ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
ESCUELA DE INGENIERÍA ELECTRÓNICA EN TELECOMUNICACIONES Y REDES

"DISEÑO DE UNA PROPUESTA DE IMPLEMENTACIÓN DE TRÁFICO DE VOIP CON SIP EN REDES DE DATOS A TRAVÉS DE FIREWALLS Y NAT."

TESIS DE GRADO

Previa la obtención del título de

INGENIERÍA ELECTRÓNICA Y COMPUTACIÓN.

Presentado por:

KATTY ELIZABETH VINUEZA CASTILLO
DANIEL FERNANDO MORALES RUEDA

RIOBAMBA - ECUADOR
2011

Agradezco a Dios, por lo esencial que ha sido en mi firme posición de alcanzar esta meta; así como a mis padres y familia con quienes he compartido grandes alegrías y que gracias a su apoyo incondicional he alcanzado todos mis logros y anhelos ya que han sabido encaminarme en los momentos de confusión y han sido mi pilar fundamental.

Danny.

Mi sincero sentimiento de gratitud a Dios por haberme regalado el hermoso don de la vida; a mis padres y familia porque siempre me apoyaron en todo momento y supieron formar una persona llena de valores; a mis amigos por sus tantas palabras de aliento y finalmente agradezco a mis maestros por haberme brindado la oportunidad de alcanzar una excelente formación académica.

Katty.

Mi tesis te la dedico a ti Dios, por haberme dado la oportunidad de vivir y de regalarme una familia maravillosa.

A mis padres, por creer en mí y alentarme en todo momento ya que han sido una gran motivación para superarme cada día. Gracias por todos sus consejos que me han hecho crecer y madurar como ser humano.

Y finalmente, a mis amigos por ser parte de mi vida y apoyarme incondicionalmente.

Danny.

Mi tesis la dedico a Dios, por haberme permitido llegar hasta este punto y haberme dado salud para lograr mis objetivos.

A mi madre Sandra, por haberme apoyado en todo momento, por sus consejos, sus valores, por la motivación constante que me ha permitido ser una persona de bien, pero más que nada, por su amor.

A mi padre Nelson, por los ejemplos de perseverancia y constancia que lo caracterizan y que me ha infundado siempre, por el valor mostrado para salir adelante y por su amor.

Y finalmente, a mis familiares, por su gran apoyo y motivación para la culminación de mis estudios profesionales y para todos aquellos que participaron directa o indirectamente en la elaboración de esta tesis.

Katty.

NOMBRE	FIRMA	FECHA
Ing. Ivan Menes **DECANO DE LA FACULTAD** **DE INFORMÁTICA Y** **ELECTRÓNICA**	……	……
Ing. Pedro Infante **DIRECTOR DE LA** **ESCUELA DE** **TELECOMUNICACIONES** **Y REDES**	……	……
Ing. Alberto Arellano **DIRECTOR DE TESIS**	……	……
Ing. Edwin Altamirano **MIEMBRO DEL TRIBUNAL**	……	……
Lcdo.. Carlos Rodríguez **DIRECTOR DPTO.** **DOCUMENTACION**	……	……
NOTA DE LA TESIS	……	

“Yo, **Daniel Fernando Morales Rueda y Katty Elizabeth Vinueza Castillo**, somos los responsables de las ideas, doctrinas y resultados expuestos en esta Tesis de Grado, y el patrimonio intelectual de la misma pertenece a la ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO”

_____

Daniel Fernando Morales Rueda

_____

Katty Elizabeth Vinueza Castillo

ÍNDICE DE ABREVIATURAS

RTC	Red Telefónica Conmutada
IAX(2)	Protocolo de Intercambio de Asterisk (versión 2)
IETF	Grupo de Trabajo de Ingeniería de la Internet
UIT	Unión Internacional de Telecomunicaciones
NAT	Traductor de Direcciones de Red (Network Address Translator)
IP	Protocolo de Internet
LAN	Red de Área Local
WAN	Red de Área Amplia
PBX	Centralita Telefónica (Automática) Privada.
PSTN	Red de Telefonía Básica (Conmutada)
QoS	Calidad de Servicio. (Quality of Service)
RFC	Documento de Trabajo de Estandarización (Internet)
RTP	Protocolo de Tiempo Real
SMTP	Protocolo Simple de Transferencia de Correo
SIP	Protocolo de Inicialización de Sesión(es)
IAX	Protocolo de intercambio de Asterisk (Inter Asterisk Exchange)
VoIP	Voz sobre IP. Telefonía IP
DSL	Línea Digital del Suscriptor
ADSL	Línea Digital Asimétrica del Suscriptor
UA	Agente de Usuario (User Agents)
B2BUA	Back-to-Back User Agents
IMS	Sub- Sistema Multimedia IP (IP Multimedia Subsystem)

INDICE GENERAL

CAPÍTULO I MARCO REFERENCIAL

CAPÍTULO II MARCO TEÓRICO

CAPÍTULO III PROPUESTA DE DISEÑO E IMPLEMENTACIÓN

CAPÍTULO IV PRUEBAS DE DESEMPEÑO

INDICE DE FIGURAS

ÍNDICE DE TABLAS

ÍNDICE DE ANEXOS

INTRODUCCIÓN

El crecimiento de las comunicaciones IP en tiempo real están en pleno auge en Internet después del correo electrónico y la Web. SIP se ha convertido rápidamente en el protocolo estándar de señalización para este tráfico, incluyendo VoIP. Sin embargo, las comunicaciones basadas en SIP no pueden llegar a los usuarios detrás de cortafuegos y NAT. Los cortafuegos están diseñados para prevenir la entrada de comunicaciones desconocidas, mientras que NAT oculta las direcciones IP privadas de la LAN, esto ofrece la comodidad de saber que sólo los usuarios autorizados pueden tener acceso a nuestras redes y la información valiosa almacenada en nuestros servidores locales y equipos.

NAT es necesario porque las direcciones IPv4 de Internet son escasas para permitir que todos los dispositivos conectados a Internet puedan tener su propia dirección IP. Con NAT, sólo el firewall o router recibe una dirección IP pública enrutable. A cada dispositivo en el interior se le asigna una dirección IP privada que sólo es conocida dentro del espacio protegido por el firewall. Esta arquitectura evita que las comunicaciones entrantes lleguen al destinatario detrás del firewall debido a que la dirección IP del dispositivo cliente es desconocida y no enrutable.

Por último, la mayoría de los cortafuegos no son compatibles con SIP. Al igual que con todos los tipos de protocolos, el firewall debe reconocer el formato de la señalización con el fin de reconocer a la red. Como la mayoría de servidores de seguridad instalados hoy en día tampoco son compatibles con SIP, el tráfico de entrada se detendrá por esta razón.

CAPÍTULO I

MARCO REFERENCIAL

1.1 Antecedentes

El fuerte crecimiento de las redes IP por un lado y el avance en el desarrollo de técnicas de digitalización de voz, control y priorización de tráfico han provocado en los último tiempos una aceleración en la adopción de la telefonía IP por empresas, instituciones y usuarios domésticos.

Las tecnologías de voz sobre IP permiten el envío de la voz codificada mediante paquetes a través de una red IP. Esto permite integrar las redes de datos y de voz en una única red convergente, dando lugar a una reducción de los costes de implantación y mantenimiento de las redes de comunicación al disminuir la complejidad de las mismas.

Las comunicaciones internas de la empresa, incluso entre diferentes sucursales dispersas geográficamente, o con otras empresas que también hayan adoptado la tecnología VoIP se pueden realizar mediante terminales VoIP. De esta forma se consigue una disminución de gastos derivados de la utilización de la conexión de datos contratada por la empresa para el curso de las conexiones telefónicas sin importar la distancia de las llamadas y, en el caso de tratarse de conexiones de datos con tarifa plana (caso más habitual), sin importar la duración de las llamadas.

Para cursar las llamadas entre la Red de Telefonía Pública y el mundo VoIP es necesario disponer de una centralita hardware o software que actué como gateway entre los dos mundos. Este servicio podría ser ofrecido y gestionado por la misma empresa o delegado a un proveedor de servicios VoIP.

Además de las ventajas puramente económicas, las soluciones VoIP presentan otro conjunto de ventajas técnicas como son la mejora de la calidad de las comunicaciones telefónicas derivadas de los diferentes códec para voz existentes hoy en día, la mayor facilidad para conseguir un servicio de voz más fiable que la telefonía convencional, hacen que la telefonía IP sea una solución cada vez más empleada para los problemas de comunicación de empresas e instituciones.

Actualmente, existen dos protocolos principales orientados a la telefonía IP. El primero en aparecer fue el conjunto de protocolos H.323 estandarizados por el ITU y que supuso una primera aproximación a la telefonía IP. En la actualidad este protocolo está siendo superado en popularidad por SIP. Se trata de un protocolo de nivel de aplicación desarrollado por el IETF que ha sido diseñado con la idea de una fácil implementación, buena escalabilidad y flexibilidad.

Hoy en día existen nuevas modalidades de comunicación aparte de las llamadas telefónicas, como son las multiconferencias, la videoconferencia, el correo electrónico, las comunicaciones móviles, la mensajería instantánea y otros servicios. Se persigue una convergencia de las comunicaciones en las que la voz, el video y los datos se transportan a través de IP, proporcionando nuevas formas de conexión, comunicación y colaboración. Al tener voz y datos integrados en una sola estructura de red, resulta más sencillo su mantenimiento y gestión, permitiendo un ahorro de costes a las compañías. Es entonces cuando surge la VoIP y como aplicación de esta tecnología, aparece la Telefonía IP, que permite la realización de llamadas telefónicas sobre redes IP u otras redes de paquetes utilizando PCs, Gateways y teléfonos IP.

El desarrollo de VoIP junto al protocolo SIP hoy en día está fuertemente ligado a Asterisk que es un programa de software libre (bajo licencia GPL) que proporciona funcionalidades de una central telefónica (PBX), Asterisk puede funcionar en muchos sistemas operativos, GNU/Linux es la plataforma más estable y en la que existe un mayor soporte.. Asterisk hace VoIP en el protocolo SIP y puede interoperar con equipos de telefonía estándar básicas usando un hardware relativamente sin costo.

El Asterisk basado en soluciones de telefonía ofrece un variado y flexible set de características y funcionalidades básicas PBX así como también ínter opera con sistemas básicos de telefonía estándar y sistemas VoIP. Asterisk ofrece voicemail, conferencias, llamadas en espera, grabado de llamadas lo cual permitirá abaratar costos debido a su instalación y a la gran prestación de servicios sin la necesidad de la Red Telefónica Pública.

1.2 Justificación

A partir de algunos años se ha dado un auge en la aparición de empresas que ofrecen VOIP la misma que reducirá los gastos que tienen las empresas o negocios por este concepto. El hecho de necesitar una conexión de DSL (Línea Digital del Suscriptor) o más bien de que se nos dé una conexión ADSL (Línea Digital Asimétrica del Suscriptor) la cual no garantiza ancho de banda y políticas proteccionistas de los proveedores hacen que este servicio no sea efectivo en su totalidad para el uso de telefonía por Internet. Pero obsérvese el caso de que las empresas grandes ya usan esta tecnología y no solo para hacer llamadas locales o de larga distancia, sino que están utilizando esta tecnología para comunicar las distintas áreas de su empresa y sus sucursales.

Para la pequeña y mediana empresa la implementación de VoIP no es una decisión fácil. Por suerte, la instalación, puede causar ahorros sustanciales y añadir nuevas capacidades a viejos sistemas telefónicos de oficina, esto se puede realizar en pequeños pasos. Para la VoIP se utiliza SIP que es un protocolo de nivel de aplicación desarrollado por el IETF que ha sido diseñado con la idea de una fácil implementación, buena escalabilidad y flexibilidad, pero el problema radica cuando la comunicación VoIP se realiza a través de Internet para enlazarse con otras sucursales o estaciones de trabajo, debido al hecho de que existen firewall y los sistemas de traducción de direcciones (NAT).

A su vez, existen problemas derivados de la propia naturaleza de las redes IP, como la falta de QoS que garantiza la calidad de las comunicaciones en tiempo real sin la necesidad de un sobredimensionamiento en los recursos de red y equipos conmutadores.

Los principales problemas que se presentan con la utilización del protocolo SIP son los siguientes:

- **Problemática del firewall:** Los cortafuegos son un serio problema a la hora de que el servicio VoIP funcione correctamente. La señalización SIP utiliza el puerto 5060 UDP, por lo que es necesario que los cortafuegos permitan el tráfico con destino al puerto 5060.

  Por otro lado, el flujo de datos con el protocolo RTP resulta más problemático, ya que los puertos se escogen dinámicamente al establecerse una nueva conexión. Esto supone un serio problema para empresas que lleven a cabo políticas restrictivas de filtrado. Una forma de evitar estos problemas con el flujo RTP consiste en no filtrar los puertos por encima del 1024, ya que RTP selecciona puertos por encima de ese número. Esto en muchos casos puede resultar una solución inaceptable desde el punto de vista de seguridad de una red privada.

  Una opción más interesante puede ser la utilización de Gateway de aplicación bien en el propio firewall o distribuidos. Se tratan de firewall dinámicos que permiten abrir y cerrar puertos en tiempo real. Un firewall con soporte del protocolo SIP procesa los paquetes SIP que reenvía pudiendo de esta forma abrir y cerrar dinámicamente los puertos RTP necesarios.

  Esta opción obliga a utilizar firewall de gama alta y con un rendimiento inferior ya que el firewall además de procesar la cabecera IP debería de procesar la cabecera SIP. Finalmente una opción más sencilla puede ser el uso de un proxy RTP, ya que todas las comunicaciones RTP deben pasar a través de él. Sin embargo, tampoco se trata de una solución óptima ya

que introduce retardo en la comunicación y requiere una máquina con recursos elevados especialmente de conectividad.

- **Problemática del NAT:** Al igual que la problemática con los cortafuegos la problemática del NAT en SIP se puede separar en dos bloques uno relativo al flujo de señalización SIP y otro relativo al flujo de datos RTP.

  El problema de la señalización SIP consiste en que la información de dirección IP y puerto obtenido de la cabecera SIP corresponderá con la dirección IP y puerto que el terminal tiene asignado dentro de la red nateada y por ello el terminal no podría recibir peticiones SIP desde el exterior. Para solventar el problema se utilizan dos mecanismos conocidos como *Symmetric Response* y *Connection Reuse*.

  Estos mecanismos se basan en el concepto *Symmetric Signalling*, por el cual un terminal envía las peticiones y recibe las respuestas a través del mismo puerto, e introducen campos adicionales en la cabecera SIP. Estos nuevos campos detallan información relativa al sistema de traducción de direcciones de forma que cuando el terminal nateado habría un agujero en el NAT se aprovechará posteriormente ese mismo agujero.

  Por otro lado, en lo que concierne al tráfico RTP, la negociación RTP se lleva a cabo mediante el protocolo de descripción de sesión SDP dentro de un intercambio petición/respuesta SIP. Durante este intercambio, cada cliente situado en su propia red especifica la dirección IP y puerto para la recepción del flujo de información durante la sesión.

El problema surge debido a que la dirección IP indicada por cada cliente corresponde con la dirección de la red privada a la que pertenece, la cual no es accesible desde fuera al otro lado del NAT correspondiente. Es decir, el problema surge cuando el tráfico RTP intenta atravesar el NAT opuesto.

La solución es el protocolo de nueva generación IPv6, que otorga mayor cantidad de direcciones, pero aún deberá pasar mucho tiempo para que se realicen las modificaciones necesarias en las redes de manera que sean compatibles con esta norma. Mientras tanto NAT es una alternativa previa al uso de IPv6, pero la utilización de NAT genera que muchas aplicaciones que trabajan bajo redes P2P no funcionen adecuadamente. Esto debido a que un NAT implica, en cierta medida, colocar un Firewall que protege a la red local, ya que al no saber los equipos externos la dirección privada de los sistemas internos, no solicita la conexión. Todo enlace hacia Internet deberá generarse bajo petición de los equipos locales.

En videoconferencia, VoIP por ej. detrás de un NAT es el equipo local el que debe iniciar la llamada, cuando el sistema remoto contesta, es el equipo NAT quien le responde, y al no tener capacidades de negociación (SIP para este caso), la llamada termina.

1.3 Planteamiento y delimitación del problema:

El surgimiento de soluciones de telefonía totalmente basada en IP y la falta de opciones, completas y/o parciales, de bajo costo para las pequeñas y medianas empresas, así como la falta de difusión de esta tecnología, son la gran barrera para el uso de VoIP.

Si se pudiera implementar una solución de bajo costo de VoIP para la pequeña y mediana empresa utilizando los activos que ésta tiene en desuso, la infraestructura de red y telefonía instalada, y la inversión en equipo de bajo costo en el mercado, los puntos importantes a tener en cuenta son:

- Instalar un sistema de VoIP que esté a un precio accesible para la micro y pequeña empresa.
- Determinar qué tan viable es poder instalar este sistema en las redes locales de estos lugares.
- Definir cuáles son los elementos básicos y dispositivos en los que hay que invertir para poder hacer operable el sistema.

En otras palabras; instalar un conmutador VoIP utilizando una computadora en desuso, instalarle software libre para uso de conmutador telefónico, con dispositivos de entrada y salida para VoIP, de bajo costo de inversión, y utilizar la infraestructura de red ya instalada en el lugar para hacer llegar el servicio a los puntos requeridos.

1.4 Objetivos.

1.4.1 Objetivos Generales

- Realizar el estudio de la tecnología de Gateway SIP aplicado a la transmisión de tráfico de Voz sobre IP a través de firewalls.

1.4.2 Objetivos Específicos

- Analizar los problemas presentados por el protocolo SIP en la transmisión de VoIP a través de firewalls.
- Estudiar las técnicas de Gateways SIP.
- Implementar el ambiente de pruebas y evaluar los resultados.
- Diseñar una propuesta de implementación de Solución a los problemas presentados en la transmisión de VOIP a través de firewalls.

1.5 Hipótesis

Las técnicas de Gateways SIP aplicadas a la transmisión de tráfico de Voz IP a través de firewalls permitirán mejorar la disponibilidad de este servicio.

CAPÍTULO II

FUNDAMENTO TEÓRICO.

2.1. Introducción

La voz sobre IP o VoIP consiste en transmitir voz sobre protocolo IP. Dicho así puede sonar simple pero las redes IP fueron diseñadas principalmente para datos y muchas de las ventajas de las redes IP para los datos resultan ser una desventaja para la voz pues ésta es muy sensible a retardos y problemas de transmisión por muy pequeños que estos sean.

Por tanto, transmitir voz sobre el protocolo IP tiene muchos problemas técnicos que resolver. Por suerte la tecnología ha evolucionado y ha resultado en que podamos abstraernos en gran medida de aquellos problemas inherentes a las redes IP que perjudican la calidad de voz.

Solo hace pocos años haciendo uso de llamadas por Internet se puedo decir que la mejora de unos 10 años para acá ha sido notable. Ahora podemos decir que la transmisión de voz por Internet ya es una alternativa rentable al alcance de la mayoría de nosotros.

2.2. Funcionalidades de la VoIP

VoIP puede facilitar tareas que serían más difíciles de realizar usando las redes telefónicas comunes:

- Las llamadas telefónicas locales pueden ser automáticamente enrutadas a un teléfono VoIP, sin importar dónde se esté conectado a la red. Uno podría llevar consigo un teléfono VoIP en un viaje, y en cualquier sitio conectado a Internet, se podría recibir llamadas.

- Números telefónicos gratuitos para usar con VoIP están disponibles en Estados Unidos de América, Reino Unido y otros países de organizaciones como Usuario VoIP.

- Los agentes de Call Center usando teléfonos VoIP pueden trabajar en cualquier lugar con conexión a Internet lo suficientemente rápida.

- Algunos paquetes de VoIP incluyen los servicios extra por los que PSTN (Red Pública Telefónica Conmutada) normalmente cobra un cargo extra, o que no se encuentran disponibles en algunos países, como son las llamadas de 3 a la vez, retorno de llamada, remarcación automática, o identificación de llamada.

2.3. Protocolos de VoIP

Hay muchos protocolos involucrados en la transmisión de voz sobre IP (ver figura II.1). Ya de por sí hay protocolos de red involucrados como el propio protocolo IP y otros protocolos de transporte como TCP o UDP.

Encima de ellos se colocan los protocolos de señalización de voz y como si esto fuera poco existen además muchas opciones de protocolos de señalización disponibles lo que puede hacer que todo suene un poco confuso al principio.

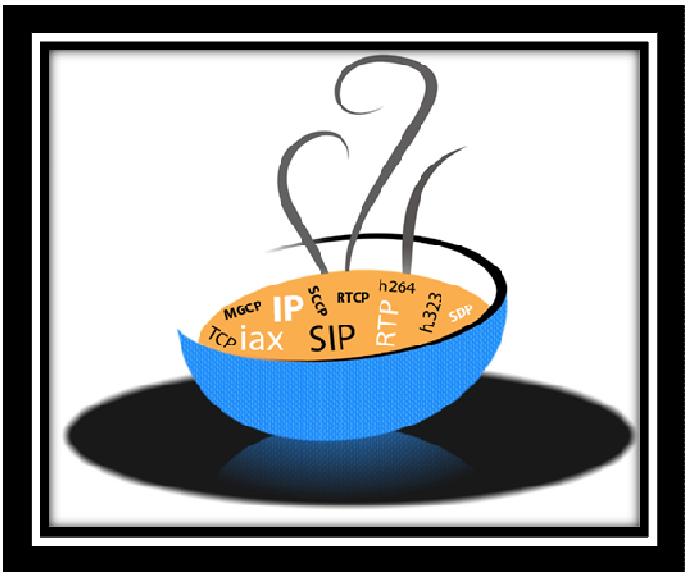


Figura II.1: Protocolos de VoIP.

2.4. Clasificación de los protocolos VoIP

Para simplificar las cosas podríamos clasificar a los protocolos utilizados en VoIP en tres grupos:

2.4.1. Protocolos de Señalización

La tarea de los protocolos de señalización en redes VoIP es la de establecer, mantener y terminar correctamente una sesión entre dos o más dispositivos. Las tareas de control implican el monitoreo, supervisión, modificación y el registro de los eventos de la sesión, entre otras, de manera que la labor no es trivial.

Para el usuario común, levantar el auricular, escuchar el tono, marcar el número deseado, y escuchar el progreso de su llamada hasta que sea contestada es una actividad de lo más natural. El no sabe, ni tiene porqué saber, todos los procesos que se desencadenan detrás de tal evento. Eso es la señalización, el conjunto de procesos y eventos necesarios para establecer la comunicación entre dos o más participantes.

Tradicionalmente, los sistemas de telefonía convencional utilizan esquemas de señalización tanto analógicos como digitales. Hoy en día existen diversos protocolos para VoIP, algunos de ellos propietarios y otros estándares. En el caso de los primeros, cada fabricante intenta que sea el suyo el que predomine en el mercado por las ventajas comerciales que ello implica. Son entonces, los organismos reguladores del sector los encargados de establecer los estándares que garanticen la interoperabilidad entre todos, sin limitar el avance de la industria.

Más importante que saber cuántos y cuáles protocolos hay, es conocer cuál es su importancia y saber diferenciarlos para elegir el que corresponda mejor a nuestras necesidades y circunstancias.

Existen algunos protocolos de señalización, que han sido desarrollados por diferentes fabricantes u organismos como la ITU o el IETF, y que se encuentran soportados por Asterisk. Algunos son:

- SIP
- IAX
- H.323
- MGCP
- SCCP

Entre estos los más populares en el ámbito de Asterisk son SIP e IAX.

2.4.2. Protocolos de transporte de Voz

No se debe confundir aquí con protocolos de transporte de bajo nivel como TCP y UDP. Nos referimos aquí al protocolo que transporta la voz propiamente dicha o lo que comúnmente se denomina carga útil. Este protocolo se llama RTP (Real-time Transport Protocol) y su función es simple: transportar la voz con el menor retraso posible. Este protocolo entra a funcionar una vez que el protocolo de señalización ha establecido la llamada entre los participantes.

2.4.3. Protocolos de plataforma IP

En esta categoría agruparemos a los protocolos básicos en redes IP y que forman la base sobre la cual se añaden los protocolos de voz anteriores (ver figura II.2). En estos protocolos podríamos mencionar a Ethernet, IP, TCP y UDP.

