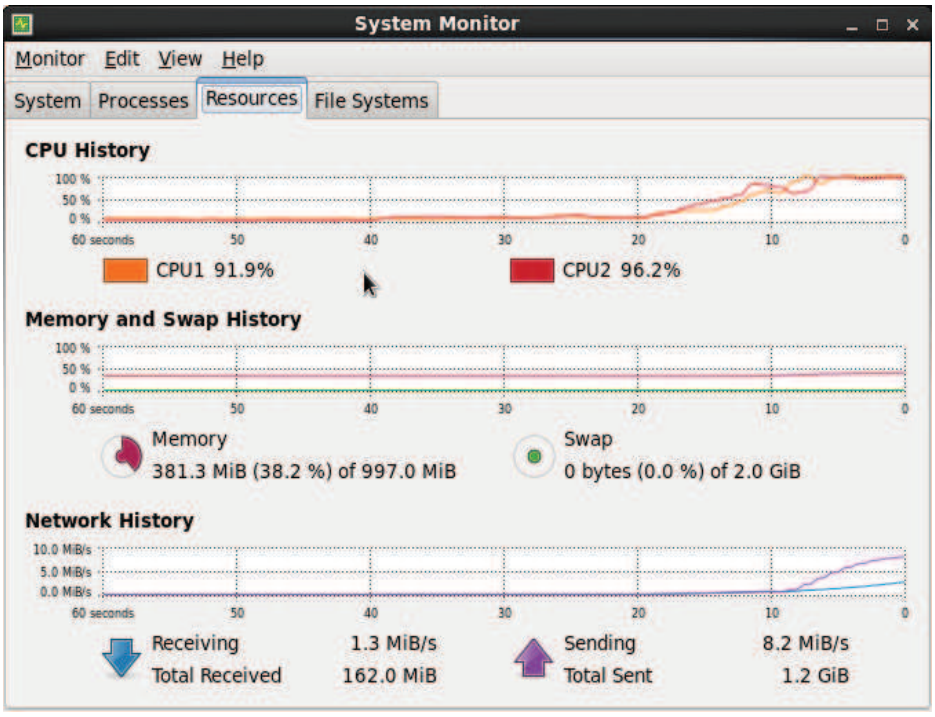


Figura 4.41 Monitor del Sistema antes de colapsar

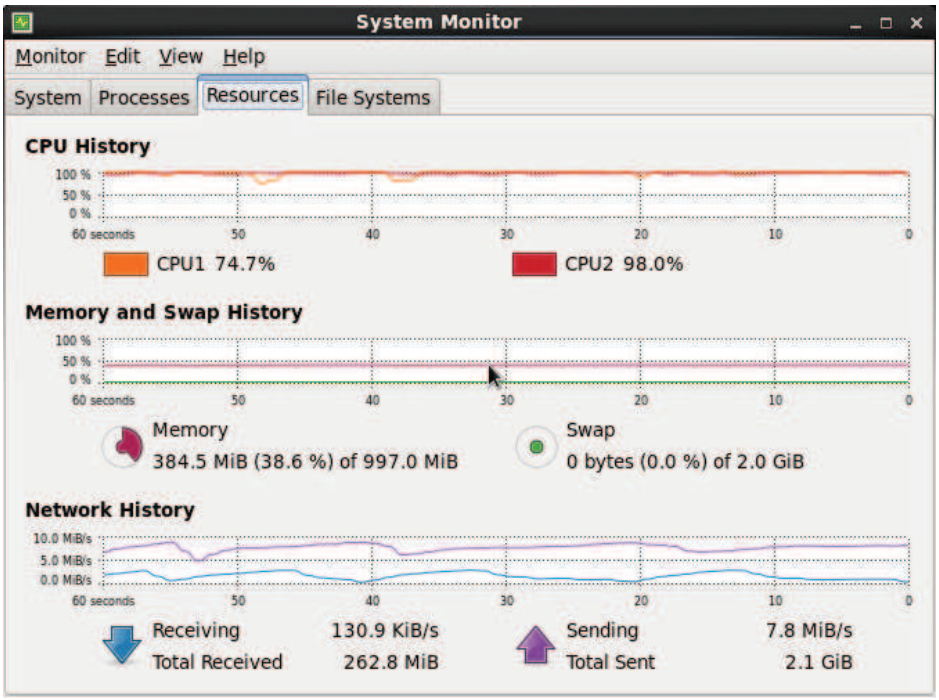


Figura 4.42 Monitor del Sistema durante la congestión

Una vez que se conoce los resultados y valores de la aplicación se puede analizar la captura en tiempo real de todo el tráfico mediante el sniffer Wireshark.

Para ello se utiliza la herramienta de análisis de *stream*s RTP que permite visualizar el valor de paquetes perdidos al enviar el tráfico telefónico desde el *host* 172.31.18.60 hasta el servidor. Se muestra a continuación un ejemplo de la captura de pérdida de paquetes y un resumen con los resultados presentados.

Wireshark: RTP Streams

Detected 900 RTP streams. Choose one for forward and reverse direction for analysis

Src IP addr	Src port	Dst IP addr	Dst port	SSRC	Payload	Packets	Lost	Max Delta (ms)
172.31.18.41	16428	172.31.18.60	6000	0x1F293DBB	g711U	401	74 (15.6%)	199.98
172.31.18.41	19208	172.31.18.60	6000	0x5E40C699	g711U	470	5 (1.1%)	40.35
172.31.18.41	17926	172.31.18.60	6000	0x3EB2CC2B	g711U	470	6 (1.3%)	79.93
172.31.18.41	13088	172.31.18.60	6000	0x4405F6A4	g711U	472	4 (0.8%)	40.97
172.31.18.41	15302	172.31.18.60	6000	0x192F54A8	g711U	472	4 (0.8%)	40.00
172.31.18.41	16138	172.31.18.60	6000	0x7ABE47F2	g711U	473	3 (0.6%)	40.02
172.31.18.41	13348	172.31.18.60	6000	0x2490C58D	g711U	472	4 (0.8%)	40.09
172.31.18.41	19994	172.31.18.60	6000	0x28178F21	g711U	474	2 (0.4%)	40.01
172.31.18.41	12968	172.31.18.60	6000	0x1EB0BC91	g711U	469	7 (1.5%)	60.37
172.31.18.41	11392	172.31.18.60	6000	0x6DBD8F27	g711U	448	28 (5.9%)	60.08
172.31.18.41	14544	172.31.18.60	6000	0x6BC2CF92	g711U	468	8 (1.7%)	40.68
172.31.18.41	11178	172.31.18.60	6000	0x3FB92E47	g711U	447	29 (6.1%)	120.13
172.31.18.41	11818	172.31.18.60	6000	0x476E1D89	g711U	434	42 (8.8%)	120.17
172.31.18.41	15682	172.31.18.60	6000	0x182FA8F6	g711U	440	36 (7.6%)	120.17
172.31.18.41	12100	172.31.18.60	6000	0x30BBC698	g711U	405	70 (14.7%)	219.94
172.31.18.41	14306	172.31.18.60	6000	0x2950DC27	g711U	471	5 (1.1%)	40.14
172.31.18.41	12484	172.31.18.60	6000	0xEE0DC48	g711U	473	3 (0.6%)	40.50
172.31.18.41	17412	172.31.18.60	6000	0xD4F26EB	g711U	464	12 (2.5%)	40.72
172.31.18.41	14890	172.31.18.60	6000	0x49D7B230	g711U	472	4 (0.8%)	40.32
172.31.18.41	10700	172.31.18.60	6000	0x135B645	g711U	471	5 (1.1%)	40.39
172.31.18.41	16532	172.31.18.60	6000	0x20611EC3	g711U	382	94 (19.7%)	259.85
172.31.18.41	12086	172.31.18.60	6000	0x6AFC59F9	g711U	455	21 (4.4%)	60.04
172.31.18.41	16174	172.31.18.60	6000	0x44E80BFC	g711U	463	13 (2.7%)	60.75
172.31.18.41	13162	172.31.18.60	6000	0x260A6583	g711U	464	12 (2.5%)	40.40
172.31.18.41	11014	172.31.18.60	6000	0x9DA774D	g711U	471	5 (1.1%)	60.31
172.31.18.41	13310	172.31.18.60	6000	0x35F3057A	g711U	468	8 (1.7%)	40.50
172.31.18.41	16062	172.31.18.60	6000	0x5C196438	g711U	467	9 (1.9%)	40.29
172.31.18.41	15242	172.31.18.60	6000	0x21168F5F	g711U	448	28 (5.9%)	100.00
172.31.18.41	16644	172.31.18.60	6000	0x551D40C2	g711U	471	5 (1.1%)	40.20

Figura 4.43 Captura de *stream*s RTP generados con SIPP

En primera instancia podemos detectar un total de 900 *stream*s de paquetes RTP, posteriormente es posible examinar en cada uno de los *stream*s: el número de paquetes exitosos y la cantidad de paquetes perdidos, en valor numérico y porcentual.

Se puede apreciar que el porcentaje de paquetes perdidos va incrementando, esto se debe a que a medida que la ejecución de la aplicación SIPP avanza el servidor se congestiona. En esta condición el servidor reduce su capacidad para atender las llamadas y procesar los paquetes, por lo cual es inevitable que estos se pierdan. El valor máximo de pérdida de paquetes se presenta en un *stream* con un total de 490 paquetes enviados, en dicho *stream* se presenta el 49.59 % de paquetes perdidos. Al analizar cada uno de los *stream*s observados en la herramienta RTP *Stream*s es posible además obtener el número total de paquetes enviados en comparación al total de paquetes perdidos.

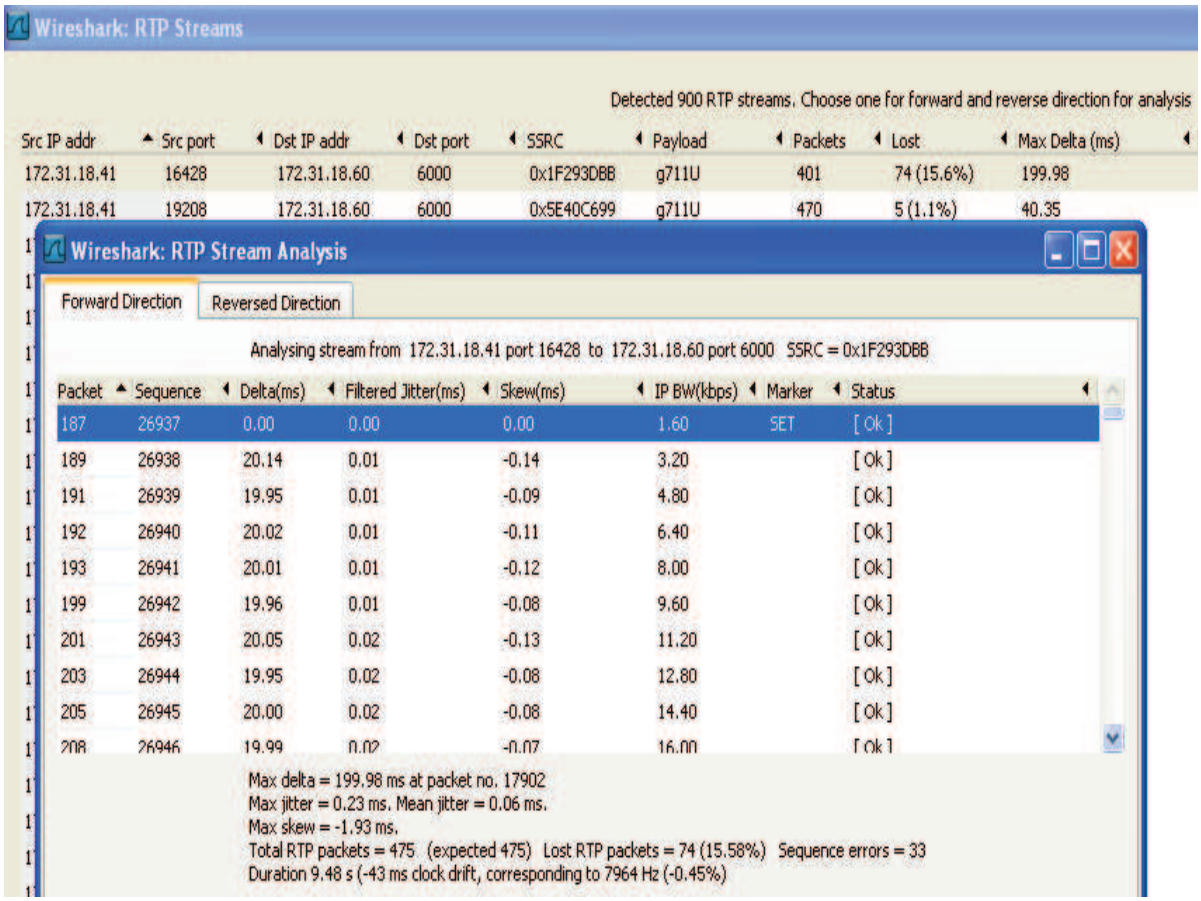


Figura 4.44 Análisis de *stream* RTP

Adicionalmente se puede visualizar los resultados obtenidos en la consola de Asterisk. En esta pantalla se puede apreciar los mensajes cuando las llamadas son cursadas exitosamente y el estado de la consola el momento en que se deniega el servicio. En esta condición que las llamadas no pueden ser cursadas una vez que se excede la capacidad del servidor.

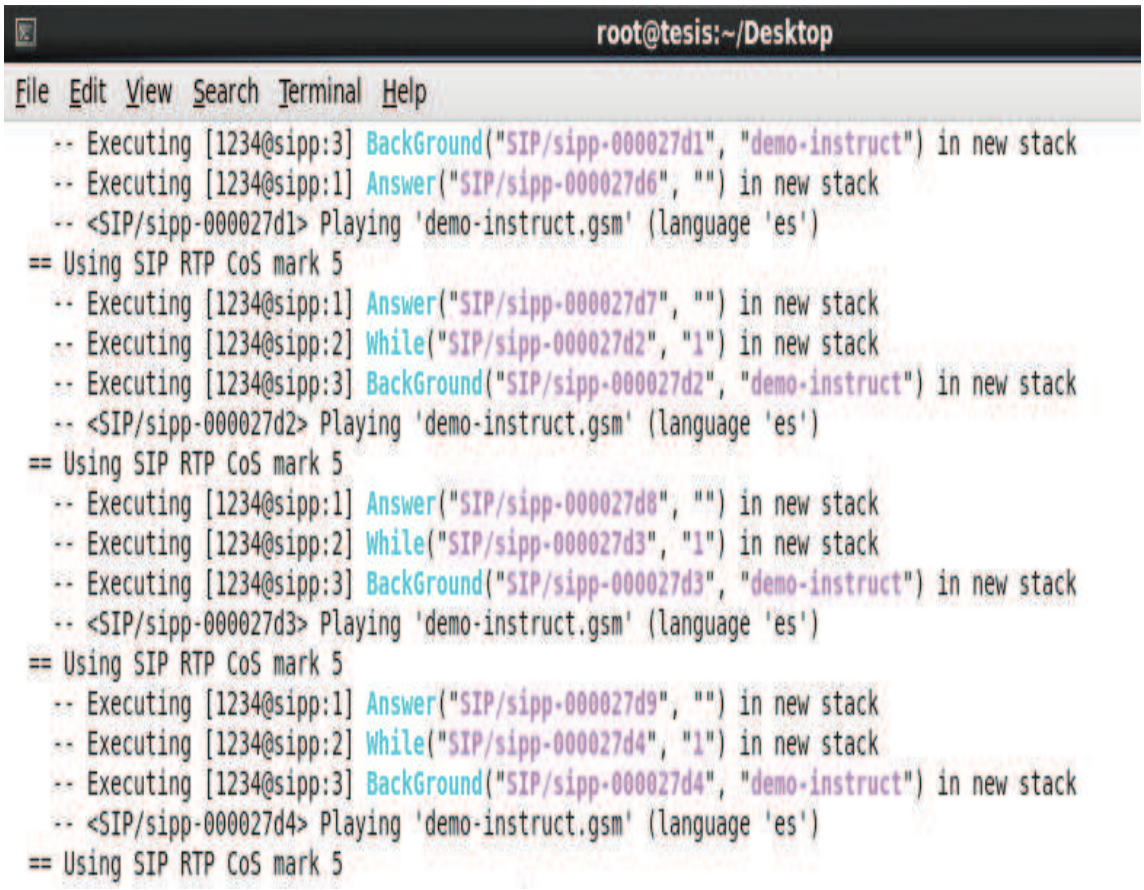


Figura 4.45 Mensajes de llamadas exitosas en la consola de Asterisk

En la imagen anterior se puede apreciar la ejecución de cada una de las líneas del dial plan para atender las llamadas en condiciones normales. La ejecución se realiza de acuerdo a la extensión, nombre aplicación y prioridad.

Durante la denegación de servicio el resultado al visualizar los mensajes en la consola revela errores y advertencias. Esta información indica de que las llamadas no están siendo completadas, además no se presenta ninguna evidencia de ejecución del dial plan por lo cual el servidor está congestionado.

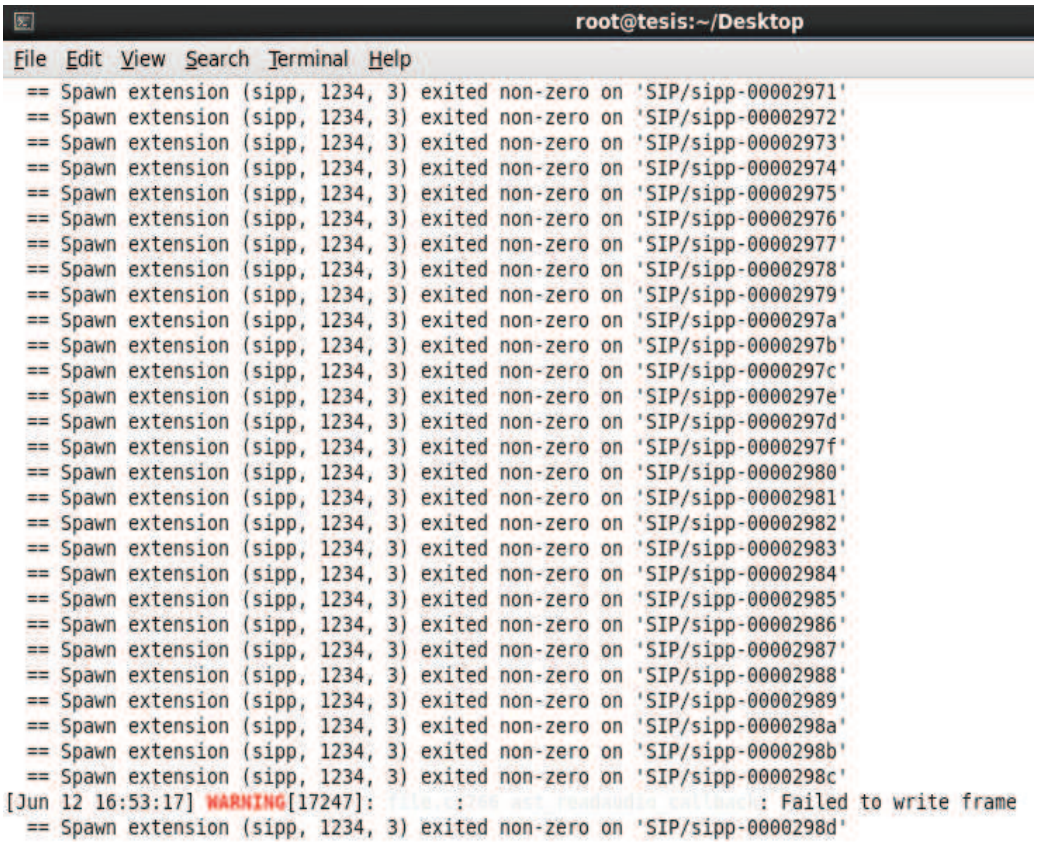


Figura 4.46 Mensajes de llamadas no completadas en la consola de Asterisk

4.9 SOLUCIÓN DE PROBLEMAS

4.9.1 PROBLEMAS DE NAT Y VOIP

El problema se presenta en las extensiones que están detrás de un dispositivo que realiza NAT y en clientes que no están en la misma red local del servidor.

El efecto de este inconveniente se presenta en la dificultad para el registro de los clientes y en una degeneración de la conversación telefónica. El problema de audio se produce al descolgar; es decir una vez que la llamada ha sido establecida entre los participantes de la conversación. Este problema tiene dos variantes: que el audio no fluye correctamente en los dos sentidos, por consiguiente solo uno de los participantes recibe el audio (*one way audio*) o en caso extremo no existe audio en absoluto.

El motivo de este problema es que en el protocolo SIP la señalización y los datos viajan de manera separada, un puerto UDP lleva la señalización de la llamada mientras que el tráfico de audio se transporta de extremo a extremo mediante 2 puertos RTP. El inconveniente surge debido a que el puerto al que se manda el audio es aleatorio, el *router* se encuentra en capacidad de dirigir correctamente la señalización pero es incapaz de determinar si el tráfico RTP corresponde a esta llamada y no sabe a dónde mandarlo.

Debido a esta razón se presentan problemas con NAT en el flujo de audio cuando este flujo debe superar los *router*s y *firewall*s. SIP utiliza el puerto 5060 para señalización y 2 puertos RTP por cada conexión de audio

Para solucionar este problema dentro del prototipo se implementan dos alternativas:

- La utilización de canales IAX puesto que la información de señalización y datos viajan por un mismo puerto.
- Configurar en el fichero **sip.conf** la opción NAT con un valor igual a yes para extensiones fuera de la red del servidor o inmersas en mecanismos de nat, y con un valor igual a not para extensiones dentro de la misma red del servidor.

4.9.2 RETARDOS EXCESIVOS

Uno de los principales inconvenientes se presenta en los valores de retardo que sobrepasan los límites permitidos. Los retardos dentro del prototipo provienen de distintas fuentes, pero en este caso particular como se menciona en la sección 4.3.2.1 la mayor parte se debe al procesamiento de la central Asterisk para la conversión de direcciones IPv4 e IPv6. Adicionalmente se presentan retardos debidos a la estructura de la red y retardos relacionados al tipo de códec escogido para la comunicación. Como posibles soluciones enfocadas a reducir los valores críticos de retardo se mencionan las siguientes alternativas:

- Implementar la central IP PBX utilizando la versión 11 de Asterisk en la cual se mejoran los problemas de procesamiento para la conversión de direcciones; sin embargo esta versión se encuentra aún en fase de

pruebas al momento de la elaboración del prototipo por lo cual se incluye únicamente como una recomendación a futuras implementaciones.

- Los retardos de red son de tipo variable dependiendo de la topología y mecanismos inmersos en la red. Dentro de este contexto para reducir el valor de retardo se presenta como opción, el establecer prioridad para los paquetes de VoIP señalizando los paquetes con valores de tipo de servicio (ToS) en el caso de IPv4 y valores del campo Clase de servicio en IPv6.
- Seleccionar los códecs con menor valor de retardos por paquetización y algoritmo; dentro del prototipo se habilitó los códecs GSM, G711 ley A y ley μ, por sus reducidos valores en los retardos mencionados.

4.9.3 REDUCCIÓN DEL JITTER

El problema se presenta en valores de *jitter* que degeneran la calidad de la comunicación, como se mencionó en las pruebas de medición de este parámetro es un efecto propio de las redes de datos.

Una de las posibles soluciones al efecto del *jitter* es la aplicación del *jitter buffer*; el cual consiste en almacenar los paquetes en una cola y reenviarlos después de un pequeño retardo que regula el tiempo de llegada, evitando así que el jitter incremente. Esta herramienta se puede modificar en los equipos como teléfonos IP, ya sea por *hardware* o *software*.

Asterisk posee soporte para esta herramienta en canales IAX y SIP para lo cual, en el caso del prototipo es necesario configurar el archivo sip.conf para habilitar la opción del jitter buffer en el lado receptor de un canal sip. De esta forma el servidor donde se aloja la central Asterisk, será en donde se almacene el contenido del buffer. Las opciones adicionales en el fichero sip.conf, para configurar el jitter buffer se muestran en la figura 4.47.

```
jbenable=yes
jbmaxsize=200 ; Opcional (define el tamanio del buffer)
```

Figura 4.47 Configuración del jitter buffer en el fichero sip.conf

El *software* Wireshark proporciona adicionalmente la opción de implementar un mecanismo de *jitter buffer*; dicha opción permite evaluar el efecto de la manipulación del buffer sobre una captura de tráfico que haya realizado el sniffer.

Esta opción se encuentra dentro del menú estadísticas para llamadas de VoIP de Wireshark, posibilita incluso desplegar una ventana que permite reproducir la captura de tráfico VoIP conforme al tamaño de *jitter buffer* que se proporcione (por defecto 50 ms). Se admite un valor de *jitter buffer* configurable por el usuario tal como se muestra en la figura 4.48.

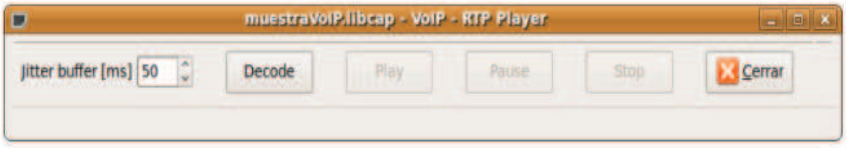


Figura 4.48 Mensajes de llamadas no completadas en la consola de Asterisk

4.9.4 PORCENTAJE DE PÉRDIDA DE PAQUETES

Una vez analizado el escenario de congestión del servidor Asterisk descrito en la sección 4.6; se propone como posibles soluciones a la pérdida de paquetes las siguientes alternativas:

- Incrementar los recursos de procesamiento y memoria del servidor en donde se instaló el servidor Asterisk.
- Seleccionar de manera adecuada los códecs a utilizarse en el prototipo de acuerdo a su tolerancia a la pérdida de paquetes y su mecanismo de compresión. De este modo para códecs de alta compresión como el G.729 se permite hasta el 1% de pérdidas y para códecs de baja compresión como el G.711 se permite un 5% de pérdidas. Cuanto mayor sea la compresión del códec más nocivo es el efecto de la pérdida de paquetes. Una pérdida del 1% degrada más la comunicación si se usa el códec G.729, en lugar del G.711; por esta razón en el prototipo, se configura la central IP PBX para seleccionar los códecs G.711 y GSM por su mejor tolerancia a pérdida de paquetes.

4.9.5 ALTERNATIVAS DE SEGURIDAD

Las consideraciones de seguridad por *software* no son suficientes para proteger al sistema de accesos no autorizados. Es recomendable incluir un dispositivo dentro de la frontera que separa la red del servidor de otras redes externas.

Entre las alternativas más utilizadas se encuentran *firewall*s que actúan como un límite entre la red local y el Internet, estos dispositivos se encargan de abrir y cerrar puertos, administrar listas de accesos de los usuarios en base a direcciones IP e incluso direcciones MAC.

Una alternativa más específica para la telefonía IP la constituye el *Session Border Controler* (SBC) que desempeña funciones similares a un *firewall* pero enfocadas al tráfico de datos multimedia. Entre sus principales características de acuerdo a las necesidades del presente prototipo se listan las siguientes:

- Protección contra ataques de denegación de servicio (DoS).
- Cifrado de tráfico de señalización y de multimedia.
- Incorpora listas de acceso y NAT.
- Provee un control de admisión de llamadas.
- Seguimiento a llamadas para facturación y generación de informes detallados.
- En general incorpora seguridad y monitoreo constante de un sistema de telefonía basado en el protocolo IP.

En el mercado existen soluciones SBC de *hardware* y *software*, sin embargo al momento de la implementación del proyecto no se dispone aún de una solución de *software* de licencia abierta. Los altos costos de estos dispositivos en *hardware* y su poca disponibilidad como *software* libre constituyen un limitante para la utilización de un SBC en el prototipo. Sin embargo es de interés para trabajos futuros mencionar que existen soluciones de *hardware* incorporadas dentro de *router*s tal como la serie 7600, además existen soluciones corporativas en las que coexisten alternativas de *hardware* y *software* como el caso de los módulos HP que brindan soporte para SBC por *software*.

CAPÍTULO 5

CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

- La integración de llamadas IPv6 en el entorno de la VoIP, se consigue mediante canales SIP; que a pesar de sus conflictos relativos a NAT, debido a la separación de señalización y datos por distintos puertos, se presenta actualmente como el único protocolo de telefonía IP que posee en su estructura las características y beneficios necesarios para soportar IPv6, tanto en el software de telefonía Asterisk como en los diferentes Softphones y teléfonos IP.

- El ancho de banda de una conversación telefónica, en una red de datos depende del códec utilizado, el protocolo de comunicación y el protocolo de transporte de datos.

- La calidad de la comunicación en una conversación telefónica utilizando telefonía IP, depende de factores como el retardo, el eco, el traslape del habla, el jitter; los cuales deben cumplir valores mínimos para que no sean un problema y tener una calidad aceptable; es así que el valor mínimo del retardo permisible en una comunicación extremo a extremo es de 150 ms mientras que el traslape del habla se presenta cuando hay retardos cercanos a 250 ms y se tendrá una calidad de voz aceptable si el jitter es menor a 50 ms.

- El protocolo IPv6 no solo permite incrementar el número de direcciones IP, sino que facilita su asignación e identificación mediante un prefijo con el que se distinguen diferentes tipos de dirección; permitiendo de esta manera asignar a una misma interfaz múltiples direcciones únicas según su tipo.

- Las dos versiones del protocolo IP mantienen una jerarquía y distribución de direcciones mediante prefijos e identificadores; enfocándose dicha jerarquía en el caso de IPv4 hacia la división en subredes; mientras que en el caso de IPv6, además de las subredes, la jerarquización se da en función del ámbito de la dirección.

- En los paquetes IPv6 analizados con el sniffer se presentan diferentes tamaños de encabezados según las funcionalidades adicionales del IPv6, que se presenten inmersas en los encabezados de extensión, anidados a la cabecera principal y que están relacionadas con una opción en particular, para funcionalidades tales como seguridad, enrutamiento, autenticación, entre otros.

- En la configuración de un túnel IPv6 sobre IPv4 el encapsulamiento requiere implementar una interfaz virtual en ambos sentidos del enlace inter routers; a la que se asigna una dirección IPv6, definiendo para ello como fuente del túnel la interfaz de salida y como destino la dirección del siguiente salto IPv4.

- En base al análisis de las dos versiones del protocolo IP se comprueba que los mecanismos para optimizar el manejo de direcciones IP en la versión 4 tales como el direccionamiento con clase, NAT y CIDR, no son una solución definitiva al agotamiento inminente de direcciones; la única solución absoluta es la migración a la nueva versión IPv6; que cuenta con

el espacio de direccionamiento suficiente para satisfacer la conectividad que demanda el mercado, cada vez creciente, de las tecnologías de la información.

- Es evidente la necesidad de implementar sistemas convergentes que puedan interactuar con tecnologías tradicionales y se adapten de manera eficaz a las necesidades futuras; en un ambiente corporativo actual, la telefonía tradicional no cumple con todas las expectativas debido a los costos elevados y limitadas prestaciones; por lo tanto se hace necesaria la implementación de un sistema de telefonía capaz de reducir costos, así como adaptarse a la red de datos existente. Un sistema de telefonía IP implementado sobre Asterisk es la solución a este problema; ya que no solo se adapta a la infraestructura, protocolos y tecnologías actuales; sino que también provee funcionalidad de acuerdo a las necesidades requeridas; puede ser integrado con la red de telefonía tradicional y además incluye la incorporación del nuevo protocolo de Internet IPv6, que poco a poco está sustituyendo al actual.

- Asterisk como PBX presenta una gran cantidad de beneficios en comparación con una central física; su característica de software libre reduce significativamente los costos de implementación; provee todas las funcionalidades de una PBX física y añade otras tan importantes como el correo de voz, la marcación automática, entre otras; convirtiéndose así en una solución de telefonía IP con las características necesarias que requieren las empresas actuales; presenta incluso la funcionalidad para prestaciones futuras, con la ventaja de requerir solamente la actualización del software para dicho efecto.

- Existen grandes similitudes entre los mecanismos de enrutamiento para IPv4 e IPv6; sin embargo la configuración de estos mecanismos en los

enrutadores son diferentes para cada versión del protocolo IP, por lo que las versiones de un mismo protocolo de enrutamiento dinámico no son compatibles entre sí y su configuración debe realizarse de manera independiente.

- El principal inconveniente que afecta a la calidad de voz en las llamadas que incluyen direcciones IPv6 es el retardo; el cual es causado por problemas que no se presentan únicamente en la capa red sino en la capa aplicación, debido al procesamiento para la conversión de tipos de direcciones y códecs que realiza el servidor Asterisk, ya que la versión 1.8 aún se encuentra en el estado de pruebas para el soporte de IPv6.

- Idealmente, una llamada, independientemente del tipo de dirección de los llamantes, no presenta paquetes perdidos; a menos que se presente una condición de congestión al sobrepasar la capacidad de los recursos de la PBX; se crea un cuello de botella para el flujo de llamadas, con lo cual se provoca que los paquetes no puedan ser atendidos y se pierdan.

- Las pruebas ejecutadas en el prototipo constituyen una etapa de planificación para la implementación de un entorno de red empresarial; y se establecen como una herramienta bien sustentada para el dimensionamiento y desarrollo de una central de comunicaciones, compatible con ambientes actuales y futuros.

5.2 RECOMENDACIONES

- En la implementación de un Sistema de Telefonía IP en base a Asterisk es recomendable escoger de manera adecuada el códec a utilizar; para así optimizar el ancho de banda y minimizar los retardos.

- Es recomendable escoger un direccionamiento de tipo global unicast a nivel IPv6 al implementar un entorno de pruebas, debido a la facilidad en el manejo, asignación e identificación que dichas direcciones presentan, además de su característica de ser globalmente ruteables.

- Para la implementación en un ambiente real se debe realizar un análisis de tráfico, dirigido a determinar el tipo y número de dispositivos de hardware y software a utilizar para tener un grado de servicio aceptable en la comunicación con la red telefónica tradicional (PSTN) y así evitar que se degeneran las características de comunicación.

- Es de gran importancia configurar adecuadamente los parámetros relativos a NAT en los archivos de configuración de Asterisk; esto permite evitar problemas de conectividad a nivel remoto, en el cual se debe habilitar dicho parámetro; mientras que en un entorno local, la configuración de dicho parámetro es transparente.

- La mejor manera de administrar el servidor de Telefonía Asterisk es mediante consola; puesto que permite mayores funcionalidades, en cuanto a autenticación, seguridad e implementación de servicios; así como posibilita evidenciar los errores que se presentan al ejecutarse alguna de las aplicaciones.

- Es fundamental que al dimensionar el servidor de Telefonía IP se considere características óptimas, en cuanto a procesamiento y capacidad de memoria; para así gestionar de manera adecuada las aplicaciones requeridas por los clientes; así como evitar el congestionamiento del mismo, brindando de esta manera una gran disponibilidad y evitando la pérdida de paquetes; es además indispensable una gran capacidad de procesamiento, para mejorar el tiempo de respuesta de la central en ambientes mixtos, en donde se realiza conversión de direcciones y protocolos.

- Para la implementación de un servidor de Telefonía, mediante el software Asterisk con todas las funcionalidades y servicios que el mismo presta, es recomendable editar cada uno de los archivos de configuración, de una manera organizada y estructurada; para así evitar líneas de programación o sentencias de ejecución redundantes e innecesarias.

BIBLIOGRAFÍA

[1] Introducción a la Telefonía.

http://www.naser.cl/sitio/Down_Papers/Introduccion%20a%20la%20telefonia.pdf. (Fecha de Revisión Octubre 2012)

[2] ANDOCILLA, Wilson. *Integración de los servicios de VoIP*.

[3] ROMERO, Adriana; MUÑOZ Cristian. "*Diseño de una Red de Telefonía IP para la ciudad comercila el Recreo"; Escuela Politécnica Nacional. Quito 2012.*

[4] CADENA, Luis; AVEIGA, Diana. *"Diseño de la Red de Telefonía IP y su Integración con la Red de Datos"*. Escuela Politécnica Nacional; Quito 2010.

[5] GUTIÉRREZ, Roberto. *Seguridad en VoIP: Ataques, Amenazas y Riesgos*.

[6] RÍOS, André; ALCOBER, Jesús; OLLER, Antoni. "*Desarrollo de una plataforma de VoIP basada en Software Libre"*; Universidad Politécnica de Cataluña. Barcelona, España.

[7] ABAD, Blacio; JIMENEZ, Edgar; LÓPEZ, Pedro. "*Diseño de una red wan para transmitir voz sobre ip y su utilización futura como red alternativa para la telefonía fija en el ecuador"*; Escuela Superior Politécnica del Litoral. Guayaquil 2004.

[8] VoIP Foro: Ejemplo de Comunicación SIP.

http://www.voipforo.com/SIP/SIPejemplo.php. (Fecha de Revisión Octubre 2012)

[9] Comunicaciones Unificadas con Elastix.

http://es.scribd.com/doc/49810258/186/Protocolo-IAX. (Fecha de Revisión Noviembre 2012)

[10] PÉREZ, Carlos; TATÉS, Germán. "*Estudio e Implementación de una*

Central de Comunicaciones Unificadas"; Escuela Politécnica Nacional. Quito 2011.

[11] VoIP Foro. Mensajes IAX.

http://www.voipforo.com/IAX/IAX-ejemplo-mensajes.php. (Fecha de Revisión Octubre 2012)

[12] VACA, Cristina. "*Estudio de VoIP aplicado en Redes Privadas". Cap1;*

[13] LÓPEZ, David. *Implementación de Protocolos de señalización de VoIP*.

[14] VoIP Foro. Codecs

http://www.voipforo.com/codec/codecs.php#g729. (Fecha de Revisión Noviembre 2012)

[15] QUIÑONEZ, Luis Fernando. "La VoIP, Una Guía Práctica"; Universidad de San Carlos de Guatemala; Guatemala. 2005

[16] Características de VoIP

http://peru.itaki.net/articulo-voip.html. (Fecha de Revisión Noviembre 2012)

[17] TATÉS, Eliécer; FUENTES, Andrés. *Telefonía IP con Asterisk. ADMINISTRADOR AVANZADO.*

[18] GONCALVES, Flavio. *"Como construir y configurar un PBX con software libreAsterisk versión 1.4".* Tercera Edición. Brasil, Rio de Janeiro 2007.

[19] CHAFFIN, Larry; LONG, Johnny. *Asterisk Hacking.*

[20] SIERRA RORIGUEZ, Antonio . *"Instalación de un sistema VoIP corporativo basado en Asterisk".* Universidad Politécnica de Cartagena. Septiembre 2008.

[21] STALLINGS, William. *Comunicaciones y Redes de Computadoras, Sexta Edición, Prentice Hall, Madrid, 2003.*

[22] CISCO Networking Academy CCNA Exploration. “Aspectos básicos sobre Networking”.

[23] SILVA BRACERO, Leonardo. *Estudio y Análisis del estado actual de la Implantación de IPv6 en los Proveedores de Servicios de internet a Nivel Nacional*. Escuela Politécnica Nacional, Ecuador, 2012.

[24] Calidad de Servicio en IPv6.

http://long.ccaba.upc.es/long/050Dissemination_Activities/alberto_lopez_QoStutorial.pdf. (Fecha de Revisión Diciembre 2012)

[25] REGIS DOS SANTOS, Rodrigo; MOREIRAS, Antonio. *Curso IPv6 Básico, Sao Paolo, 2010.*

[26] IPv6 y Calidad de servicio en Redes Convergentes BGP+IPv6.

[27] VAZQUES CLAVIJO, Jenny. *Análisis de las funcionalidades de los protocolos de Seguridad IPSec, IKE, ISAKMP sobre IPv6 e Implementación en una red prototipo bajo infraestructura CISCO, Universidad Politécnica Salesiana 2011.*

[28] Características de IPv6.

http://technet.microsoft.com/es-es/library/cc780593(v=ws.10).aspx. (Fecha de Revisión Diciembre 2012)

[29] Fundamentos de IPv6.

http://www.ipv6.mx/index.php/informacion/fundamentos/ipv6. (Fecha de Revisión Diciembre 2012)

[30] Fundamentos de IPv6

http://www.ipv6.mx/index.php/informacion/fundamentos/ipv6. (Fecha de Revisión Diciembre 2012)

[31] TANENBAUM, Andrew. *Redes de computadoras, Cuarta Edición, Prentice Hall, México, 2003.*

[32] David NÚÑEZ LARA, *Estudio para la migración de IPv4 a IPv6 para la empresa proveedora de Internet MILLTEC S.A.*

[33] Direccionamiento IPv6

http://www.paratorpes.es/cisco/RFC%203513%20DIRECCIONAMIENTO%20IPV6.pdf (Fecha de Revisión Diciembre 2012)

[34] Introducción a IPv6

http://lacnic.net/documentos/lacnicxii/presentaciones/introduccion_ipv6_v12.pdf (Fecha de Revisión Diciembre 2012)

[35] Evolución de Redes Fijas del Protocolo IPv4 a IPv6 en Guatemala.

http://es.scribd.com/doc/41595709/23/Figura-4-Formato-de-una-direccion-IPv6-compatible-con-IPv4. (Fecha de Revisión Diciembre 2012)

[36] Tutorial de Ipv6: Introducción.

http://long.ccaba.upc.es/long/050Dissemination_Activities/jordi_palet_tutorial ipv6introduccion.pdf. (Fecha de Revisión Diciembre 2012)

[37] RFC 3315: Dynamic *Host* Configuration Protocol for IPv6 (DHCPv6).

http://www.ietf.org/rfc/rfc3315.txt. (Fecha de Revisión Diciembre 2012)

[38] CCNA Exploration CISCO Networking Academy. “Introducción al enrutamiento y reenvió de paquetes”.

[39] RFC 2080 RIPng for IPv6.

http://www.ietf.org/rfc/rfc2080.txt. (Fecha de Revisión Diciembre 2012)

[40] RFC 2740: OSPF for IPv6.

http://www.ietf.org/rfc/rfc2740.txt. (Fecha de Revisión Diciembre 2012)

[41] RFC 1933: IPv6 Transition Mechanisms.

http://www.ietf.org/rfc/rfc1933.txt. (Fecha de Revisión Diciembre 2012)

[42] Modelo de Seguridad en Asterisk.

http://asteriskmx.com/modelo-de-seguridad-en-asterisk-parte-1-de-3/. (Fecha de Revisión Septiembre 2013)

[43] Routers Cisco serie 2900..

http://cloudnetwork.mx/images/cisco%201941.pdf. (Fecha de Revisión Junio 2013)

[44] Equipos HP.

http://pro-networking-h17007.external.hp.com/us/en/products/routers/HP_6600_Router_Series/index.aspx. (Fecha de Revisión Septiembre 2013)

[45] Encaminadores H3C.

http://pro-networking h17007.external.hp.com/us/en/products/routers/HP_6600_Router_Series/index.aspx. (Fecha de Revisión Septiembre 2013)

[46] DES-1008D, 8 PORT 10/1000 Mbps, Dual Speed Ethernet *Switch*. User's Guide.

http://www.dlink.com/-/media/Business_Products/DES/DES%201008D/Manual/DES_1008D_Manual_EN_SE.pdf. (Fecha de Revisión Agosto 2013)

[47] MADSEN, Leif; MRGGELEN, Jim; BRYANT, Russell . *Asterisk The Definitive Guide*. Tercera Edición. U.S.A 2011

[48] Cómo configurar un servidor de nombres de dominio.

http://www.alcancelibre.org/staticpages/index.php/como-dns. (Fecha de Revisión Mayo 2013)

[49] Centos 6.0: Crear un Servidor WEB.

http://www.taringa.net/posts/linux/12974646/Centos-6_0_-Crear-un-servidor-web.html. (Fecha de Revisión Mayo 2013)

[50] Servidor de Correo.

http://servilinux.galeon.com. (Fecha de Revisión Mayo 2013)

[51] SquirrelMail en Centos.

http://www.linuxparatodos.net/portal/staticpages/index.php?page=como-squirrelmail. (Fecha de Revisión Mayo 2013)

[52] ¿Qué es el Squirrelmail?.

http://es.scribd.com/doc/56231327/Tutorial-Squirrel-Mail. (Fecha de Revisión Mayo 2013)

[53] Libro Asterisk 1.8.

www.voztovoice.org. (Fecha de Revisión Abril 2013)

ANEXO A

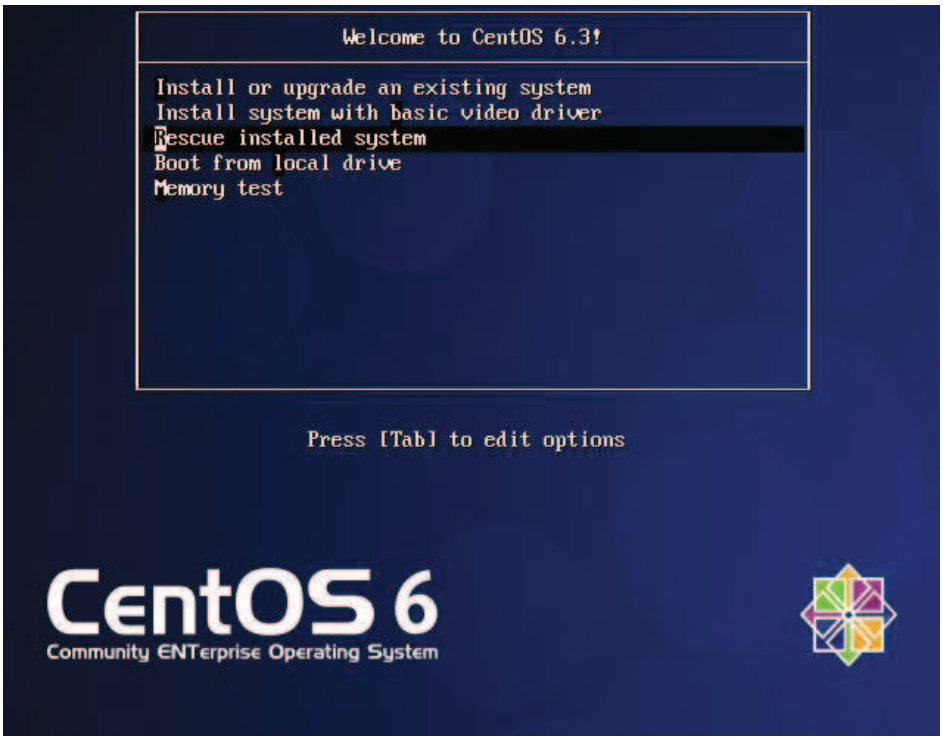
INSTALACIONES

A.1 INSTALACIÓN DE LINUX

Para la implementación de Asterisk 1.8 se utiliza como Sistema Operativo Centos 6.3 (última versión estable al momento de la realización del proyecto).