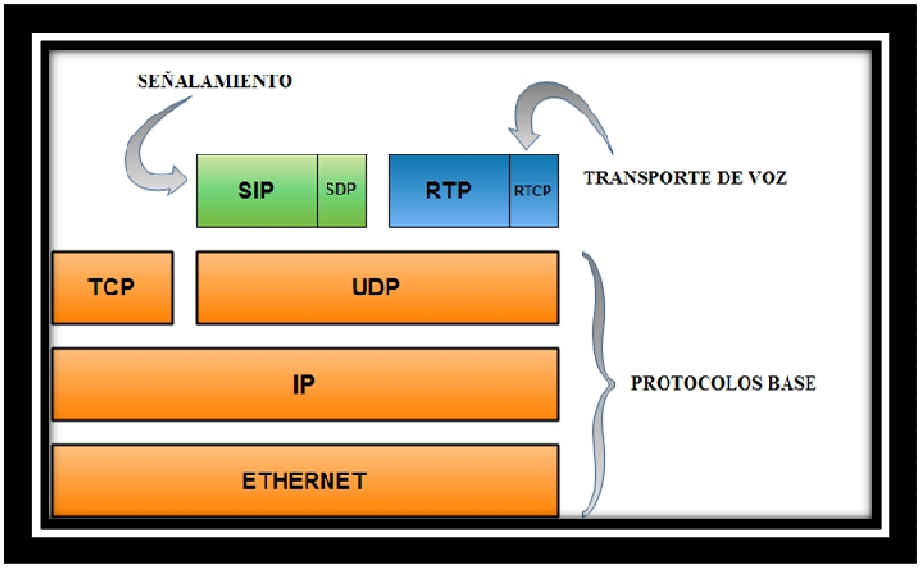


Figura II.2. Protocolos en una llamada SIP.

En la figura II.2 podemos observar un hecho curioso y es que pese a que SIP soporta tanto UDP como TCP sólo lo vemos posado sobre UDP. No se trata de un error sino más bien que en Asterisk la implementación de SIP solo está disponible para UDP.

2.5. Porque la Señalización

¿Cuál es la necesidad de emplear protocolos de señalización en una red de telefonía IP? Aunque inocente, esta pregunta sirve como preámbulo para describir cuán importante es la labor que desempeña la señalización en una llamada telefónica por Internet.

En primer lugar, en telefonía convencional, los clientes estaban asociados a un circuito y su localización física era fija. En VoIP los clientes pueden ser hardphones o softphones asociados a una dirección de red que puede cambiar dependiendo de la localización del dispositivo, es decir, el cliente puede o no ser fijo. Lo anterior

deja de manifiesto que la localización de los clientes es un punto que la señalización debe resolver.

En segundo lugar, para poder ser localizado, un usuario VoIP requiere, además del cliente, de una cuenta asociada que lo identifique en el sistema. El proceso de registro del cliente ante el sistema usando un nombre de usuario y contraseña o cualquier otro método también cae en la lista de capacidades que debe tener un protocolo de señalización. Una vez registrado y localizado el cliente VoIP, estaremos en condiciones de utilizar los recursos del sistema para realizar llamadas, conferencias, consultar el buzón de voz, etc. Es justamente la administración de los recursos lo que nos lleva a otro punto de comparación entre los diferentes protocolos.

La administración de los recursos es un punto clave que diferencia a los protocolos de señalización existentes. Existen protocolos con control de llamada distribuido, es decir, permiten que los dispositivos que conforman la red de VoIP tomen sus propias decisiones respecto al establecimiento de la llamada, sin consultar a una entidad central. Por otro lado, existen protocolos que obligan a todos los dispositivos de su red a consultar primero con una entidad central antes de tomar una decisión, MGCP es el más claro ejemplo de este tipo de esquemas. Otros como SIP y H.323, pueden trabajar en ambos esquemas ya que su arquitectura así se los permite.

Otro punto a considerar es el soporte de aplicaciones multimedia. Protocolos como H.323 fueron diseñados pensando en un entorno totalmente multimedia, con capacidad de transmisión de texto, voz, video y datos en multiconferencias, de ahí

su complejidad. Otros como SIP e IAX son menos complejos que H.323, mas, no por ello superiores en desempeño.

2.6. SIP

SIP (Protocolo de Inicialización de Sesión), es un protocolo de señalización definido por el “Internet Engineering Task Force” o IETF que permite el establecimiento, la liberación y la modificación de sesiones multimedia (RFC3261).

Este protocolo hereda ciertas funcionalidades de los protocolos “Hyper Text Transport Protocol” o “http”, utilizado para navegar sobre la WEB y “Simple Mail Transport Protocol” o “SMTP”, utilizado para transmitir mensajes electrónicos (e-mails). SIP se apoya sobre un modelo transaccional cliente / servidor como http. El direccionamiento utiliza el concepto “Uniform Resource Locator” o “URL SIP” parecido a una dirección e-mail. Cada participante en una red SIP es entonces alcanzable vía una dirección, por medio de una URL SIP. Por otra parte, los requerimientos SIP son satisfechos por respuestas identificadas por un código digital. De hecho, la mayor parte de los códigos de respuesta SIP han sido tomados del protocolo http. Por ejemplo, cuando el destinatario no está ubicado, un código de respuesta «404 Not Found» es devuelto.

Un requerimiento SIP está constituido de “headers” o encabezamientos, al igual que un comando SMTP. Por fin, SIP, al igual de SMPT es un protocolo textual. SIP ha sido extendido con el fin de soportar numerosos servicios tales como la presencia, la mensajería instantánea (similar al servicio SMS en las redes

móviles), la transferencia de llamada, la conferencia, los servicios complementarios de telefonía, etc…

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

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

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

2.6.1 Entidades SIP

SIP define dos tipos de entidades: los clientes y los servidores. De manera más precisa, las entidades definidas por SIP son (ver figura II.3):

- *El Servidor Proxy:* El recibe solicitudes de clientes que el mismo trata o encamina hacia otros servidores después de haber eventualmente, realizado ciertas modificaciones sobre estas solicitudes.

- *El Servidor de Redireccionamiento:* Se trata de un servidor quien acepta solicitudes SIP, traduce la dirección SIP de destino en una o varias direcciones de red y las devuelve al cliente. De manera contraria al Servidor Proxy el Servidor de Redireccionamiento no encamina las solicitudes SIP. En el caso de la devolución de una llamada, el Servidor Proxy tiene la capacidad de traducir el número del destinatario en el mensaje SIP recibido, en un número de reenvio de llamada y encaminar la llamada a éste nuevo destino, y eso de manera transparente para el cliente de origen; para el mismo servicio, el Servidor de Redireccionamiento devuelve el nuevo número (numero de reenvio) al cliente de origen quien se encarga de establecer una llamada hacia este nuevo destino.

- *El Agente Usuario o "UA":* Se trata de una aplicación sobre un equipo de usuario que emite y recibe solicitudes SIP. Se materializa por un software instalado sobre un « Equipo de Usuario » o UE: una PC, un teléfono IP o una estación móvil UMTS.

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

"Registrar"actualiza entonces una base de datos de localización. El registrador es una función asociada a un Servidor Proxy o a un Servidor de Redireccionamiento. Un mismo usuario puede registrarse sobre distintos UAs SIP, en este caso, la llamada le será entregada sobre el conjunto de estas UAs.

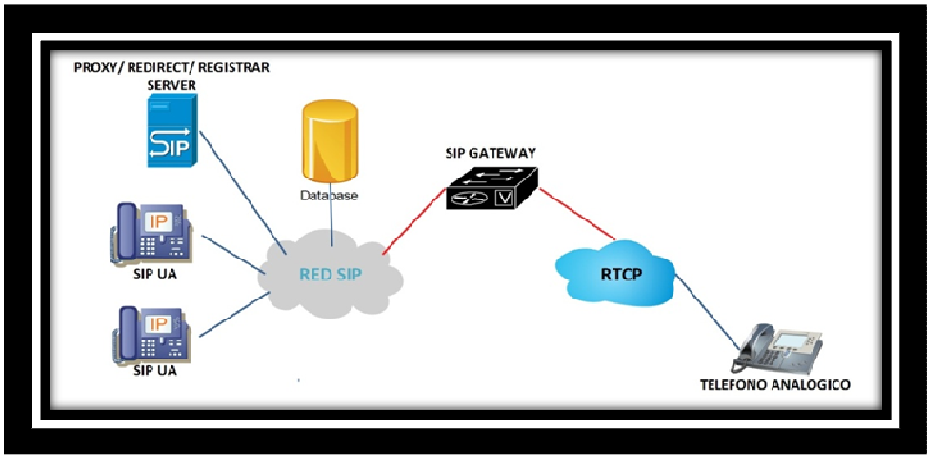


Figura II.3. Entidades de una red SIP

2.6.2. Métodos y Respuestas SIP

2.6.2.1. Métodos SIP

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

- El método "*INVITE*" es usado con el fin de establecer una sesión entre UAs. INVITE corresponde al mensaje ISUP IAM o al mensaje Q.931 SET UP y contiene las informaciones sobre el que genera la llamada y el destinatario así como sobre el tipo de flujos que serán intercambiados (voz, video,...). Cuando un UA que emitió el método SIP INVITE recibe una respuesta final a la invitación (ejemplo: 200 OK), el confirma la recepción de ésta respuesta por medio de un método "ACK". Una respuesta del tipo "busy" o "answer" es considerada como final mientras

que una respuesta del tipo “ringing” significa que el destinatario ha sido avisado o es una respuesta provisoria.

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

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

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

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

2.6.2.2. Respuestas SIP

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

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

2.6.3. Funcionamiento del protocolo SIP

2.6.3.1 Inscripción a la red SIP

El método “REGISTER” es utilizado por un “UA” con el fin de indicar a la función Registrar (físicamente implantada en un Servidor Proxy o un Servidor de Redireccionamiento) la correspondencia entre su dirección SIP (ejemplo: katty.vinueza@electronica.com) y su dirección IP (ejemplo: katty.vinueza@192.168.1.20). La dirección IP puede ser estática u obtenida de modo dinámico por DHCP. La función Registrar actualiza entonces una base de datos de localización.

Desde este momento, el UA puede recibir llamadas ya que se encuentra ubicado. Si un usuario SIP desea reenviar sus llamadas de su dominio corriente hacia otro dominio, (ejemplo: del dominio electronica.com al dominio elastix.com), solo tendrá que indicar a la función Registrar de electronica.com su dirección SIP en el dominio elastix.com. Cuando un mensaje INVITE debe ser entregado por el Servidor Proxy del dominio orange.com a sip: katty.vinueza@electronica.com, la base de datos actualizada por la función Registrar indica al Servidor Proxy que el mensaje tiene que ser relevado a sip: katty.vinueza@elastix.com. Entonces, el Servidor Proxy efectúa una búsqueda por el DNS de la dirección IP del Servidor Proxy del dominio elastix.com, con el fin de relevar el mensaje SIP y encaminar al destino apropiado (sip: katty.vinueza@elastix.com).

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

2.6.3.2. Establecimiento y liberación de sesión SIP

En el ejemplo siguiente (ver figura II.4), el que llama tiene como URL SIP sip: daniel.morales@electronica.com, mientras la URL SIP del destinatario de la llamada es sip: katty.vinueza@electronica.com.

Un mensaje de establecimiento de llamada SIP INVITE es emitido por parte del UA SIP del que llama al Servidor Proxy. Este último interroga la base de datos de localización para identificar la localización del que está llamado (dirección IP) y encamina la llamada a su destino. El mensaje INVITE contiene distintos “headers” o encabezamientos obligatorios, entre los cuales tenemos: la dirección SIP de la persona que llama “From”, la dirección SIP de la persona que recibe la llamada “To”, una identificación de la llamada “Call-ID”, un número de secuencia “Cseq”, un número máximo de saltos “max-forwards”. El encabezamiento “Via” está actualizado por todas las entidades que participaron en el enrutamiento del requerimiento INVITE. Eso asegura que la respuesta seguirá el mismo camino que el requerimiento.

Por otra parte, el requerimiento SIP INVITE contiene una sintaxis “Session Description Protocol” o SDP. Esta estructura consiste en varias líneas que describen las características que “Daniel” necesita para la llamada.

Daniel Morales indica que la descripción SDP utiliza la versión 0 del protocolo, que se trata de una sesión telefónica (m = audio), que la voz constituida en paquetes le debe ser entregada a la dirección de transporte (puerto UDP = 45450, dirección IP =192.168.1.20) con el protocolo RTP y utilizando un formato de codificación definido en el RFC “Audio Video Profile” o AVP, pudiendo ser G. 711 law o G.728.

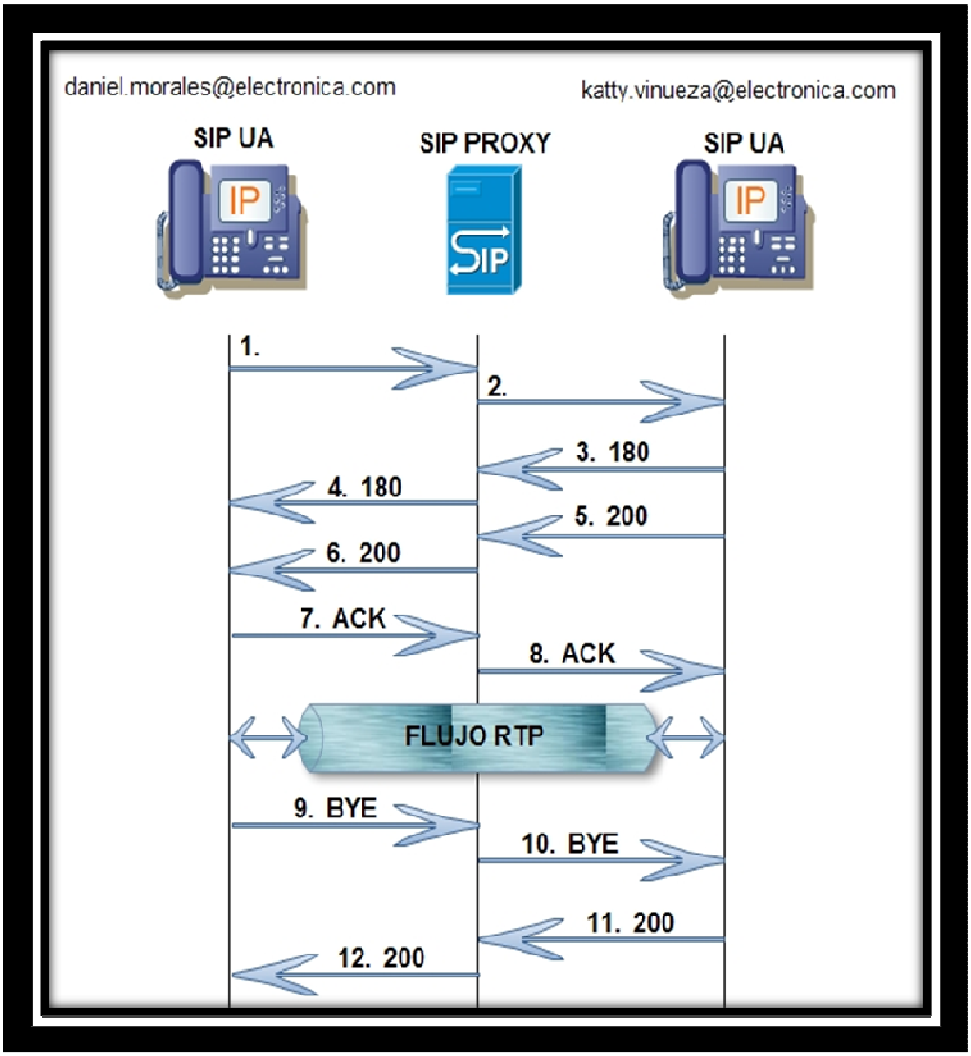


Figura II.4. Establecimiento y liberación de sesión SIP

INVITE sip: katty.vinueza@electronica.com SIP/2.0

Via: SIP/2.0/UDP station1.electronica.com:5060

Max-Forwards: 20

To: Daniel Morales <sip:daniel.morales@electronica.com>

From: Katty Vinueza <sip: katty.vinueza@electronica.com >

Call-Id: 23456789@station1.electronica.com

CSeq: 1 INVITE

Contact: Daniel.morales@192.168.1.20

Content-Type: application/sdp

Content-Length:162

v = 0

c = IN IP4 192.168.1.20

m = audio 45450 RTP/AVP 0 15

La respuesta 180 RINGING es devuelta por el destinatario a la UA del que genera la llamada. Cuando el destinatario acepta la sesión, la respuesta 200 OK esta emitida por su UA y encaminada hacia la UA del que genera la llamada.

SIP/2.0 200 OK

Via: SIP/2.0/UDP ps1.electronica.com:5060

Via: SIP/2.0/UDP station1.electronica.com:5060

Max-Forwards : 20

To: Katty Vinueza <sip: katty.vinueza@electronica.com >

From: Daniel Morales <sip: daniel.morales@electronica.com >

Call-Id: 23456789@station1.electronica.com

CSeq: 1 INVITE

Contact: Katty.vinueza@192.168.1.27

Content-Type: application/sdp

Content-Length:162

v = 0

c = IN IP4 192.168.1.27

m = audio 22220 RTP/AVP 0

La UA del que genera la llamada devuelve un método ACK al destinatario, relevada por la entidad llamada Servidor Proxy. La entidad Servidor Proxy participa en el encaminamiento de la señalización entre UAs mientras que las UAs establecen directamente canales RTP para el transporte de la voz o del video en forma de paquetes sin implicación del Servidor Proxy en este transporte.

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

BYE sip: katty.vinueza@electronica.com SIP/2.0

Via : SIP/2.0/UDP station1.electronica.com:5060

Max-Forwards : 20

To : Katty Vinueza <sip: katty.vinueza@electronica.com >

From : Daniel Morales <sip: daniel.morales@electronica.com >

Call-Id: 23456789@station1.electronica.com

Copyright EFORT 2005 7

CSeq: 2 BYE

SIP/2.0 200 OK

Via : SIP/2.0/UDP ps1.electronica.com:5060

Via : SIP/2.0/UDP station1.electronica.com:5060

Max-Forwards : 20

To : Katty Vinueza <sip: katty.vinueza@electronica.com >

From : Daniel Morales <sip: daniel.morales@electronica.com >

Call-Id: 23456789@station1.electronica.com

CSeq: 2 BYE

2.6.4. Extensiones del protocolo SIP

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

El método REFER (RFC3515) reenvía el receptor hacia un recurso identificado en el método. REFER permite emular distintos servicios o aplicaciones incluyendo la transferencia de llamada. Contemplamos T1, la entidad que origino la transferencia, T2 la entidad transferida y T3, el destinatario de la transferencia. La transferencia de llamada permite a T1 transformar una llamada en curso entre T1 y T2 en una nueva llamada entre T2 y T3, elegida por T1. Si la transferencia de llamada se lleva a cabo, T2 y T3 podrán comunicarse mientras que T1 no podrá seguir dialogando con T2 o T3.

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

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

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

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

2.6.5 Interfuncionamiento entre SIP y RTC

Para el interfuncionamiento entre la Red Telefónica Conmutada RTC y SIP, es necesario introducir una pasarela o Gateway RTC/SIP que se comunique por una parte al RTC y por otra parte a una red SIP. Este Gateway cumple con dos funciones:

- Traducción de la señalización ISDN User Part o ISUP en señalización SIP y recíprocamente.
- Conversión de señales de audio en paquetes RTP y recíprocamente; en efecto, este Gateway establece canales lógicos RTP con la terminal SIP y establece circuitos de palabras con un switch o conmutador Class 4 o

Class 5. El Switch Clase 5 representa un conmutador telefónico de acceso mientras el Switch Clase 4 es un conmutador telefónico de tránsito.

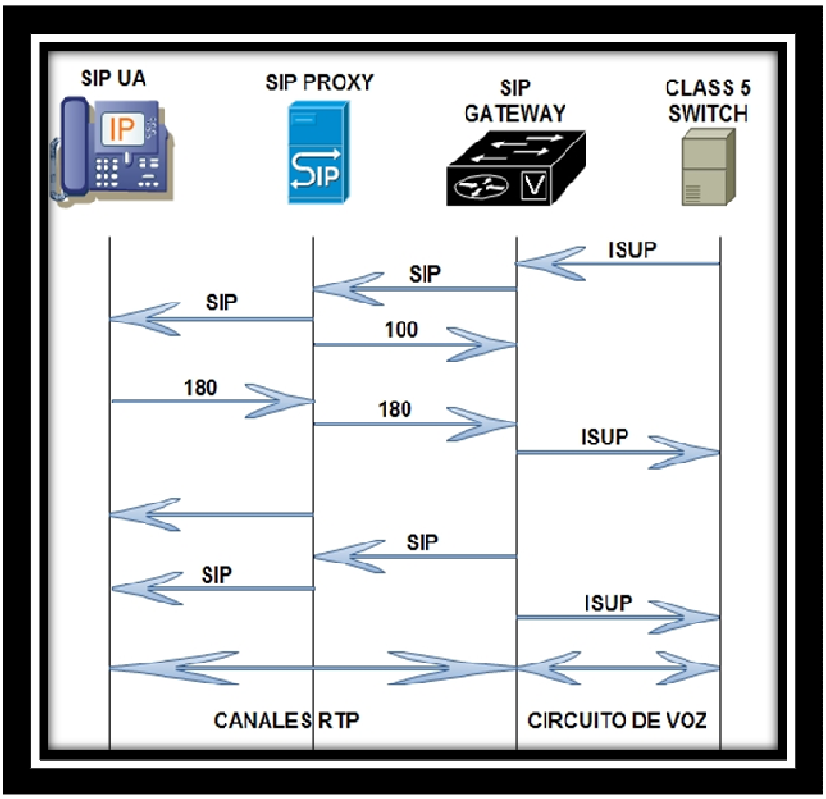
En el ejemplo contemplado en la figura II.5, un terminal conectado a la RTC llama un UA SIP. El Switch Clase 5 al cual está conectado, es el que genera la llamada y emite un mensaje ISUP IAM al Gateway RTC/SIP. Este mensaje contiene el número del destinatario, el identificador del circuito elegido por el Switch Clase 5 para la llamada (Circuit Identification Code o CIC) así como informaciones indicando la naturaleza de la llamada (palabras, fax, datos, etc..)

El Gateway RTC/SIP traduce este mensaje en un requerimiento SIP INVITE que contiene una dirección de destino SIP del cual el campo “user” es un número telefónico. Pasa el mensaje al Servidor Proxy SIP que obtiene la dirección IP del destinatario con la dirección SIP por medio de la interrogación de una base de datos o de un servidor de localización. El mensaje INVITE está relevado a la UA SIP. En paralelo, el Servidor Proxy notifica al Gateway la recepción del requerimiento INVITE por medio de la respuesta 100 Trying. El terminal SIP devuelve al Servidor Proxy una respuesta 180 Ringing para informar de la llamada, mensaje relevado por el Servidor Proxy al Gateway.

El Gateway traduce esta respuesta en un mensaje ISUP “Address Complete Message” o ISUP ACM enviado al Switch Clase 5. Este mensaje es traducido por el Switch Clase 5 en un mensaje “Alerting” si el terminal que origina la llamada es una terminal RDSI o en una señal “Ringing Tone” en el caso de una terminal analógica. Cuando el destinatario descuelga, una respuesta 200 OK esta devuelta al Servidor Proxy quien la releva al Gateway. El Gateway pone el

recibí de esta respuesta por un requerimiento ACK encaminado por el Proxy Server al destinatario.

En paralelo, el Gateway genera un mensaje ISUP Answer Message o ISUP ANM emitido al Switch Clase 5. Este intercambio de señalización ha permitido el establecimiento de canales RTP entre el terminal SIP y el Gateway así como la colocación de un circuito de voz entre el Gateway y el Switch Clase 5. Durante la fase de transferencia de información, el Gateway convierte las señales de audio recibidas sobre el circuito de voz en paquetes RTP enviados sobre los canales RTP y viceversa.



.

Figura II.5. Interfuncionamiento RTC/SIP

2.6.6. Arquitectura de servicios SIP

La arquitectura de servicios SIP está constituida de servidores de aplicación, de servidores de media y de S-CSCF. El servidor de aplicación SIP ejecuta servicios (ejemplo: Push To Talk, Presence, Prepaid, Instant messaging etc...) y pueden influenciar en el desempeño de la sesión a pedido del servicio. El servidor de aplicación corresponde al SCP de la Red Inteligente.