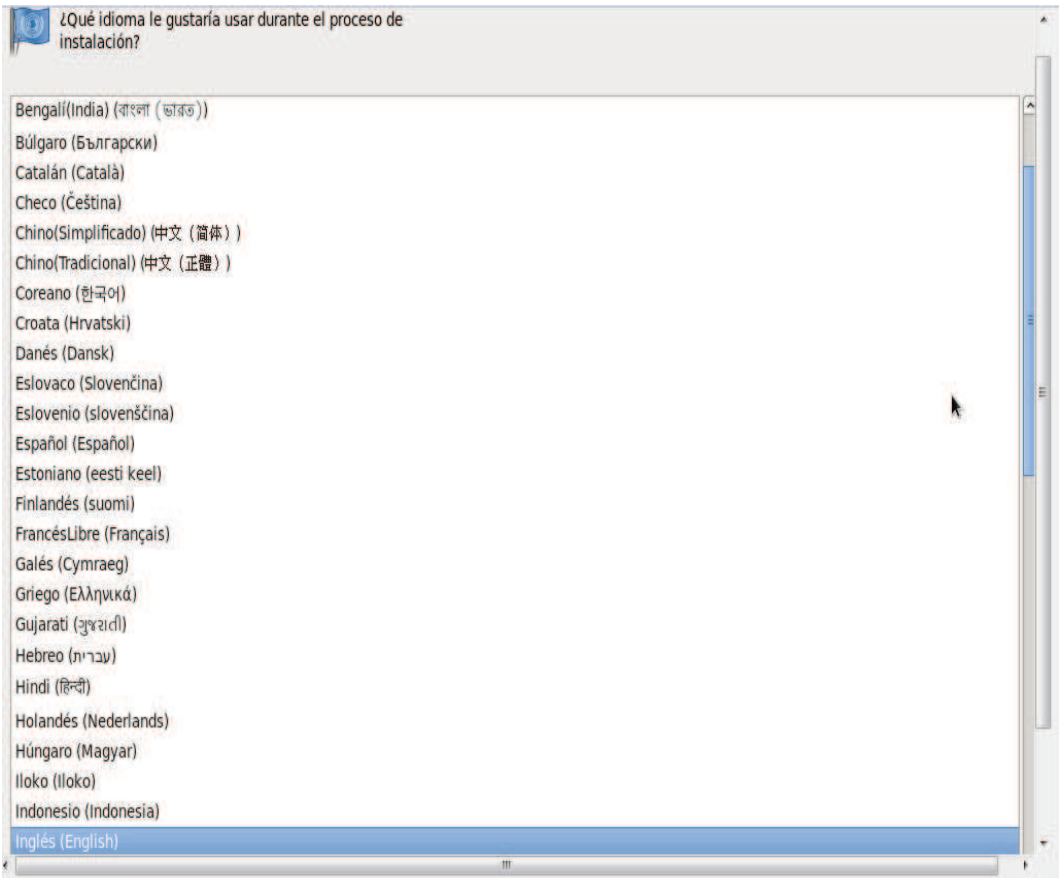
La instalación de este sistema operativo se la realiza de la siguiente manera:

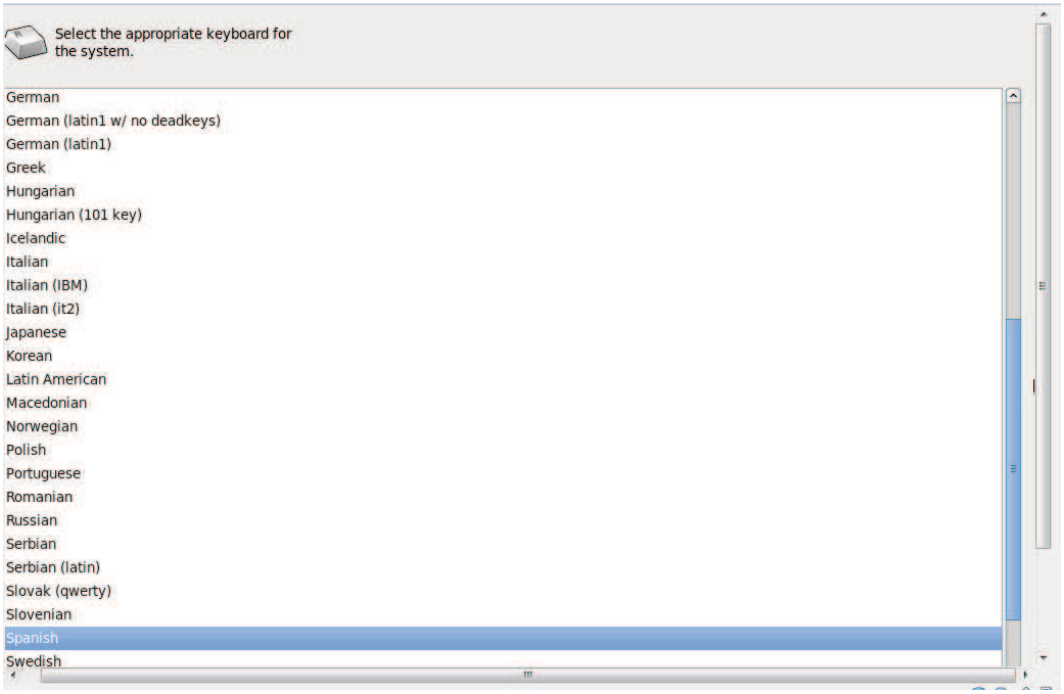
- Se configura la BIOS para gestionar el arranque desde el DVD en el cual estará la imagen del S.O (obtenida previamente de www.centos.org). Una vez que arranca el sistema desde el DVD, se presenta una imagen en la que se puede seleccionar el tipo de instalación; se selecciona entonces la primera opción (Install or upgrade an existing system).



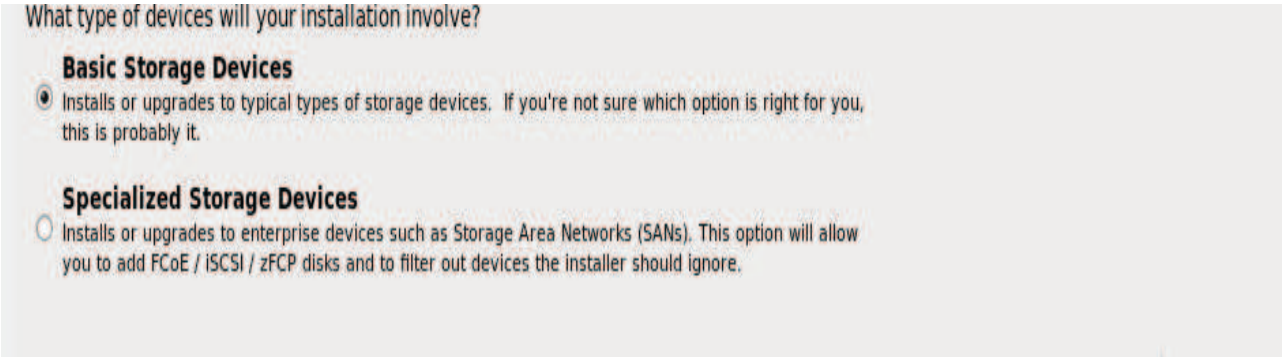
- La siguiente pantalla que aparece pregunta si se desea verificar la integridad del medio de instalación; aquí se selecciona *SKIP* para obviar este paso (si se está seguro de la integridad del DVD) y empezar con la instalación; aparecerá entonces una pantalla de bienvenida de Centos 6, posteriormente se presiona *Siguiente.*

- A continuación se configura el tipo de idioma con el cual se desea que se instale el S.O. y el idioma del Teclado. Es recomendable para el idioma de instalación escoger inglés ya que los archivos de ayuda y soporte de los programas son más fáciles de entender, para el idioma del teclado se selecciona español.

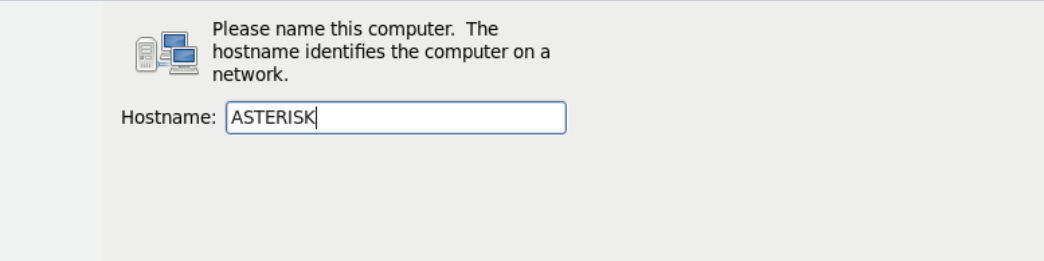


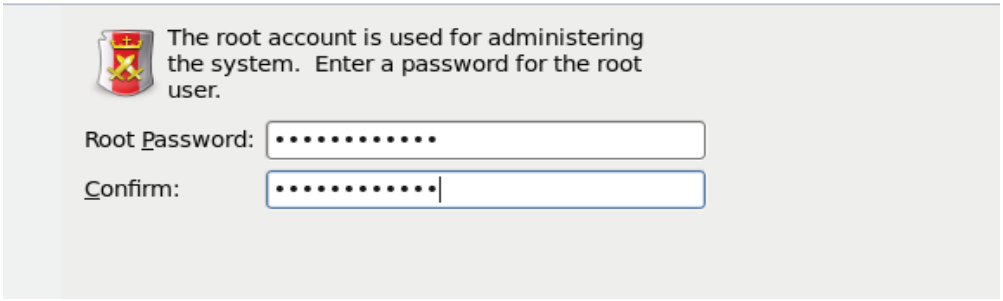


- En la siguiente pantalla, si sólo dispone de discos duros en el equipo donde se realizará la instalación, se elige la opción *Dispositivos de almacenamiento básicos* y se hace clic sobre el botón *Siguiente.*

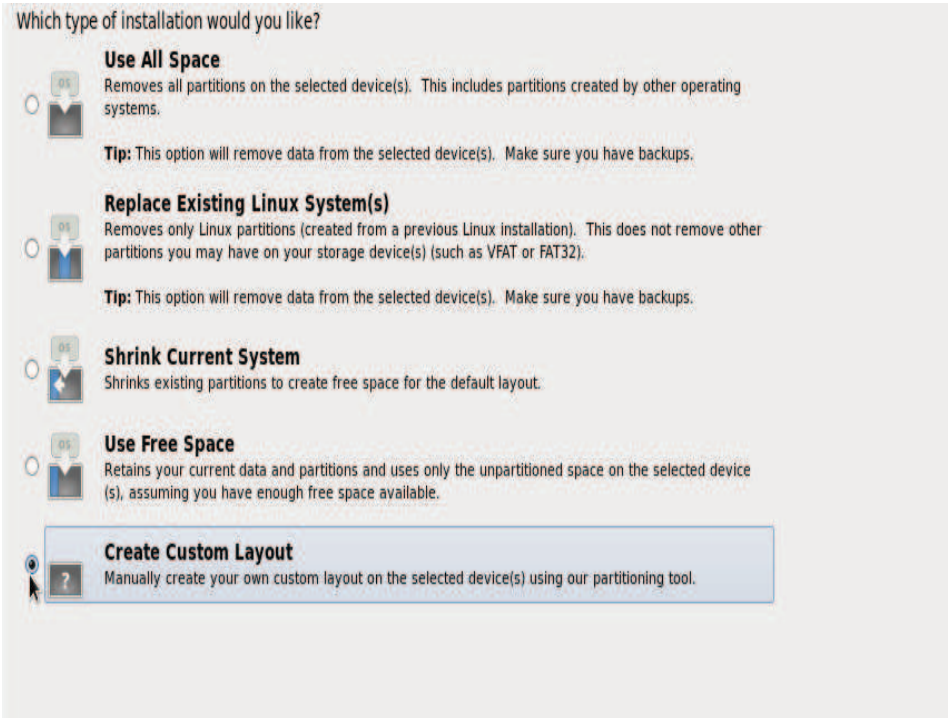


- Para continuar se procede a configurar el nombre del *host*, la ubicación geográfica (para el caso de Ecuador la zona aparece como Guayaquil-América) y la contraseña de *root* (super usuario, con todos los permisos).





- La configuración siguiente es el particionamiento del disco duro, para poder gestionar de manera adecuada este particionamiento, en la pantalla que aparece a continuación se selecciona *Create Custom Layout.*



La opción escogida crea una plantilla de particionado basada LVM, que tiene como ventaja el permitir unir varios discos o particiones (conocidos como Volúmenes físicos PV) formando un gran conjunto de almacenamiento de datos conocido como Volume Group (VG) que puede ser posteriormente "subdividido" en volúmenes lógicos en caso de que se haya terminado el espacio en estas particiones.

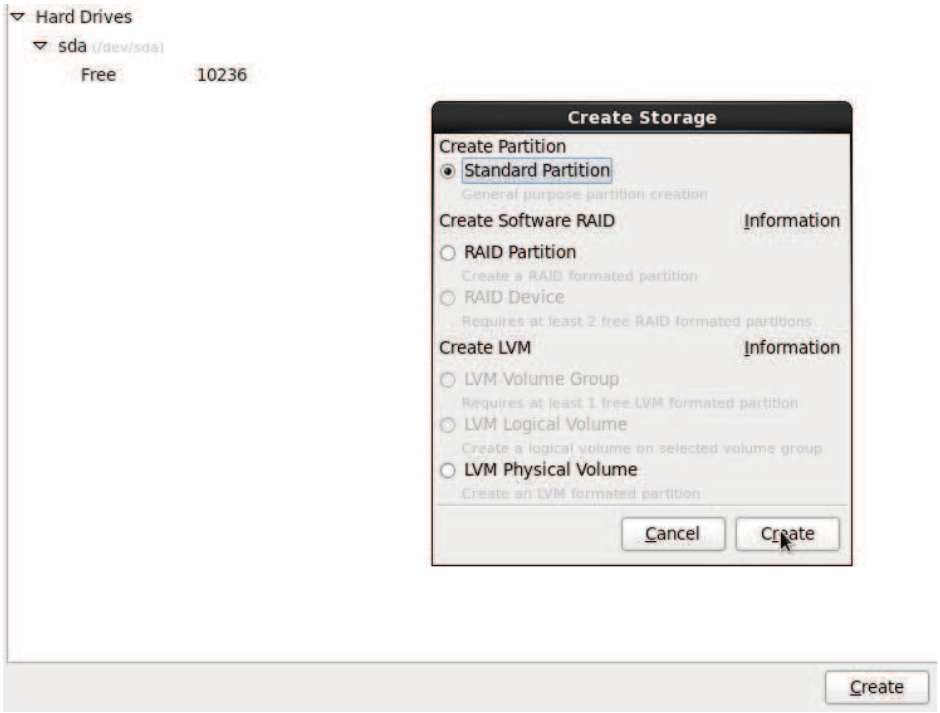
Para la instalación de Centos se requiere como mínimo tres particiones:

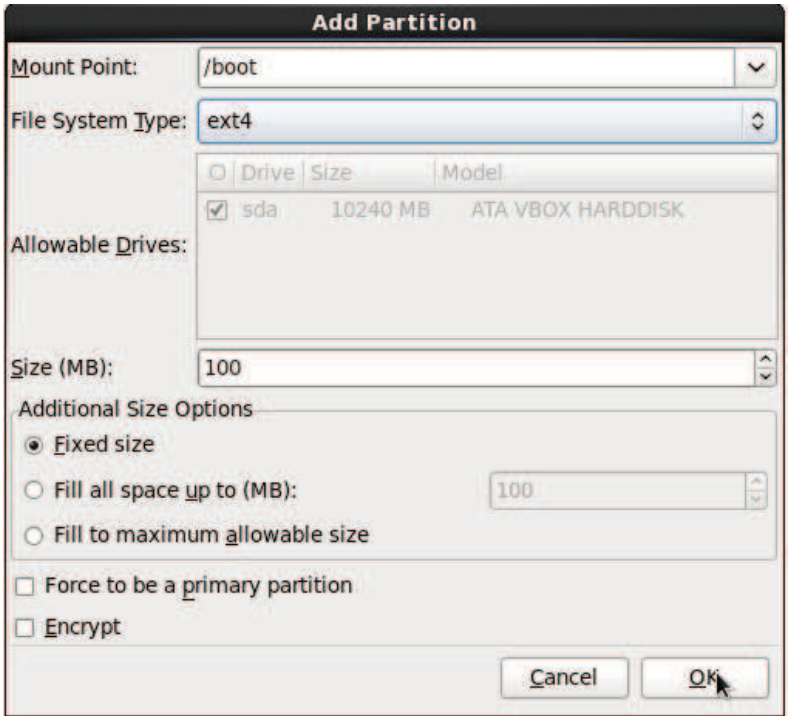
/boot Requiere al menos 100 MB. Tipo de archivo de Sistema ext4

Swap Es recomendable asignar el **doble del tamaño de la RAM (físico)** siempre y cuando la RAM sea menor o igual a 4GB caso contrario se asigna el mismo valor de la RAM. No se le asigna ningún punto de montaje

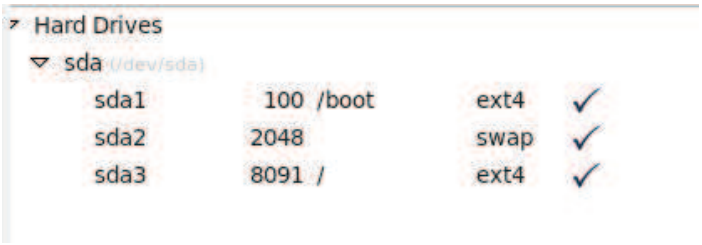
/ Completar hasta el tamaño máximo permitido.

- Para crear un partición simplemente se presiona clic en la pestaña *create,* luego en *estándar partition* y en create nuevamente, tal como muestra la siguiente figura:

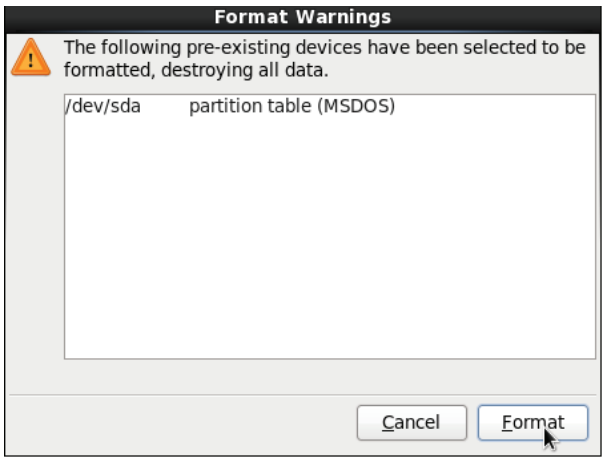




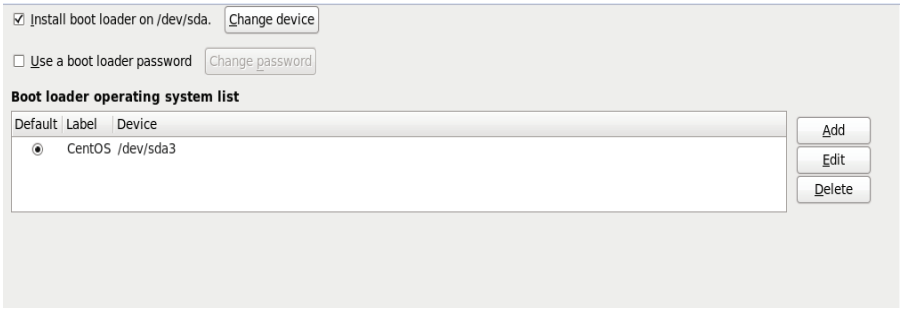
- Al terminar de crear las particiones se tiene una pantalla en la que aparece un resumen del particionamiento realizado.



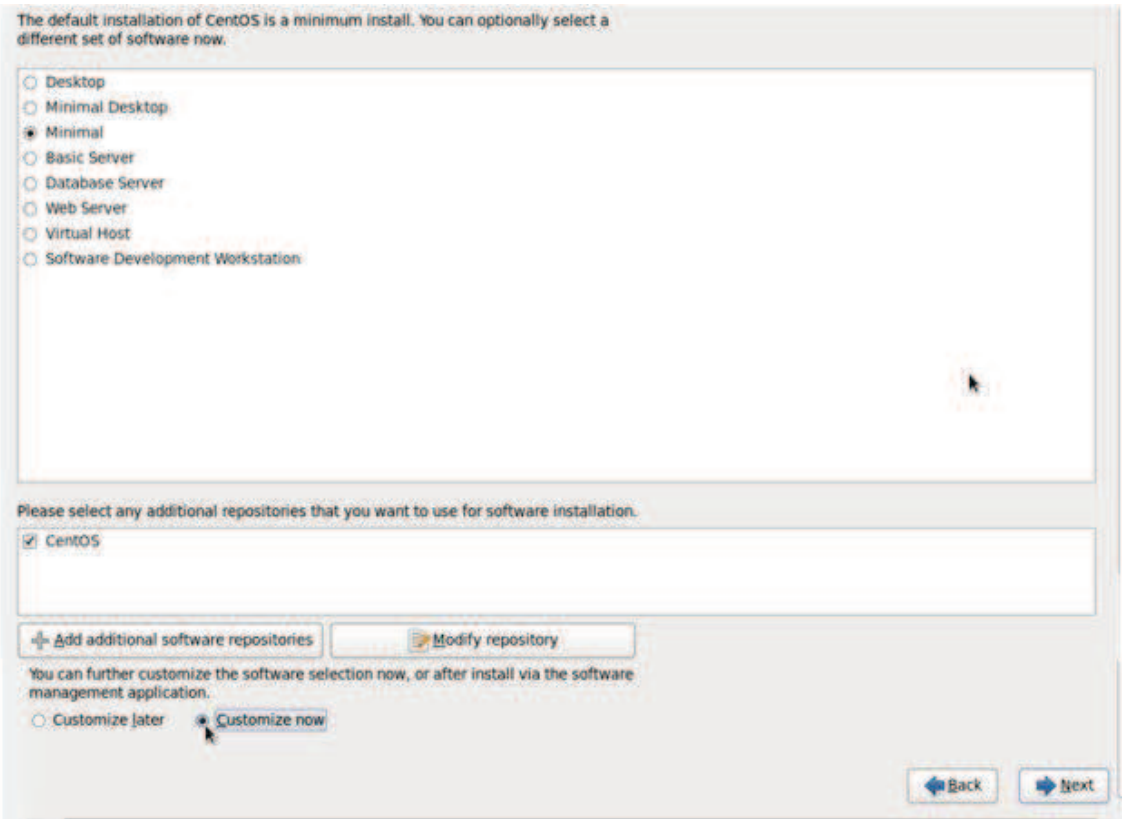
- Luego de crear todas las particiones y dar click en el botón *siguiente;* cuando se pregunte si se desea formatear cada una de las particiones se selecciona la opción *Format.*

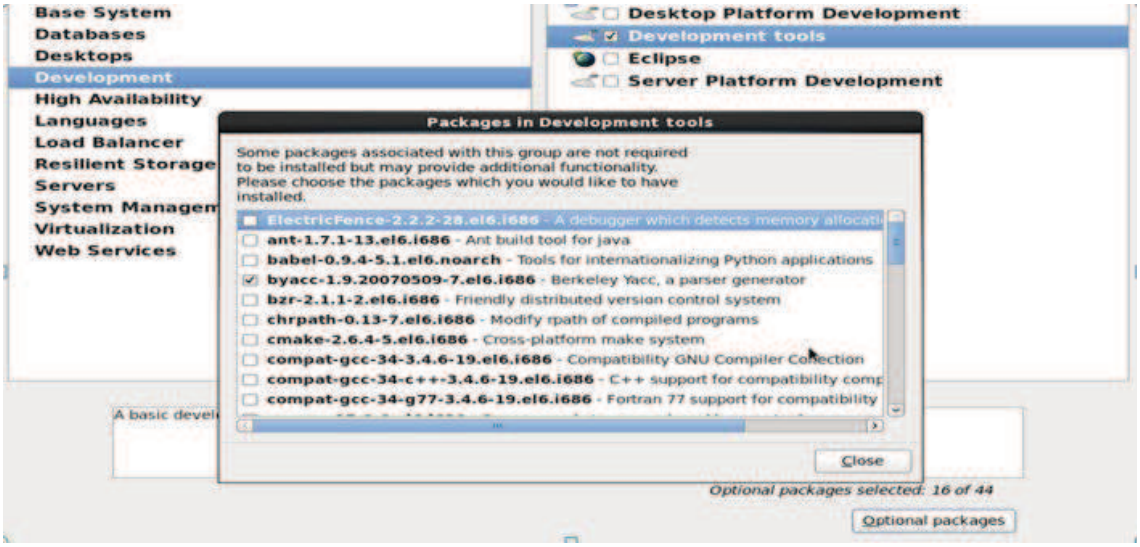


- La siguiente pantalla que aparece está relacionada a la ubicación en dónde se instalará el Gestor de arranque (programa que se instala en los primeros sectores del disco duro). GNU/Linux usa como gestor de arranque a GRUB (GRand Unifier Bootloader); su instalación se realiza juntamente con la instalación del SO, por defecto se instala de manera tal que toma el control del arranque del SO por lo cual es recomendable dejar la configuración predeterminada tal y como se muestra en la figura.



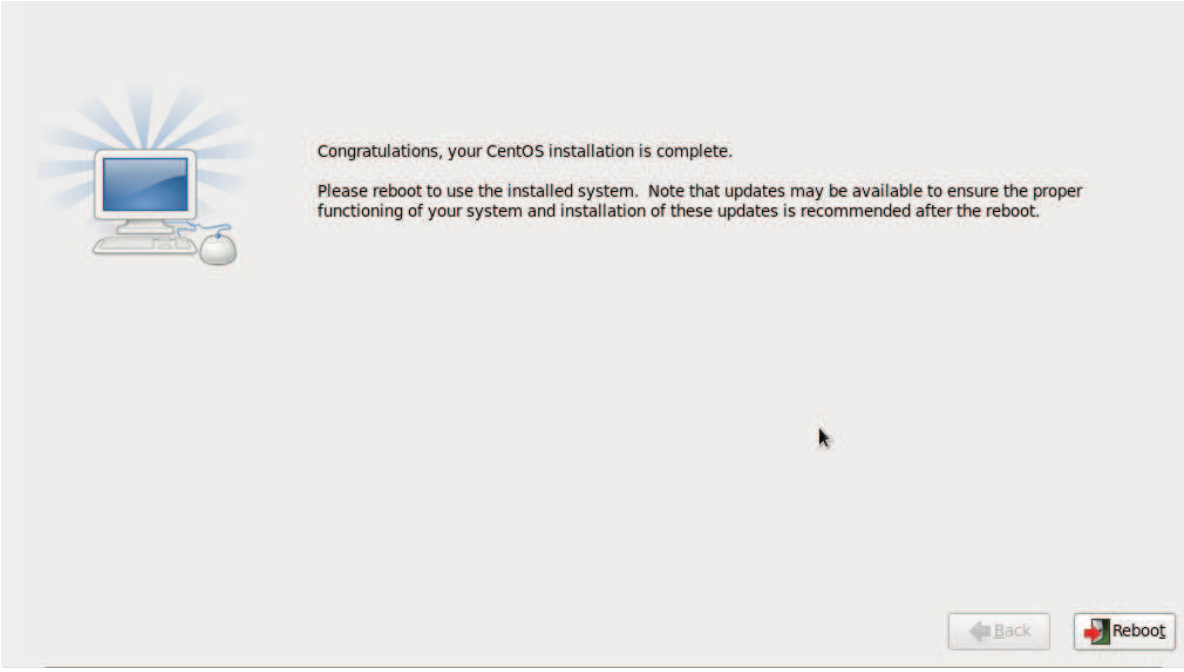
- El siguiente paso de la instalación es seleccionar los paquetes a instalar, las principales categorías a escoger son las siguientes:





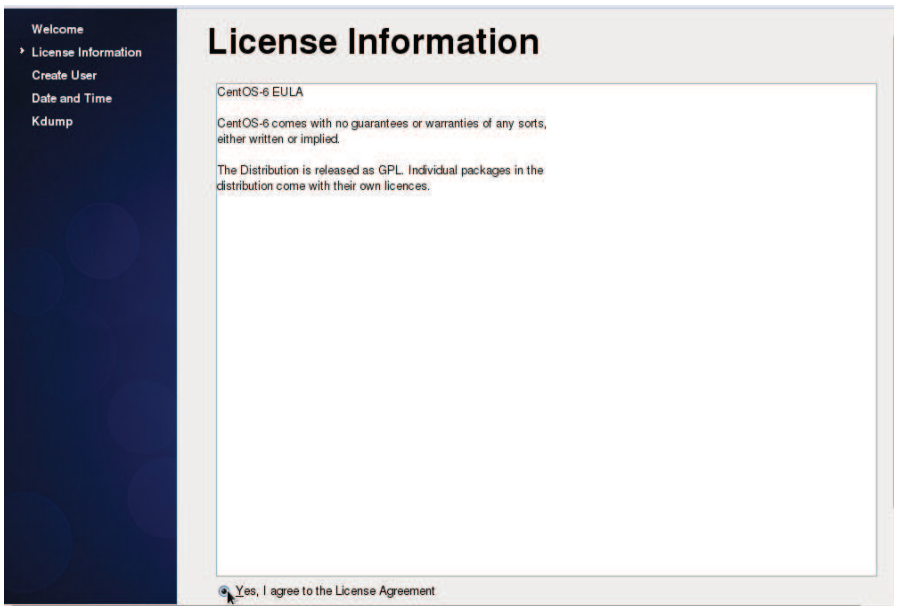
- Finalmente se realiza la instalación del SO la cual toma alrededor de 20 a 25 minutos dependiendo de los paquetes escogidos.

Una vez que se realizado la instalación del SO se obtiene una pantalla tal como se muestra en la figura siguiente, la cual solicita que se reinicie el sistema.

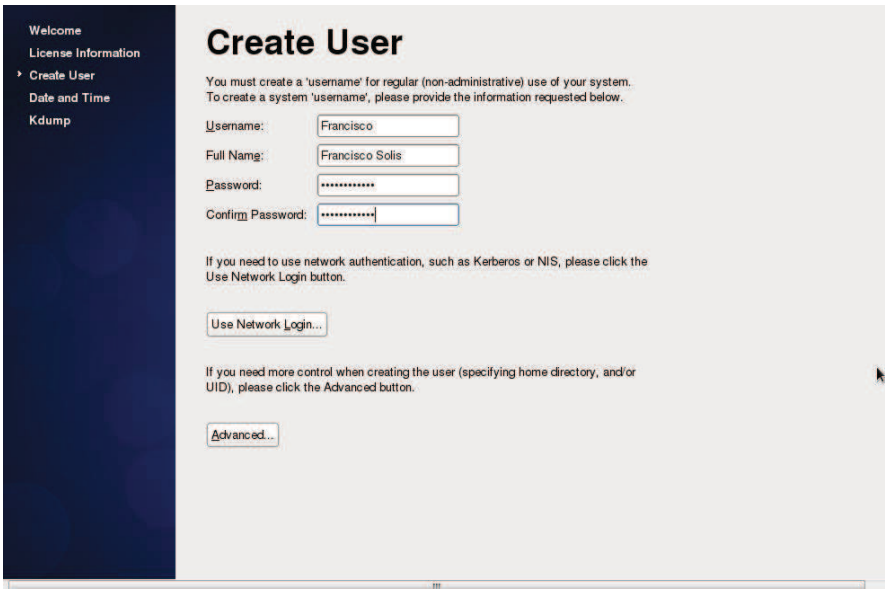


- Para terminar con el proceso de instalación luego de reiniciar el sistema, se solicita que se realice algunas configuraciones como: Aceptar el acuerdo de licencia, creación de un usuario, configuración de la fecha y hora y el Kdump el cual es un mecanismo de protección del Kernel y el cual reserva una memoria de 128 MB.

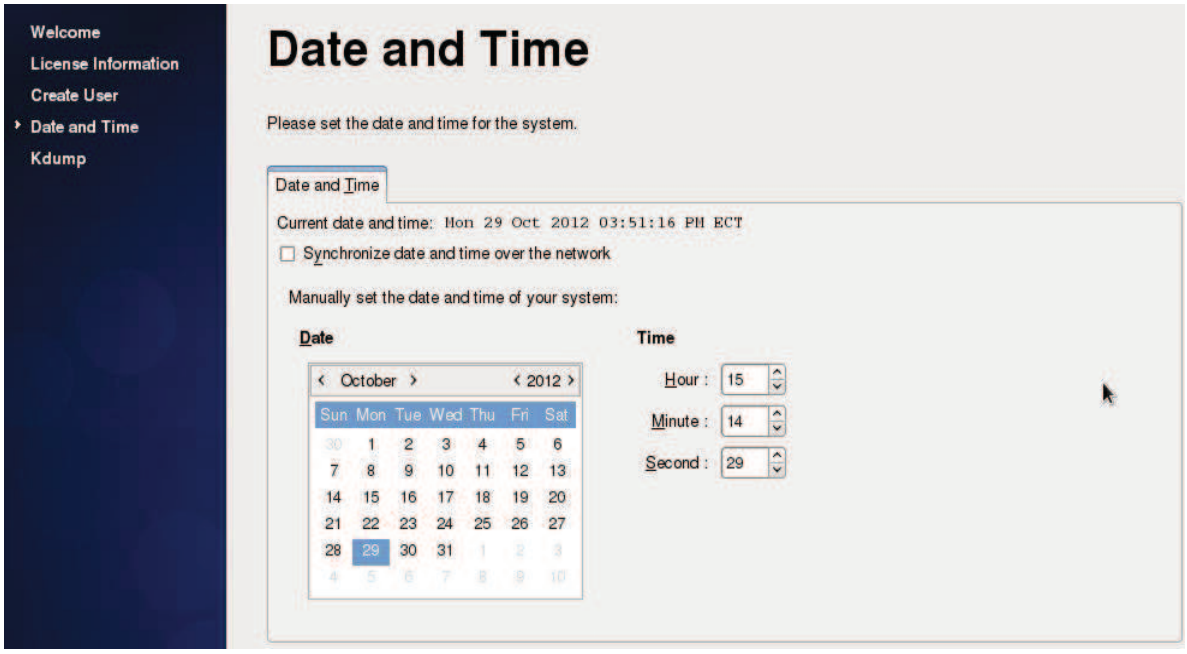
Se acepta el acuerdo de licencia y se da clic en *Next.*



Se realiza la configuración de una cuenta de usuario normal.



Se configura la fecha y hora actual:



Posteriormente se asigna el espacio predeterminado de memoria para el Kdump.

Al concluir todos estos pasos el sistema operativo se reiniciará y la aparecerá una pantalla como se muestra en la siguiente figura, en la cual se puede iniciar sesión con cualquiera de las tres cuentas (Invitado, usuario configurado o como superusuario root), de esta manera se ingresa por primera vez al SO ya instalado.

A.2 INSTALACIÓN DE ASTERISK

Existen dos maneras de instalar Asterisk mediante la compilación del código fuente y mediante repositorios, al instalar Asterisk mediante compilación se tiene un control más estricto sobre las herramientas que se instalan es decir se puede elegir las que se adapten a las necesidades requeridas; mientras que si se instala mediante repositorios es posible que al momento de configurar una aplicación específica se requiera un módulo adicional para que funcione. Por esta razón se opta por instalar Asterisk mediante la instalación del código fuente.

A continuación se explica los pasos a seguir para esta instalación:

- Conseguir el código fuente de una versión estable y en el caso del prototipo a instalar que tenga soporte sobre IPv6. Esta versión se puede descargar desde el sitio http://www.asterisk.org/downloads.

Para dichas necesidades se hace uso del paquete asterisk-1.8.12.2.tar.gz

Para la instalación de asterisk es recomendable copiar el archivo de código fuente en el directorio /*usr/src* del servidor., para tener una referencia de instalación de las carpetas de Asterisk.

- Una vez que el paquete con el código fuente se encuentra en el directorio especificado se procede a descomprimirlo mediante consola en el mismo directorio con permisos de superusuario editando:

 # tar xvfz asterisk-1.8.12.2.tar.gz

Con esto se crea el directorio asterisk-1.8.12.2.

- Se ingresa al directorio que se creó: *cd /usr/src/asterisk-1.8.12.2*
- A continuación se ejecuta los siguientes comandos:

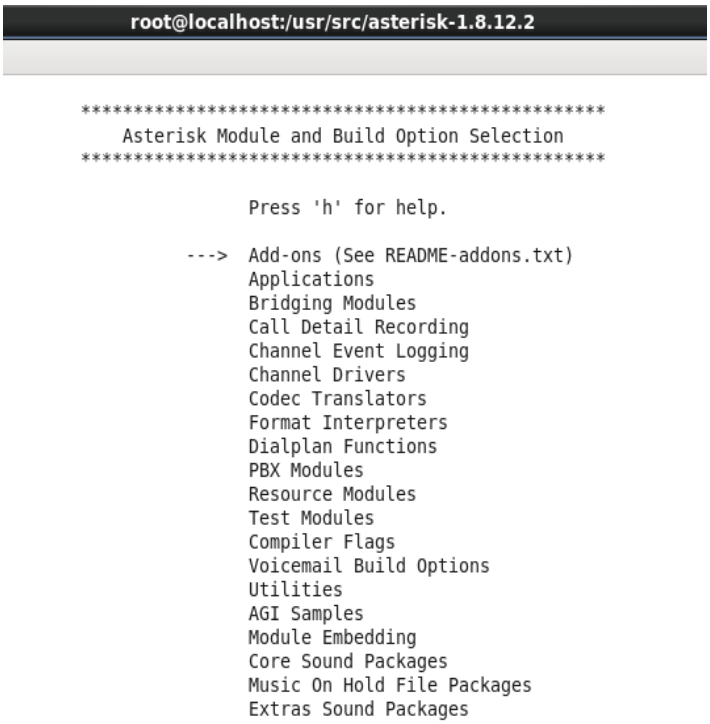
1. ***./configure***

   Se debe ejecutar este comando con permisos de superusuario; este comando realiza un chequeo de verificación de las dependencias de los paquetes necesarios para la instalación de Asterisk (paquetes de

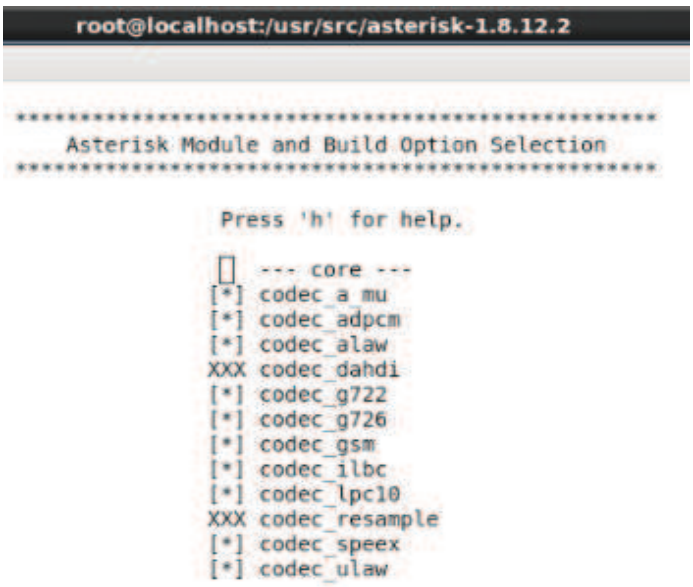
desarrollo necesarios: *ncurses, make, gcc, bison, libxml, curl*) además de configurar las fuentes de Asterisk para compilarse de acuerdo a la arquitectura del sistema. Si todas las dependencias requeridas se encuentran en el sistema y no ha existido ningún tipo de error en la configuración de los archivos se obtiene lo siguiente:

2. ***make menu select***

   Con este comando se obtiene una ventana de configuración en la cual se puede seleccionar características, módulos, funciones y aplicaciones que Asterisk tendrá al momento de compilarse e instalarse. La ventana que se obtiene se presenta en la siguiente figura:

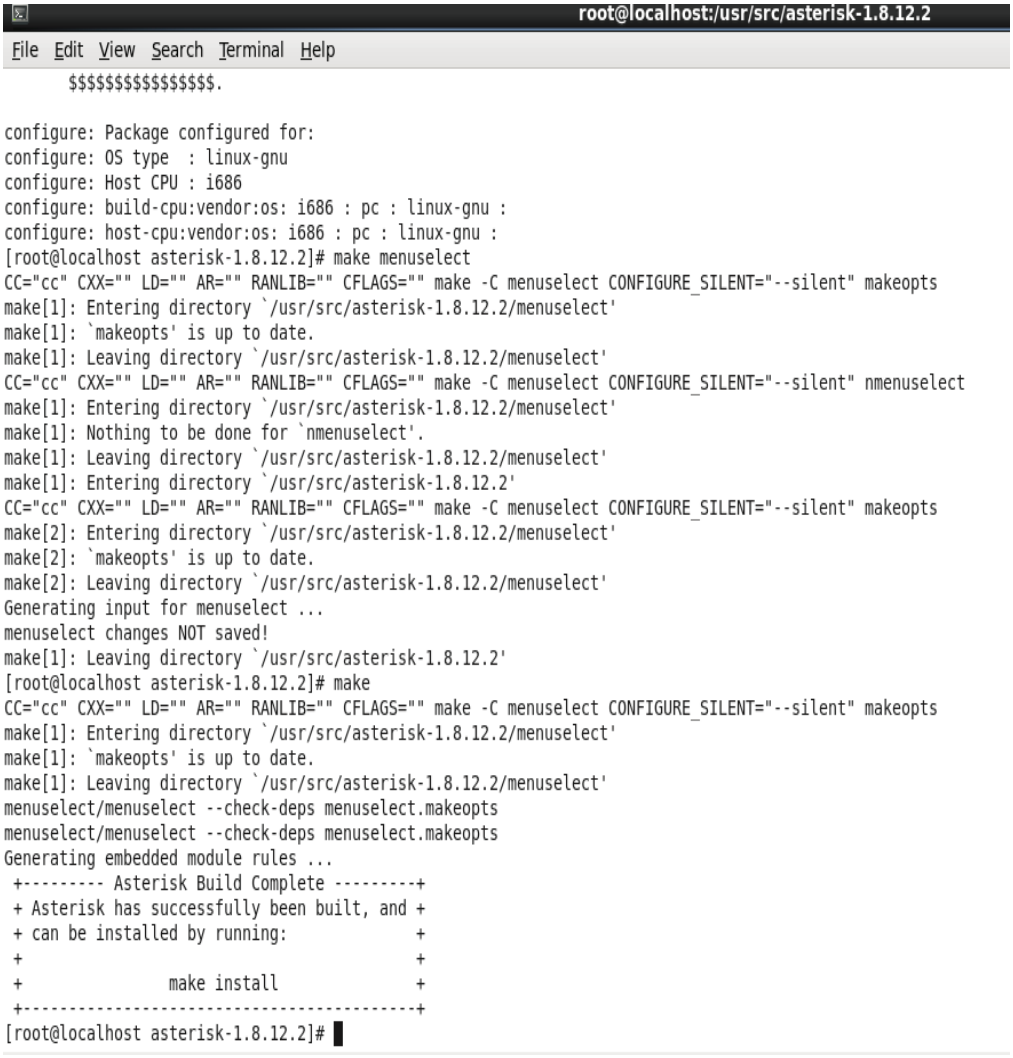


A continuación se despliega una ventana en la cual se puede observar los codecs que se van a instalar.



3. ***make***

   Este comando ordena la compilación del programa; permite crear los ejecutables del programa. Si todo ha salido bien se sugiere que se edite *make install*



4. ***make install***

   Este comando copia los binarios ejecutables de Asterisk (ya compilado) a cada directorio del sistema para su ejecución. Si el procedimiento es correcto9- se despliega una ventana como la siguiente:

```
+                                                     +
+    YOU MUST READ THE SECURITY DOCUMENT              +
+                                                     +
+ Asterisk has successfully been installed.          +
+ If you would like to install the sample            +
+ configuration files (overwriting any               +
+ existing config files), run:                       +
+                                                     +
+                make samples                         +
+                                                     +
+-----------------  or ----------------------------+
+                                                     +
+ You can go ahead and install the asterisk          +
+ program documentation now or later run:            +
+                                                     +
+               make progdocs                         +
+                                                     +
+ **Note** This requires that you have               +
+ doxygen installed on your local system             +
+-----------------------------------------------------+
[root@localhost asterisk-1.8.12.2]#
```

5. ***make config (opcional)***

   Crea los scripts de inicio y detención del servicio Asterisk.

6. ***Make samples (opcional)***

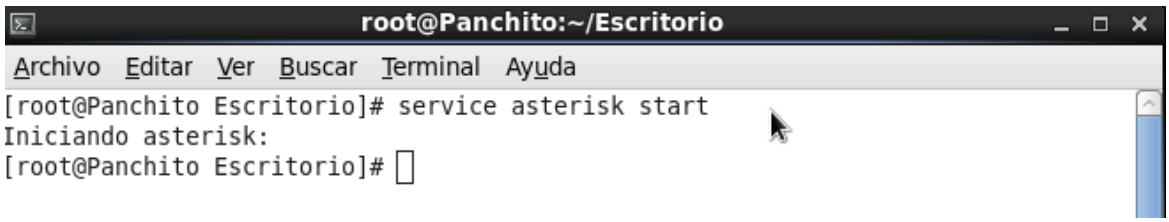
   Crea archivos de configuración de ejemplo para Asterisk.

```
            echo "Updating asterisk.conf" ; \
            sed -e 's|^astetcdir.*$|astetcdir => /etc/asterisk|' \
                  -e 's|^astmoddir.*$|astmoddir => /usr/lib/asterisk/modules|' \
                  -e 's|^astvarlibdir.*$|astvarlibdir => /var/lib/asterisk|' \
                  -e 's|^astdbdir.*$|astdbdir => /var/lib/asterisk|' \
                  -e 's|^astkeydir.*$|astkeydir => /var/lib/asterisk|' \
                  -e 's|^astdatadir.*$|astdatadir => /var/lib/asterisk|' \
                  -e 's|^astagidir.*$|astagidir => /var/lib/asterisk/agi-bin|' \
                  -e 's|^astspooldir.*$|astspooldir => /var/spool/asterisk|' \
                  -e 's|^astrundir.*$|astrundir => /var/run/asterisk|' \
                  -e 's|^astlogdir.*$|astlogdir => /var/log/asterisk|' \
                  "/etc/asterisk/asterisk.conf" > "/etc/asterisk/asterisk.conf.tmp" ; \
            /usr/bin/install -c -m 644 "/etc/asterisk/asterisk.conf.tmp" "/etc/asterisk/asterisk.conf" ; \
            rm -f "/etc/asterisk/asterisk.conf.tmp" ; \
      fi ; \
      /usr/bin/install -c -d "/var/spool/asterisk/voicemail/default/1234/INBOX"
Updating asterisk.conf
build_tools/make_sample_voicemail "//var/lib/asterisk" "//var/spool/asterisk"
Config file phoneprov/000000000000.cfg is unchanged
Config file phoneprov/000000000000-directory.xml is unchanged
Config file phoneprov/000000000000-phone.cfg is unchanged
Config file phoneprov/polycom_line.xml is unchanged
Config file phoneprov/polycom.xml is unchanged
Config file phoneprov/snom-mac.xml is unchanged
[root@localhost asterisk-1.8.12.2]#
```

Una vez instalado asterisk se puede hacer uso de los siguientes comandos para el inicio, parada y consola del servicio Asterisk:

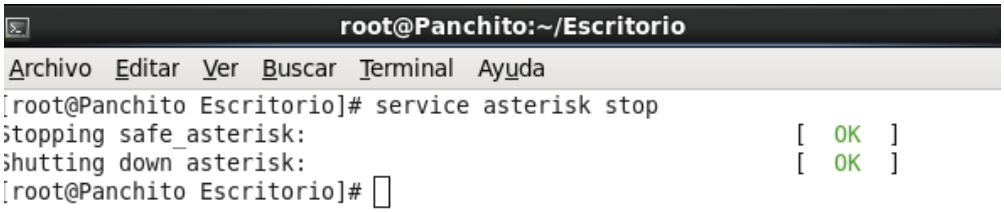
- Iniciar el servicio:

  # service asterisk start



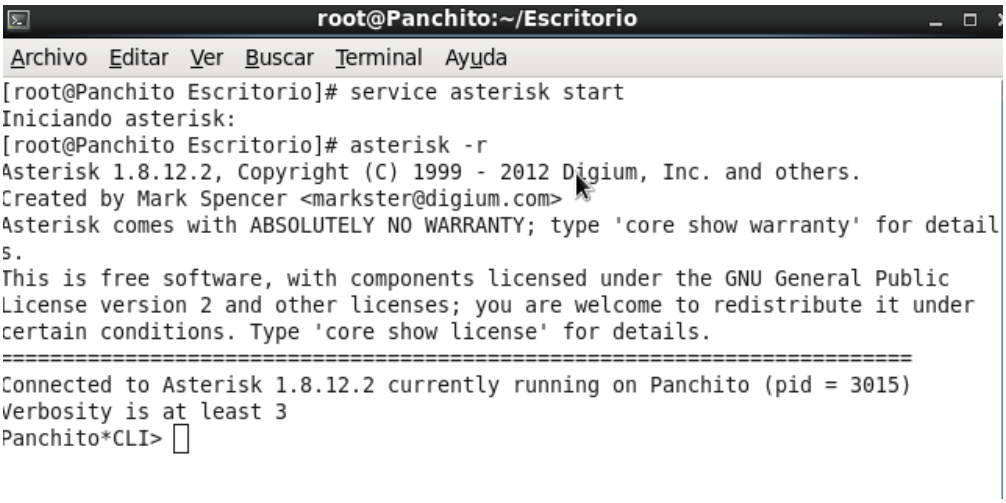
- Parar el servicio:

  # service asterisk stop



- Entrar en la consola de Asterisk

 # asterisk -r



A.3 INSTALACIÓN DE SOFTPHONES

A.3.1 INSTALACIÓN DE LINPHONE

A.3.1.1 Sobre Centos 6.2

Lo primero que se debe hacer para instalar linphone en el sistema operativo Linux (Centos 6.2) es obtener el código fuente del softphone desde la página www.linphone.org (para el caso del prototipo debido a problemas de compilación con la última versión de linphone se opta por descargarse la 3.0.0 la cual funciona de manera adecuada).