El servidor de media SIP (llamado también el Multimedia Resource Function o MRF) establece conferencias multimedia, emite anuncios vocales o multimedia y recolecta la información del usuario. Se trata de la evolución de la entidad Specialized Resource Point o SRP en el mundo multimedia. El servidor de llamada SIP (Servidor Proxy) es el que dispone del perfil de servicio del abonado, así como de los servicios suscritos y bajo qué condiciones invocar estos servicios. Corresponde al SSP de la arquitectura de Red Inteligente.

2.6.6.1 Servidor de aplicación

Un servidor de aplicación SIP provee un ámbito de ejecución para aplicaciones llamado “Service Logic Execution Environment” o SLEE. El provee un conjunto de servicios que permite simplificar las tareas de los desarrolladores de aplicaciones así como de los administradores. El objetivo es el de disponer de una plataforma que pone en marcha todas las funcionalidades permitiendo así al desarrollador enfocarse únicamente en la lógica “profesional” de la aplicación.

Las funciones de un servidor de aplicación son las siguientes:

- *La gestión de recursos:* el servidor de aplicaciones controla la creación y la utilización de recursos tales como los threads, las conexiones de transporte, los componentes aplicativos (ejemplo: scripts CPL, servlets SIP) así como las sesiones de aplicaciones.
- *La gestión de aplicaciones:* la aplicación puede ser asociada a un perfil de configuración durante su despliegue. Este perfil puede contener parámetros que pueden ser modificados a través de una interface administrativa durante el despliegue de la aplicación o durante su ejecución.
- *La composición de aplicación:* el servidor de aplicación debe permitir la ejecución de varias aplicaciones por un mismo requerimiento SIP. Eso provee una capacidad de modularización. De hecho, elementos de servicio pueden ser desarrollados independientemente y pueden ser combinados según las necesidades de aplicación. Eso permite por otra parte un mejor control de las interacciones de servicio.
- *La integración WEB:* Se utiliza con el fin de proveer un GUI Web para la administración y permitir el interfuncionamiento con servidores WEB.
- *La programación:* el servidor de aplicación provee un soporte para el desarrollo de aplicación, APIs (JAIN API, SIP Servlet API, etc.) así como lenguajes de script. Los scripts pueden ser creados con el apoyo de ámbitos de creación de servicio.
- *El interfuncionamiento:* El servidor de aplicación se comunica usando el protocolo SIP con el servidor de media, permitiendo las interacciones con el usuario y con el servidor de llamada (CSCF) para el encaminamiento y la señalización.

- *La seguridad:* El servidor de aplicación debe proveer mecanismos de encriptación, de autentificación y de autorización con el fin de asegurar un acceso seguro a los servicios.
- *Las capacidades no funcionales:* Alta disponibilidad, reparto de carga, tolerancia a los errores. Estas características son similares a las características exigidas por un SCP en la arquitectura de Red Inteligente.

2.6.6.2. El servidor media SIP

El servidor media SIP es una plataforma poderosa y evolutiva para el desarrollo de servicios de portales vocales y servicios vocales / video interactivos capaces de soportar centenares y hasta millares de sesiones simultáneas en un amplio rango de configuraciones.

El servidor de media SIP es un equipo físico y pone en marcha la entidad funcional “Multimedia Resource Function” o “MRF” definido por el “IMS”. El servidor de media SIP provee las funciones permitiendo interacciones entre usuarios y aplicaciones a través de recursos vocales / video.

Por ejemplo, el puede responder a una llamada y difundir un anuncio, o leer un mensaje electrónico usando funciones de síntesis vocales o recolectar la información del usuario (ejemplo: clave, voto, número) y devolverla a la aplicación. El servidor de media SIP pone en obra dos tipos de funciones:

- Las funciones de recursos del servidor media tales como las funciones de detección de tonalidad, de síntesis vocal, de reconocimiento vocal, de traducción de media etc. Es la función “Multimedia Resource Function Processor” o “MRFP”.
- Las funciones de control del servidor media que proveen a las aplicaciones los medios de controlar recursos media tales como, tocar un mensaje, colectar un voto, grabar un mensaje etc. Y eso, a través del protocolo SIP. Es la función “Multimedia Resource Function Controller” o “MRFC”.

La arquitectura distribuida del servidor de media SIP / servidor de aplicación separa las aplicaciones voz / video del control de medias, lo que permite a los operadores reducir los costos de los recursos de red y albergar con costos menores las aplicaciones clientes. El servidor de media IP soporta el protocolo de control SIP.

Además del servidor de media IP y del servidor de aplicación, las entidades siguientes pueden ser contempladas: *Browser Voice XML:* Este componente integrado en el servidor de media IP provee un ejemplo de ámbito de ejecución de aplicaciones vocales. Las aplicaciones desarrolladas según las especificaciones Voice XML pueden ser interpretadas y ejecutadas por el Browser Voice XML. Este Browser solo interpreta y determina las etapas atómicas del call flow, ya que es el servidor de media IP el que interactúa con el usuario. *Servidor ASR:* Este componente provee el servicio “Automatic Speech Recognition” o ASR. El flujo de audio del usuario es transportado sobre RTP del Media Gateway o

del teléfono IP del usuario al servidor ASR. El Browser Voice XML contacta el servidor ASR cuando un reconocimiento de palabra es necesario.

Servidor TTS: Este componente provee el servicio “Text-To-Speech” o TTS. Cuando una cadena de caracteres es emitida hacia este componente es convertida en un aviso vocal que puede ser emitida al usuario bajo la forma de flujo RTP. El browser Voice XML contacta el servidor TTS cuando un texto debe ser traducido en un mensaje vocal y entregado al usuario.

Servidor WEB: Este componente es un servidor estándar http. Esta utilizado con el fin de albergar el contenido vocal. Este contenido consiste en escrito Voice XML, anuncios vocales / video, mensajes de recepción y gramáticas de reconocimiento de la palabra. Los escritos Voice XML definen la lógica de aplicación. Mensajes de recepción apoyan el usuario en su navegación dentro de una aplicación. Las gramáticas contienen las palabras autorizadas o las frases que un usuario puede pronunciar cuando la aplicación le pide ingresar sus informaciones.

2.6.6.2.1. Funcionalidades del servidor media SIP

Las funcionalidades del servidor de media SIP incluyen las funciones de control media y de recursos media:

- *Anuncios:* La mayor parte de los servicios evolucionados utilizan formas de anuncios, bien sea un mensaje de bienvenida durante el acceso a su buzón de mensajes unificado o de un mensaje de introducción a un portal. La utilización de un servidor de media SIP para realizar servicios de anuncios permite no tener que

desplegar un nuevo servidor de anuncios.; reduciendo así el número de elementos de red y simplificando la gestión de la red. Un equipo de almacenamiento externo puede ser utilizado para almacenar anuncios creando así una solución confiable y escalable. El protocolo RTP esta utilizado para entregar el anuncio al usuario.

- *Automated Speech Recognition (ASR):* El reconocimiento de la palabra es un componente de la mayor parte de los servicios al usuario tales como mensajería vocal (voicemail), la mensajería unificada, juegos interactivos y portales vocales.
- *Generación de información de tasación:* Una tasación precisa y justa es una exigencia por los operadores del servicio con el fin de ofrecer servicios de voz y datos con fuerte valor agregado. El servidor de media SIP genera informaciones de tasación.
- *Interactive Voice Response (IVR):* El servidor de media SIP debe soportar la detección de tonalidades DTMF enviadas en la banda así como los dígitos recibidos vía SIP INFO.
- *Grabación:* El servidor de media SIP tiene capacidades de grabación y de restitución (playback).
- *Gestión del multipartes:* El servidor de media SIP debe ser capaz de proveer todos los mecanismos de control de las llamadas con varios participantes. Esta funcionalidad es utilizada dentro de numerosas aplicaciones tales como conferencias o el Push-To-Talk.
- *Transcodificación:* la transcodificación permite convertir un esquema de codificación digital en otro. En el caso de una

conferencia donde los participantes no disponen de un mismo codificador común, el servidor de media SIP asegurará entonces las traducciones de media necesarias.

- *Interfaces estándares abiertos*: El servidor de media SIP debe poder ser controlado a través el protocolo SIP y debe poder ejecutar escritos Voice XML

Numerosas aplicaciones tales como la mensajería vocal, la mensajería unificada, el pushto-talk y la conferencia utilizan esta función es decir la grabación de la llamada para que sea restituida. El servidor de media SIP utiliza servidores de almacenamiento en los operadores de servicios.

Text-To-Speech: La tecnología “text-to-speech” está estrechamente asociada a la funcionalidad IVR. El “text-to-speech” es utilizado en aplicaciones tales como la mensajería unificada a fin de leer un e-mail o fax a través del teléfono. La traducción puede ser realizada en varios idiomas.

2.6.7. SIP Clientes y Servidores

2.6.7.1. Agentes de Usuario.

Uno de los propósitos del protocolo SIP es permitir el establecimiento de sesiones entre los UA. Como el nombre lo implica, un Agente de Usuario es aquel que toma la dirección o la entrada de un usuario y actúa como agente en el set-up y en la gestión de sesiones de medios con otros

Agentes de Usuarios. En la mayoría de los casos, el usuario será un ser humano, pero podría ser otro protocolo, como en el caso de un gateway. Debe ser capaz de establecer una sesión de medios con otro UA y debe mantener el estado en las llamadas que inicia o participa.

Un sistema mínimo de estado de llamada incluye en el header, las etiquetas locales y remotas, Call-ID, CSeq local y remota, junto con el seteo de la ruta e información de estado necesaria para los medios. Esta información se utiliza para almacenar la información del diálogo y para la confiabilidad. El almacenaje de CSeq remoto es necesario para distinguir entre un *re-INVITE* y una retransmisión. Un *re-INVITE* se utiliza para cambiar los parámetros de sesión de una llamada existente o pendiente. Utiliza el mismo *Call-ID*, pero *CSeq* se incrementa porque es una nueva petición. Un *INVITE* retransmitido contendrá el mismo *Call- ID* y *CSeq* que el anterior *INVITE.*

Incluso después de haber terminado una llamada, el estado de la misma debe ser mantenido por el Agente de Usuario por lo menos 32 segundos en caso de mensajes perdidos. Los agentes de usuario desechan un *ACK* para peticiones desconocidas. Peticiones a un URI desconocidas reciben *404 Not Found Response*. Una recepción a un pedido de diálogo desconocido responde con *481 Dialog/Transaction Does Not Exist.* Las respuestas a un diálogo desconocido serán también descartadas. Estos descartes son necesarios por seguridad. Si no, un agente malévolo podía obtener información sobre otros agentes del SIP mediante el Spanning de peticiones o respuestas falsas.

En la práctica debe poder interpretarse cualquier respuesta desconocida basada en la clase (primer dígito) de la respuesta. Es decir, tratar el código *498* como, *400 Client Error*.

El agente responde a una petición no soportada con *501 Not Implemented response.* El UA SIP debe soportar transporte UDP y TCP si envía mensajes mayores a 1.000 octetos, además contiene tanto la aplicación cliente (UAC) como la de servidor (UAS). El UAC inicia peticiones mientras que el UAS genera respuestas. Durante una sesión, el UA funcionará como UAC y UAS, debe también soportar SDP (sesión description protocol) y entender cualquier extensión *Require* en una petición, aunque los campos desconocidos del header pueden ser ignorados.

El UA debe anunciar sus capacidades y características en cualquier petición que envíe, esto permite que el otro UAs aprenda de ellos. Por ejemplo, los métodos que un UA soporta dentro del header son: en *Allow*, las extensiones en *Supported* y los tipos de cuerpo de mensaje en *Accept*.

2.6.7.2. Agentes de presencia

Es un dispositivo capaz de recibir peticiones de suscripción y generar notificaciones de estado. El agente de presencia soporta presence Event package, responde a la petición de *SUBSCRIBE*, y envía peticiones *NOTIFY.* Puede recoger información de presencia de un número de dispositivos la cual puede venir de, SIP device registering, o SIP device publishing, o de otras fuentes no-SIP.

Un servidor de presencia actúa a veces como agente y brinda información de presencia y otras veces como Proxy, enviando peticiones *SUSCRIBE* a otros agentes de presencia, primero autentica una petición de suscripción, si la autentificación pasa, establece un diálogo y envía las notificaciones. La suscripción puede ser refrescada recibiendo nuevamente una petición *SUSCRIBE.*

2.6.7.3 Agentes de Usuario (B2BUA)

Recibe peticiones SIP, las reformula y las envía como nuevas peticiones, las respuestas también son reformuladas y enviadas de regreso. Por ejemplo B2BUA puede ser utilizada para implementar servicios anónimos en el cual dos UAs SIP puedan comunicarse sin aprender información de la otra parte (URI, IP address). Para lograr esto, B2BUA debe reformular las peticiones con un nuevo "From, Via, Contact, Call-ID y SDP", además de eliminar cualquier campo en el header SIP que contenga información de la parte llamante.

La respuesta también debe cambiar (Contact y SDP information de la parte llamada). Este SDP modificado apunta al B2BUA el cual realizará forward de los paquetes RTP desde la parte llamante a la llamada y viceversa. De esta forma ninguno obtiene información de la otra parte durante el establecimiento de la sesión. (La parte llamante necesita conocer el URI de la parte llamada). La aplicación más común en SIP networks es actuar como layer gateways (ALG). Algunos firewalls tienen funcionalidades ALG.

2.6.7.4. Gateways SIP

Es una aplicación que interconecta una red SIP a una red que utiliza otro protocolo de señalización. En los términos del protocolo SIP, un gateway es un Cliente y servidor SIP. Puede terminar el recorrido de la señal y puede también terminar la trayectoria de los medios, aunque éste no es siempre el caso.

Por ejemplo (ver figura II.6), un gateway SIP a H.323 termina el recorrido de la señal del SIP y convierte la señalar a H.323, pero el UA del SIP y el terminal H.323 pueden intercambiar la información de los medios RTP directamente sin pasar a través del gateway. Un gateway SIP a red de telefonía pública (PSTN) termina ambas, las trayectorias de señalización y de los medios. SIP puede traducirse con, los protocolos comunes de PSTN tales como Integrated Services Digital Network (ISDN).

Un gateway PSTN también convierte los medios RTP de la red IP en una línea estándar de telefonía. La conversión de las trayectorias de señalización y de los medios permite llamar a y desde la PSTN usando el SIP. El cuadro siguiente muestra una red SIP conectada vía gateways con la PSTN y una red H.323.

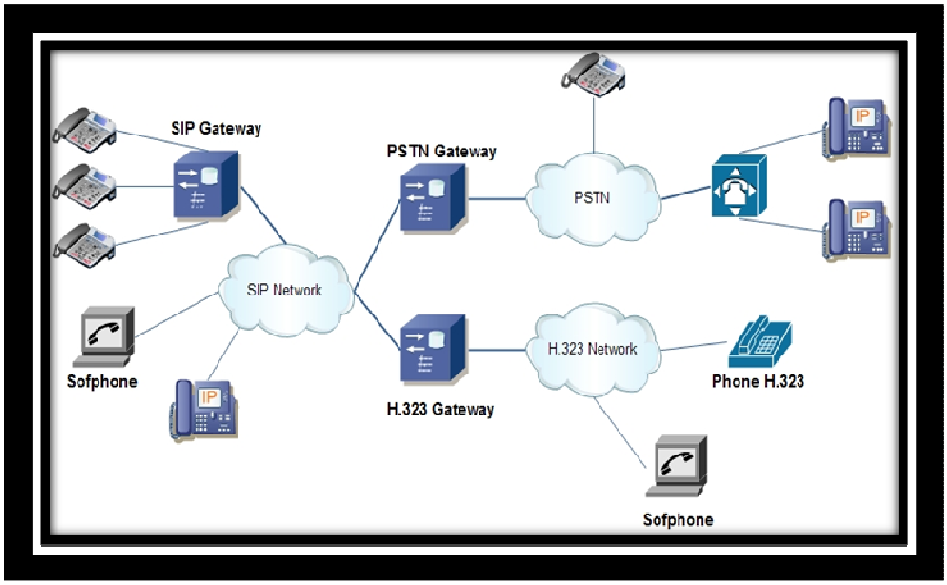


Figura II.6. Gateways SIP

Los gateways se descomponen a veces en un media gateway (MG) y un media Gateway controller (MGC). El MGC es llamado agente de llamada porque maneja los protocolos de control de la llamada (señalización), mientras que el MG maneja la conexión de los medios.

Esta descomposición es transparente al SIP. Otra diferencia entre un UA y un gateway es el número de usuarios, mientras que un UA soporta típicamente a un solo usuario, un Gateway puede apoyar centenares o a millares de usuarios. Un gateway PSTN podía dar soporte a un cliente corporativo grande, o un área geográfica entera. Un protocolo Non-SIP se puede utilizar para informar a un Proxy sobre gateways y para asistir al enrutamiento (se ha propuesto para esto Telephony Routing over IP (TRIP)) en donde una tabla de ruteo de gateways en el interdomain es desarrollada, otro protocolo utilizado es el llamado (TGREP)).

2.6.7.5. Servidor SIP

Aceptan peticiones SIP y responden a ellas (ver figura II.7). Un servidor SIP no se debe confundir con un servidor de UA. Los tipos de servidores SIP que veremos son entidades lógicas. El cuadro siguiente muestra la interacción de los UA, servidores, y servicios de localización.

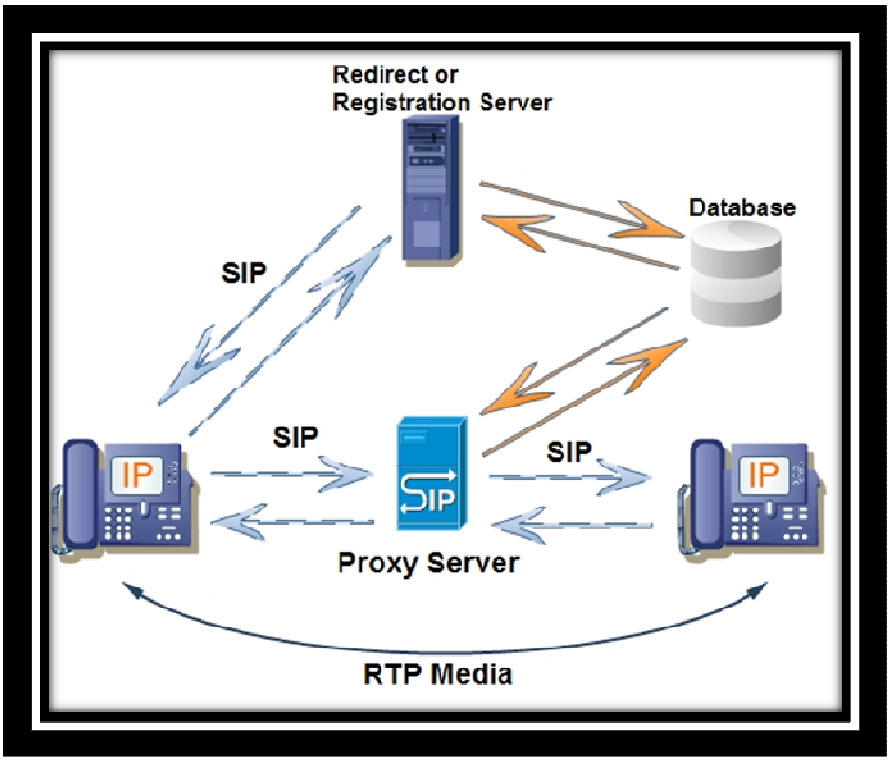


Figura II.7. Servidor SIP

2.6.7.6. Servidor Proxy

Recibe una petición SIP de un UA o de otro Proxy y actúa en nombre del UA fordwardiando o respondiendo a la petición. El Proxy no es un B2BUA puesto que este solo permite modificar peticiones y respuestas según las reglas terminantes precisadas en RFC 3261.

Un Servidor Proxy tiene típicamente acceso a una base de datos o a un servicio de localización para ayudarle en el proceso de la petición (determinando el salto siguiente). La interfaz entre el Proxy y el servicio de localización no está definido por el protocolo SIP. Un Proxy puede utilizar distintos tipos de bases de datos para ayudar en el proceso de una petición. Las bases de datos podrían contener registros del SIP, la información de presencia, o cualquier otro tipo de información acerca de donde localizan a un usuario.

Se diferencia de un UA o gateway ya que:

- No publica peticiones; responde solamente a las peticiones de un UA. (Una petición de CANCEL es una excepción a esta regla.)
- No tiene capacidad de los medios.
- No analiza cuerpos de mensaje; confía exclusivamente en campos del Header.

En la figura II.8. puede verse como un par de UA en distintos dominios establecen una sesión usando un par de Proxy, uno en cada dominio. El trapezoide se refiere a la forma de la señalización y mensajes de los medios. En esta configuración, cada UA se configura con un Servidor Proxy de salida por defecto, al cual envía todas las peticiones, este autenticará al UA, para así levantar el perfil de usuario y aplicar servicios de salida de enrutamiento. En un intercambio del interdomain, las preguntas del DNS SRV serán utilizadas para localizar un Servidor Proxy en el otro dominio.

Este Proxy, a veces llamado inbound puede aplicar servicios de enrutamiento, también tiene acceso a información de registro del usuario, y puede enrutar la petición al abonado llamado. En general, a futuro las peticiones del SIP serán enviadas directamente entre los dos UA, a menos que uno o ambos Proxy's inserten Record-Route en el header.

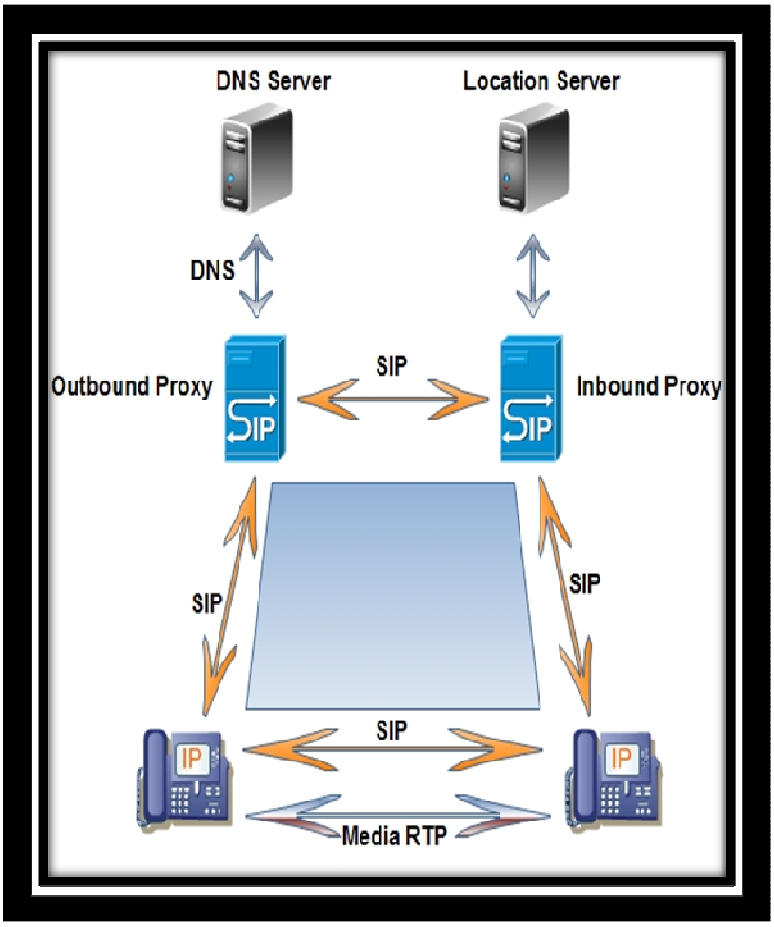


Figura II.8: Servidor Proxy

Un Proxy Server puede ser stateless o stateful.

- *Stateless:* Procesa cada petición o la respuesta SIP basada solamente en el contenido del mensaje. Una vez que el mensaje se haya analizado, procesado, remitido o se haya respondido, no queda

almacenado ningún tipo de información, además nunca retransmite un mensaje, y no utiliza contadores de tiempo SIP.

- *Stateful:* Mantiene las peticiones y las respuestas recibidas en el pasado y usa esa información en futuras solicitudes y respuestas. Por ejemplo, comienza un contador de tiempo en que se remite una petición. Si no se recibe ninguna respuesta a la petición dentro del período del contador de tiempo, el Proxy retransmitirá la petición, relevando al UA de esta tarea, también puede requerir la autentificación del UA.