Una vez descargado el código fuente del softphone es recomendable guardarlo en */usr/src.*

Ahora con el paquete almacenado en el directorio señalado se procede a su instalación de la siguiente manera.

- Puesto que el archivo descargado es un archivo comprimido *.tar.gz se* descomprime el mismo desde consola en el directorio en el cual se encuentra con el comando ***tar xvfz nombre del paquete****.*
- Ahora se procede con la compilación del código de la siguiente manera:

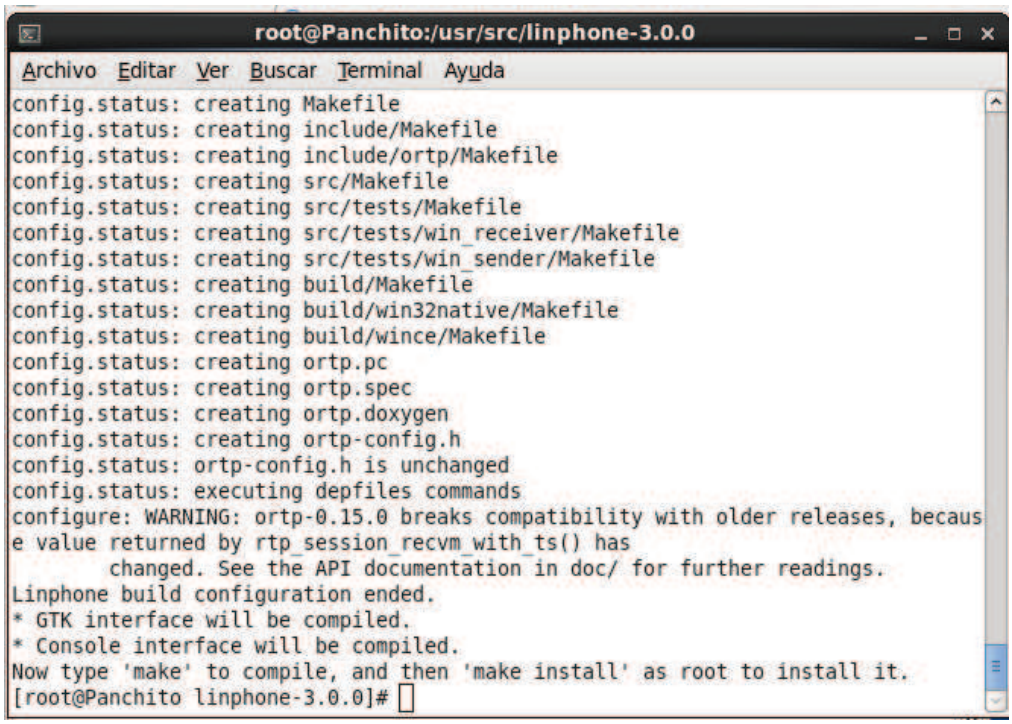
1. Se ejecuta desde consola y en el directorio en el cual se encuentra el paquete la orden: ***./configure*** mediante la cual el compilador configura el código fuente revisando que se tengan todas las dependencias necesarias para que el programa pueda ser instalado. (Para este caso con el linphone 3.0.0 necesita tener los siguientes paquetes: libosip2-3.0.3, libeXosip2-3.03, speex-1.1.6, códec gsm, gtk-2.4.0, SDL-1.2.10, ffmpeg).

   # ./configure

2. Si todas las dependencias necesarias para poder instalar linphone se encuentran presentes en el sistema operativo se muestra un mensaje de

indicación de lo que se hace a continuación, como se observar en la figura anterior. Entonces se procede a ejecutar desde consola la orden ***make.***

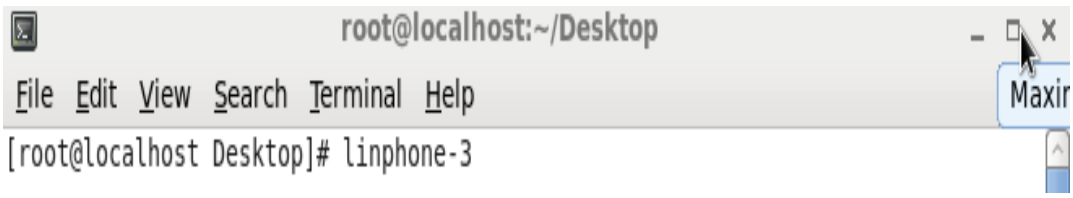
 # make



3. Si no se ha producido ningún tipo de error se procede finalmente a ejecutar la orden ***make install*** para instalar el programa.

 # make install

Una vez que el programa está instalado se puede ejecutar mediante la siguiente ruta: ***Aplicaciones/Internet/Linphone-3.0.0.0*** o editando desde consola el comando: ***linphone-3***.

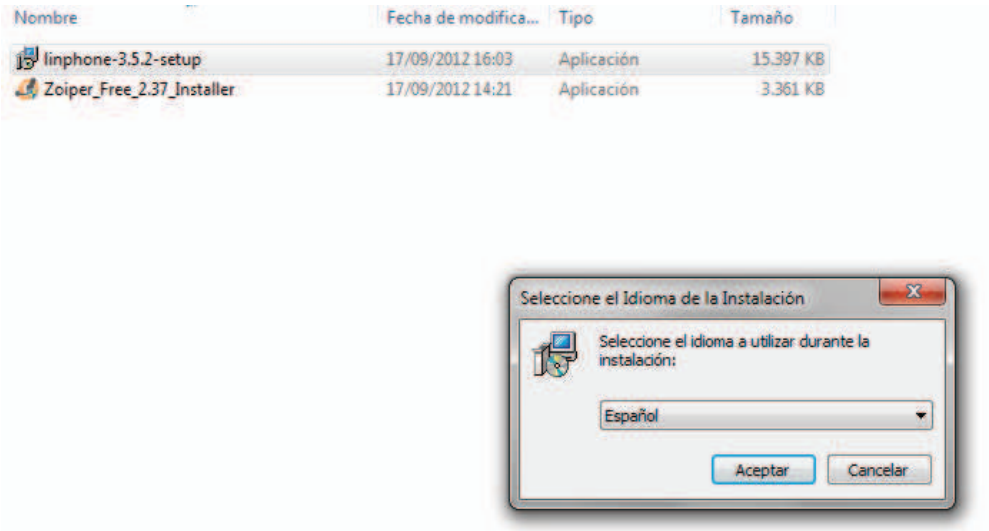


A.3.1.2 Sobre Windows

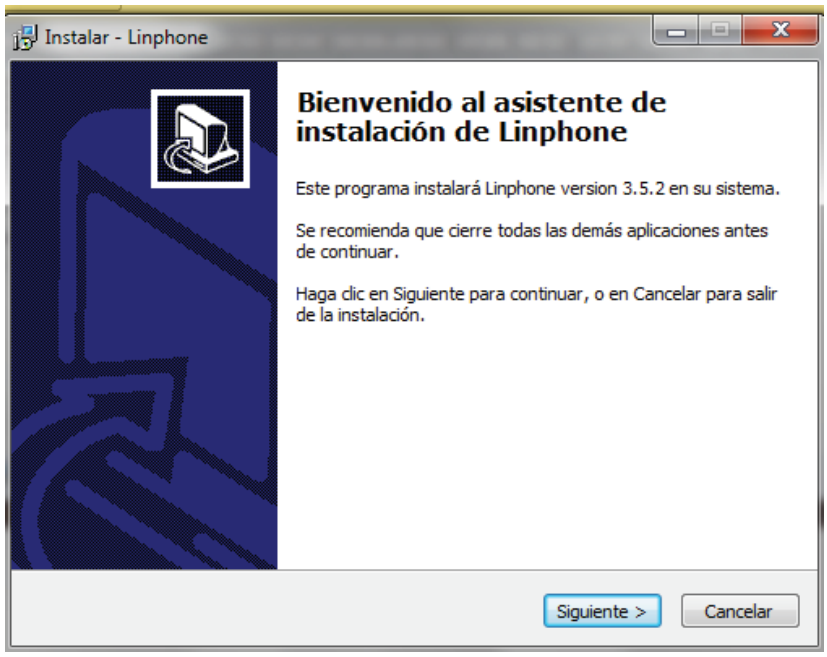
La instalación sobre Windows es mucho más sencilla ya que se cuenta con el asistente de instalación el cual indica paso a paso el procedimiento a seguir.

Los pasos a seguir son:

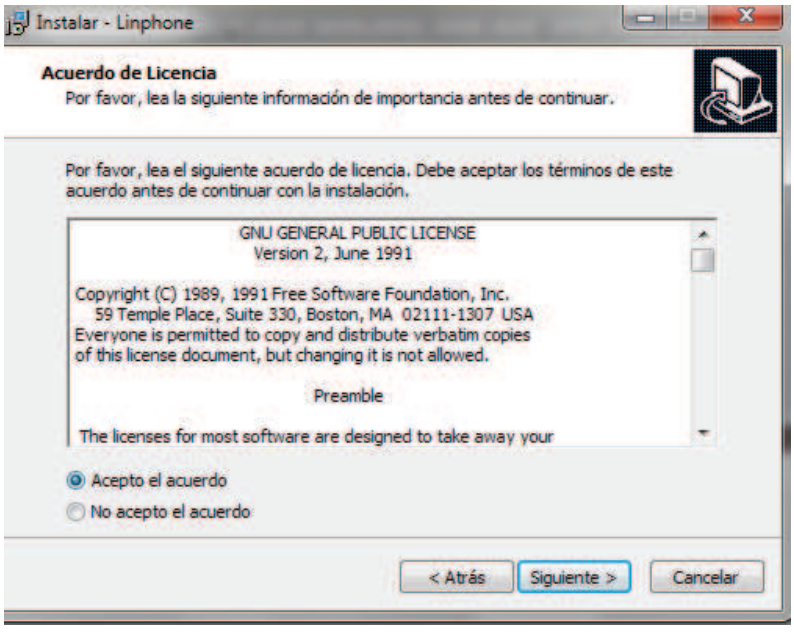
- Descargar el instalador desde la página *www.linphone.org*
- Ejecutar el instalador



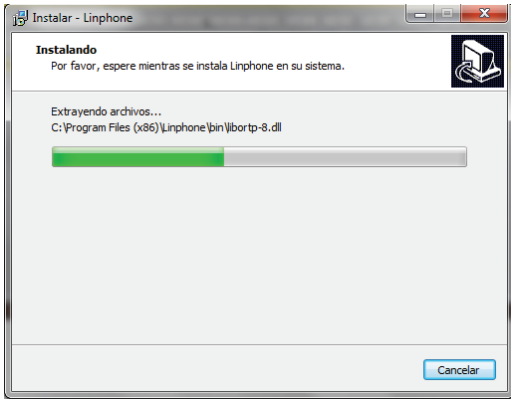
- Se escoge el idioma y se presiona en aceptar, a continuación se muestra la pantalla del asistente de instalación y para lo cual se selecciona la opción siguiente.

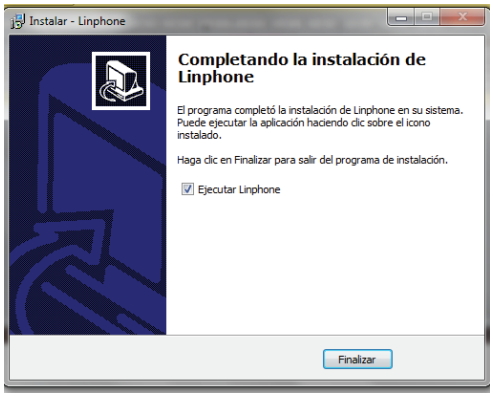


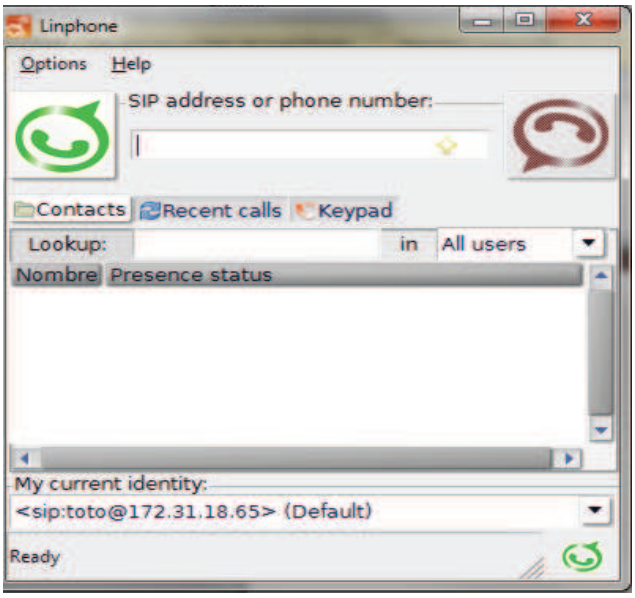
- Para continuar con la instalación se acepta el acuerdo de licencia que se presenta.



- Se continúa con el proceso de instalación siguiendo los pasos del asistente de instalación; y una vez finalizada la instalación, al dar clic en ***finish*** el softphone se ejecuta y queda instalado.





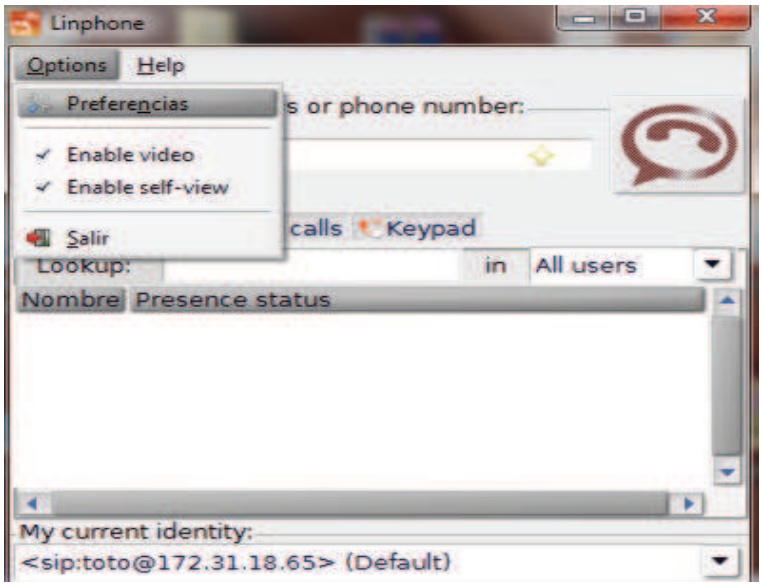


A.3.1.3 CONFIGURACIÓN DE LINPHONE PARA REGISTRARSE EN EL SERVIDOR ASTERISK

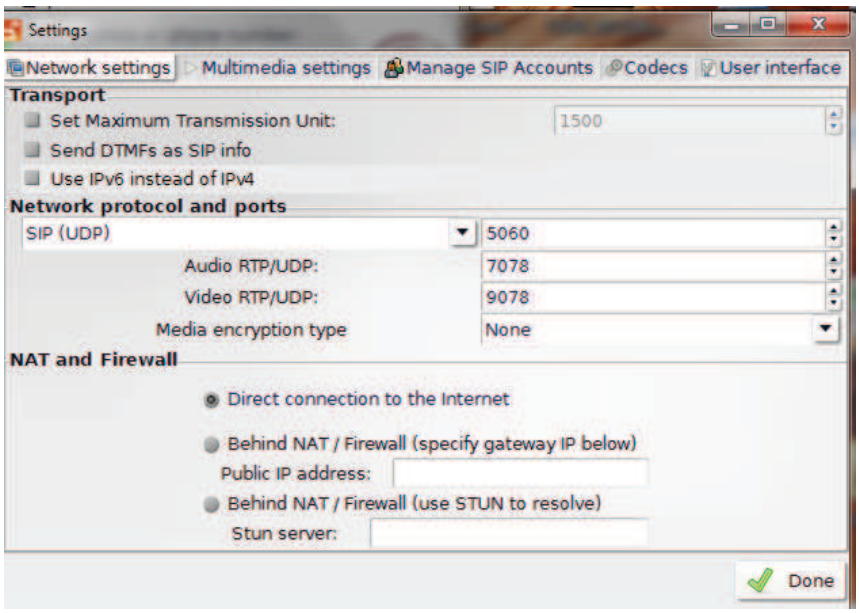
Linphone es un softphone que utiliza el protocolo estándar SIP para comunicaciones VoIP y su principal característica es que tiene soporte sobre IPv6.

Se ejecuta el softphone y en la ventana principal se procede de la siguiente manera:

- Se da un clic en la pestaña: ***Opciones/Preferencias***

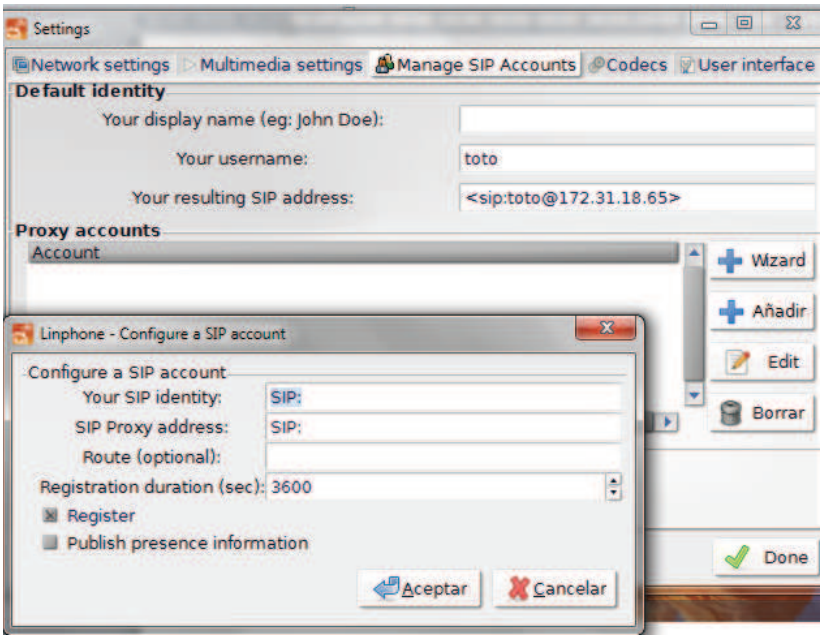


- En la ventana que aparece se configura los puertos del protocolo SIP. En la sección de ***Transport*** se configura las opciones para trabajar ya sea con IPv4 o con IPv6, si se desea que trabaje con IPv4 se deja la configuración por defecto, pero si se quiere que trabaje con IPv6 se escoge la opción ***use IPv6 instead of IPv4***.



- Se configura las cuentas en la pestaña ***Manage SIP Accounts*** *y* luego en la parte inferior izquierda en la sección (Proxy accounts) se añade una

cuenta al presionar la opción **Add** (puesto que se va a registrar al servidor asterisk).

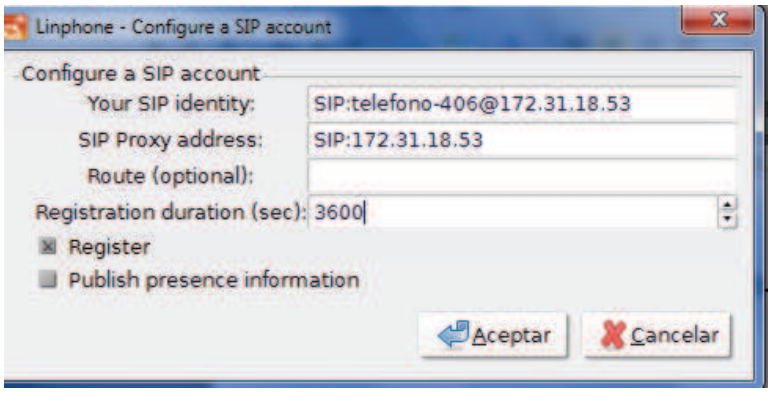


- Se completa los parámetros que se presentan de acuerdo a la siguiente manera.

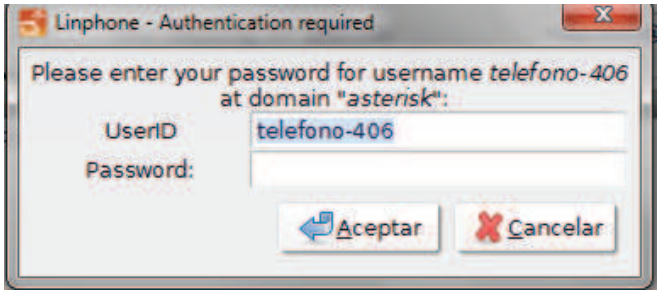
  Your SIP identity: Se escribe el nombre de usuario (como en el archivo **sip.conf** para esta cuenta) a continuación el símbolo @ seguido de la dirección IP del servidor asterisk.

  SIP Proxy addres: Se escribe la dirección IP en la cual se encuentra el servidor asterisk.

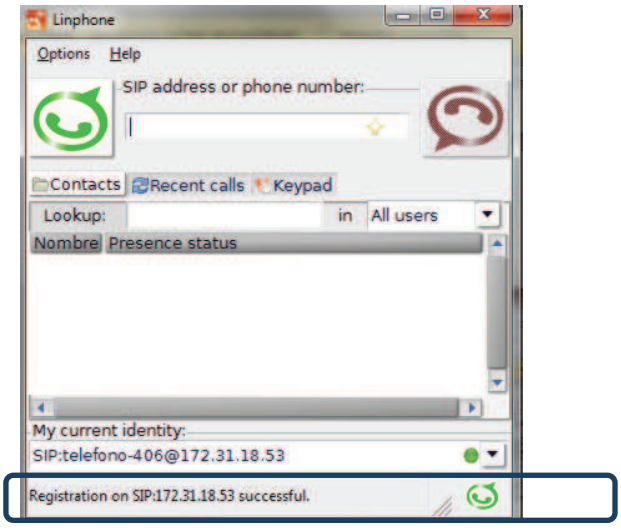
  Para el caso propuesto se tiene la siguiente salida:



- Se da un clic en aceptar e inmediatamente aparecerá una pantalla para la autenticación en la cual se escribe la misma contraseña que se configura en el archivo sip.conf para esta cuenta SIP.



Si la autenticación es correcta el softphone quedará registrado en el servidor asterisk y ya se puede hacer llamadas con este softphone cruzando por la central asterisk cuando así se requiera.



A.3.2 INSTALACIÓN Y CONFIGURACIÓN DE ZOIPER

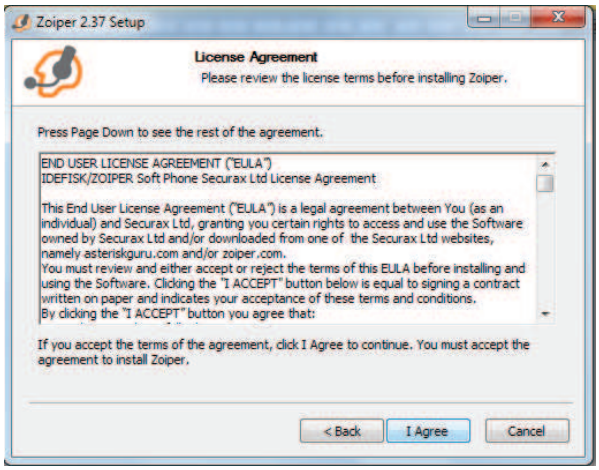
A.3.2.1 ***Instalación***

Para la instalación del softphone zoiper se procede de la siguiente manera:

- Descargar el *software* de la página de zoiper www.zoiper.com en la sección dowloads (La versión estable al momento de nuestra instalación fue la 2.37)

- Una vez descargado el ejecutable del softphone se procede a instalarlo ejecutando el instalador, mediante el cual se abrirá la pantalla del asistente de instalación, se da click en ***Next***.

- Para continuar con la instalación se acepta el acuerdo de licencia que se presenta.

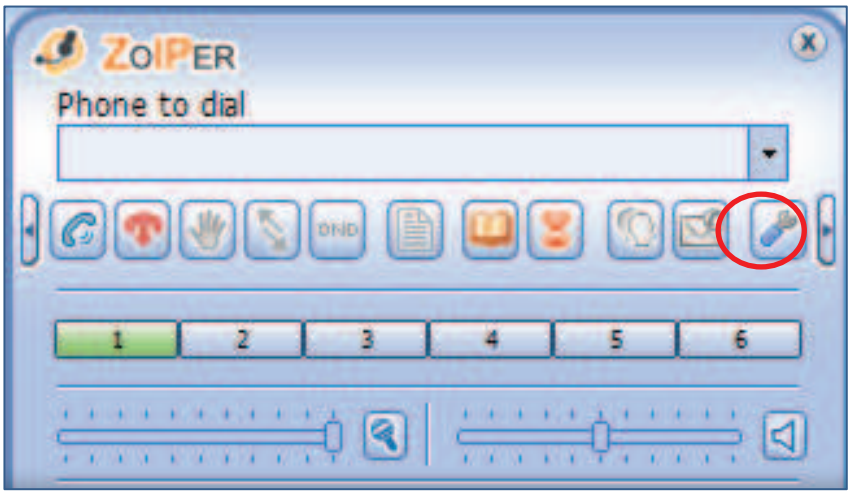


- Se continúa con el proceso de instalación siguiendo los pasos que se dan con el asistente de instalación; y una vez finalizada la instalación, al dar click en *finish* el softphone queda instalado.

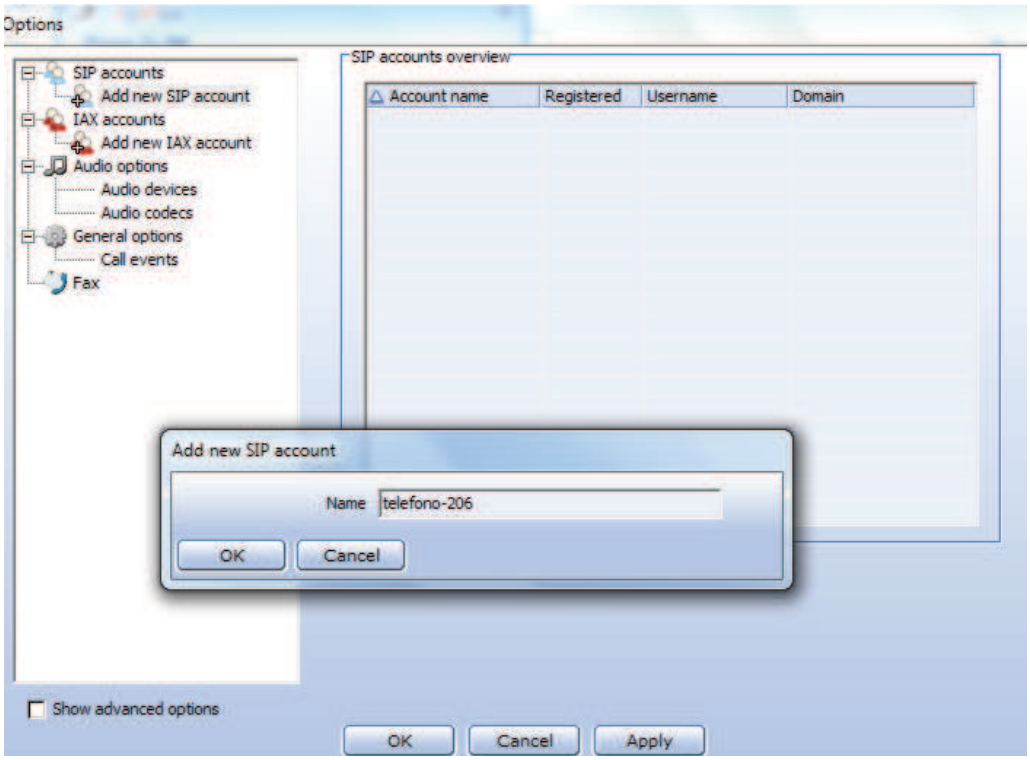
A.3.2.2 Configuración

Para registrar el softphone al servidor Asterisk se ejecuta el softphone y en la ventana principal se procede de la siguiente manera:

- Click en el icono de configuración:



- Aparecerá una pantalla en la cual seleccionaremos la Pestaña *SIP Accounts/ Add New SIP Account,* y en la pantalla que aparecerá ingresaremos el nombre de la cuenta. Para el caso de una cuenta IAX la ruta es: *IAX Accounts/ Add New IAX Account*



- Al presionar *OK* aparecerá una imagen en la cual se configura lo siguiente:

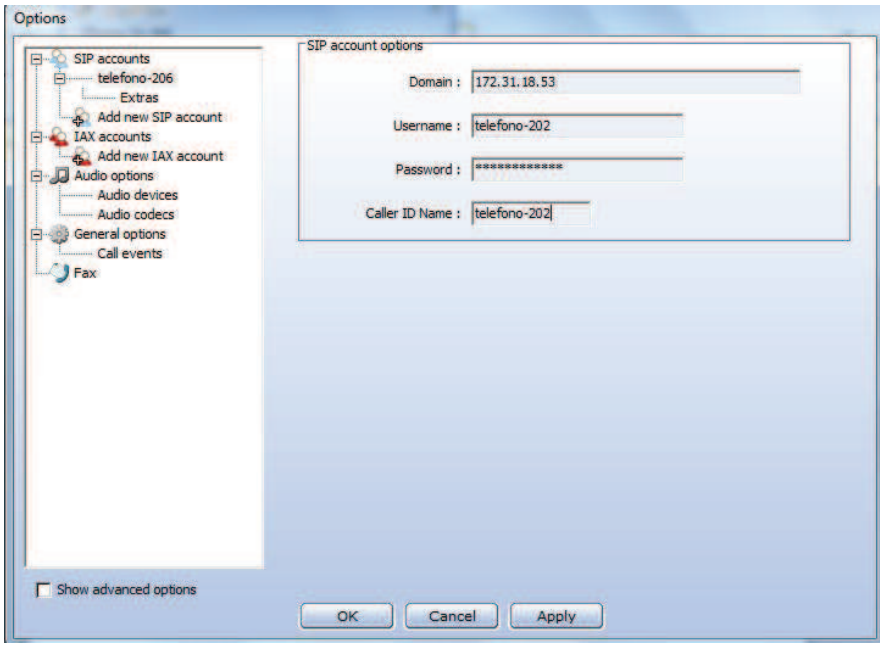
  Domain: En este campo se coloca la dirección IP del servidor de telefonía IP.

  Username: El nombre de usuario para la cuenta SIP, con el cual se realizará la autenticación (debe ser el mismo colocado en el archivo sip.conf o iax.conf)

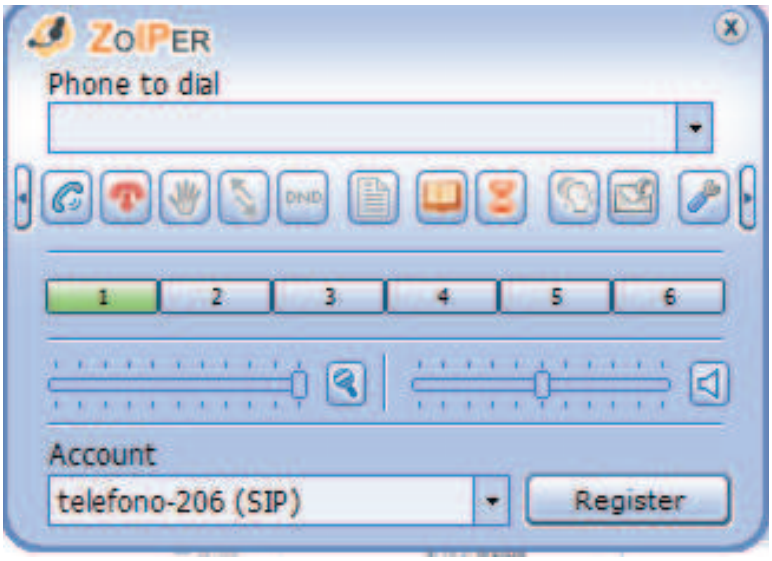
  Password: Se deberá poner el mismo que se encuentra configurado en el servidor Asterisk en el archivo de configuración sip.conf o iax.conf.

Caller ID Name: El nombre que saldrá en el localizador de los otros softphones cuando este les realice una llamada.

Se llenan dichos campos y se da un click en *OK*



- Una vez creada la cuenta hay que hacerle que se registre en el servidor dando click en la pestaña *Register.*



Si el procedimiento de registro tuvo éxito la pestaña *Register* cambiará a *Unregister.*

NOTA: Para que el proceso de registro de cualquier softphone o teléfono IP en el servidor de telefonía tega éxito se debe en el servidor desactivar el *firewall* del S.O. Para este caso particular en la consola de Centos se escribe el comando:

```
# service ip tables stop
```

A.4 INSTALACIÓN DE CLIENTE DE CORREO SQUIRRELMAIL

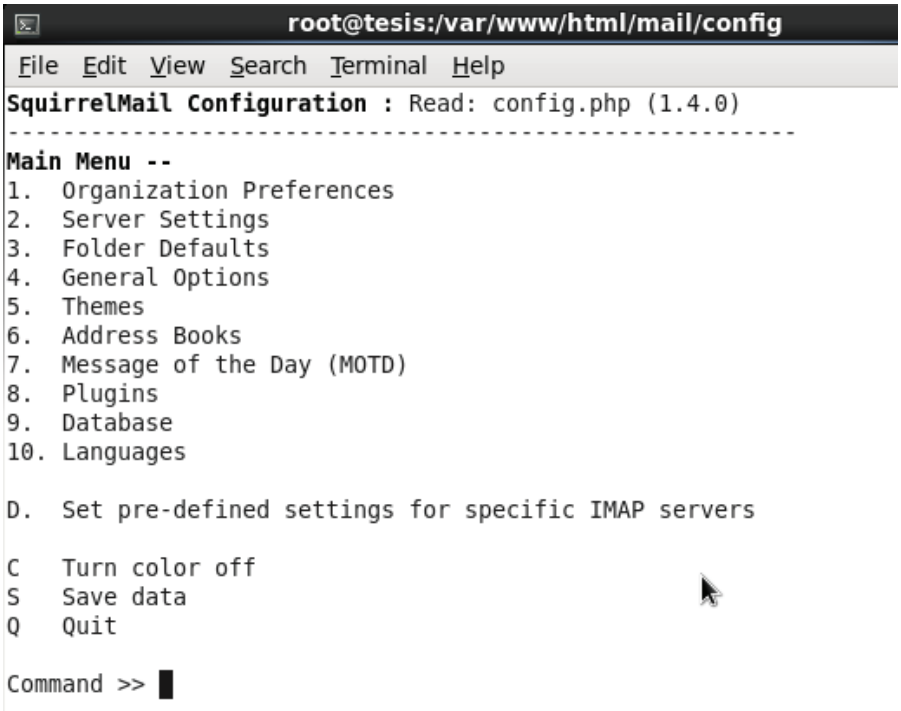
Se procede de la siguiente manera:

- *Instalación del software requerido*

```
yum -y install squirrelmail httpd
```

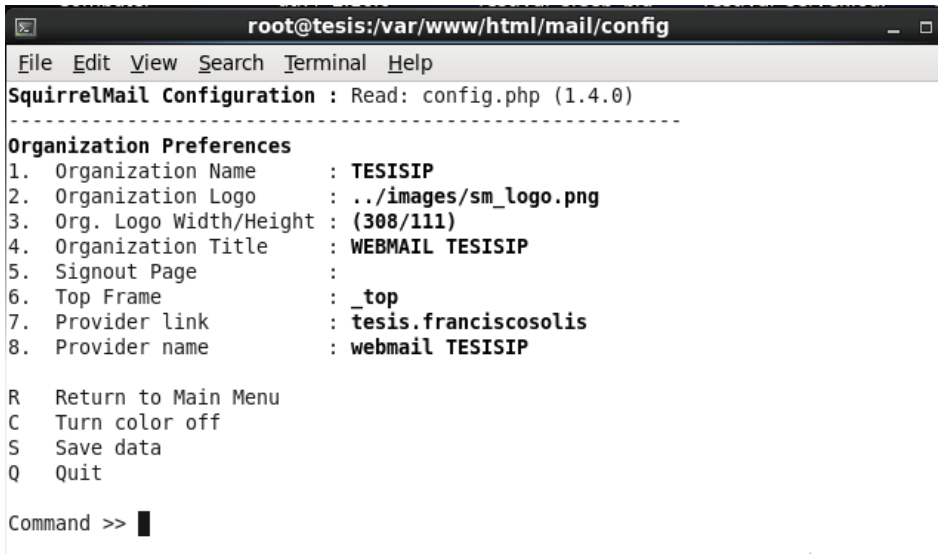
- *Configuración de Squirrelmail:* Para acceder mediante el navegador por el servidor web copiamos la carpeta squirrelmail que se encuentra en la ruta ***/usr/share/squirrelmail*** hacia la ruta ***/var/www/html/.*** Una vez copiada la carpeta procedemos a editar el archivo conf.pl

```
# cd /usr/share/squirrelmail/config/
# ./conf.pl
```



En la opción 1 (Preferencias de la organización) editamos el nombre de la empresa, el logotipo, sus dimensiones y el título de la página principal del servidor de red.

La configuración para el cliente generado resulta de acuerdo a la siguiente figura:



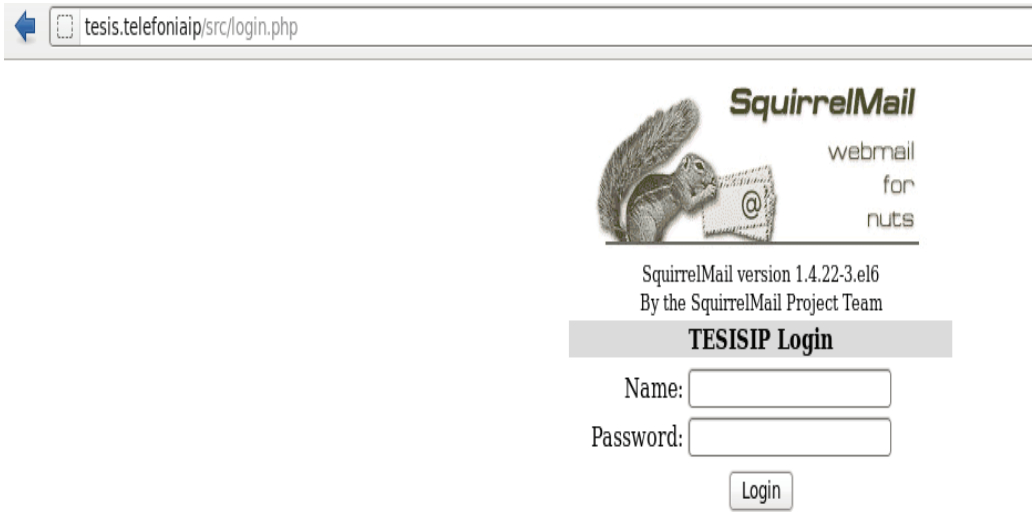
En la opción 2 (Herramientas de Servidor) se configura el dominio del servidor, el servidor SMTP e IMAP se configuran por defecto.

- Editamos con vim o gedit el archivo ***config.php*** en la carpeta de squirrelmail y modificamos los valores de las variables siguientes:

```
$domain= 'example.com'; colocar el dominio "tesis.telefoniaip" en este caso.

$smtpServerAddress = 'localhost'; colocar la IP del servidor "172.31.18.41" en este caso.

$trash_folder = 'INBOX.Trash';cambiar por 'Papelera';

$sent_folder = 'INBOX.Sent';cambiar por 'Enviados';

$draft_folder = 'INBOX.Drafts';cambiar por 'Borradores';
```

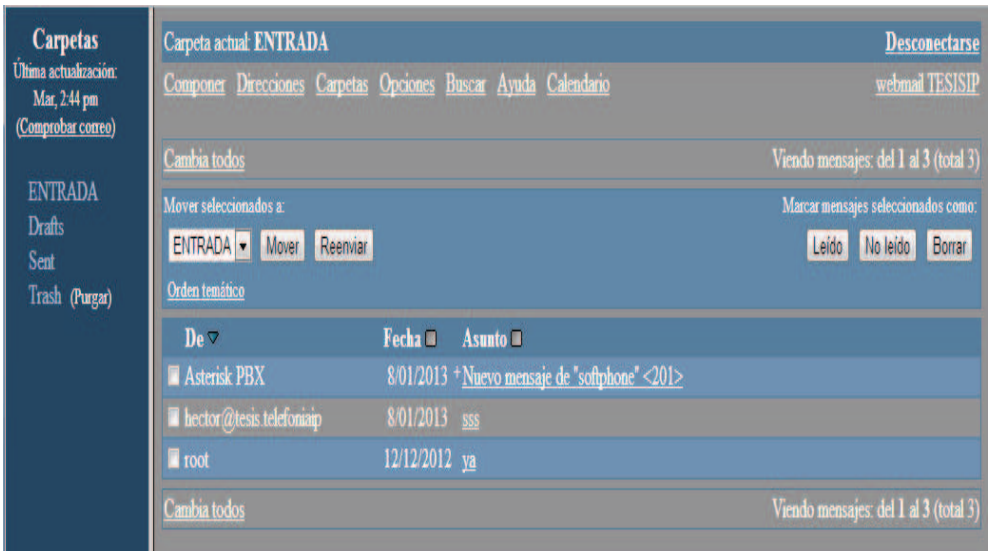
- Finalmente reiniciamos los servicios e ingresamos en el navegador y editamos http://tesis.telefoniaip o la dirección IP del servidor.

```
# service network restart
# service httpd restart
# service named restart
# service postfix restart
# service dovecot restart
```



La interfaz de usuario se la puede observar al loguearse con un nombre de usuario y una contraseña que existan en el servidor de correo (Usuarios añadidos en linux).

Una vez autenticado el cliente con un nombre de usuario y una contraseña válidos se muestra la interfaz de dicho usuario de Squirrelmail.



ANEXO B

ARCHIVOS DE CONFIGURACIÓN DE ASTERISK

B.1 ARCHIVO SIP.CONF

En este archivo se realiza la configuración de los canales que utilizarán el protocolo de comunicación SIP.