  El tipo más común de proxy SIP es un proxy stateful de la transacción el cual guarda el estado sobre una transacción pero solamente mientras la petición está pendiente. Por ejemplo (ver figura II.9), un proxy stateful de la transacción guardaría el estado cuando recibe una petición INVITE hasta que recibía una 200 OK o una respuesta final de la falta (Ej. 404 not found). Después de ésa, destruiría la información del estado.

  Un ejemplo de servicio de búsqueda es un Servidor Proxy que recibe una petición INVITE, y la retransmite a un número de localizaciones en el mismo tiempo. Este Proxy Server bifurca y no pierde de vista cada una de las peticiones excepcionales y tampoco la respuesta a cada uno.

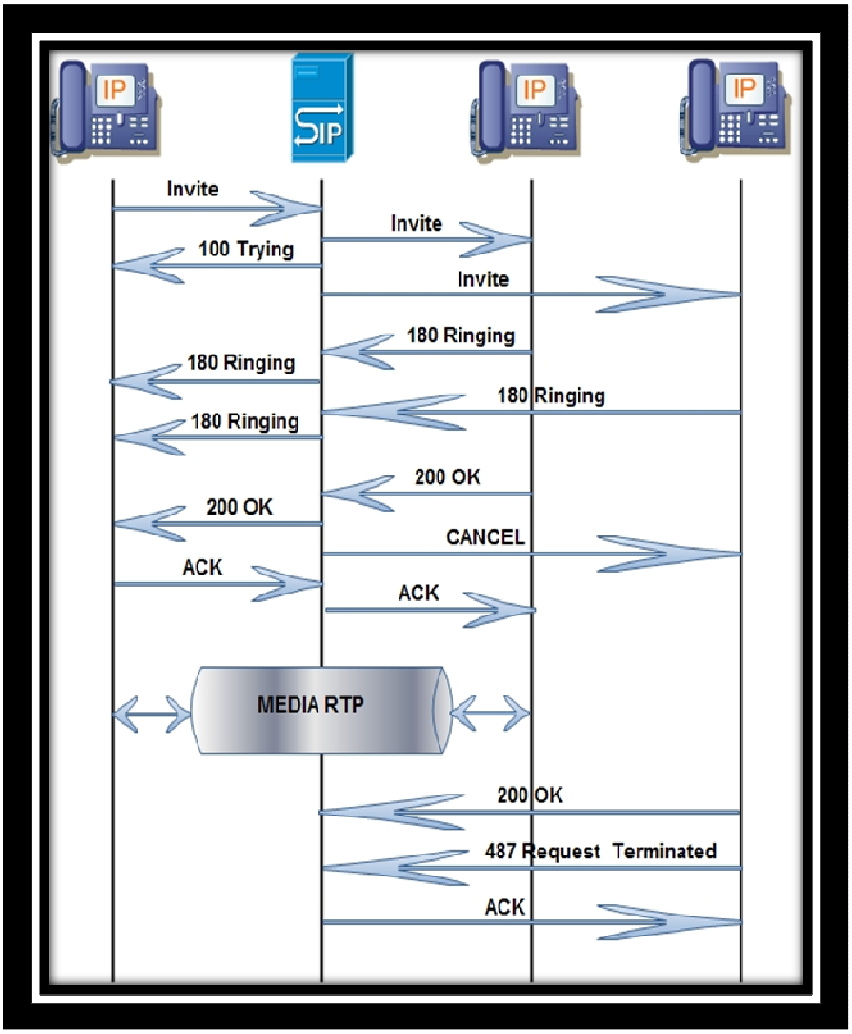


Figura II.9. Servicio de Búsqueda de un Servidor Proxy

El Proxy bifurcador envía CANCEL al segundo UA para detener la alerta. Si ambo UAs hubiera contestado, el Proxy habría remitido a ambos 200 OK y el originador de la llamada habría tenido que elegir, probablemente aceptando una y enviando un BYE a la otra. Un Proxy stateful envía generalmente una respuesta 100 Trying cuando recibe un INVITE, Proxy stateless nunca envía una respuesta 100 Trying.

Un Proxy que maneja peticiónes TCP debe ser stateful, puesto que un UA asumirá que el transporte es confiable para las retransmisiones en cualquier salto UDP. El único límite al número de Proxy es que pueden remitir un mensaje controlado por Max- Forwards del header, que se decrementa en cada Proxy que toque la petición, si Max- Forward va a cero, el Proxy desecha el mensaje y envía 483 Too Many Hops. Un contador de tiempo limita el excedente de tiempo en el que un Proxy stateful mantiene la información de estado.

En el INVITE inicial, *Session-Expires* en el header indica el intervalo después del cual los Proxy pueden desechar la información de estado sobre la sesión, el que llama puede enviar un *re-INVITE* para restaurar el contador de tiempo, permitiendo un mecanismo de Keep Alive para el SIP. Esto soluciona el problema del tiempo de almacenamiento de la información en casos donde una petición BYE se pierde o es misdirected.

2.6.7.7. Servidor Redirector

Es un tipo de servidor SIP el cual responde, pero no remite peticiones, es como un Proxy que usa una base de datos o servicio de localización para buscar un usuario (ver figura II.10).

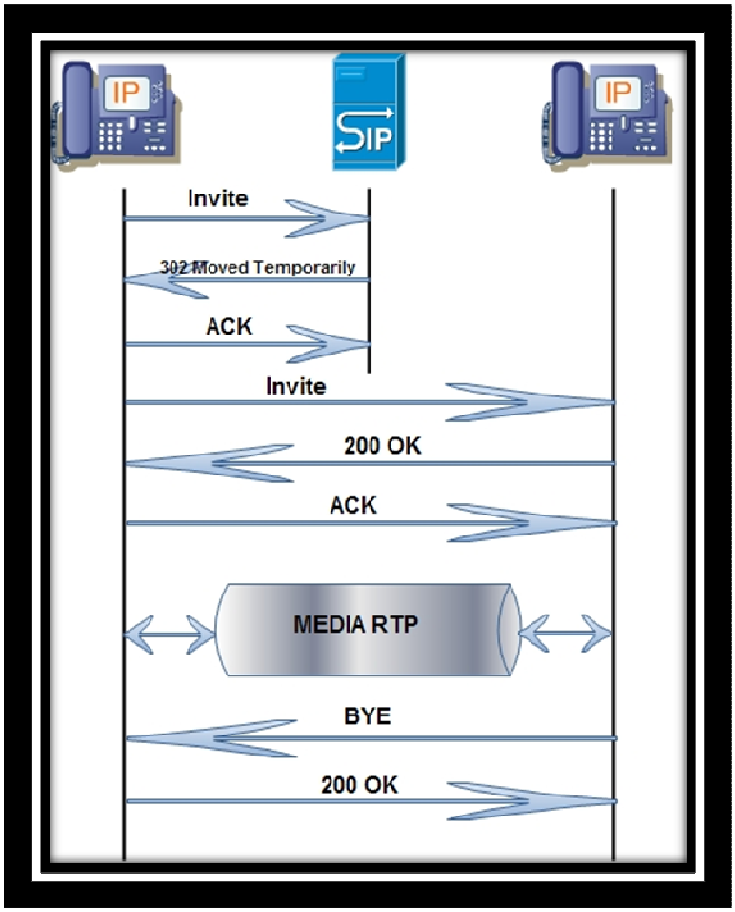


Figura II.10: Servidor de Redireccionamiento

2.6.7.8. Servidor de Registración

Un servidor de registración, acepta peticiones SIP REGISTER; el resto de las peticiones reciben una respuesta 501 Not Inplemented. La información del contacto se pone a disposición de otros servidores SIP dentro del mismo dominio administrativo, tal como los servidores Proxy y de Redireccionamiento. En una petición del registro, el campo *To* contiene el nombre del recurso que es registrado, y los campos *contact* contienen las direcciones o los alias alternativos. Generalmente

requieren a los usuarios estar autenticados, para evitar capturas por usuarios desautorizados.

2.7. Problemas del SIP con NAT

NAT, también conocida como Network Address Translation, fue la solución encontrada para resolver la escasez de direcciones IP en el pronóstico de mediados de los 90. La solución consistió en utilizar un pequeño rango de direcciones IP (en la mayoría de los casos una sola dirección IP) en el puerto fuera del firewall y un rango de direcciones reservadas (direcciones no registradas se define en el RFC1918) en el puerto dentro del firewall. Por desgracia, NAT rompe la comunicación SIP.

NAT se implementa normalmente en los routers y firewalls. A veces NAT es también conocido como (Traducción de dirección de puerto) PAT. PAT mantiene una tabla de asignación de "ip: puerto" pares que permite una única dirección externa para ser utilizado por varias direcciones internas.

En el RFC1918 se define la asignación de direcciones para redes privadas. El espacio de direcciones privado, puede ser definido por:

10.0.0.0 a 10.255.255.255 (10.8)

172.16.0.0 a 172.31.255.255 (172.16/12)

192.168.0.0 a 192.168.255.255 (192.168/16)

2.7.1. Traducción de direcciones NAT

El crecimiento actual de las redes y los masivos cambios en seguridad han creado nuevas vías que hacen difícil el funcionamiento de algunas aplicaciones. La arquitectura original de Internet donde cada equipo se podía comunicar directamente con otro utilizando una IP única y global, ha sido remplazada por una nueva arquitectura de direcciones, que consiste en una dirección global y muchas direcciones privadas interconectadas por NAT.

Esta nueva arquitectura como se ve en la figura II.11, sólo los equipos que tengan dirección pública pueden ser fácilmente conectados por otros en la red, ya que tienen una única, global y ruteable dirección IP. Los equipos de las redes privadas pueden conectarse entre sí, o pueden establecer conexiones TCP o UDP con equipos que permanezcan en la red pública.

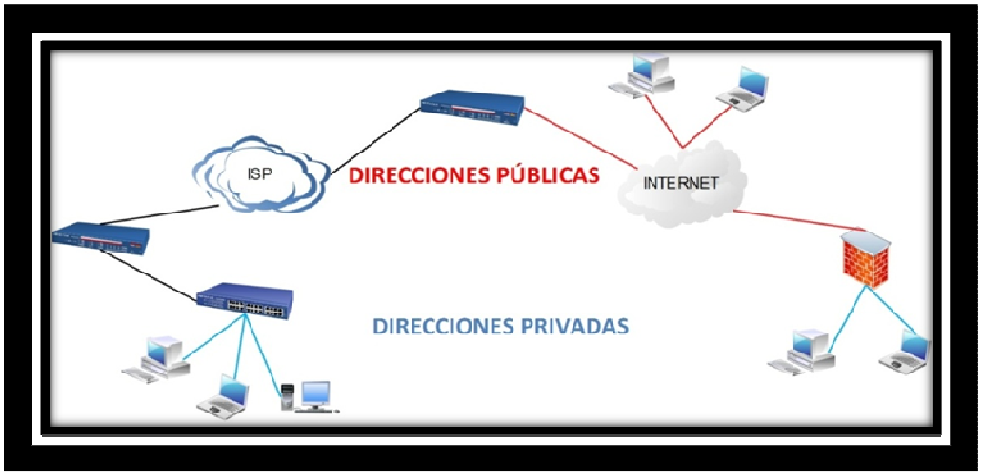


Figura II.11. Esquema de conectividad de Internet

Los dispositivos que realizan el NAT traducen la dirección y puertos de los paquetes que provienen de las redes privadas hacia las redes públicas, por lo tanto los dispositivos NAT generalmente permiten sólo conexiones salientes y

bloquean cualquier tráfico entrante, a no ser que este configurado específicamente de otra manera.

Esta nueva arquitectura de direcciones permite la comunicación cliente / servidor en el caso típico donde el cliente se encuentra en una red privada y el servidor se encuentra en la red pública. Esta arquitectura hace difícil la comunicación directa entre dos equipos (que es importante para las redes P2P) que pertenezcan a distintas redes privadas. Con esto tenemos que muchas aplicaciones P2P que comparten archivos, realizan videoconferencia, y juegos en línea que no funcionan a través de entornos con NAT.

2.7.2. Cómo funciona el NAT

Cuando un cliente en la red interna contacta a un equipo público, envía paquetes IP destinados a ese equipo. Estos paquetes contienen toda la información de direccionamiento necesaria para que puedan ser llevados a su destino. NAT se encarga de esta información:

- Dirección IP de origen (Ej. 192.168.0.1)
- Puerto TCP o UDP de origen (Ej. 12345)

Cuando los paquetes pasan a través de la pasarela de NAT, son modificados para que parezca que se han originado y provienen de la misma pasarela de NAT. La pasarela de NAT registra los cambios que realiza en su *tabla de estado*, para así poder:

a) Invertir los cambios en los paquetes devueltos.
b) Asegurarse de que los paquetes devueltos pasen a través del cortafuego.

Por ejemplo, podrían ocurrir los siguientes cambios:

- IP de origen: Sustituida con la dirección externa de la pasarela (Ej. 10.0.0.1)
- Puerto de origen: Sustituido con un puerto no en uso, puerto de la pasarela (Ej. 50050)

Ni la máquina interna ni el otro equipo de Internet se da cuenta de estos pasos de traducción. Para el equipo interno, el sistema NAT es simplemente una pasarela a Internet. Para el equipo público, los paquetes parecen venir directamente del sistema NAT; ni siquiera se da cuenta de que existe la estación interna.

Cuando el anfitrión de Internet responde a los paquetes internos de la máquina, los direcciona a la IP externa de la pasarela NAT (10.0.0.1) y a su puerto de traducción (50050). La pasarela de NAT busca entonces en la tabla de estado para determinar si los paquetes de respuesta concuerdan con alguna conexión establecida.

2.7.3. Tipos de NAT

Existen cuatro tipos de NAT definidos como

- Full conexión
- Conexión Restringida

- Conexión Restringida por Puerto
- NAT Simétrico

Para una dirección interna, los tres primeros tipos de NAT mantienen un mapeo de su dirección interna que es independiente de la dirección de destino. El cuarto tipo de NAT localizará un nuevo mapeo para cada dirección de destino independiente. A menos que haya una tabla de mapeo estático.

El mapeo que se abre cuando el primer paquete es enviado de un cliente a través de NAT puede ser válido apenas por cierta cantidad de tiempo, (típicamente algunos minutos), a menos que los paquetes continúen, siendo enviados y recibidos en un puerto IP.

2.7.3.1. Full Conexión

En el caso de la conexión completa o llena, cuando la traducción se encuentra establecida, cualquier equipo que quiera alcanzar al cliente detrás del NAT, necesita sólo conocer la dirección y el puerto de donde el tráfico está siendo enviado. Por ejemplo en la figura II.12. un equipo detrás de un NAT con dirección 192.168.0.2 enviando y recibiendo en el puerto 5000, se ha traducido a la dirección externa 10.0.0.1 en el puerto 12345. Cualquier equipo de Internet puede enviar tráfico a esta IP externa y este tráfico será traspasado a la dirección cliente 192.168.0.2:5000.

Es el caso de Firewalls sin control de sesión. Normalmente implementado a través de filtrado de paquetes y es el tipo más inseguro de Firewall y cada vez menos común.

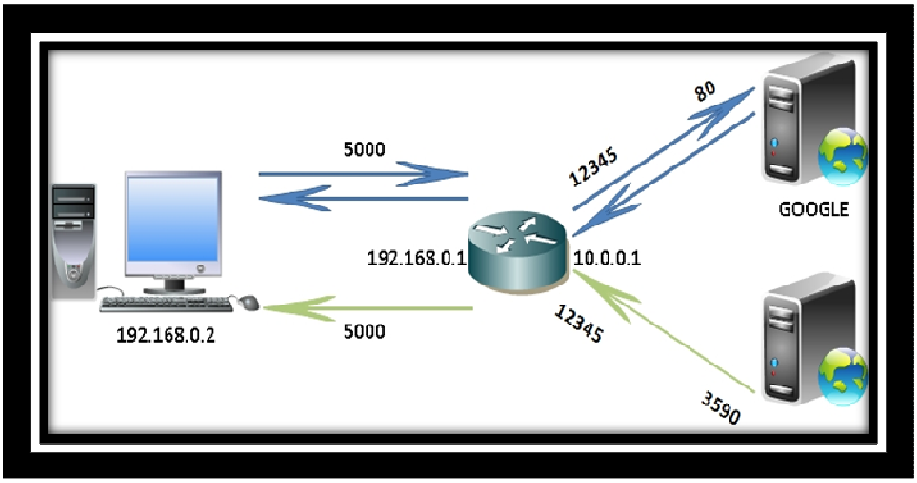


Figura II.12. Full Conexión

2.7.3.2. Conexión Restringida

En este caso de la conexión restringida, ver figura II.13, la IP y puerto externo son abiertos cuando el equipo de la red privada envía tráfico saliente a una dirección IP específica. Por ejemplo si el cliente envía paquetes hacia el equipo A, el NAT traduce la dirección privada 192.168.0.2:5000 a la dirección pública 10.0.0.1:12345, dejando que solamente el equipo A pueda enviar tráfico a esa dirección pública. El NAT bloqueará cualquier otro tráfico que venga de una dirección distinta.

Por lo tanto el equipo B podrá enviar tráfico hacia la red privada, solamente si antes el equipo que se encuentra detrás del NAT ha enviado tráfico a este equipo B. Note que en este caso el Firewall tiene

control sobre la sesión, esperando paquetes pertenecientes a una sesión, pero una vez abierto, aquel computador puede iniciar cualquier sesión independiente del puerto (200.210.1.1:3000, 200.210.1.1:3001...).

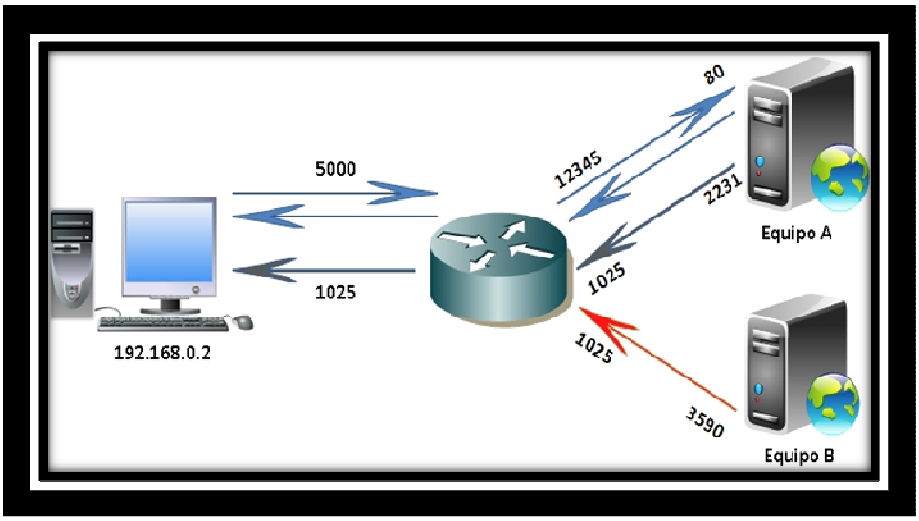


Figura II.13. Conexión Restringida

2.7.3.3. Conexión Restringida por Puerto

Una conexión restringida por puerto, ver figura II.14, es casi idéntica a la conexión restringida, pero en este caso el NAT bloqueará todo el tráfico, a menos que el cliente haya enviado antes tráfico a una IP y *puerto específico,* solo entonces esa IP:PUERTO tendrán acceso a la red privada. Entonces en nuestro ejemplo si el cliente envía tráfico al equipo A puerto 80, el NAT sólo permitirá tráfico proveniente desde Equipo A: 80, ahora si el cliente envía a múltiples direcciones y puertos entonces estas direcciones podrán responder al cliente a la dirección 10.0.0.1:12345.

En este caso el Firewall tiene un control mayor de sesión, solo permitiendo que paquetes pertenecientes a aquella sesión puedan retornar, al final de la sesión, si el computador de destino resuelve enviar paquetes de un puerto diferente estos no serán aceptados.

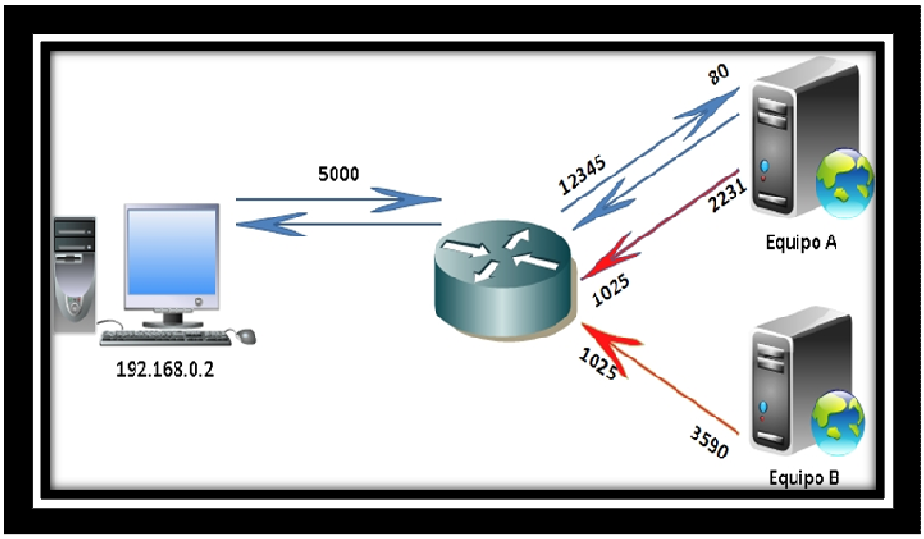


Figura II.14. Conexión Restringida por Puerto

2.7.3.4. NAT Simétrico

El último tipo de NAT simétrico, ver figura II.15, es diferente de los otros tres. Específicamente debido a que la traducción de la IP pública a la privada depende de la IP de destino donde ha sido enviado el tráfico. Para nuestro ejemplo si el cliente envía tráfico desde la dirección privada 192.168.0.2:5000 a Google, la dirección traducida sería la 10.0.0.1 con el puerto 12345, y si el equipo privado envía tráfico a otra IP pública distinta, por ejemplo Yahoo la dirección traducida para ese equipo sería la 10.0.0.1 con el puerto 45678.

Google puede solamente responder a su dirección traducida (10.0.0.1:12345) y Yahoo a su dirección (10.0.0.1:45678). Si cualquier otro equipo envía tráfico a cualquiera de estas dos direcciones éste será rechazado.

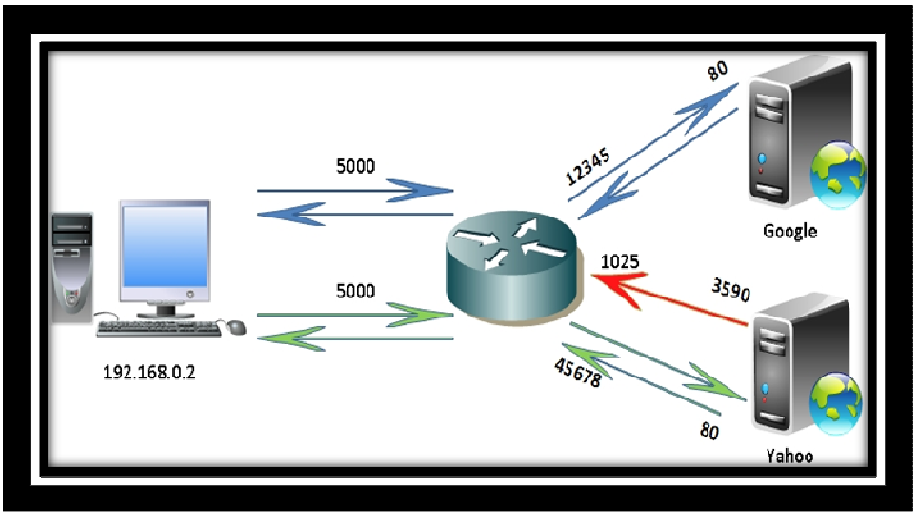


Figura II.15. NAT Simétrico

2.7.4. NAT en la transversabilidad de la señalización SIP

Existen dos partes de una llamada basada en SIP. La primera es la señalización, que es un protocolo de mensajes para establecer una llamada. La segunda es realmente el flujo de media. Los paquetes de RTP viajan directamente entre los dispositivos finales.