En este archivo se definen opciones generales y específicas de todos aquellos dispositivos que utilizarán dicho protocolo.

```
[general]               ;Etiqueta en la cual se introducirá la parte
                        general  de      todos los canales SIP
context=teléfonos      ;Contexto al cual llegarán las llamadas no
                        autenticadas
allowguest=no           Se impiden llamadas entrantes de usuarios no
                        autenticados
srvlookup=yes           Permitir búsqueda DNS de llamadas salientes
udpbindaddr=[::]        ;Para que acepte tanto conexiones IPv4 como
                        conexiones IPv6
tcpenable=no            ;No se permite señalización SIP sobre TCP.
qualify=yes             ;Para mantener la activa conexión de una
                        extensión que se conecta a Asterisk detrás de
                        un NAT.
allowtransfer=yes       ;Poder hacer uso de la transferencia de
                        llamadas
language=es             ;Idioma predefinido para las locuciones
rtptimeout=60           ; Tiempo de espera para cerrar una llamada si
                        no hay flujo de audio
rtpholdtimeout=300
directmedia=no
dtmfmode=rfc2833
callcounter=yes

#include sip_trunks.conf    ;Incluye en este archivo el archivo
                            mencionado
#include sip_custom.conf
insecure=invite,port        ; No se le solicita autenticación para
                            poder registrarse en el sistema y
                            realizar las llamadas; además se ignora
                            el puerto por el cual se generen las
                            llamadas

[softphones](!)         ; Etiqueta a ser aplicada a cada uno de las
                        extensiones, el signo de admiración lo
                        convierte en una plantilla que se puede aplicar
                        a una determinada etiqueta y así evitar repetir
                        dicha configuración.
```

```
type=friend             ;La extensión se autentica como usuario (usando
                        el campo From) o como peer (Utilizando la IP y
                        el puerto)
context=telefonos
host=dynamic            ;Para extensiones que se conectan al servidor
                        de manera remota
secret=asterisk2012     ;Contraseña para la autenticación
disallow=all            ;Desactivar todos los codecs
allow=gsm,ulaw,alaw     ;Activar los codecs mencionados
bindport=5061           ;Puerto de escucha
qualify=yes             ;Para mantener  activa la conexión de una
                        extensión que se conecta a Asterisk detrás de
                        un NAT.
nat=yes                 ;Impone siempre el uso del parámetro rport y
                        envía el flujo audio/video por el mismo puerto
                        utilizado por el dispositivo remoto. Es
                        necesario colocar en yes para dispositivos que
                        atraviesan routers o firewalls.

[telefono-201](softphones)  ;Se aplica la configuración de la
                            plantilla a la etiqueta mencionada
callerid=softphone <201>    ; Identificador de llamada
mailbox=201@tesis           ; Identificador de Conexión para el buzón
                            de voz asociado al archivo voicemail.conf
callgroup=1                 ; Grupo de llamado
pickupgroup=1               ; Grupo para realizar la funcionalidad de
                            pickup. Estos dos parámetros definen una
                            de las funcionalidades avanzadas de
                            Asterisk, es decir la posibilidad de
                            contestar una llamada de una extensión
                            que está timbrando, desde otra extensión.

[telefono-202](softphones)
callerid=softphone <202>
mailbox=202@tesis
callgroup=1
pickupgroup=1

[telefono-203](softphones)
callerid=softphone <203>
mailbox=203@tesis
callgroup=1
pickupgroup=1

[telefono-204](softphones)
callerid=softphone <204>
mailbox=204@tesis
callgroup=1
pickupgroup=1

[telefono-205](softphones)
callerid=softphone <205>
mailbox=205@tesis
callgroup=1,2
pickupgroup=1

[telefono-206](softphones)
```

```
callerid=softphone <206>
mailbox=206@tesis
callgroup=1
pickupgroup=1

[telefono-401](softphones)
callerid=softphone <401>
callgroup=2
pickupgroup=1,2

[telefono-402](softphones)
callerid=softphone <402>
callgroup=2
pickupgroup=2

[telefono-403](softphones)
callerid=softphone <403>
callgroup=2
pickupgroup=2

[telefono-404](softphones)
callerid=softphone <404>
callgroup=2
pickupgroup=2

[telefono-405]
callerid=softphone <405>
callgroup=2
pickupgroup=2
type=friend
context=telefonos
host=dynamic
secret=asterisk2012
disallow=all
allow=gsm,ulaw,alaw
bindport=5061
qualify=yes
nat=yes

[sipp]
type=friend
context=sipp
host=dynamic
user=sipp
insecure=invite,port
canreinvite=no
disallow=all
allow=ulaw
allow=alaw
```

B.1.1 ARCHIVO SIP_TRUNKS.CONF

```
[cnt]
type=peer
context=entrada_trunk
host=172.31.18.53
disallow=all
allow=g723.1,g729,gsm
qualify=yes
```

B.1.2 ARCHIVO SIP_CUSTOM.CONF

```
[asterisk_test]
type=friend
context=congestion
host=dynamic
user=sipp
directmedia=no
disallow=all
allow=ulaw
```

B.2 ARCHIVO IAX.CONF

```
[general]
disallow=all
allow=gsm,ulaw,alaw
qualify=yes
requirecalltoken=no    ; No se requiere el calltoken por parte de la
                       troncal IAX

maxregexpire=1200      ; Tiempo máximo en segundos luego del cual
                       expira el registro de troncales IAX

language=es
qualify=yes
context=entrada

[telefono-302]
type=friend
context=entrada
host=dynamic
secret=asterisk2012
disallow=all
allow=gsm,ulaw,alaw
callerid=anonimo <302>

[telefono-303]
type=friend
context=entrada
host=dynamic
secret=asterisk2012
disallow=all
allow=gsm,ulaw,alaw
callerid=anonimo <303>
```

B.3 ARCHIVO EXTENSIONS.CONF

```
[general]
language=es
static=yes                    ; Si static es yes y writeprotect es no,
                              para guardar los cambios hechos desde la
                              consola de asterisk tendremos que
                              escribir el comando dialplan reload.
writeprotect=no
autofallthrough=yes           ; Si una llamada por algún motivo se sale
                              del plan, ésta se cerrará automáticamente

#include extensions_database.conf

[telefonos]
include =>pruebas
include =>redsip
include =>iaxred
include =>buzon
include =>directorio
include =>database
include =>consulta-notas
include =>redipv6
include =>agitest
include =>analogicos
include =>PSTN
include =>celulares
include =>laboratorios
include =>FXO
include =>IVR
include =>conferencias
include =>sipp

[entrada]
include =>iaxred
include =>pruebas
include =>buzon
include =>directorio
include =>database
include =>consulta-notas

[subscribe]
exten => _70Z,hint,park:${EXTEN}@parkedcalls
exten => 710,hint,park:710@parkedcalls

[pruebas]
exten => 500,1,Playback(demo-congrats)   ;Se marca la extensión y se
                                         reproduce un sonido (demo-
                                         congrats)
```

```
exten => 500,2,HangUp()                    ;Se cuelga la llamada

;exten => 600,1,Festival(hola)             ; Se marca la extensión y se
                                           usa la aplicación Festival
                                           (para convertir texto plano en
                                           voz)
exten => 600,1,Playback(IVR1)
same => n,HangUp()

exten => 700,1,Festival(Esta? es? una? prueba?)
exten => 700,2,HangUp()

[redsip]
include =>parkedcalls
exten => _20x,1,Set(ESTADO_TELEFONO=${DEVICE_STATE(SIP/telefono-
${EXTEN})})            ; Se muestra el estado de la extensión marcada
(                       puede ser de la 200 a la 299).


same => n,Verbose(El canal está ${ESTADO_TELEFONO})      ;Verbose para
que en la consola de asterisk se muestre la variable ESTADO_TELEFONO

same => n,Gotoif($[${ESTADO_TELEFONO} = NOT_INUSE]?:buzon)    ;
Aplicación condicional, en este caso si se cumple se cumple la
siguiente línea de programación caso contrario se envía a la etiqueta
buzon.

same => n,Dial(SIP/telefono-${EXTEN},20,kKtTxX) ; Marcar a la
extensión, ocupando el canal SIP con nombre teléfono- (variable),
marcar por 20 segundos.

same => n,Gotoif($[${DIALSTATUS} = NOANSWER] | $[${DIALSTATUS} =
BUSY]?buzon:terminar) ;Verificar la condición mostrada (Estado de la
marcación: no responde u ocupado)

same => n(buzon),Voicemail(${EXTEN}@tesis,ub); Utiliza la aplicación
voicemail con los parámetros establecidos

same => n(terminar),Hangup()

[iaxred]
include =>parkedcalls
exten => _30x,1,Dial(IAX2/telefono-${EXTEN},20,kKtTxX)
same => n,Hagup()

[redipv6]
include =>parkedcalls
exten => _40x,1,Set(ESTADO_TELEFONO=${DEVICE_STATE(SIP/telefono-
${EXTEN})})
same => n,Verbose(El canal está ${ESTADO_TELEFONO})
same => n,Gotoif($[${ESTADO_TELEFONO} = NOT_INUSE]?:buzon)
same => n,Dial(SIP/telefono-${EXTEN},20,kKtTxX)
same => n,Gotoif($[${DIALSTATUS} = NOANSWER] | $[${DIALSTATUS} =
BUSY]?buzon:terminar)
same => n(buzon),Voicemail(${EXTEN}@tesis,ub)
same => n(terminar),Hangup()
```

```
 [analogicos]
include =>parkedcalls
exten => 2514,1,Dial(DAHDI/4,20,kKtTxX)
same => n,Gotoif($[${DIALSTATUS} = NOANSWER] | $[${DIALSTATUS} =
BUSY]?buzon:terminar)
same => n(buzon),Voicemail(${EXTEN}@tesis,ub)
same => n(terminar),Hangup()


[buzon]
exten => 99,1,Answer()
exten => 99, n,VoiceMailMain(${CALLERID(num)}@tesis)
exten => 99,n,HangUp()

exten => 98,1,Answer()
exten => 98,n,VoiceMailMain(@tesis)
exten => 98,n,HangUp()

[directorio]
exten => 97,1,Answer()
same => n,Directory(tesis,redsip,eb)
same => n,HangUp()

[PSTN]
exten => _xxxxxxx,1,Dial(DAHDI/1/${EXTEN},20)   ; Marcar a la
extensión mencionada utilizando la tecnología DAHDI a través del
canal 1 (FXS en este caso). Marcar por 20 segundos.

same => n,Hangup()

[celulares]
exten => _0xxxxxxxxx,1,Dial(DAHDI/1/${EXTEN},20)
same => n,Hangup()

[laboratorios]
exten => _2xxx,1,Dial(DAHDI/1/${EXTEN},20)
same => n,Hangup()

[FXO]
exten => s,1,Goto(IVR,s,1)

[IVR]
exten => s,1(menu),Background(IVR-TESIS)     ;Extensión de inicio
utilizada para entrar en un contexto con cualquier extensión. Se
reproduce el sonido mostrado mediante la aplicación Background (si se
presiona alguna tecla el sonido se pausa y se pasa a la siguiente
línea de programación).

exten => s,n,Set(TIMEOUT(digit)=5)
exten => s,n,Set(TIMEOUT(response)=5)
exten => s,n,Wait(1)                    ;Esperar 1 segundo
exten => s,n,Waitexten(5)               ;Esperar la recepción de
                                        señalización de la extensión
                                        por 5 segundos.

;
```

```
exten => 1,1,Goto(analogicos,2514,1)    ; Se cumple si se marca el
dígito 1. Se envía a la etiqueta analógicos con extensión 2514 y
prioridad 1.

exten => 1,n,Hangup()

;
exten => 2,1,Goto(redsip,205,1)   ;Se cumple si se marca el dígito 2.
Se envía a la etiqueta redsip, con 205  y prioridad 1.

exten => 2,n,HangUp()

;
exten => 3,1,Goto(redipv6,405,1)
exten => 3,n,HangUp()

;
exten => i,1,Festival(Opcion? invalida?)     ; Se cumple si se marca
un dígito no existente en la programación.

exten => i,n,Festival(Use?)
exten => i,n,Goto(IVR,s,menu)
exten => i,n,HangUp()

;
exten => t,1,Festival(use las siguientes? extensiones)  ; Se cumple
si luego de un tiempo establecido no se ha recibido la marcación de
ningún dígito existente en la programación del IVR.

exten => t,n,Goto(IVR,s,menu)     ;Se ejecuta la programación en el
contexto IVR con extensión s, en la etiqueta menu.

exten => t,n,Hangup()


[conferencias]
exten => _350[012],1,Meetme(${EXTEN},scM(default))      ;Se admiten
las extensiones 3500, 3501, 3502. Se usa la aplicación meetme con las
opciones scM explicadas en el apartado 3.4.2.3

same => n,HangUp()

[sipp]
exten => 1234,1,Answer()
same => n,While(1)
same => n,Background(demo-instruct)
same => n,HangUp()
```

B.3.1 ARCHIVO EXTENSIONS_DATABASE.CONF

```
[consulta-notas]
exten => 511,1,Answer()
same => n,Festival(Ingrese su clave)
same => n(clave),Read(CLAVES,beep,5,,2,6)    ;Recibe los dígitos
introducidos por el usuario y lo guarda en una variable llamada
CLAVES. Permite hasta 5 dígitos y espera por estos hasta 6 sgundos.

same => n,Set(CLAVEDB=${DB(profesores/David-Vaca)})     ; Setea el
valor del campo almacenado en la base de datos de asterisk
perteneciente a dicha familia, llave y valor y lo asocia a la
variable llamada CLAVEDB.

same => n,Gotoif($[${CLAVES}=${CLAVEDB}]?correcto:fin)  ; Compara las
variables CLAVES y CLAVEDB. Si se cumple la condición el proceso de
ejecución pasa a la etiqueta correcto, caso contrario pada a la
etiqueta fin.

same => n(correcto),Festival(Ingrese el numero de cedula del
estudiante)
same => n,Read(CI,beep,10,,2,12)
same => n(cedula),Gotoif($[${DB_EXISTS(Alumnos/${CI})}]?v:f)
     ;Verifica si el número de cédula ingresado existe

same => n(v),Festival(Ingrese la nota)
same => n,Read(NOTA,beep,2,,2,3)
same => n,Set(DB(notas/${CI})=${NOTA}) ; Asocia  la nota ingresada a
la base de datos de asterisk (en la familia notas, llave C.I).

same => n(fin),Festival(Usted ha finalizado el proceso)
same => n,HangUp()
same => n(f),Festival(Cedula no existe)
same => n,Goto(correcto)

exten => 512,1,Answer()
same => n(ingrese),Festival(Ingrese su cedula de identidad)
same => n,Read(CI,beep,10,,2,12)
same => n,Gotoif($[${DB_EXISTS(Alumnos/${CI})]?v:f)
same => n(v),Set(NOMBRE=${DB(Alumnos/${CI})})
same => n,Festival(Usted es? ${NOMBRE})
same => n,Festival(Presione? uno si asi lo es)
same => n,Read(CONFIRMAR,beep,1,2,,3)
same => n,Gotoif($[${CONFIRMAR}=1]?si:no)
same => n(si),Festival(Su nota es? ${DB(notas/${CI})})
same => n(no),HangUp()
same => n(f),Goto(ingrese)
```

B.4 ARCHIVOS DE CONFIGURACIÓN DE DAHDI

B.4.1 CHAN_DAHDI.CONF

```
[trunkgroups]

[channels]
usecallerid=yes
callwaiting=yes
usecallingpres=yes
callwaitingcallerid=yes
threewaycalling=yes
transfer=yes
canpark=yes
cancallforward=yes
callreturn=yes
echocancel=yes
echocancelwhenbridged=yes
busydetect=yes
faxdetect=both
language=es
#include dahdi-channels.conf        ;Es la única línea a
                                    editar en este archivo,
                                    las otras se las deja con
                                    la configuración por
                                    defecto

callwaitingcallerid=yes
;
; Support three-way calling
;
threewaycalling=yes
transfer=yes
canpark=yes

callreturn=yes
echocancel=yes
echocancelwhenbridged=yes
group=1
callgroup=1
pickupgroup=1
```

B.4.2 DAHDI-CHANNELS.CONF

Al instalar el modulo DADHI y las tarjetas FXS y FXO estas configuraciones se crean por defecto. Simplemente editaríamos el ***callerid*** (con el número de extensión correspondiente). Así como el ***mailbox.***

```
;;; line="1 WCTDM/4/0 FXSKS"
signalling=fxs_ks
callerid=asreceived
group=5
context=telefonos
channel => 1
callerid=
group=
context=default

;;; line="4 WCTDM/4/3 FXOKS"
signalling=fxo_ks
callerid="Channel 4" <2514>
mailbox=2514@tesis
group=5
context=telefonos
channel => 4
```

B.5 FEATURES.CONF

```
[general]
parket => 700              ;Extensión a dónde transferir una
                           llamada para parquearla

parkpos => 701-709         ;Extensiones reservadas para
                           parquear las llamadas.

parkinghints=no            ;Para configurar manualmente en el
                           dialplan las prioridades Hint para
                           monitorear el estado de las
                           extensiones dedicadas al parqueo.

parkingtime => 45          ;Tiempo en segundos que quedará
                           parqueada la llamada. Luego de este
                           tiempo la llamada se transfiere a la
                           extensión definida en el próximo
                           parámetro.

comebacktoorigin=yes       ;Para que la llamada parqueada se
                           transfiera a la extensión que la
                           parqueó.

courtesytone=beep          ; Locución que se enviara al canal
                           parqueado cuando alguien lo llama o
                           cuando se activa/desactiva la
                           grabación de la llamada

parkedplay=both            ;Para que envié el courtesytone
                           tanto al canal parqueado como al que
                           llama al canal parqueado.

parkedcalltransfers=caller    ;Activa o desactiva la
                              secuencia de tonos para
                              transferir una llamada cuando
                              es una llamada parqueada. En
                              este caso se aplica al
                              llamante.

parkedcallreparking=caller    ;Activa o desactiva la
                              secuencia de tonos para
                              parquear una llamada cuando es
                              una llamada parqueada. En este
                              caso se aplica al llamante.

parkedcallhangup=caller       ;Activa o desactiva la
                              secuencia de tonos para
```

```
                                terminar una llamada cuando es
                                una llamada parqueada.  En este
                                caso se aplica al llamante.

parkedcallrecording=caller      ;Activa   o   desactiva   la
                                secuencia de tonos para grabar
                                una   llamada   cuando   es   una
                                llamada   parqueada.    En   este
                                caso se aplica al llamante.

parkedcallmusicclass=default    ;Clase de música que escuchará
                                el canal parqueado.

transferdigittimeout => 5       ;Tiempo de espera en segundos
                                entre los dígitos cuando se
                                está transfiriendo una llamada.

xfersound=beep                  ;Sonido que notificará que la
                                transferencia    de    llamada
                                asistida ha sido exitosa.

xferfailsound=beeper            ;Sonido que notificará que la
                                transferencia    de    llamada
                                asistida no ha sido exitosa.

pickupexten=*8                  ;Combinación de dígitos para
                                jalar   la   llamada   de   otro
                                teléfono.

pickupsound=beep                ;Sonido que notificará que la
                                captura   de   llamada   ha   sido
                                exitosa.

pickupfailsound=beeper          ;Sonido que notificará que la
                                captura de llamada no ha sido
                                exitosa.

pickupdigittimeout=2000         ;Tiempo de espera de secuencia
                                de dígitos para capturar una
                                llamada.

atxfernoanswertimeout=15        ;Tiempo máximo en segundos para
                                contestar   una   transferencia
                                asistida.
```

```
atxferdropcall=no          Si quien transfiere una llamada con
                           el método "asistido" cuelga antes
                           que la llamada sea transferida
                           completamente, Asterisk devuelve la
                           llamada a quien la estaba
                           transfiriendo.

atxferloopdelay=10         ;Tiempo de espera en segundos antes
                           de intentar devolver la llamada.

atxfercallbackretries=2    ;Número de veces que se intentara
                           devolver una llamada transferida a
                           quien la transfirió sin éxito.


[featuremap]
blindxfer => *4      ;Secuencia de dígitos para activar la
                     transferencia ciega.

disconect => *0      ;Colgar la llamada presionando la
                     secuencia indicada.

automon => *1        ;Secuencia de dígitos para grabar la
                     llamada (en dos archivos de audio, uno
                     para cada canal).

atxfer => *2         ;Secuencia de dígitos para activar la
                     transferencia asistida.


parkcall => *7       ;Secuencia de dígitos para parquear una
                     llamada.


automixmon => *3     ;Secuencia de dígitos para grabar la
                     llamada (en un solo archivo de audio).


[applicatiomap]
test1 => *9,peer,Playback,tt-monkeys,default
```

B.6 MEETME.CONF

```
[general]
audiobuffers=32     ;Número de paquetes de audio de 20ms que
                    serán guardados en un buffer de memoria
                    cuando pertenecen a canales que no son
                    DADHI. Esto permite sincronizar el audio
                    de los distintos participantes y evitar
                    retrasos

[rooms]
conf => 3500
conf => 3501,1234         ;Extensión con PIN de usuario.
conf => 3502,1234,5678    ;Extensión con PIN de usuario y PIN
                          de administrador.
```

B.7 VOICEMAIL.CONF

```
[general]
format=wav49                    ;Formato de audio en el cual se
                                grabará el mensaje de voz.

servermail=asterisk@tesis.telefoniaip   ;Correo electrónico
                                del remitente en las
                                notificaciones de la
                                presencia de un nuevo
                                mensaje de voz.

attach=yes                      ;Para que se adjunte el archivo
                                de audio que contiene el
                                mensaje de voz al mensaje de
                                correo.

delete=no
maxmsg=100                      ;Número máximo de mensajes de
                                voz guardados para cada buzón
                                de voz.

maxsecs=180                     ;Tiempo máximo en segundos que
                                puede durar un mensaje de voz.

minsecs=2                       ;Tiempo mínimo en segundos para
                                que un mensaje de voz sea
                                reconocido como tal.

skipms=3000
maxsilence=10                   ;Tiempo máximo en segundos que
                                puede existir silencio en un
```

```
                                mensaje de voz. Luego de este
                                tiempo se termina la grabación
                                del mensaje.

silencethreshold=128            ;Determina el nivel de ruido

maxlogins=3                     ;Máximo número de intentos para
                                editar la clave del buzón de
                                voz

emailsubject=Nuevo mensaje de ${VM_CALLERID};Asunto      del
correo electrónico.

emailbody=Buenos dias ${VM_NAME}, \n\nHemos recibido un
mensaje en su buzon de voz...     ;Contenido del correo
electrónico.

emaildateformat=%A, %B %d, %Y at %r    ;Formato de la hora y
fecha del correo electrónico.

[tesis]   ;Etiqueta que debe llevar el mismo nombre que el
          configurado en el parámetro mailbox en el archivo
          de configuración sip.conf.


201 => 1234,Diego Garcia,diego@tesis.telefoniaip

;Extensión=> clave de buzón de voz, Nombre Apellido,
dirección de correo electrónico

202 => 2345,Hector Moyon,hector@tesis.telefoniaip
205 => 3456,David Vaca,david@tesis.telefoniaip
2514 => 5678,Camilo Calle,camilo@tesis.telefoniaip
402 => 6789,Cristian Tintin,tintin@tesis.telefoniaip
403 => 5678,Julian Jaramillo,julian@tesis.telefoniaip
```

ANEXO C

ARCHIVOS DE CONFIGURACIÓN DE LOS ROUTERS

C.1 ROUTER 1

```
!
version 15.1
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname R1
!
boot-start-marker
boot-end-marker
!
enable secret 4 zKpMfAFG13ztVIBEdSXL8VdJLc/95MQnRJNxAlPSQIE
!
aaa new-model
!
aaa authentication login LOCAL_AUTH local
!
aaa session-id common
!
no ipv6 cef
ip source-route
ip cef
!
multilink bundle-name authenticated
!
crypto pki token default removal timeout 0
!
license udi pid CISCO1941/K9 sn FTX1641817V
!
username ccna password 0 ciscoccna
!
interface Loopback0
 ip address 192.168.4.1 255.255.255.0
!
interface Loopback1
 ip address 192.168.5.1 255.255.255.0
!
interface Loopback2
 ip address 192.168.6.1 255.255.255.0
!
interface Embedded-Service-Engine0/0
 no ip address
 shutdown
!
interface GigabitEthernet0/0
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface GigabitEthernet0/1
 ip address 192.168.10.1 255.255.255.0
 duplex auto
 speed auto
```

```
!
interface Serial0/1/0
 ip address 10.1.1.1 255.255.255.252
 no fair-queue
 clock rate 64000
!
interface Serial0/1/1
 ip address 172.16.100.1 255.255.255.252
 clock rate 64000
!
router rip
 version 2
 network 10.0.0.0
 network 192.168.10.0
!
ip forward-protocol nd
!
no ip http server
no ip http secure-server
!
control-plane
!
line con 0
 login authentication LOCAL_AUTH
line aux 0
line 2
 no activation-character
 no exec
 transport preferred none
 transport input all
 transport output pad telnet rlogin lapb-ta mop udptn v120 ssh
 stopbits 1
line vty 0 4
 login authentication LOCAL_AUTH
 transport input all
!
scheduler allocate 20000 1000
end
```

C.2 ROUTER 2

```
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname R2
!
boot-start-marker
boot-end-marker
!
no aaa new-model
ip subnet-zero
ip cef
ip audit po max-events 100
ipv6 unicast-routing
!
interface Ethernet0/0
 no ip address
 shutdown
 half-duplex
!
interface Serial0/0
 no ip address
 ipv6 address 2000::12:2/112
 ipv6 enable
 ipv6 rip AS1 enable
 clock rate 56000
!
interface Serial0/1
 no ip address
 ipv6 address 2000::23:1/112
 ipv6 enable
 ipv6 rip AS1 enable
 clock rate 56000
!
ip http server
no ip http secure-server
ip classless
!
ipv6 router rip AS1
!
line con 0
line aux 0
line vty 0 4
!
End
```