La señalización SIP puede atravesar el NAT de una forma bastante directa, desde que exista un proxy, a un salto de distancia de NAT, que reciba los mensajes SIP del cliente (a través de NAT) y entonces retorne los mensajes para el mismo lugar. El proxy precisa retornar los paquetes de SIP para el mismo puerto de donde este recibió los paquetes (En el puerto SIP 5060). El

SIP tiene etiquetas (tags) que le dicen al proxy para hacer esto – La etiqueta “recibida” dice al proxy para retornar un paquete para una IP específica y la etiqueta “rport” guarda el puerto a donde retornar. La mayoría de los proxys todavía no implementan la etiqueta “rport”, y algunos clientes no van a procesar los mensajes SIP correctamente. Si estas etiquetas estuviesen presentes por lo menos en un principio, en ese caso si existirá un mecanismo para atravesar el NAT. Otro modo simple de atravesar el NAT es usar TCP para la señalización SIP entre el cliente y el proxy.

Desde que la conexión TCP es abierta a través de NAT directamente del cliente para el proxy, la señalización procederá sin bloqueo. Nuevamente, muchos proxys todavía no implementan la opción TCP y trabajan apenas usando UDP. Hay que notar que la señalización SIP debería estar apta a atravesar cualquiera de los cuatro tipos de NAT si el proxy retorna los mensajes de SIP en el mismo puerto fuente que él recibió el mensaje inicial. El mensaje inicial SIP, enviado para el proxy IP:Port, abre el mapeamiento de NAT, y el proxy retorna los paquetes de NAT para el mismo IP:Puerto. Esto es permitido en cualquier escenario de NAT.

Registrar un cliente que está atrás de un NAT requiere un *Registrar* que pueda salvar el IP:Puerto en la información de registro basada en el puerto e IP que se ve como fuente de mensajes SIP o un cliente que sepa de la dirección mapeada externamente y de su puerto para así insertarlo en la información de contacto como IP:Puerto y recibir los mensajes SIP. Es preciso tener cuidado en usar un intervalo de registro menor que el “keepalive” para el mapeamiento de NAT.

2.7.5. Formas de transversabilidad por el NAT

Existen innumerables mecanismos creados por las formas de transversabilidad por el NAT. La mayoría funciona para los NATs de tipo Full Conexión, Conexión Restringida y Conexión Restringida por Puerto. Entre tanto, apenas el RTP Relay funciona para los NATs de tipo simétrico, felizmente Asterisk puede actuar como un RTP Relay usando la opción “canreinvite=no” para aquella extensión en el archivo sip.conf. Podemos dividir los métodos de transversabilidad de NAT en Near-End-Nat Tarversal (Soluciones en los clientes) y Far-End-Nat-Tarversal (Soluciones en el servidor).

Soluciones en los clientes:

- UPnP
- ALG
- STUN
- Configuración Manual
- ICE

Soluciones en el servidor:

- Comedia (Conexion Oriented Media)
- TURN – Traversal of UDP using Relay NAT

2.7.5.1. Soluciones en los clientes

- **UPnP:**

  Un cliente puede preguntar al NAT como es que está mapeado para un par IP:Puerto a través de un protocolo llamado Universal Plug and Play.

Esta es una solución que está siendo promovida por Microsoft (entre otros). El cliente pregunta al NAT vía UPnP que mapeamiento el debe usar si él quisiera recibir en el puerto x. NAT responde con el par IP:Puerto que alguien en la red pública debería usar para alcanzar el cliente en aquel puerto. Muchos fabricantes de dispositivos NAT ya incluyen UPnP en sus productos. Un problema es que el UPnP no va a funcionar en el caso de NATs Simétricos.

- **ALG – Aplication Layer Gateway**

Esta técnica se vale de la instalación de un Firewall/NAT mejorado llamado un gateway de capa de aplicación (ALG) que entiende la relación entre los flujos de media y los mensajes de señalización. El ALG procesa los flujos de media y señalización de forma que refleja la dirección pública y puertos en la comunicación para fuera del Firewall, en otras palabras toda la traducción necesaria es hecha en el Gateway y ruteadores Cisco más recientes con IOS/Firewall ya que permiten estos recursos.

- **STUN – Simple Traversal of UDP NAT**

En la ausencia de un mecanismo para comunicarse con el dispositivo NAT, el mejor medio para que el cliente determine su par IP:Puerto externo es preguntar al servidor situado en la Internet Pública como él ve su dirección. En este escenario existe un servidor que permanece esperando estos paquetes. Cuando el recibe un paquete el retorna un mensaje del mismo puerto para la fuente del paquete recibido conteniendo el par IP:puerto que él ve

en el encabezado del paquete enviado. En todos los casos (todos los 4 casos de NAT), el cliente irá a recibir un paquete de retorno. El cliente entonces va a determinar:

- Si el está atrás de un NAT (El par IP:Puerto contenido es diferente del par IP:Puerto que él piensa que está)
- Cual par IP:Puerto público él debería usar para colocar en el mensaje SDP de forma que lo alcance. Por ejemplo, si el cliente quiere ser alcanzado en 10.0.0.1:8000, el primero va a enviar una consulta a NAT por el puerto 8000. NAT irá realmente a recibir una consulta del paquete 200.180.4.168:1234 y así va a responder para el par IP:puerto con el paquete contenido 200.180.4.168:1234. El cliente entonces podrá colocar esto en su SDP “m=AUDIO 1234” y “c=200.180.4.168”, así el cliente continua escuchando en el puerto 10.0.0.1:8000.

  Esto funciona en las siguientes situaciones:

  - El cliente debe enviar y recibir el RTP en el mismo puerto.
  - El cliente debe enviar un mensaje SIP logo después de enviar la consulta para la prueba de NAT. Si existe un largo atraso NAT puede tener un timeout.
  - En el caso de Conexión Restringida y Conexión Restringida por puerto, el cliente debe enviar los paquetes para el punto destino antes que el NAT permita el paso de paquetes que van del punto destino al cliente. Esto no va a funcionar en el caso de NAT simétrico, porque la dirección de la prueba de

NAT es diferente de aquel punto destino y de este modo el mapeamiento de NAT que se ve es diferente de aquel que el punto destino usa para enviar paquetes hasta el cliente en aquel par IP:Puerto.

STUN - (Travesía simple de UDP sobre el NAT). Es un protocolo para configurar el tipo de prueba NAT como fue descrito. El realmente hace un poco más que apenas retornar el par IP:Puerto público, el puede también determinar el tipo de NAT del que usted está detrás. Clientes que usan el protocolo STUN ya existen como el XTEN, por ejemplo. Los pedidos de STUN especifican los siguientes parámetros:

RESPONSE-ADDRESS: El servidor STUN enviará su respuesta para el par IP:Puerto especificado en el atributo RESPONSE-ADDRESS. Si este campo no estuviese presente, entonces el servidor envía su respuesta en el par IP:Puerto de donde él recibió el pedido. Si ambas “flags” Change IP y Change Port no estuviesen seteadas, el STUN responde de el par IP:Puerto que el paquete inicial fue enviado. Si el Change IP estuviese seteado, el servidor responde de un IP diferente y si el Change Port estuviese seteado entonces él responde de un puerto diferente. La respuesta del STUN contiene las siguientes informaciones:

MAPPED-ADDRESS: Es el par IP puerto del cliente como es visto en el primer servidor STUN fuera de NAT al recibir el pedido.

CHANGED-ADDRESS: La Dirección IP que debería ser la fuente de respuesta retornada si el pedido fue hecho con el “flag” **Change IP** seteado.

SOURCE-ADDRESS: El Par IP puerto de donde la respuesta STUN fue enviada. Usando una combinación de diferentes pedidos el servidor STUN, un cliente puede determinar si él está en el Internet abierto o si está atrás de un Firewall que bloquea el UDP o si él está atrás de un NAT y de qué tipo.

- **Configuración manual**

En este método el cliente es manualmente configurado con los detalles de las direcciones públicas IP y puertos que el NAT irá a usar para la señalización y media. En este caso el NAT debe ser configurado manualmente con mapeamientos estáticos en el ruteador. Asterisk permite ser configurado de forma manual cuando está atrás de un NAT.

En el archivo sip.conf en la sesión general, las instrucciones:

  - Externip=Dirección IP Externo
  - Localnet=Dirección da Red Local Interna

Permiten que cuando Asterisk está enviando paquetes SIP para afuera de la red la dirección sea substituida por la dirección definida en el comando *Externip*. La línea *Localnet* define cuales direcciones pertenecen a la red local. Todas las redes que no estuviesen en la franja definida en localnet son externas. Con esto el Asterisk sabe

cuándo debe sustituir las direcciones de los encabezados dependiendo del peer de destino.

En el archivo RTP.CONF es posible definir en que puertos RTP el Asterisk va a trabajar.

```
; RTP Configuration
;
[general]
;
; RTP start and RTP end configure start and end addresses
;
rtpstart=10000
rtpend=20000
```

- **ICE – Interactivity Connectivity Establishment**

El ICE está siendo desarrollado por la IETF en el grupo de trabajo MMUSIC y experimenta la forma de unificar varias técnicas de atravesar NAT. Esto va a permitir que el cliente VoIP atraviese con éxito una gran variedad de firewalls que existiesen entre el usuario remoto y la red.

ICE define una estandarización en los clientes SIP para que éstos puedan determinar qué tipo de firewall existe entre ellos, los servidores y a su vez determinar una dirección IP a través de la cual ellos se puedan comunicar. Usando mecanismos como STUN, TURN, RSIP se obtienen direcciones localmente configuradas que

van a proveer una dirección donde el cliente se podrá comunicar. La gran ventaja del ICE es la unificar los métodos de transversabilidad por NAT.

2.7.5.2. Soluciones para el servidor

- **COMEDIA Conexión Oriented Media**

  La solución de arriba funciona bien (Servidor STUN) para los tres primeros tipos de NAT. El cuarto caso (NAT simétrico) no va a permitir este esquema, pues este tiene diferentes mapeamientos dependiendo de la dirección IP objeto. De esta forma el mapeamiento que el NAT designó entre el cliente y la prueba NAT es diferente de aquella entre el cliente y el gateway. En el caso de NAT simétrico el cliente deberá enviar y recibir el RTP de vuelta de la misma dirección IP. Cualquier conexión RTP entre un punto destino fuera de NAT y uno dentro de NAT debe ser establecido punto a punto y así, el punto destino fuera de NAT debe esperar hasta recibir un paquete de un cliente antes que él pueda saber para donde responder. Esto es conocido como *"Media orientada a conexión".*

  Si se desea que ambos, los UACs que están atrás de NATs y los UACs en la Internet abierta se comuniquen, entonces ellos deben saber si pueden confiar en el mensaje SDP que ellos reciben en el mensaje SIP y cuando ellos precisan esperar recibir un paquete directamente antes que el cliente abra un canal de vuelta para el par IP:puerto fuente de aquel paquete.

- **TURN – Traversal using Relay NAT.**

  Si un dispositivo soporta media orientada a conexión, entonces el problema de atravesar un NAT simétrico está resuelto. Dos escenarios todavía son problemáticos.

  - Si el punto destino soporta la directiva a=direction:active tag.
  - Si los dos puntos destinos están atrás de NATs simétricos.

  En cualquiera de los dos casos, una solución es tener un Relay de RTP en el medio del flujo entre los puntos destinos. El Relay RTP actúa como un segundo punto destino para el cual los dispositivos reales se intentan comunicar uno con el otro. Típicamente, existiría un servidor en medio del flujo que va a manipular el SDP de forma de instruir los puntos destino para enviar el RTP para el Relay. El Relay establecerá su propio mapeamiento de una sesión, guardando el par IP:puerto de cada punto destino para donde el debería enviar los paquetes RTP.

2.7.6. Soluciones Prácticas para la PBX

Lo más difícil en lo concerniente a NAT en Asterisk o Elastix es entender que existen diversos casos o situaciones y que cada uno debe ser tratado individualmente.

En primer lugar vamos a tratar esto como dos soluciones separadas.

• PBX detrás de NAT

• Clientes detrás de NAT

Obviamente existen diversas situaciones intermediarias y la cosa se complica pensando que tenemos diferentes tipos de NAT. Para aumentar la complejidad tenemos clientes que soportan diferentes tipos de soluciones para NAT (TURN, STUN, ICE, ALG). Esto puede tornar el problema realmente complejo.

2.7.7. Parámetros de PBX usados para atravesar NAT

NAT

• *nat=yes (puede ser true, t, y, 1 y on)*

Es la combinación de los modos route + modo rfc3581

• *nat=route*

Asterisk enviará el audio para el puerto en vez de confiar en las informaciones contenidas en los encabezados SIP y SDP. Esto solo va a funcionar si el teléfono detrás de NAT envía y recibe el audio del mismo puerto (RTP).

• *nat=rfc3581*

Este es el estándar, en este caso Asterisk o Elastix va a adicionar el “rport” al encabezado SIP informando al cliente en que puerto recibió el pedido y va a encaminar el flujo de los mensajes para el cliente en la dirección de donde vino y no en la dirección descripta en los encabezados. El cliente en este punto tiene condiciones de saber su dirección externa (campo Via received) y ahora su puerto externo.

• *nat=never*

En este caso la PBX no va a adicionar el rport en la línea VIA del encabezado como en la RFC3581

QUALIFY

Esta opción tiene dos funciones, mantener el NAT abierto y certificarse que Asterisk no intente enviar una llamada para un teléfono que está inalcanzable.

- *Qualify=yes*

Esta opción usa el valor estándar de 2 segundos.

- *Qualify=no*

Deshabilita el chequeo del peer

- *Qualify=x ms*

Setea el tiempo en ms entre los chequeos

EXTERNIP

Esta es la opción usada dentro de la sesión general del archivo sip.conf y puede ser colocada o como una ip o como un hostname apuntando la dirección externa de su dispositivo NAT.

Ex: externip=200.180.4.168

Solo precisa usar esta opción si la PBX está detrás de NAT intentándose comunicar con dispositivos fuera de NAT.

LOCALNET

Usada dentro de la sesión [general] del archivo sip.conf. Indica la red privada interna donde la PBX no va a usar la dirección externa provista por el parámetro externip.

Ex: localnet=10.1.0.0/255.255.0.0

2.7.8. Escenarios de Asterisk con NAT

Para simplificar, vamos a usar dos situaciones que son las más típicas. PBX está detrás de un Firewall sub dominio del área técnica de la empresa. Los clientes son externos y no tienen dominio sobre la configuración de los firewalls de estos clientes.

- **PBX detrás de NAT**

Cuando la PBX está detrás de NAT podemos usar las configuraciones localnet y externip en el archivo sip.conf más allá de redireccionar los puertos en el Firewall. Suponiendo (ver figura II.16) que la dirección IP externa fuese 200.184.7.1 y que la red local interna fuese 192.168.1.0/24. Esto quedaría así:

[general]

nat=yes

externip = 200.84.7.1

localnet = 192.168.1.0/255.255.255.0

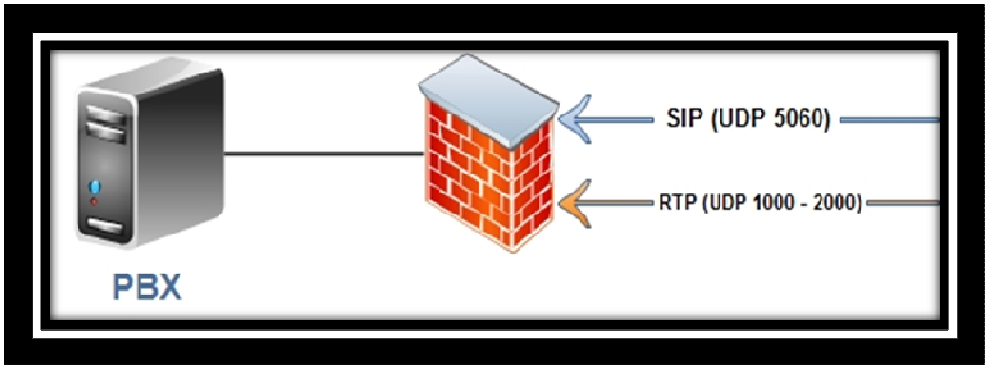


Figura II.16. PBX detrás de NAT

Más allá de esto, es preciso redireccionar los puertos UDP 5060 y RTP de 10000 a 20000 en el Firewall. Si usted quisiera reducir esta franja puede editar el archivo rtp.conf.

- **Cliente detrás de NAT**

Cuando un cliente está detrás de un NAT, normalmente este NAT es dinámico, principalmente cuando es de uso doméstico. Con esto, solo restan las opciones de que el cliente soporte STUN o UPnP para que pueda aprender la dirección de una fuente externa o a partir del ruteador respectivamente. Otra forma de operar con un cliente, que está detrás de un NAT, y la PBX es el uso de un túnel basado en PPTP, IPIP o IPSec, esto puede ser hecho a través de un ruteador (Cisco o Linux).

Cuando opera un cliente detrás de un NAT y configura STUN en el cliente, se coloca los siguientes parámetros en la configuración del cliente en el archivo sip.conf. Los clientes tienen obligatoriamente que registrarse.

nat=yes ; Ignora el encabezado VIA y usa la dirección de donde llega el paquete.
canreinvite=no ; Fuerza el flujo de media por la PBX
qualify=500 ; Fuerza a que un paquete exploratorio que mantiene el NAT abierto

2.8. PROBLEMAS DE SIP CON RTP

El protocolo en tiempo real del transporte (RTP) define un formato estandarizado del paquete para entregar audio y vídeo sobre el Internet.

Fue desarrollado por el grupo de funcionamiento de transporte de Audio-Vídeo del IETF, primero publicado en 1996 como RFC 1889, el cual quedó obsoleto en el año 2003 como RFC 3550. El Protocolo en Tiempo Real de Transporte se puede también

utilizar conjuntamente con RTSP que es un protocolo que realza el campo de los usos multimedia.

RTP no tiene un estándar TCP o UDP puerto en el cual se comunica. El único estándar que obedece es que las comunicaciones de UDP estén hechas vía un puerto uniforme y que el puerto impar más alto siguiente se utilice para el protocolo de control de comunicaciones RTP (RTCP). Aunque no hay estándares asignados, RTP se configura generalmente para utilizar los puertos del 16384 al 32767. Además puede llevar cualquier dato en tiempo real, tales como audio y video interactivos. La disposición de llamada y el desmontaje para los usos de VoIP es realizada generalmente por cualquiera de los dos protocolos SIP o H.323. RTP utiliza una gama de puertos dinámicos por lo que hace difícil el atravesar los cortafuegos. Para solucionar este problema, a menudo es necesario instalar un Servidor de Media o un Proxy RTP.

El protocolo fue diseñado originalmente como multicast, pero se ha aplicado desde entonces en muchos usos unicast. Se utiliza con frecuencia en medios que involucran sistemas (conjuntamente con H.323 o SIP) así como en videoconferencias. Va junto con el RTCP y está sobre el User Datagram Protocol (UDP). Los usos que se le atribuye a RTP son menos sensibles a la pérdida del paquete, pero típicamente muy sensible al retraso, así que UDP es una mejor opción que TCP para tales usos.

Según el RFC 1889, los servicios proporcionados por RTP incluyen:

- *Carga-tipo identificación*: Aquí se informa el tipo de contenido que se está transportando.
- *Número de serie*: Se indicará el número de serie de la PDU.

- *El estampar de Tiempo*: Permitirá los cálculos de la sincronización.
- *Supervisión de la entrega.*

Los protocolos por si mismos no proporcionan mecanismos para asegurar la entrega oportuna. También no dan calidad del servicio y garantías (de QoS). Estos parámetros tienen que ser proporcionados por otros mecanismos.

Sin embargo, los protocolos entregan los datos y se cercioraran de que los paquetes recibidos se encuentren en el orden correcto. También, RTCP proporciona información sobre la calidad de la recepción de los paquetes para así hacer ajustes locales. Por ejemplo si una congestión se acaba de formar, el protocolo podría decidir el disminuir el flujo de datos.

RTP también fue publicado por ITU-T como H.225.0, pero quitado más adelante una vez que el IETF tuviera un RFC estable ya publicado. Además tiene un perfil específico para las conferencias audio y video con control mínimo.

2.8.1. NAT en el flujo de media RTP

El RTP para atravesar un NAT no tiene una solución tan fácil como la señalización SIP. En este caso de RTP, el cuerpo del mensaje SIP contiene informaciones sobre los puntos destino, necesarios para permitir la comunicación de uno con el otro. Esta información es contenida en el mensaje SDP. Los dispositivos rellenan esta información de acuerdo con lo que ellos saben sobre sí mismos.

Un cliente situado detrás de un NAT conoce apenas su puerto interno IP:Puerto y es esto lo que él coloca en el cuerpo SDP del mensaje SIP. Cuando el punto de destino quiere enviar paquetes para el punto origen, el usará la información SDP recibida, la misma que contiene la dirección IP interna del punto origen y los paquetes nunca van a llegar. A continuación se encuentra un ejemplo de un “trace” de un mensaje INVITE de un cliente SIP atrás de un NAT recibida por el Gateway:

001 INVITE sip:12125551212@211.123.66.222 SIP/2.0

002 Via: SIP/2.0/UDP 211.123.66.223:5060;branch=a71b6d57-507c77f2

003 Via: SIP/2.0/UDP 10.0.0.1:5060;received=202.123.211.25;rport=12345

004 From: <sip:2125551000@211.123.66.223>;tag=108bcd14

005 To: sip: 12125551212@211.123.66.222

006 Contact: sip: 2125551000@10.0.0.1

007 Call-ID: 4c88fd1e-62bb-4abf-b620-a75659435b76@10.3.19.6

008 CSeq: 703141 INVITE

009 Content-Length: 138

010 Content-Type: application/sdp

011 User-Agent: HearMe SoftPHONE

012

013 v=0

014 o=deltathree 0 0 IN IP4 10.0.0.1

015 s=deltathree

016 c=IN IP4 10.0.0.1

017 t=0 0

018 m=audio 8000 RTP/AVP 4

019 a=ptime:90

020 a=x-ssrc:00aea3c0

En el tráfico capturado, la Dirección IP en la línea 003 del encabezado SIP es la dirección IP donde el cliente piensa que está (10.0.0.1). Pero el Proxy sabe la dirección IP real por la que recibió el paquete. Entonces él adicionará las etiquetas “received” y “rport” con la dirección IP y el puerto después del mapeamiento de NAT. Estas etiquetas permiten al proxy encaminar los mensajes SIP de vuelta al cliente vía NAT.

La información que es usada para pasar los datos de voz (conexión RTP) es mantenida en el mensaje de las líneas 014 y 016. El cliente espera recibir en el puerto 8000 (m=) en la IP 10.0.0.1 (c=), que es el puerto que él mira como propio, y como existe un segundo punto final retornará los paquetes. El resultado es que una vez que la llamada esté establecida (la señalización SIP pasa) el audio no es recibido.

Si el cliente estuviese atrás de uno de los tres primeros tipos de NAT, entonces la solución de atravesar el NAT es simple. El cliente debe descubrir como su IP:Puerto aparece para el mundo y entonces debe colocar esta información en los mensajes SDP en vez de la información de su IP:Puerto interno. Existen dos métodos para que un cliente pueda determinar la dirección pública mapeada para el IP:Puerto. El primero es preguntar al ruteador con NAT, el segundo es preguntar a alguien fuera de NAT en la red pública.

Una configuración incorrecta del firewall causará que las llamadas realizadas a través de un proveedor de VoIP o extensiones remotas, no tengan audio o tengan audio en un solo sentido.

Las reglas del firewall para permitir el tráfico SIP y RTP son las siguientes ver tabla II.I:

```
iptables –I INPUT 1 –p udp –m udp –dport 5060 –j ACCEPT
iptables –I INPUT 4 –p udp –m udp –dport 10000:20000 –j
ACCEPT
```

Tabla II.I: Reglas del firewall

2.9. Proxy SIP y RTP

La mayoría de las LAN corporativas se conectan a Internet a través de un Proxy. Un Proxy es un software usado para proteger la LAN de varios tipos de ataques y accesos no autorizados. Algunas veces son usados para prevenir que usuarios accedan a ciertos recursos de Internet. Un Proxy puede ser pensado como una puerta de un solo sentido que permite paquetes salientes de la LAN hacia Internet, pero bloquea paquetes entrantes desde Internet a menos que estos sean respuestas a requerimientos. Solo algunos ciertos tipos de requests desde Internet serán permitidos a pasar a la LAN, puede ser http request hacia el Server Web corporativo.

El Proxy hace esto manteniendo un rastro de las conexiones TCP abiertas y también presentan un desafió para las sesiones SIP. Debido a que SIP puede usar UDP y un puerto bien conocido, configurando el firewall para dejar pasar SIP no es muy dificultoso. Sin embargo esto no ayuda a la sesión del audio, la cual usa RTP sobre UDP en varios puertos que serán bloqueados por el firewall.

Un Proxy necesita entender SIP, para leer el contenido del INVITE y 200 OK response, y así extraer las direcciones IP y los números de puertos desde la descripción de la

Sesión SPD y abrir esos puertos para permitir que pase ese tráfico. Esos puertos pueden ser cerrados cuando un BYE es enviado o un temer expire.

El NAT también puede causar serios problemas para SIP. Nat es básicamente usado para conservar direcciones IPv4. Es usado en un router o firewall que provee la única conexión de la LAN a Internet. NAT permite a IPs privadas ser usadas internamente dentro de la LAN. Cuando un paquete es enviado desde la LAN hacia Internet, el NAT cambia la IP privada en el heather del datagrama por una IP pública. Esto significa que cada nodo en la red no tiene una dirección pública (única). Respuestas desde Internet son traducidas hacia atrás a las IP privadas. Sin embargo NAT no es completamente transparente para niveles superiores.

Debido a que las respuestas en SIP son ruteadas usando el header "Via", es decir un dispositivo detrás de un NAT, se colocará su no ruteable ip privada en el campo Via de los mensajes que origina. Cuando los requerimientos son enviados al exterior de la LAN, los encabezados IP y UDP serán reescritos con una IP pública y otro puerto aleatorio. El NAT mantendrá un registro de las IPs privadas, públicas y los puertos. Sin embargo, la dirección IP en el mensaje SIP, así como también los campos Via y Contact o las IPs en el campo SPD no serán reescritas y no serán ruteables.

Para parcialmente resolver este problema, SIP tiene un mecanismo para detectar la presencia de NAT. Cada Proxy o Agente de Usuario que recibe un request chequea la IP recibida con la IP en el campo Via. Si las direcciones son diferentes, entonces hay un NAT entre ellos. El no ruteable Via header es arreglado agregándole un received tag que contiene la IP pública con la que se recibió el paquete. Fuera del NAT, la respuesta es ruteada usando la IP recibida. Dentro del NAT, la dirección del campo "Via" es usada. Pero esto no resuelve el problema de la transmisión del audio/video.

Para solucionar los problemas del NAT y del tráfico RTP para establecer la transmisión del audio se puede utilizar diferentes soluciones:

- OpenSER
- Media RTP
- Servidor STUN
- Siproxd

2.9.1. OpenSER

SER (SIP Express Router) ha sido uno de los precursores de este cambio y del éxito del SIP. Muchos nuevos productos se han basado en este software para crear nuevos servicios en torno al protocolo SIP, como SipFoundry u OpenSER. Este último está ganando amplia popularidad en el mercado, considerándose una evolución del SER llevada a liderar el mercado.

OpenSER es un excelente servidor SIP cuya arquitectura flexible acoge una amplia colección de extensiones. El servidor implementa un proxy, herramientas de registro, redirección y por fin, localización de servicios SIP/VoIP. Un servidor SIP es un elemento primordial en la constitución de una red VoIP basada en el estándar SIP, un elemento de control de sesión entre usuarios y servicios. Por el Proxy SIP pasan todas las peticiones de un usuario, le permite acceder a la red, localizar a otros usuarios y establecer la comunicación VoIP SIP.

En resumidas cuentas, un servidor SIP es una especie de router, proxy y cortafuegos a la vez pero, en realidad es algo más complicado. OpenSER brinda soporte para los estándares UDP/TCP/TLS, ENUM, AAA vía base de datos, RADIUS, DIAMETER, puertas de enlace hacia SMS y XMPP, NAT transversal, etc.

A pesar de esta increíble carta de presentación, OpenSER no puede ser considerada una centralita tradicional. A diferencia de Asterisk, OpenSER no tiene prestaciones de “Media Gateway”, por lo que de por sí solo, no nos vale para sustituir a una centralita avanzada. Sin embargo, está considerado como el servidor SIP más avanzado del mercado ya que permite una gran escalabilidad y se trata de un entorno sumamente optimizado, basado en sistemas abiertos y posibilitando la conexión de miles de usuarios de forma concurrente.

Un servidor con un procesador de doble núcleo o de núcleo cuádruple y con memoria de 4 GHz debe ser capaz de procesar una media de 1000 llamadas simultáneas (Modo Proxy) en las llamadas SIP/H.323

Éstas características hacen de OpenSER el sistema habitual para los proveedores de VoIP del mercado, o para grandes multinacionales. Al estar basado en sistemas abiertos, es habitual encontrar proveedores de VoIP que utilicen OpenSER como corazón del sistema, integrándolo con sistemas como Asterisk para interactuar con la red PSTN o efectuar labores de Contestador Automático.

Características:

- Modular
- Escalable
- Proxy SIP (registrar, localizar, redireccionar)
- Transacción de estado
- SIP servidor de redirección
- SIP agente de presencia
- SIP back-to-back "de agente de usuario
- SIP servidor de mensajería instantánea
- SIP to SMS gateway (bidireccional)
- SIP to XMPP gateway for presence and IM (bidireccional)
- NAT Traversal
- Soporta J2EE y Perl

2.9.2. Media RTP

El RTP Proxy SIP es un software de alto rendimiento del servidor proxy de RTP que pueden trabajar junto con el SIP Express Router (SER).

El objetivo principal de RTP Proxy originalmente es permitir que la comunicación entre los agentes de usuario SIP detrás de NAT sea posible. Varios casos existen cuando la comunicación directa de extremo a extremo no es posible y RTP retransmite a través de otro host. El RTP Proxy se puede utilizar para configurar un host de retransmisión.

Más tarde se hizo evidente que hay muchos otros usos posibles para este software. Puede ser utilizado en combinación con elementos de señalización

(SIP Proxy o B2BUA SIP) para la construcción de complejas redes de VoIP, optimizar el flujo de tráfico, recopilar información de calidad de voz y así sucesivamente. Se pueden realizar varias funciones adicionales de RTP, incluyendo grabación de llamadas, jugando anuncios pre-codificado, copia de flujo en tiempo real y reformulación RTP carga útil.

El Proxy RTP apoya algunas características avanzadas, tales como el modo de control remoto, permitiendo la construcción de redes escalables distribuidos SIP VoIP.

El módulo nathelper incluidos en el SIP Express Router (SER), Elastix o Kamailio así SIP B2BUA permiten el uso de múltiples instancias RTP Proxy que se ejecutan en máquinas remotas para tolerancia a fallos y efectos de equilibrio de carga.

RTP Proxy fue desarrollado por Maxim Sobolev y ahora está siendo activamente mantenida por el Software Sippy, Inc.

¿Por qué RTP Proxy?

Paquetes tradicionales de VoIP PBX actúan como un hub, fijando los medios de comunicación y señalización en un solo lugar. Tal lugar a menudo se convierte en cuello de botella o el punto único de fallo.

2.9.3. Servidor STUN

Un servidor STUN permite a los clientes NAT (tal como computadores detrás de un firewall), configurar llamadas telefónicas a un proveedor VoIP alojado afuera de su red local.

En una llamada de VoIP con SIP los usuarios suelen encontrar problemas, como audio unidireccional o fallos al intentar registrarse con un proveedor de VoIP o con una centralita IP que no esté en la misma red. Esto suele deberse a cómo funcionan los protocolos SIP cuando las entidades están detrás de un router o firewall (escenarios bastante usuales). STUN ayuda a resolver esos problemas.

STUN es un protocolo de red del tipo cliente/servidor que tiene como principal objetivo permitir a los clientes encontrar sus direcciones públicas, el tipo de NAT del cual ellos están atrás y el puerto Internet asociado por el NAT con el puerto local específico. Esta información es usada para configurar comunicación UDP entre el cliente y el proveedor VOIP para así establecer una llamada. El protocolo STUN está definido en el RFC 3489.

Además utiliza el puerto UDP 3478, sin embargo, el servidor indicará a los clientes que realicen pruebas en IP alternativas y también con los números de puertos (servidores STUN tienen 2 direcciones IP). El RFC indica que este puerto e IP son arbitrarios.

STUN también se usa para refrescar las relaciones que establezca el NAT, en particular en los casos de *Conexión Restringida NAT* o *Conexión Restringida por puerto NAT*. En esos tipos de NAT, los puertos y direcciones internas se mapean a un puerto y dirección externa concreto; pero si no hubiese tráfico

durante un tiempo (configurable), se pierde dicho mapeo. En ese caso cuando el dispositivo interno intenta conectarse de nuevo con una entidad externa (que bien pudiera ser la misma a la que se conectó antes) usando la misma IP y puerto, el router le asignará el mapeo correcto, posiblemente un puerto e IP diferentes de los de antes.

STUN es usado principalmente por teléfonos o software VoIP. Éste incorpora un cliente que envía una petición a un servidor. El servidor STUN informa entonces al cliente de la IP pública de este último y qué puerto ha sido abierto por NAT para permitir el tráfico entrante a la red del cliente.

La respuesta permite además al cliente STUN determinar el tipo de NAT en uso, ya que diferentes tipos de NAT manejan los paquetes UDP entrantes de manera diferente. STUN soporta tres de los cuatro tipos principales de NAT existentes: *Full Conexión*, C*onexión Restringida* y C*onexión Restringida por Puerto*. No soporta, sin embargo, *NAT Simétrico*, también conocido como NAT bidireccional.

2.9.4. Siproxd

Siproxd es un proxy / demonio disfrazado para el protocolo SIP. Maneja los registros de los clientes SIP en una red IP privada y realiza la reescritura de los cuerpos de los mensajes SIP para hacer las conexiones posible a través de un firewall enmascarado (ver figura II.17). Se permite a los clientes SIP (como kphone, linphone) trabajar detrás de un cortafuegos o enmascaramiento de IP del router.

SIP es utilizado por Softphones y hardphones (Voz sobre IP) para iniciar la comunicación. Por sí solo, SIP no funciona a través de cortafuegos como enmascaramiento de los datos transferidos, contiene direcciones IP y números de puerto.

Existen los llamados servidores STUN que permiten a un cliente SIP averiguar su dirección IP pública visible y el uso de este lugar. Como inconveniente, y por lo general el servidor de seguridad está disfrazado dentro de un rango de puertos muy amplio el mismo que debe estar abierto para el tráfico entrante de RTP. El cliente debe ser compatible con SIP STUN (que en la mayoría de ellos sucede).

Siproxd utiliza otro enfoque (capa de aplicación proxy) y se coloca como proxy de salida entre los clientes SIP locales y el cliente remoto o registrador. Lo hace volver a escribir el tráfico SIP sobre la marcha y también incluye un proxy para el tráfico RTP de entrada y salida (los datos de audio real).

El rango de puertos que se utilizarán para la recepción de datos RTP es configurable, por lo que el servidor de seguridad sólo debe permitir el tráfico entrante para un rango de puertos pequeños. Un escenario normal se vería así (ver figura II.17):

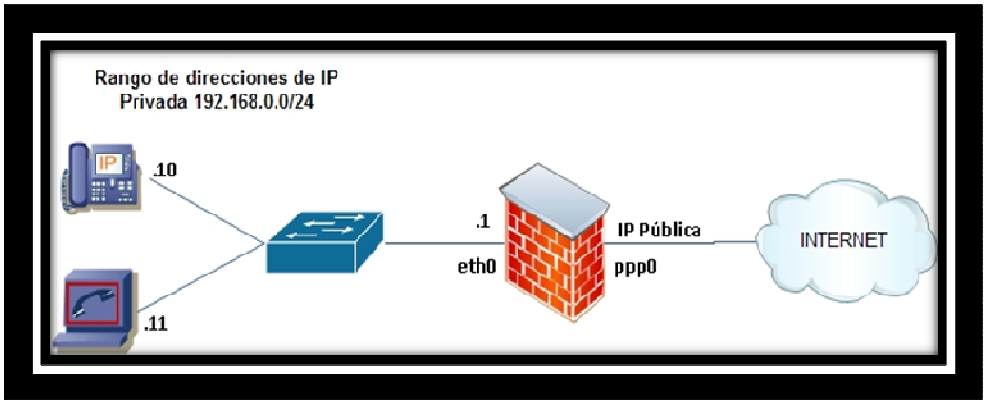


Figura II.17. Escenario Estándar Siproxd

2.10. Análisis comparativo para las soluciones de SIP y RTP

Para definir las métricas, se analizaron distintos puntos que pudieran permitir la comparación de las posibles soluciones para el NAT y el tráfico RTP mencionados en el capítulo anterior. De ellos se lograron destacar seis criterios para la definición de las métricas, objeto de este trabajo, los criterios escogidos fueron los siguientes: Llamadas simultáneas [10], Tipos de NAT soportados [10], Números de IPs públicas [8], Proxy SIP [6], Proxy RTP [6] y Complejidad [6]. La sumatoria de todos los puntajes da un total de 46, así que la mejor solución será la que más se aproxime a este valor.

- **Llamadas Simultáneas**

  Una llamada simultánea es el número de llamadas que se puede realizar al mismo tiempo en el servidor de registro, se califica con el valor de [10] cuando una de las soluciones para el NAT y el RTP soporte la mayor cantidad de llamadas simultáneas. En este caso, Siproxd al soportar 300000 llamadas simultáneas obtiene un puntaje de [10] y el Openser un puntaje de [0.33] al soportar 1000 llamadas, mientras que el Media RTP al igual que el Servidor STUN tienen un puntaje de [0] ya que no soportan llamadas simultáneas.

- **Tipos de NAT**

  Existen 4 tipos de NAT. Los routers vienen configurados con uno de estos 4 tipos: Full Conexión, Conexión restringida, Conexión restringida por puerto y NAT Simétrico. De los 4 tipos de NAT existente, la solución que mayor número de NATs soporte tendrá un valor de [10]. En este caso Siproxd y Openser obtienen un valor de [10] al soportar los 4 tipos de NATs, el servidor STUN [7.5] al soportar 3 de los 4 tipos de NATs y el Media RTP obtiene [0] ya q no soporta ningún tipo de NAT.

- **Números de IPs Públicas**

  Las IPs públicas son direcciones que se pueden utilizar en internet, para solucionar el problema del NAT y el tráfico RTP del protocolo SIP, se puede ocupar una o dos IPs públicas y la solución que ocupe una IP pública obtendrá un valor de [8] como es el caso del Servidor STUN y del Siproxd. Mientras que el Openser y el Media RTP reciben un puntaje de [4] al utilizar 2 direcciones IP públicas.

- **Proxy SIP**

  Un Proxy SIP es una aplicación que permite que cualquier usuario SIP envíe un comando a otro usuario SIP. Si una de las posibles soluciones es un proxy SIP se calificará con un valor de [6] tal es el caso del Openser y del Siproxd, mientras q el Media RTP y el Servidor STUN reciben un puntaje de [0] por no ser un Proxy SIP.

- **Proxy RTP**

  Originalmente el Proxy RTP permite que la comunicación entre los agentes de usuario SIP detrás de NAT sea posible. Si una de las soluciones es un Proxy RTP su valor será de [6], tal es el caso del Media RTP y del Siproxd, mientras que el Openser y el Servidor STUN reciben un puntaje de [0] por no ser un Proxy RTP.

- **Complejidad**

  Se medirá el nivel de dificultad en la configuración de cada uno de las posibles soluciones, se obtendrá el valor de [6] si se trata de una configuración sencilla, como es el caso del Servidor STUN, en el caso del Media RTP y del Siproxd tienen un puntaje de [4] porque la configuración es algo avanzada pero no tan compleja, mientras que el Openser recibe un puntaje de [1] porque la configuración es complicada.

La tabla de comparación para elegir el software que se va utilizar para solucionar los problemas del NAT y RTP ver la tabla II.II:

	OpenSER	**Media RTP**	**Servidor STUN**	**Siproxd**
Llamadas Simultáneas	0.33	0	0	10
Tipos de NAT	10	0	7.5	10
Utiliza una IP pública	4	4	8	8
Proxy SIP	6	0	0	6
Proxy RTP	0	6	0	6
Complejidad	1	4	6	4
Total	**21.33/46**	**14/46**	**21.5/46**	**44/46**

Tabla II.II: Comparación de las posibles soluciones

El software elegido para solucionar los problemas del SIP cuando la PBX y los clientes se encuentran detrás de Firewalls y NAT es SIPROXD, puesto que cumple con la mayoría de los puntos analizados, puede soportar alrededor de 30000 llamadas simultaneas, soporta los cuatro tipos de NAT, utiliza una IP publica, es un proxy SIP, es un proxy RTP y su configuración no es tan compleja obteniendo un puntaje de 44/46.

OpenSER también sería una buena elección ya que soporta 1000 llamadas simultáneas, también resuelve los cuatro tipos de NAT, el problema que tiene es que solo es un Proxy SIP y necesita del Media RTP que es un Proxy RTP, al utilizar Media RTP por separado se necesita dos IPs públicas, también su configuración es demasiado compleja.

2.11. Utilización de Siproxd

2.11.1. Prerrequisitos de Siproxd

El sistema operativo:

- Linux (debería funcionar con cualquier núcleo)
- FreeBSD
- Solaris

Paquetes adicionales requeridos:

- Paquete Libosip2

2.11.2. Configuración de Siproxd

El archivo de configuración de Siproxd puede estar en las siguientes ubicaciones dependiendo la distribución de Linux que se esté ocupando:

- *$HOME/.siproxdrc*

- *<buildingprefix>/etc/siproxd.conf*
- */etc/siproxd.conf*
- */usr/etc/siproxd.conf*
- */usr/local/etc/siproxd.conf*

La siguiente es una lista de las directivas que existen. Tener en cuenta que los valores de cadena no debe contener espacios o tabuladores. Para empezar siproxd se debe cambiar la definición de interfaz para las interfaces de red entrante y saliente *(if_inbound y if_outbound).*

Definir las interfaces para la red de entrada (red local donde está conectado el cliente SIP, esta red normalmente utiliza direcciones IP de los rangos de IP privadas como 192.168.xx) y la red de salida (la conexión a Internet, Normalmente esta interfaz tiene una IP pública asignada por el proveedor).

if_inbound = eth0

if_outbound = ppp0

Por lo general, sólo las directivas if*_inbound y if_outbound* se utiliza, La directiva *host_outbound* entra en juego cuando se ejecuta siproxd "frente a un router NAT. “

Host_outbound = <my_public_ip_address>

En las listas de control de acceso se permite que las redes se puedan registrar en el Siproxd y a su vez permitir el paso de trafico SIP. Estas son listas separadas de la forma <IP> / <máscara>, teniendo en cuenta que no

se permiten espacios dentro de la lista (el analizador de archivos de configuración no puede manejar espacios).

 # Hosts_allow_reg = 192.168.1.0/24, 192.168.2.0/24

 # Hosts_allow_sip = 123.45.0.0/16, 123.46.0.0/16

 # Hosts_deny_sip = 10.0.0.0 /8, 11.0.0.0 / 8

Puerto para escuchar los mensajes entrantes SIP. 5060 es generalmente la opción correcta, no cambiar a menos que haya una razón para hacerlo.

sip_listen_port = 5060

Por lo general Siproxd funciona como demonio que es la opción correcta. Si desea que Siproxd no corra como demonio y que funcione en primer plano se debe ubicar daemon = 0

daemon = 1

Los logs de Siproxd utiliza syslog () lo cual facilita cuando se ejecuta como un demonio. Esta configuración controla la cantidad de registro

- *0 - DEBUGs, INFOs, WARNINGs and ERRORs*
- *1 - INFOs, WARNINGs and ERRORs*
- *2 - WARNINGs and ERRORs*
- *3 - only ERRORs*
- *4 - absolutely nothing*

Si siproxd se inicia como root, puede cambiar los privilegios de root y su ID de usuario en el inicio.

user = nobody

 # chrootjail = /var/lib/siproxd/

Activar / desactivar el proxy RTP. Esto siempre debe estar habilitado.

Este es uno de los puntos más importantes en la configuración ya que junto al proxy sip y al proxy rtp se puede solucionar el problema del SIP a través de firewalls

rtp_proxy_enable = 1

Rango de puertos (UDP) que siproxd utilizará para el tráfico RTP entrantes y salientes. Un servidor de seguridad debe estar configurado para permitir el tráfico desde y hacia los puertos UDP. Por defecto, el rango de 7070 hasta el 7089 se utiliza. Esto permite hasta 10 llamadas simultáneas (dos puertos por llamada). Si necesita más llamadas simultáneas, aumentar la gama.

rtp_port_low = 7070

rtp_port_high = 7089

Tiempo de espera para un flujo RTP. Si para el número especificado de segundos no hay datos que se transmitan, se considera muerto y finalizara la llamada

rtp_timeout = 300

Si siproxd se utiliza como servidor de registro y autenticación, se define la siguiente directiva proxy_auth_realm, los clientes se verán obligados a autenticarse con el proxy (sólo para el registro). Para deshabilitar la autenticación, simplemente comentar esta línea. Por defecto está desactivada.

 # Proxy_auth_realm = Authentication_Realm

 # Proxy_auth_passwd = alguna_contraseña

Siproxd por sí mismo puede enviar todo el tráfico a otro proxy de salida. Se puede utilizar esta función para "cadena" proxies siproxd múltiples si tiene varios servidores de seguridad disfrazado de cruzar. Normalmente con discapacidad.

2.11.3. Escenarios de configuración

2.11.3.1. Clientes detrás del Router que ejecuta Siproxd

Siproxd se está ejecutando en la misma máquina que el firewall la cual se encarga también del ruteo, como se puede observar en la figura II.18.