C.3 ROUTER 3

```
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname R3
!
boot-start-marker
boot-end-marker
!
memory-size iomem 10
no aaa new-model
ip subnet-zero
ip cef
!
ip audit po max-events 100
ipv6 unicast-routing
!!
interface Ethernet0/0
 no ip address
 shutdown
 half-duplex
!
interface Serial0/0
 no ip address
 shutdown
 no fair-queue
!
interface Serial0/1
 no ip address
 ipv6 address 2000::23:2/112
 ipv6 enable
 ipv6 rip AS1 enable
!
interface Serial0/2
 no ip address
 ipv6 address 2000::34:1/112
 ipv6 enable
 ipv6 rip AS1 enable
!
ip http server
no ip http secure-server
ip classless
!
!
ipv6 router rip AS1
!
line con 0
line aux 0
line vty 0 4
 login
!
End
```

C.4 ROUTER 4

```
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname R4
!
boot-start-marker
boot-end-marker
!
no aaa new-model
!
resource policy
!
ip subnet-zero
!
ip cef
no ip dhcp use vrf connected
!
no ip ips deny-action ips-interface
!
ipv6 unicast-routing
no ftp-server write-enable
!
no crypto isakmp ccm
!
interface Ethernet0/0
 no ip address
 half-duplex
 ipv6 address 2000::4C:1/112
 ipv6 enable
 ipv6 rip AS1 enable
!
interface Serial1/0
 no ip address
 ipv6 address 2000::34:2/112
 ipv6 enable
 ipv6 rip AS1 enable
 clockrate 56000
 no dce-terminal-timing-enable
 no fair-queue
!
interface Serial1/1
 no ip address
 shutdown
 no dce-terminal-timing-enable
!
interface Serial1/2
 no ip address
 shutdown
 no dce-terminal-timing-enable
!
interface Serial1/3
 no ip address
```

```
 shutdown
 no dce-terminal-timing-enable
!
ip http server
no ip http secure-server
ip classless
!
ipv6 router rip AS1
!
control-plane
!
line con 0
line aux 0
line vty 0 4
 login
!
!
end
```

C.5 ROUTER 5

```
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname R5
!
boot-start-marker
boot-end-marker
!
no aaa new-model
ip subnet-zero
!
ip cef
ip audit po max-events 100
ipv6 unicast-routing
no ftp-server write-enable
!
interface Ethernet0/0
 ip address 172.31.18.2 255.255.0.0
 ip nat inside
 half-duplex
 ipv6 address 2000::AA:2/112
 ipv6 rip AS1 enable
 no clns route-cache
!
interface Serial0/0
 ip address 200.200.56.1 255.255.255.0
 ip nat outside
 clockrate 56000
 no fair-queue
 no clns route-cache
!
interface TokenRing0/0
 no ip address
 shutdown
 ring-speed 16
 no clns route-cache
!
router ospf 1
 log-adjacency-changes
 network 200.200.56.0 0.0.0.255 area 0
!
ip nat pool publicasnat 199.99.9.40 199.99.9.62 netmask
255.255.255.224
ip nat inside source list 1 pool publicasnat
ip nat inside source static 172.31.18.41 199.99.9.33
ip nat inside source static 172.31.18.3 199.99.9.34
ip http server
no ip http secure-server
ip classless
```

```
ip route 0.0.0.0 0.0.0.0 200.200.56.2
!
access-list 1 permit 172.31.0.0 0.0.255.255
ipv6 router rip AS1
!
line con 0
line aux 0
line vty 0 4
 login
!
!
End
```

C.6 ROUTER 6

```
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname R6
!
boot-start-marker
boot-end-marker
!
no aaa new-model
ip subnet-zero
!
ip cef
ip audit po max-events 100
ipv6 unicast-routing
no ftp-server write-enable
!
interface Tunnel0
 no ip address
 ipv6 address 2000::56:2/112
 no clns route-cache
 tunnel source Serial0/0
 tunnel destination 200.200.56.1
 tunnel mode ipv6ip
!
interface Ethernet0/0
 ip address 200.200.6.1 255.255.255.0
 half-duplex
 no clns route-cache
!
interface Serial0/0
 ip address 200.200.56.2 255.255.255.0
 no fair-queue
 no clns route-cache
!
interface Ethernet0/1
 no ip address
 half-duplex
 ipv6 address 2000::6E:1/112
 no clns route-cache
!
router ospf 1
 log-adjacency-changes
 network 200.200.6.0 0.0.0.255 area 0
 network 200.200.56.0 0.0.0.255 area 0
!
ip http server
no ip http secure-server
ip classless
ip route 199.99.9.32 255.255.255.224 200.200.56.1
!
ipv6 route 2000::AA:0/112 2000::56:1
!
```

```
line con 0
line aux 0
line vty 0 5
 password cisco
 login
!
!
end
```

ANEXO D

HOW TO

INSTALACIÓN DEL SERVIDOR DE TELEFONÍA IP ASTERISK E IMLEMENTACIÓN DE SERVICIOS

En el caso particular se implementa el servidor de telefonía IP Asterisk en el S.O. Linux (Centos 6).

Por lo tanto los pasos para la implementación del servidor de telefonía son los siguientes:

1. Instalación del S.O. Centos 6, el mismo que se lo detalla en el Anexo A.1

2. Instalación del *Software* Asterisk sobre el sistema operativo antes mencionado, el cual se explica a detalle en el Anexo A.2

3. Una vez instalado el servidor se procede a editar los archivos de configuración para establecer los canales de comunicación (SIP, IAX2, DAHDI) y poder implementar terminales telefónicos clientes con dichos protocolos de comunicación. Los ejemplos de configuración aplicados al prototipo implementado se los encuentra en el Anexo B, en el cuál además se explica el por qué de dichas configuraciones. Cabe destacar que luego de editar cualquier archivo de configuración de Asterisk, para que surtan los efectos de dichas configuraciones, en la consola de asterisk se edita el comando:

```
cli> module reload
```

El comando anterior reinicia todos los módulos de Asterisk y aplica la configuración final.

En el caso particular del prototipo implementado para que se puedan aceptar clientes IPv6 con el protocolo de comunicación SIP se establece la configuración explicada en el apartado 3.3.2.2.1

4. Una vez establecidos los canales de comunicación es posible registrar a los clientes IPv4 con los diferentes protocolos de comunicación; a los clientes IPv6 con el protocolo de comunicación SIP pero con softphones y/o teléfonos IP con capacidad para soportar dicho protocolo de Internet (el softphone Linphone para el presente caso y el teléfono IP D-LINK DPH-150SE). La instalación de los softhones así como el registro de los mismos al servidor de telefonía IP se los explica en el Anexo A.3. Para verificar los usuarios registrados al servidor de telefonía se editan los siguientes comandos:

```
CLI> sip show peers

CLI> iax2 show peers

CLI> dahdi show peers
```

5. Para registrar clientes con tecnología DAHDI es necesario colocar las tarjetas FXO y FXS en un puerto PCI del CPU del servidor de telefonía IP y luego instalar el *software* de dichas tarjetas y el módulo dahdi en el servidor de telefonía IP. La instalación de dicho módulo se la explica en el apartado 3.4.4.8.1 y la configuración del archivo de configuración de dicho módulo se la explica en el apartado 3.4.4.8.2

6. Una vez registrados los usuarios se debe crear un plan de marcación (DIAL PLAN) el mismo que permita la comunicación interna entre los diferentes clientes, así como la comunicación desde y hacia otras redes las cuales pueden ser: redes IPv4, redes IPv6 e incluso la red de telefonía pública PSTN. Dicho plan de marcación se lo crea en el archivo de configuración *extensions.conf* el cual para una mayor facilidad de administración y una mejor organización, se lo configura de acuerdo a diferentes contextos. El ejemplo del archivo de configuración aplicado al prototipo implementado se lo explica en el Anexo B, en dicho plan de marcación se utilizan aplicaciones de Asterisk las cuales son explicadas en el apartado 3.3.2.2.2

7. Para poder utilizar el servicio de correo de voz es necesario implementar el servidor de correo en el sistema operativo, para dicho efecto se debe involucrar en el archivo de configuración de zonas ubicado en la ruta: ***/var/named/chroot/var/named/data/*** a un registro MX. Una vez establecido dicho servidor de correo es necesario editar el archivo de configuración ***voicemail.conf*** en el cual se especifica el dominio del servidor de correo electrónico así como las extensiones asociadas a una determinada cuenta de correo. La implementación del servidor DNS y del servidor de correo se encuentran especificadas en los apartados 3.4.3.1 y 3.4.3.2 respectivamente. El archivo de configuración ***voicemail.conf*** se lo presenta de una manera detallada en el anexo B. Para hacer uso del correo de voz se instaló un cliente de correo; la instalación y configuración de dicho cliente de correo se la especifica en el anexo A.4

8. Para hacer uso de servicios como transferencia de llamadas, grabación de llamadas y parqueo de llamadas se edita el archivo de configuración ***features.conf*** el cual se detalla en el anexo B; y se permite dichos servicios habilitando en la configuración de las extensiones las opciones que se detallan en los apartados 3.4.4.1 y 3.4.4.2.

9. Para poder habilitar el servicio de conferencias de voz en la central, se debe editar el archivo de configuración ***meetme.conf*** en el cual se habilitan las salas de conferencias con determinados parámetros, ya sean salas abiertas o salas con autenticación de usuarios y se lo habilita en el archivo de configuración ***extensions.conf*** haciendo uso de la aplicación meetme en las extensiones establecidas en el archivo de configuración meetme.conf. Los archivos de configuración meetme.conf y extensions.conf se los detalla en el Anexo B.

10. La implementación del IVR se la realiza en el archivo de configuración ***extensions.conf*** en el contexto llamado ***IVR*** (este contexto lo crea el

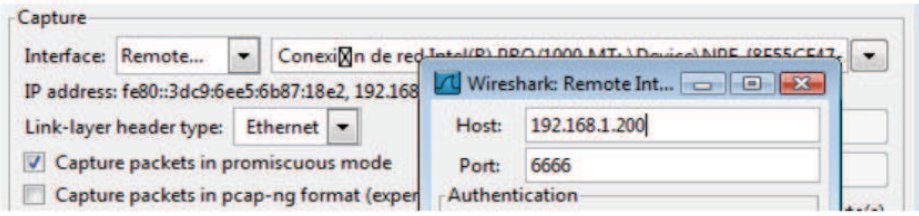
administrador) de acuerdo a configuraciones con aplicaciones y funciones propias de Asterisk que permitan establecer una interacción con el usuario en espera de respuestas (marcación de alguna extensión) para redireccionar la petición del mismo en forma de un menú de opciones. El ejemplo de la programación aplicada para la interacción con el usuario de una llamada entrante de la PSTN se la puede observar en el contexto ***IVR*** del archivo de configuración ***extensions.conf*** que se detalla en el Anexo B.

ANÁLISIS DE TRAFICO DE VOZ CON WIRESHARK

El sniffer wireshark puede emplearse para monitorear las características de retardos, jitter, ancho de banda y pérdida de paquetes en una llamada telefónica IP.

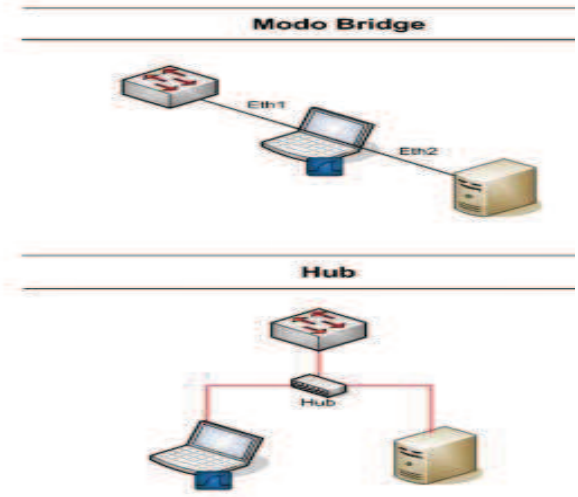
1. Es necesario definir el modo de captura que Wireshark va a utilizar:

Modo Remoto: se puede especificar la dirección IP, puertos y parámetros de autenticación para monitorear remotamente el trafico de una estación. Para ello es necesario instalar adicionalmente un sistema RPCAP (Remote Packet Capture System). El cliente tendrá que especificar dirección, puerto y la interface desde la cual se desean capturar paquetes. En Wireshark se utiliza la herramienta Capture >> Options y especificando en Interface el tipo Remote.



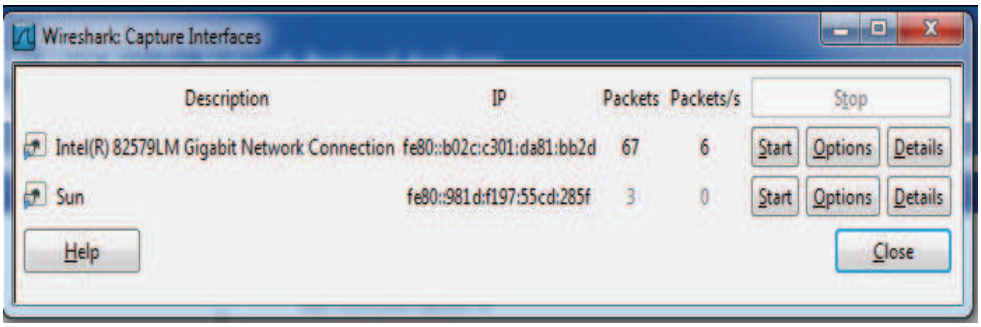
Modo Bridge: una estación con el sniffer se coloca en medio del *switch* y el servidor

Modo Hub: se conecta un hub en el mismo segmento de red que el servidor. Al tratarse de un medio compartido, todo el tráfico entre el *switch* y el servidor podrá analizarse en nuestro equipo.



2. Para iniciar la captura se selecciona la interfaz que se va a analizar.

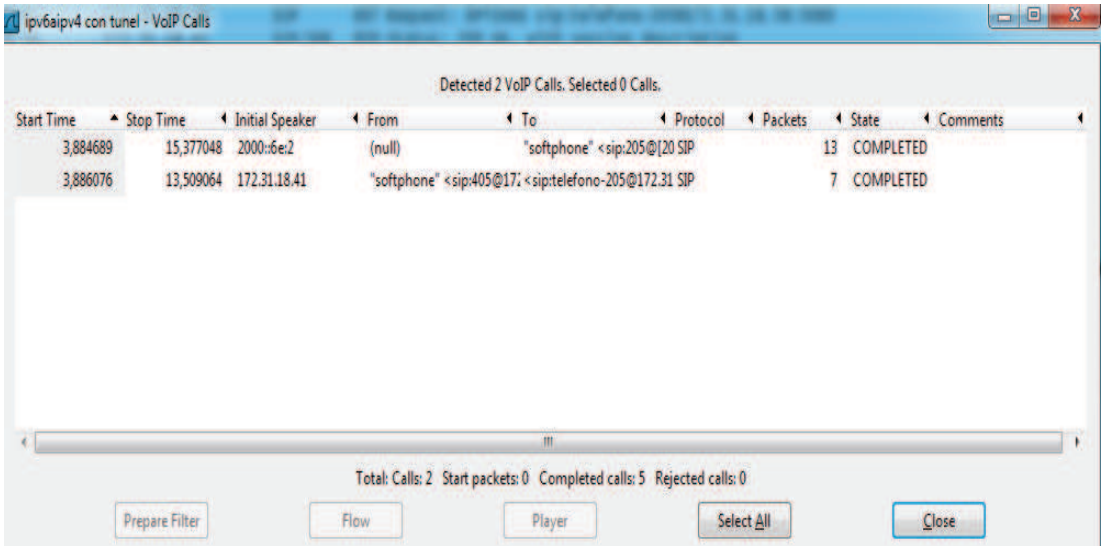
Se accede al menú Capture>>Interfaces y se selecciona la interfaz deseada de la lista de interfaces disponibles.



Una vez seleccionada la interfaz de interés se da click en Start para iniciar la captura.

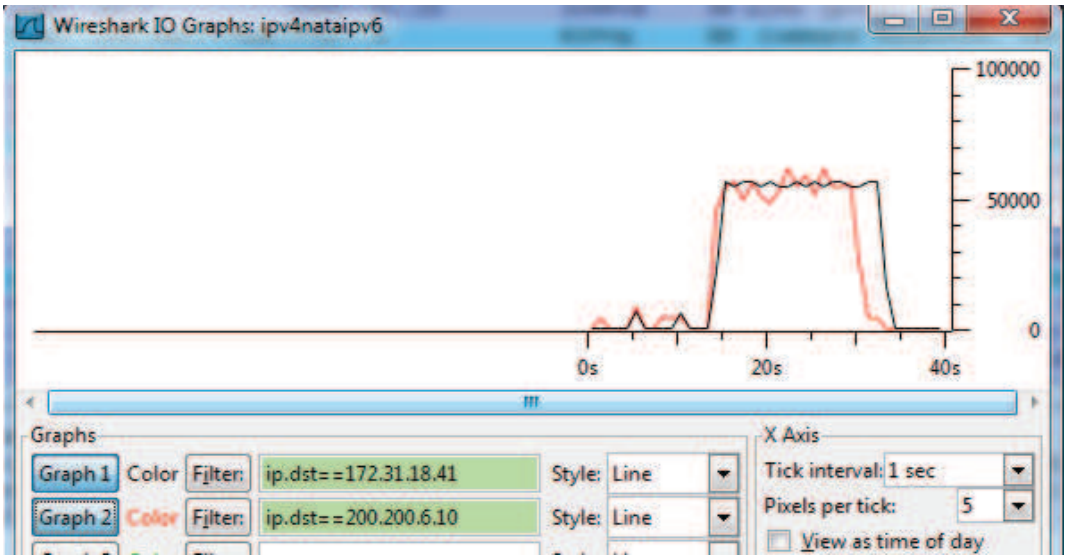
3. Cuando se finaliza la captura de tráfico se permite el acceso al menú para parámetros de telefonía IP

Para visualizar el registro de llamadas capturadas se accede al menú Telephony y se selecciona VoIP Calls.

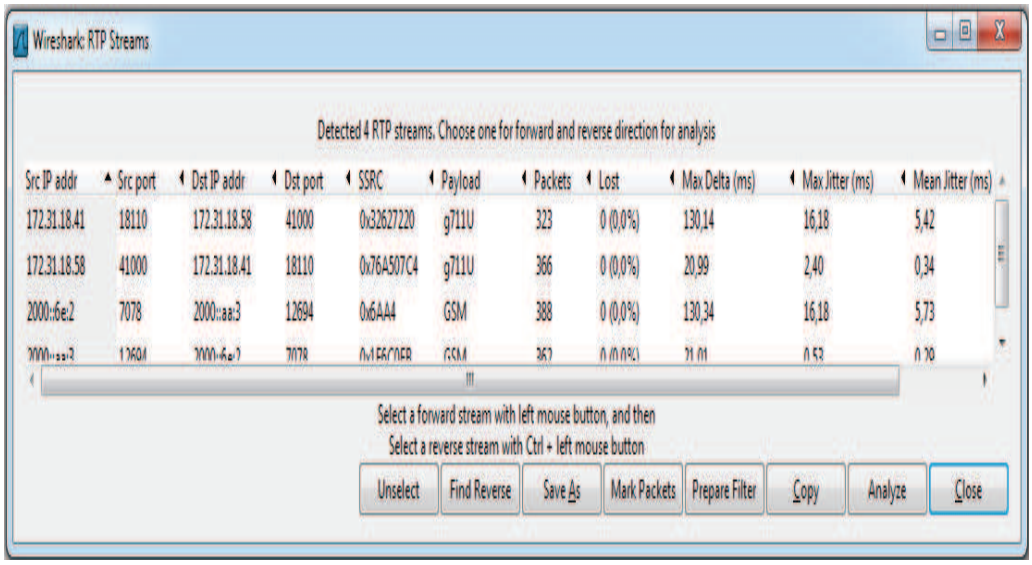


Se selecciona la llamada de interés y se da clic en la opción Flow (Graph en otras versiones) con la cual se visualiza el intercambio de paquetes para establecer la llamada (señalización) y los paquetes de VoIP (Audio RTP).

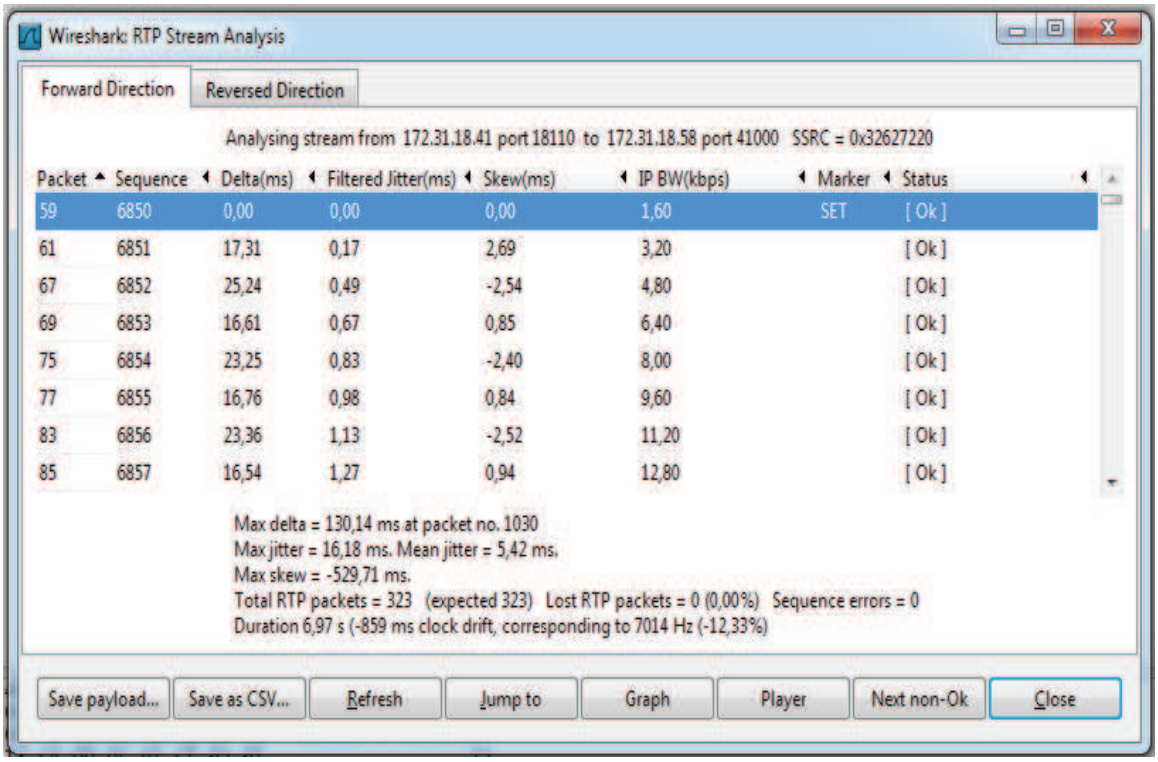
El sniffer permite además generar gráficos de estadística de cualquier parámetro, para lo cual se accede al menú Statistics y se da clic en la opción IO Graphs. Esta herramienta permite graficar por ejemplo el consumo del ancho de banda vs el tiempo; esta herramienta permite además filtrar la información seleccionada para realizar un grafico más personalizado.



Para analizar los parámetros de retardos, jitter y pérdida de paquetes se accede al menú Telephony, se da clic en RTP y se selecciona Show all *stream*s. La pantalla que aparece muestra los resultados para los diferentes *stream*s de audio:



Dentro de esta ventana se puede seleccionar un *stream* para visualizar a detalle la información de retardos, jitter e incluso se visualiza el ancho de banda (en bps) correspondiente a cada *stream*.



Es importante destacar que la medición de ancho de banda de Wireshark presenta solamente el valor de un canal de ida o de vuelta pero no el valor total. Por esta razón se debe sumar acumulativamente el valor del canal de ida y el de vuelta.