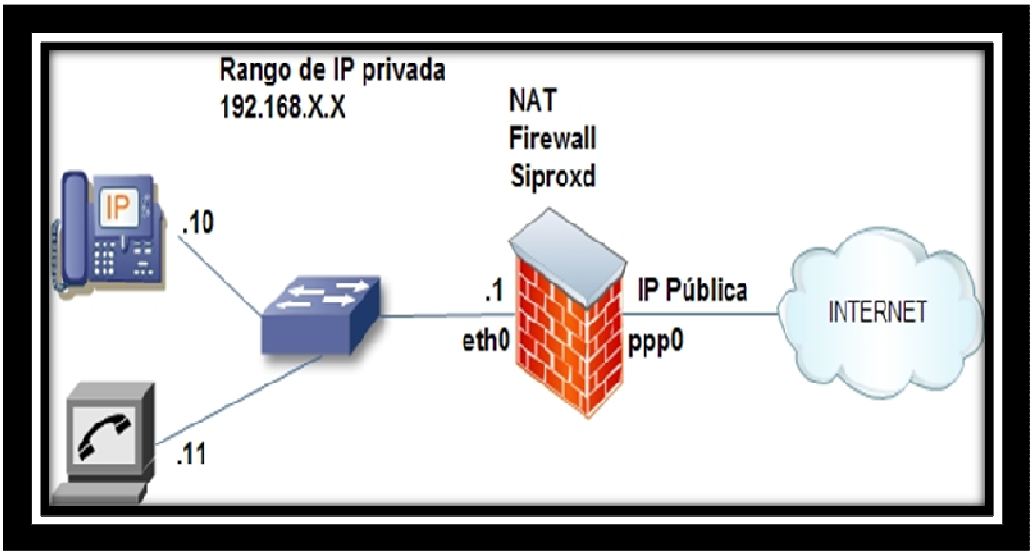


Figura II.18: Clientes detrás del Router que ejecuta Siproxd

El archivo de configuración del demonio Siproxd en este caso el siproxd.conf que se encuentra en la ubicación /etc/Siproxd.con debe quedar como se muestra en la tabla II.III:

```
if_inbound = eth0
if_outbound = ppp0
hosts_allow_reg = 192.168.0.0/24
sip_listen_port = 5060
daemonize = 1
silence_log = 1
user = siproxd
registration_file = /var/lib/siproxd_registrations
pid_file = /var/run/siproxd/siproxd.pid
rtp_proxy_enable = 1
rtp_port_low = 7070
rtp_port_high = 7089
rtp_timeout = 300
default_expires = 600
debug_level = 0
debug_port = 0
```

Tabla II.III. Archivo siproxd.conf en Escenario Estándar

Configuración de Iptables para el firewall, obsérvese tabla II.IV:

```
# Permitir la entrada de tráfico SIP y RTP
iptables -A INPUT -m udp -p udp -i ppp0 --dport 5060 -j
ACCEPT
iptables -A INPUT -m udp -p udp -i ppp0 --dport 7070:7089 -j
ACCEPT
```

Tabla II.IV: Configuración de Iptables en Escenario Estándar

Configuración del Teléfono IP, tal como se muestra en la tabla II.V:

```
IP Address: 192.168.0.10
Subnet Mask: 255.255.255.0
Default Router: 192.168.0.1
DNS Server 1: <Servidor DNS del proveedor de Internet>
SIP Server: <IP Publica de la PBX>
Outbound Proxy: 192.168.0.1
SIP User ID: 2500
Authenticate ID: 2500
Authenticate Passwd: ****
Name: Your Name Here
Use DNS SRV: no
User ID is phone #: no
Sip Registration: yes
Unregister on reboot:no
Register expiration: 60
Early Dial: no
local SIP port: 5060
local RTP port: 5004
Use random port: yes
NAT traversal: no
Use NAT IP: <empty>
Subscribe for MWI: No
Send DTMF: via RTP (RFC2833)
```

Tabla II.V. Configuración del Teléfono IP en Escenario Estándar

2.11.3.2. Siproxd ejecutándose detrás de un Router NAT

Siproxd se está ejecutando en la máquina 192.168.0.2. El router NAT (por ejemplo un router ADSL que no puede ejecutar aplicaciones de usuario), ver figura II.19:

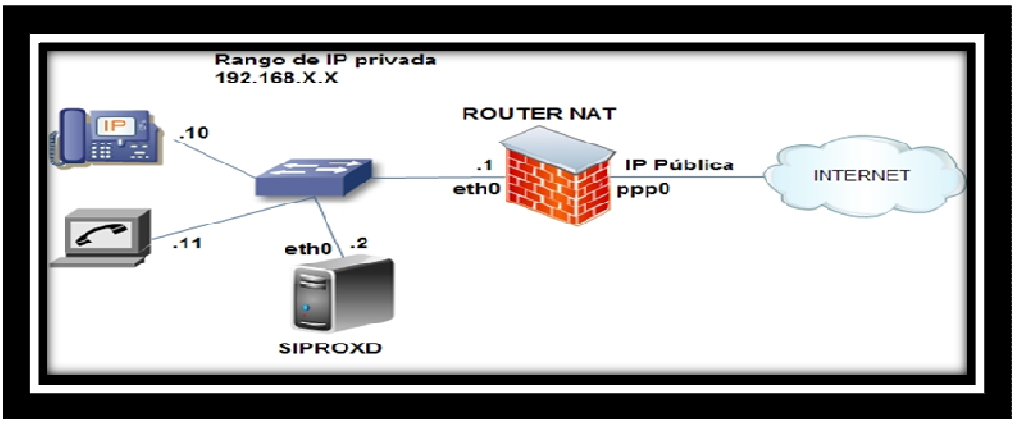


Figura II.19. Siproxd ejecutándose detrás de un Router NAT

El archivo de configuración del siproxd.conf debe quedar como se muestra en la tabla II.VI:

```
if_inbound = eth0
if_outbound = eth0
host_outbound = <IP Publica>
hosts_allow_reg = 192.168.0.0/24
sip_listen_port = 5060
daemonize = 1
silence_log = 1
user = siproxd
registration_file = /var/lib/siproxd_registrations
pid_file = /var/run/siproxd/siproxd.pid
rtp_proxy_enable = 1
rtp_port_low = 7070
rtp_port_high = 7089
rtp_timeout = 300
default_expires = 600
debug_level = 0
debug_port = 0
```

Tabla II.VI. Siproxd está detrás de un router NAT

Configuración de forwarding para el Router NAT, ver tabla II.VII:

```
forward all incoming traffic on 5060/udp to 192.168.0.2
forward all incoming traffic from 7070 – 7089 udp to
192.168.0.2
```

Tabla II.VII. Forwarding para el Router NAT

Configuración del Teléfono IP, tal como se muestra en la tabla II.VIII:

```
IP Address: 10.0.0.10
Subnet Mask: 255.255.255.0
Default Router: 192.168.0.1
DNS Server 1: <Servidor DNS del proveedor de Internet>
SIP Server: <IP Publica de la PBX>
Outbound Proxy: 192.168.0.2
SIP User ID: 2500
Authenticate ID: 2500
Authenticate Passwd: ****
Name: Your Name Here
Use DNS SRV: no
User ID is phone #: no
Sip Registration: yes
Unregister on reboot:no
Register expiration: 60
Early Dial: no
local SIP port: 5060
local RTP port: 5004
Use random port: yes
NAT traversal: no
Use NAT IP: <empty>
Subscribe for MWI: No
Send DTMF: via RTP (RFC2833)
```

Tabla II.VIII. Configuración del teléfono IP

2.11.3.3. Proxy Transparente

Se puede tener un teléfono SIP que no permite la especificación de un proxy de salida. Si siproxd se está ejecutando en el router NAT, la configuración siguiente hará que se transforme en un proxy transparente. El firewall redirige los mensajes salientes de SIP a siproxd, sin embargo, el cliente local no es consciente de ello, ver tabla II.IX:

```
if_inbound = eth0
if_outbound = ppp0
hosts_allow_reg = 192.168.0.0/24
sip_listen_port = 5060
daemonize = 1
silence_log = 1
user = siproxd
registration_file = /var/lib/siproxd_registrations
pid_file = /var/run/siproxd/siproxd.pid
rtp_proxy_enable = 1
rtp_port_low = 7010
rtp_port_high = 7019
rtp_timeout = 300
default_expires = 600
debug_level = 0
debug_port = 0
```

Tabla II.IX. Configuración de Siproxd como Proxy Transparente

Configuración de iptables para el proxy transparente, obsérvese tabla II.X:

```
# redirect outgoing SIP traffic to siproxd (myself)
iptables -t nat -A PREROUTING -m udp -p udp -i eth0 \
--destination-port 5060 -j REDIRECT
# allow incoming SIP and RTP traffic
iptables -A INPUT -m udp -p udp -i ppp0 --dport 5060 -j
ACCEPT
iptables -A INPUT -m udp -p udp -i ppp0 --dport 7070:7089 -j
ACCEPT
```

Tabla II.X. Configuración de Iptables para Proxy Transparente

2.11.3.4. PBX detrás de un Router NAT

Siproxd también puede ser utilizado para enmascarar un servidor Asterisk o Elaastix ver figura II.20. La PBX se registrará como un SIP UA (cliente) a un monitor de registro SIP. Como la PBX no permite especificar un proxy SIP de salida se utiliza la misma configuración de proxy transparente. El contexto de los valores de la configuración de la PBX probablemente se deba adaptar a sus necesidades.

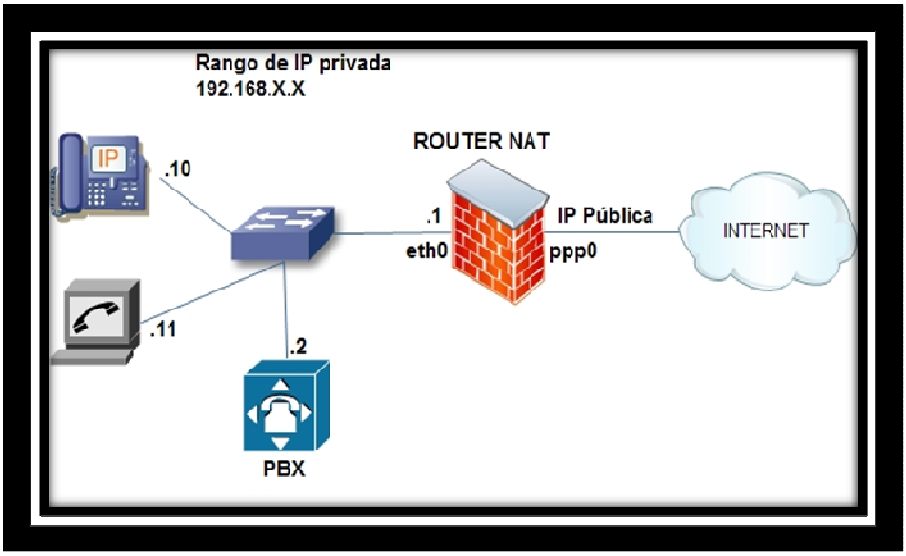


Figura II.20. PBX detrás de un Router NAT

El archivo de configuración del siproxd.conf debe quedar como se muestra en la tabla II.XI:

```
if_inbound = eth0
if_outbound = ppp0
hosts_allow_reg = 192.168.0.0/24
sip_listen_port = 5060
daemonize = 1
silence_log = 1
user = siproxd
registration_file = /var/lib/siproxd_registrations
pid_file = /var/run/siproxd/siproxd.pid
rtp_proxy_enable = 1
rtp_port_low = 7070
rtp_port_high = 7089
rtp_timeout = 300
default_expires = 600
debug_level = 0
debug_port = 0
```

Tabla II.XI. Siproxd.conf

Configuración de Iptables para el proxy transparente, obsérvese tabla II.XII:

```
# redirect outgoing SIP traffic to siproxd (myself)
iptables -t nat -A PREROUTING -m udp -p udp -i eth0 \
--source 192.168.0.2 --destination-port 5060 -j REDIRECT
# allow incoming SIP and RTP traffic
iptables -A INPUT -m udp -p udp -i ppp0 --dport 5060 -j
ACCEPT
iptables -A INPUT -m udp -p udp -i ppp0 --dport 7070:7080 -j
ACCEPT
```

Tabla II.XII. Configuración de Iptables para PBX detrás de Router NAT

Configuración de Asterisk o Elastix del archivo sip.conf (parte relacionadas con el SIP), ver tabla II.XIII:

```
IP Address: 192.168.0.10
Subnet Mask: 255.255.255.0
Default Router: 192.168.0.1
DNS Server 1: <Servidor DNS del proveedor de Internet>
SIP Server: <IP Publica de la PBX>
Outbound Proxy: 192.168.0.1
SIP User ID: 2500
Authenticate ID: 2500
Authenticate Passwd: ****
Name: Your Name Here
Use DNS SRV: no
User ID is phone #: no
Sip Registration: yes
Unregister on reboot:no
Register expiration: 60
Early Dial: no
local SIP port: 5060
local RTP port: 5004
Use random port: yes
NAT traversal: no
Use NAT IP: <empty>
Subscribe for MWI: No
Send DTMF: via RTP (RFC2833)
```

Tabla II.XIII. sip.conf para PBX detrás del Router NAT

CAPÍTULO III

PROPUESTA DE DISEÑO E IMPLEMENTACIÓN

3.1. Introducción

En el presente capítulo se mostrará el diagrama de red sobre el cual se realizó la implementación, así como también la instalación, configuración detallada de Quagga (Router) y del software Siproxd sobre Ubuntu Server.

Además se muestran los diferentes escenarios de red, en los cuales se explicará donde está ejecutándose el Siproxd con sus respectivas configuraciones tanto de la aplicación como de las reglas del firewall para su funcionamiento óptimo.

3.2. Arquitectura de red

La plataforma sobre la cual funciona la red de voz sobre IP está conformada por un servidor principal que es la PBX, la cual se encuentra detrás de un firewall y sus clientes se sitúan tanto en la red de la PBX como en diferentes sucursales, cada una de las cuales tiene que pasar el NAT transversal a través de firewalls, en el diagrama son la oficina B y C. La conexión de los tres Routers simula el internet y se implemento sobre maquinas Linux. El diagrama de red propuesto se muestra en la figura III.21:

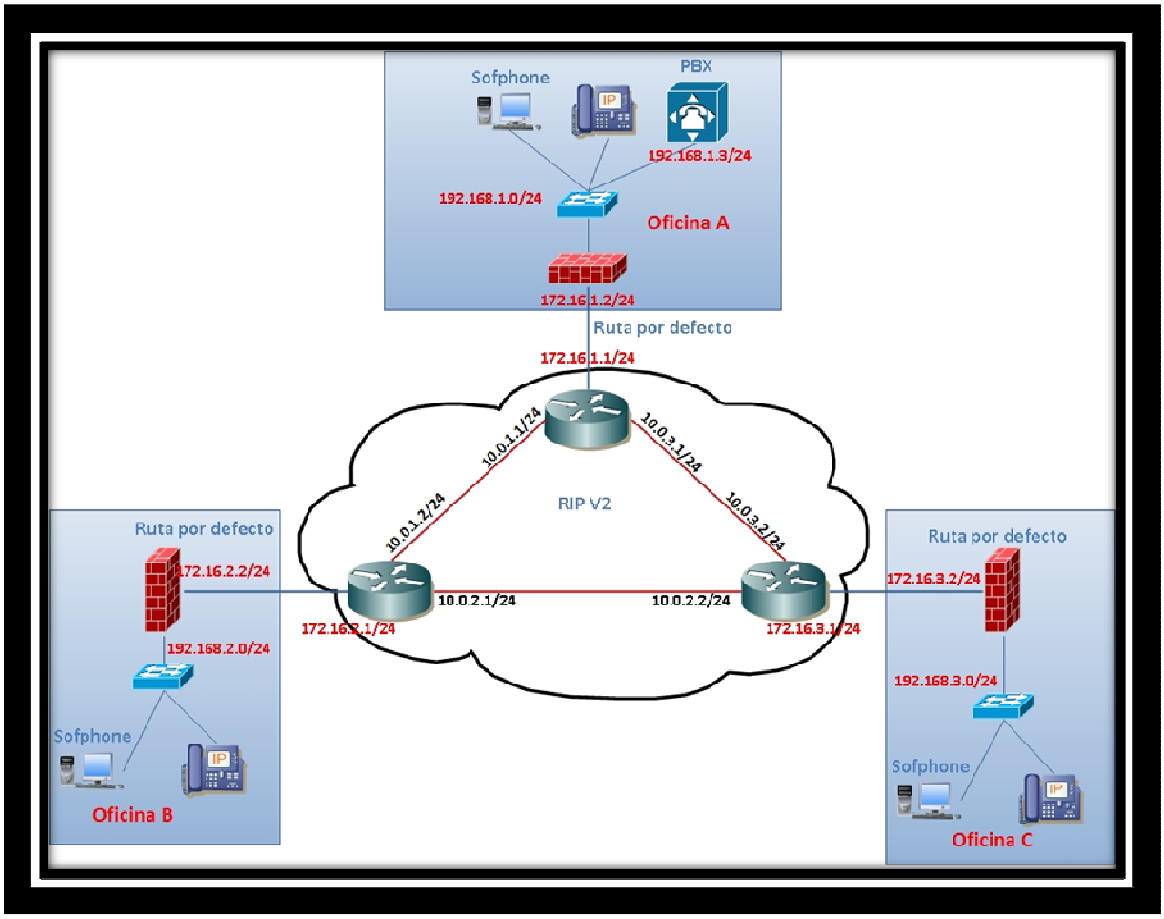


Figura III.21. Arquitectura de red

3.3. Sistema Operativo

Se eligió el sistema Operativo GNU/Linux Ubuntu Server 10.10 debido a su capacidad de reconocimiento en los repositorios del software a ser usado, como por ejemplo quagga encargado del ruteo y Siproxd que se encarga de resolver los problemas de SIP a través de firewalls. Para la PBX se eligió Elastix el mismo que se encuentra sobre un Centos.

3.4. Software

El software a utilizar para la realización del presente trabajo de tesis es el siguiente:

- Elastix: Servidor de VoIP
- Quagga: Es software libre para poder usar la familia de sistemas operativos Unix como enrutadores. Está diseñado especialmente para NetBSD, FreeBSD, Solaris y Linux.
- Siproxd: Es una aplicación proxy SIP para softphones o harphones basados en SIP que se encuentran detrás de un enmascaramiento IP (NAT). Incluye un conjunto de datos RTP, flujo de audio entrante y saliente y los datos de vídeo. Toda la configuración se realiza mediante un simple archivo de texto ASCII. Su objetivo principal es realizar NAT para el tráfico SIP y RTP.

.

3.5. Instalación y Configuración

3.5.1. Quagga

Es un software libre de enrutamiento avanzado. Proporciona todos los protocolos de encaminamiento (routing) basados en TCP/IP:

- RIP v1/v2 (Routing Information Protocol)
- OSPF v2/v3 (Open Shortest Path First)

- BGP -4 y BGP -4+ (Border Gateway Protocol)
- IS/IS (Intermediate system-to-intermediate system)

Además de soportar ipv4, también soporta ipv6. Posee una arquitectura avanzada que le proporciona una gran calidad y potencia.

3.5.1.1. Instalación Quagga

Se instala quagga usando apt-get Primero se ubica en modo root para no ingresar sudo constantemente, tal como se muestra en la tabla III.III:

```
Fdanny_24@Proxy:~$ sudo su
root@Proxy:/home/fdanny_24# apt-get install quagga
Leyendo lista de paquetes... Hecho
Creando Ã¡rbol de dependencias
Leyendo la informaciÃ³n de estado... Hecho
Paquetes sugeridos:
  snmpd
Se instalarÃ¡n los siguientes paquetes NUEVOS:
  quagga
0 actualizados, 1 se instalarÃ¡n, 0 para eliminar y 91 no
actualizados.
Necesito descargar 1547kB de archivos.
Se utilizarÃ¡n 5329kB de espacio de disco adicional despuÃ©s
de esta operaciÃ³n.
Des:1    http://ec.archive.ubuntu.com/ubuntu/    maverick/main
quagga i386 0.99.17-1 [1547kB]
Descargados 1547kB en 34s (45,0kB/s)
Preconfigurando paquetes ...
Seleccionando el paquete quagga previamente no seleccionado.
(Leyendo la base de datos ...  00%
42504 ficheros y directorios instalados actualmente.)
Desempaquetando quagga (de .../quagga_0.99.17-1_i386.deb) ...
Procesando disparadores para ureadahead ...
Procesando disparadores para man-db ...
Configurando quagga (0.99.17-1) ...
Loading capability module if not yet done.
Starting Quagga daemons (prio:10):.
Procesando disparadores para libc-bin ...
ldconfig deferred processing now taking place
root@Proxy:/home/fdanny_24#
```

Tabla III. XIV: Instalación de Quagga

Poner los archivos de configuración de quagga en su sitio, como se indica en la tabla III.XV:

```
root@Proxy:/home/fdanny_24#cd /usr/share/doc/quagga/examples/
root@Proxy:/usr/share/doc/quagga/examples# cp * /etc/quagga/
```

Tabla III.XV: Ubicar Archivos de Configuración de Quagga

Renombrar los archivos de configuración de ejemplo, para usar y activar Quagga con la configuración por defecto, ver tabla III.XVI:

```
root@Proxy:/usr/share/doc/quagga/examples# cd /etc/quagga/
root@Proxy:/etc/quagga# cp zebra.conf.sample zebra.conf
root@Proxy:/etc/quagga# cp ripd.conf.sample ripd.conf
```

Tabla III.XVI: Renombrar Archivos de Quagga

Configurar el archivo daemon que se encuentra en /etc/quagga para que se active zebra, y RIP, ya que en este caso se utilizara el protocolo RIP para el ruteo, se hace de igual modo para los demás protocolos. El archivo debe quedar como lo muestra la tabla III.XVII:

```
# This file tells the quagga package which daemons to start.
# Entries are in the format: <daemon>=(yes|no|priority)
#   0, "no"  = disabled
#   1, "yes" = highest priority
#   2 .. 10  = lower priorities
# Read /usr/share/doc/quagga/README.Debian for details.
# Sample configurations for these daemons can be found in
# /usr/share/doc/quagga/examples/.
# ATTENTION:
#
# When activation a daemon at the first time, a config file,
even if it is
# empty, has to be present *and* be owned by the user and
group "quagga", else
# the daemon will not be started by /etc/init.d/quagga. The
permissions should
# be u=rw,g=r,o=.
# When using "vtysh" such a config file is also needed. It
should be owned by
# group "quaggavty" and set to ug=rw,o= though. Check
/etc/pam.d/quagga, too.
zebra=yes
bgpd=no
ospfd=no
ospf6d=no
ripd=yes
ripngd=no
isisd=no
```

Tabla III.XVII: Archivo de Configuración Daemon de Quagga

Para que tengan efecto los cambios reiniciamos quagga. Tal como muestra la tabla III.XVIII:

```
root@Proxy:/etc/quagga# /etc/init.d/quagga restart
Stopping Quagga daemons (prio:0): (ripd) (zebra) (bgpd)
(ripngd) (ospfd) (ospf6d) (isisd).
Removing all routes made by zebra.
Loading capability module if not yet done.
Starting Quagga daemons (prio:10): zebra ripd.
```

Tabla III.XVIII: Reiniciar el Demonio Quagga

3.5.1.2. Configuración Quagga

Se puede acceder por separado con una interfaz interactiva a cada uno de los demonios. Para acceder a Zebra (**Password** por defecto zebra) obsérvese la tabla III.XIXI:

```
root@Proxy:~# telnet localhost 2601
Trying ::1...
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
Hello, this is Quagga (version 0.99.17).
Copyright 1996-2005 Kunihiro Ishiguro, et al.
User Access Verification
Password: zebra
Router>
```

Tabla III.XIX. Ingresar al Demonio Zebra

Para acceder al demonio de RIP (ripd) y configurarlo, ver tabla III.XX:

```
root@Proxy:~# telnet localhost 2602
Trying ::1...
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
Hello, this is Quagga (version 0.99.17).
Copyright 1996-2005 Kunihiro Ishiguro, et al.
User Access Verification
Password: zebra
ripd>
```

Tabla III.XX. Ingresar al Demonio RIP

3.5.1.2.1. Configuración Zebra

Es el demonio general. Los demonios que se encargan de los protocolos de enrutamiento son: ripd, ripngd, ospfd, ospf6d, bgpd. Podemos configurar Zebra editando el fichero de configuración zebra.conf o bien accediendo al demonio y por diversos comandos.

Antes que nada con el signo de interrogación (?) obtenemos una ayuda contextual de las posibilidades que se va clasificando, es decir, si ponemos "?" aparecen los posibles comandos que tenemos.

Se configurara el primer router con el Nombre de RouterA, la interfaz eth0 con la IP 172.16.1.1/24 que va hacer la dirección del ISP, la interfaz eth1 puede tener cualquier IP ya que es la parte que simula al Internet en este caso se ha colocado la dirección 10.0.1.1/24, la interfaz eth2 se configuro con la dirección 10.0.3.1/24como se observa los comandos son muy parecidos a los que ocupa Cisco, ver la tabla III.XXI:

```
Router> enable
Password:zebra
Router# configure terminal
Router(config)# hostname RouterA
RouterA(config)# interface eth0
RouterA(config-if)# ip address 172.16.1.1/24
RouterA(config-if)# no shutdown
RouterA(config-if)# exit
RouterA(config)# interface eth1
RouterA(config-if)# ip address 10.0.1.1/24
RouterA(config-if)# no shutdown
RouterA(config)# interface eth2
RouterA(config-if)# ip address 10.0.3.1/24
RouterA(config-if)# no shutdown
RouterA# show interface
Interface eth0 is up, line protocol detection is disabled
  index 2 metric 1 mtu 1500
  flags: <UP,BROADCAST,RUNNING,MULTICAST>
  HWaddr: 00:0c:29:64:14:de
  inet 172.16.1.1/24 broadcast 172.16.1.255
  inet6 fe80::20c:29ff:fe64:14de/64
Interface eth1 is up, line protocol detection is disabled
  index 3 metric 1 mtu 1500
  flags: <UP,BROADCAST,RUNNING,MULTICAST>
  HWaddr: 00:0c:29:64:14:e8
  inet 10.0.1.1/24 broadcast 10.0.1.255
  inet6 fe80::20c:29ff:fe64:16e9/64
Interface eth2 is up, line protocol detection is disabled
  index 3 metric 1 mtu 1500
  flags: <UP,BROADCAST,RUNNING,MULTICAST>
  HWaddr: 00:0c:29:64:16:e9
  inet 10.0.3.1/24 broadcast 10.0.3.255
  inet6 fe80::20c:29ff:fe64:14e8/64
Interface lo is up, line protocol detection is disabled
  index 1 metric 1 mtu 16436
  flags: <UP,LOOPBACK,RUNNING>
  inet 127.0.0.1/8
  inet6 ::1/128
RouterA# copy running-config startup-config
Configuration saved to /etc/quagga/zebra.conf
```

Tabla III.XXI. Configuración del Demonio Zebra en el RouterA

El archivo de configuración de quagga en el RouterA es el zebra.conf y es el siguiente, ver tabla III.XXII:

```
root@RouterA:/home/fdanny_24# cat /etc/quagga/zebra.conf
!
! Zebra configuration saved from vty
!   2011/01/25 12:14:52
!
hostname RouterA
password zebra
enable password zebra
!
interface eth0
 ip address 172.16.1.1/24
 ipv6 nd suppress-ra
!
interface eth1
 ip address 10.0.1.1/24
 ipv6 nd suppress-ra
!
interface eth2
 ip address 10.0.3.1/24
 ipv6 nd suppress-ra
!
interface lo
!
!
!
line vty
!
```

Tabla III.XXII. Archivo de Configuración de zebra.conf en el RouterA

Se configurara el segundo router con el Nombre de RouterB, la interfaz eth0 con la IP 172.16.2.1/24 que va hacer la dirección del ISP, la interfaz eth1 puede tener cualquier IP ya que es la parte que simula al Internet en este caso se ha colocado la dirección 10.0.1.2/24, la interfaz eth2 se configuro con la dirección 10.0.2.1/24, ver la tabla III.XXIII:

```
Router> enable
Password:zebra
Router# configure terminal
Router(config)# hostname RouterB
RouterB(config)# interface eth0
RouterB(config-if)# ip address 172.16.2.1/24
RouterB(config-if)# no shutdown
RouterB(config-if)# exit
RouterB(config)# interface eth1
RouterB(config-if)# ip address 10.0.1.2/24
RouterB(config-if)# no shutdown
RouterB(config)# interface eth2
RouterB(config-if)# ip address 10.0.2.1/24
RouterB(config-if)# no shutdown
RouterB# show interface
Interface eth0 is up, line protocol detection is disabled
  index 2 metric 1 mtu 1500
  flags: <UP,BROADCAST,RUNNING,MULTICAST>
  HWaddr: 00:0c:29:64:14:de
  inet 172.16.2.1/24 broadcast 172.16.2.255
  inet6 fe80::20c:29ff:fe64:14de/64
Interface eth1 is up, line protocol detection is disabled
  index 3 metric 1 mtu 1500
  flags: <UP,BROADCAST,RUNNING,MULTICAST>
  HWaddr: 00:0c:29:64:14:e8
  inet 10.0.1.2/24 broadcast 10.0.1.255
  inet6 fe80::20c:29ff:fe64:16e9/64
Interface eth2 is up, line protocol detection is disabled
  index 3 metric 1 mtu 1500
  flags: <UP,BROADCAST,RUNNING,MULTICAST>
  HWaddr: 00:0c:29:64:16:e9
  inet 10.0.2.1/24 broadcast 10.0.2.255
  inet6 fe80::20c:29ff:fe64:14e8/64
Interface lo is up, line protocol detection is disabled
  index 1 metric 1 mtu 16436
  flags: <UP,LOOPBACK,RUNNING>
  inet 127.0.0.1/8
  inet6 ::1/128
RouterB# copy running-config startup-config
Configuration saved to /etc/quagga/zebra.conf
```

Tabla III.XXIII. Configuración del Demonio Zebra en el RouterB

El archivo de configuración de quagga en el RouterB es el zebra.conf y es el siguiente, ver tabla III.XXIV:

```
root@RouterB:/home/fdanny_24# cat /etc/quagga/zebra.conf
!
! Zebra configuration saved from vty
!   2011/01/25 12:19:29
!
hostname RouterB
password zebra
enable password zebra
!
interface eth0
 ip address 172.16.2.1/24
 ipv6 nd suppress-ra
!
interface eth1
 ip address 10.0.1.2/24
 ipv6 nd suppress-ra
!
interface eth2
 ip address 10.0.2.1/24
 ipv6 nd suppress-ra
!
interface lo
!
!
!
line vty
!
```

Tabla III.XXIV. Archivo de Configuración de zebra.conf en el RouterB

Se configurara el tercer router con el Nombre de RouterC, la interfaz eth0 con la IP 172.16.3.1/24 que va hacer la dirección del ISP, la interfaz eth1 puede tener cualquier IP ya que es la parte que simula al Internet en este caso se ha colocado la dirección 10.0.3.2/24, la interfaz eth2 se configuró con la dirección 10.0.2.2/24, ver la tabla III.XXV:

```
Router> enable
Password:zebra
Router# configure terminal
Router(config)# hostname RouterC
RouterC(config)# interface eth0
RouterC(config-if)# ip address 172.16.3.1/24
RouterC(config-if)# no shutdown
RouterC(config-if)# exit
RouterC(config)# interface eth1
RouterC(config-if)# ip address 10.0.3.2/24
RouterC(config-if)# no shutdown
RouterC(config)# interface eth2
RouterC(config-if)# ip address 10.0.2.2/24
RouterC(config-if)# no shutdown
RouterC# show interface
Interface eth0 is up, line protocol detection is disabled
  index 2 metric 1 mtu 1500
  flags: <UP,BROADCAST,RUNNING,MULTICAST>
  HWaddr: 00:0c:29:64:14:de
  inet 172.16.3.1/24 broadcast 172.16.3.255
  inet6 fe80::20c:29ff:fe64:14de/64
Interface eth1 is up, line protocol detection is disabled
  index 3 metric 1 mtu 1500
  flags: <UP,BROADCAST,RUNNING,MULTICAST>
  HWaddr: 00:0c:29:64:14:e8
  inet 10.0.3.2/24 broadcast 10.0.3.255
  inet6 fe80::20c:29ff:fe64:16e9/64
Interface eth2 is up, line protocol detection is disabled
  index 3 metric 1 mtu 1500
  flags: <UP,BROADCAST,RUNNING,MULTICAST>
  HWaddr: 00:0c:29:64:16:e9
  inet 10.0.2.2/24 broadcast 10.0.2.255
  inet6 fe80::20c:29ff:fe64:14e8/64
Interface lo is up, line protocol detection is disabled
  index 1 metric 1 mtu 16436
  flags: <UP,LOOPBACK,RUNNING>
  inet 127.0.0.1/8
  inet6 ::1/128
RouterC# copy running-config startup-config
Configuration saved to /etc/quagga/zebra.conf
```

Tabla III.XXV. Configuración del Demonio Zebra en el RouterC

El archivo de configuración de quagga en el RouterC es el zebra.conf y es el siguiente, ver tabla III.XXVI:

```
root@RouterC:/home/fdanny_24# cat /etc/quagga/zebra.conf
!
! Zebra configuration saved from vty
!   2011/01/25 12:23:12
!
hostname RouterC
password zebra
enable password zebra
!
interface eth0
 ip address 172.16.3.1/24
 ipv6 nd suppress-ra
!
interface eth1
 ip address 10.0.3.2/24
 ipv6 nd suppress-ra
!
interface eth2
 ip address 10.0.2.2/24
 ipv6 nd suppress-ra
!
interface lo
!
!
!
line vty
!
```

Tabla III.XXVI. Archivo de Configuración de zebra.conf en el RouterC

3.5.1.2.2. Configuración RIP

RIP (Routing Information Protocol) es un protocolo de pasarela interior ampliamente desplegado. RIP fue desarrollado en la década de 1970 en los laboratorios Xerox como parte del protocolo de encaminamiento XNS. RIP

es un protocolo vector-distancia y está basado en el algoritmo Bellman-Ford.

Como protocolo de vector-distancia, un router funcionando con RIP envía actualizaciones a sus vecinos periódicamente, permitiendo la convergencia asi a una topología conocida. En cada actualización, la distancia a una red dada será comunicada a todos los demás routers vecinos del mismo. Para anunciar las redes mediante el protocolo RIP obsérvese la tabla III.XXVII:

```
RouterA> enable
Password:zebra
ripd# configure terminal
ripd(config)# router rip
ripd(config-router)# network 172.16.1.0/24
ripd(config-router)# network 10.0.1.0/24
ripd(config-router)# network 10.0.3.0/24
ripd(config-router)# network eth0
ripd(config-router)# network eth1
ripd(config-router)# network eth2
ripd(config-router)# exit
ripd# copy running-config startup-config
Configuration saved to /etc/quagga/ripd.conf
```

Tabla: III.XXVII. Configuración del Demonio RIP en RouterA

Para la configuración del RouterB ver Tabla III.XXVIII:

```
RouterB> enable
Password:zebra
ripd# configure terminal
ripd(config)# router rip
ripd(config-router)# network 172.16.2.0/24
ripd(config-router)# network 10.0.1.0/24
ripd(config-router)# network 10.0.2.0/24
ripd(config-router)# network eth0
ripd(config-router)# network eth1
ripd(config-router)# network eth2
ripd(config-router)# exit
ripd# copy running-config startup-config
Configuration saved to /etc/quagga/ripd.conf
```

Tabla III.XXVIII: Configuración del Demonio RIP en RouterB

Para la configuración del RouterC ver Tabla III.XXIX:

```
RouterC> enable
Password:zebra
ripd# configure terminal
ripd(config)# router rip
ripd(config-router)# network 172.16.3.0/24
ripd(config-router)# network 10.0.2.0/24
ripd(config-router)# network 10.0.3.0/24
ripd(config-router)# network eth0
ripd(config-router)# network eth1
ripd(config-router)# network eth2
ripd(config-router)# exit
ripd# copy running-config startup-config
Configuration saved to /etc/quagga/ripd.conf
```

Tabla III.XXIX: Configuración del Demonio RIP en RouterC

Para la implementación del ambiente de pruebas, hemos utilizado el protocolo de ruteo RIP pero también es posible utilizar otros protocolos de enrutamiento como son: OSPF, EIGRP, ISIS o rutas estáticas.

3.5.2. Siproxd

El software utilizado para resolver los problemas del SIP a través de firewalls y NAT es Siproxd, ya que es un proxy SIP y RTP.

3.5.2.1. Instalación de Siproxd

Se instala Siproxd usando apt-get install Siproxd. Se ubica en modo root para no ingresar sudo constantemente, tal como se muestra en la tabla III.XXX:

```
root@Proxy:~# apt-get install siproxd
Leyendo lista de paquetes... Hecho
Creando Ã¡rbol de dependencias
Leyendo la informaciÃ³n de estado... Hecho
Se instalarÃ¡n los siguientes paquetes extras:
  libosip2-4
Paquetes sugeridos:
  linphone kphone asterisk
Se instalarÃ¡n los siguientes paquetes NUEVOS:
  libosip2-4 siproxd
0 actualizados, 2 se instalarÃ¡n, 0 para eliminar y 91 no
actualizados.
Necesito descargar 563kB de archivos.
Se utilizarÃ¡n 1331kB de espacio de disco adicional despuÃ©s
de esta operaciÃ³n.
Â¿Desea continuar [S/n]?
0% [Trabajando]
Des:1    http://ec.archive.ubuntu.com/ubuntu/    maverick/main
libosip2-4 i386 3.3.0-1ubuntu1 [91,7kB]
Des:2  http://ec.archive.ubuntu.com/ubuntu/  maverick/universe
siproxd i386 1:0.7.2-1 [472kB]
Descargados 563kB en 13s (43,3kB/s)
Seleccionando   el   paquete   libosip2-4   previamente   no
seleccionado.
(Leyendo la base de datos ...  00%
42624 ficheros y directorios instalados actualmente.)
Desempaquetando    libosip2-4    (de    .../libosip2-4_3.3.0-
1ubuntu1_i386.deb) ...
Seleccionando el paquete siproxd previamente no seleccionado.
Desempaquetando siproxd (de .../siproxd_1%3a0.7.2-1_i386.deb)
...
Procesando disparadores para man-db ...
Procesando disparadores para ureadahead ...
ureadahead will be reprofiled on next reboot
Configurando libosip2-4 (3.3.0-1ubuntu1) ...
Configurando siproxd (1:0.7.2-1) ...
Adding system-user for siproxd
To enable siproxd, the sip proxy, modify /etc/default/siproxd
Procesando disparadores para libc-bin ...
ldconfig deferred processing now taking place
root@Proxy:~#
```

Tabla III.XXX. Instalación de Siproxd

3.5.2.2. Configuración del archivo Siproxd.conf

En el capitulo anterior se explicó los diferentes parámetros para la configuración del software Siproxd, con sus diferentes escenarios, los mismos que pueden variar según la arquitectura de la red.

Para el desarrollo de ésta tesis se utilizó tres firewalls, el firewall de la oficina A con IP 172.16.1.2, el firewall de la oficina B con IP 172.16.2.2 y el

firewall de la oficina C con la IP 172.16.3.2, cada uno de los cuales tienen instalado Siproxd. El archivo de configuración de Siproxd en el firewall A (172.16.1.2) es el siguiente, ver tabla III.XXXI:

```
if_inbound  = eth0
if_outbound = eth1
hosts_allow_reg = 192.168.1.0/24 #### IP de la red local
hosts_allow_sip = 0.0.0.0/0
sip_listen_port = 5060
daemonize = 1
silence_log = 1
user = siproxd
chrootjail = /var/lib/siproxd/
registration_file = /var/lib/siproxd/siproxd_registrations
autosave_registrations = 300
pid_file = /var/run/siproxd/siproxd.pid
rtp_proxy_enable = 1
rtp_port_low  = 7000
rtp_port_high = 20000
rtp_timeout = 300
rtp_dscp = 46
sip_dscp = 0
rtp_input_dejitter  = 0
rtp_output_dejitter = 0
default_expires = 600
debug_level =      0
debug_port = 0
plugindir=/usr/lib/siproxd/
load_plugin=plugin_logcall.la
plugin_demo_string =
This_is_a_string_passed_to_the_demo_plugin
plugin_shortdial_akey = *00
plugin_shortdial_entry = 17474743246
plugin_shortdial_entry = 17474745000
plugin_defaulttarget_log = 1
plugin_defaulttarget_target = sip:internal@dddd:port
plugin_fix_bogus_via_networks =
10.0.0.0/8,172.16.0.0/12,192.168.0.0/16
```

Tabla III.XXXI. Archivo Siproxd.conf en el firewall de la oficina A

El archivo de configuración de Siproxd en el firewall B (172.16.2.2) es el siguiente, ver tabla III.XXXII:

```
if_inbound  = eth0
if_outbound = eth1
hosts_allow_reg = 192.168.2.0/24  #### IP de la red local
sip_listen_port = 5060
daemonize = 1
silence_log = 1
user = siproxd
chrootjail = /var/lib/siproxd/
registration_file = /var/lib/siproxd/siproxd_registrations
autosave_registrations = 300
pid_file = /var/run/siproxd/siproxd.pid
rtp_proxy_enable = 1
rtp_port_low  = 7000
rtp_port_high = 20000
rtp_timeout = 300
rtp_dscp = 46
sip_dscp = 0
rtp_input_dejitter  = 0
rtp_output_dejitter = 0
default_expires = 600
debug_level =      0
debug_port = 0
plugindir=/usr/lib/siproxd/
load_plugin=plugin_logcall.la
plugin_demo_string =
```

```
This_is_a_string_passed_to_the_demo_plugin

plugin_shortdial_akey = *00

plugin_shortdial_entry = 17474743246

plugin_shortdial_entry = 17474745000

plugin_defaulttarget_log = 1

plugin_defaulttarget_target = sip:internal@dddd:port

plugin_fix_bogus_via_networks =
10.0.0.0/8,172.16.0.0/12,192.168.0.0/16
```

Tabla III.XXXII. Archivo siproxd.conf en el firewall de la oficina B

El archivo de configuración de Siproxd en el firewall C (172.16.3.2) es el siguiente, ver tabla III.XXXIII:

```
if_inbound  = eth0

if_outbound = eth1

hosts_allow_reg = 192.168.3.0/24  #### IP de la red local

sip_listen_port = 5060

daemonize = 1

silence_log = 1

user = siproxd

chrootjail = /var/lib/siproxd/

registration_file = /var/lib/siproxd/siproxd_registrations

autosave_registrations = 300

pid_file = /var/run/siproxd/siproxd.pid

rtp_proxy_enable = 1
```

```
rtp_port_low  = 7000

rtp_port_high = 20000

rtp_timeout = 300

rtp_dscp = 46

sip_dscp = 0

rtp_input_dejitter  = 0

rtp_output_dejitter = 0

default_expires = 600

debug_level =      0

debug_port = 0

plugindir=/usr/lib/siproxd/

load_plugin=plugin_logcall.la

plugin_demo_string =
This_is_a_string_passed_to_the_demo_plugin

plugin_shortdial_akey = *00

plugin_shortdial_entry = 17474743246

plugin_shortdial_entry = 17474745000

plugin_defaulttarget_log = 1

plugin_defaulttarget_target = sip:internal@dddd:port

plugin_fix_bogus_via_networks =
10.0.0.0/8,172.16.0.0/12,192.168.0.0/16
```

Tabla III.XXXIII. Archivo siproxd.conf en el firewall de la oficina C

3.5.3. Iptables

Son las reglas necesarias para configurar tanto el firewall y el NAT. Este es un punto importante ya que si las reglas del firewall no son configuradas correctamente la comunicación VoIP no se establecerá.

La configuración de las reglas del firewall de la oficina A son las siguientes, ver tabla III.XXXIV:

```
#!/bin/sh
iptables -F
iptables -X
iptables -Z
iptables -t nat -F

iptables -P INPUT ACCEPT
iptables -P OUTPUT ACCEPT
iptables -P FORWARD ACCEPT
iptables -t nat -P PREROUTING ACCEPT
iptables -t nat -P POSTROUTING ACCEPT

iptables -A INPUT -s 192.168.1.0/24 -i eth0 -j ACCEPT
iptables -t nat -A POSTROUTING -s 192.168.1.0/24 -o eth1 -j
SNAT --to 172.16.1.2

# redirect outgoing SIP traffic to siproxd (myself)
iptables -t nat -A PREROUTING -m udp -p udp -i eth0 --
destination-port 5060 -j REDIRECT

# allow incoming SIP and RTP traffic
iptables -A INPUT -m udp -p udp -i eth1 --dport 5060 -j
ACCEPT
iptables -A INPUT -m udp -p udp -i eth1 --dport 7000:20000 -j
ACCEPT

iptables -t nat -A PREROUTING -i eth1 -p tcp --dport 443 --j
DNAT --to 192.168.1.3:443

iptables -t nat -A PREROUTING -i eth1 -p udp -m udp --dport
7000:20000 --j DNAT --to-destination 192.168.1.3
iptables -t nat -A PREROUTING -i eth1 -p udp -m udp --dport
5060 --j DNAT --to-destination 192.168.1.3

echo 1 > /proc/sys/net/ipv4/ip_forward
```

Tabla III.XXXIV. Iptables del firewall de la Oficina A

Como se puede observar se está realizando un Nateo de la IP privada 192.168.1.0 a la IP publica 172.16.1.2. Las reglas del firewall están configuradas en modo de proxy transparente, el puerto 5060 UDP para el protocolo SIP y los puertos del 7000 al 20000 UDP para el flujo RTP. También

se realiza una redirección hacia la PBX (192.168.1.3) para administrar la página de Elastix a través del puerto 443.

La configuración de las reglas del firewall de la oficina B son las siguientes, ver tabla III.XXXV:

```
#!/bin/sh
iptables -F
iptables -X
iptables -Z
iptables -t nat -F

iptables -P INPUT ACCEPT
iptables -P OUTPUT ACCEPT
iptables -P FORWARD ACCEPT
iptables -t nat -P PREROUTING ACCEPT
iptables -t nat -P POSTROUTING ACCEPT

iptables -A INPUT -s 192.168.2.0/24 -i eth0 -j ACCEPT
iptables -t nat -A POSTROUTING -s 192.168.2.0/24 -o eth1 -j
SNAT --to 172.16.2.2

# redirect outgoing SIP traffic to siproxd (myself)
iptables -t nat -A PREROUTING -m udp -p udp -i eth0 --
destination-port 5060 -j REDIRECT

# allow incoming SIP and RTP traffic
iptables -A INPUT -m udp -p udp -i eth1 --dport 5060 -j
ACCEPT
iptables -A INPUT -m udp -p udp -i eth1 --dport 7070:7089 -j
ACCEPT

echo 1 > /proc/sys/net/ipv4/ip_forward
```

Tabla III.XXXV. Iptables del firewall de la Oficina B

Las reglas tanto para el firewall de la Oficina B y C son similares, la única diferencia es que no se está realizando una redirección de puertos a la PBX (192.168.1.3), también se encuentran configurados en modo de proxy transparente, debido a que no en todos los softphones se puede especificar el proxy SIP que van utilizar.

La configuración de las reglas del firewall de la oficina C son las siguientes, ver tabla III.XXXVI:

```
# !/bin/sh
iptables –F
iptables –X
iptables –Z
iptables –t nat –F

iptables –P INPUT ACCEPT
iptables –P OUTPUT ACCEPT
iptables –P FORWARD ACCEPT
iptables –t nat –P PREROUTING ACCEPT
iptables –t nat –P POSTROUTING ACCEPT

140ptables –A INPUT –s 192.168.3.0/24 –i eth0 –j ACCEPT
140ptables –t nat –A POSTROUTING –s 192.168.3.0/24 –o eth1 –j
SNAT –to 172.16.3.2

# redirect outgoing SIP traffic to 140ptable (myself)
iptables –t nat –A PREROUTING –m udp –p udp –i eth0 –
destination-port 5060 –j REDIRECT

# allow incoming SIP and RTP traffic
140ptables –A INPUT –m udp –p udp –i eth1 –dport 5060 –j
ACCEPT
140ptables –A INPUT –m udp –p udp –i eth1 –dport 7000:20000 -
j ACCEPT

echo 1 > /proc/sys/net/ipv4/ip_forward
```

Tabla III.XXXVI. Iptables del firewall de la Oficina C

3.5.4. Elastix

Para el correcto funcionamiento de la PBX (192.168.1.3) detrás de firewall de la oficina A se debe configurar ciertos archivos, que son el sip.conf, sip_nat.conf y el rtp.conf. Para la configuración del archivo sip.conf, ver la tabla III.XXXVII:

```
 [General] context= from –sip
Bindport=5060
Bindaddr=0.0.0.0
Srvlookup=yes
Disallow= all
language= es

nat=yes

canreinvite= no

insecure= very

relaxdtmf= yes

dtmfmode= info

externip= 172.16.1.2

localnet= 192.168.1.0/24
```

Tabla III.XXXVII: Archivo de configuración sip.conf de la PBX

El archivo sip.conf que permite a la PBX en este caso Elastix conocer su IP pública, esto es cuando las extensiones se encuentran registradas fuera de la red local, caso contrario ocupa la IP que se especifica en localnet.
La opción canreinvite = no es un punto importante ya que permite trabajar a la PBX como un proxy RTP.

Para la configuración del archivo sip_nat.conf, ver la tabla III.XXXVIII:

```
externip= 172.16.1.2

localnet= 192.168.1.0/24

qualify= yes

relaxdtmf= yes
```

Tabla III.XXXVIII: Archivo de configuración sip_nat.conf de la PBX

Es un archivo similar al sip.conf, únicamente es utilizado en sistemas Elastix, si se utiliza Asterisk no existe este archivo.

Para la configuración del archivo rtp.conf, ver la tabla III.XXXIX:

```
[General]
Rtpstart=7000
Rtpend=20000
```

Tabla III.XXIX: Archivo de configuración rtp.conf de la PBX

En este archivo se especifica los puertos RTP que utilizara la PBX, debe tener el mismo rango de puertos que en el archivo siproxd.conf

CAPÍTULO IV

PRUEBAS DE DESEMPEÑO

4.1. Introducción

Este capítulo consiste en comprobar que mediante la utilización de Gateways SIP se soluciona el problema del NAT y RTP cuando se utiliza el protocolo SIP. Se realizó pruebas, para determinar lo importante que es el Siproxd en VoIP cuando la PBX y los clientes se encuentran detrás de NAT y de firewalls.

A continuación se detallan los 4 cuatro tipos de pruebas realizados:

- Total Operatividad.
- PBX sin Siproxd.
- En el transcurso de una llamada se detiene el servicio del Siproxd.
- Sin ejecución del Siproxd en el Firewall.

Todas las pruebas se realizo en el siguiente Diagrama de red, ver figura IV.22:

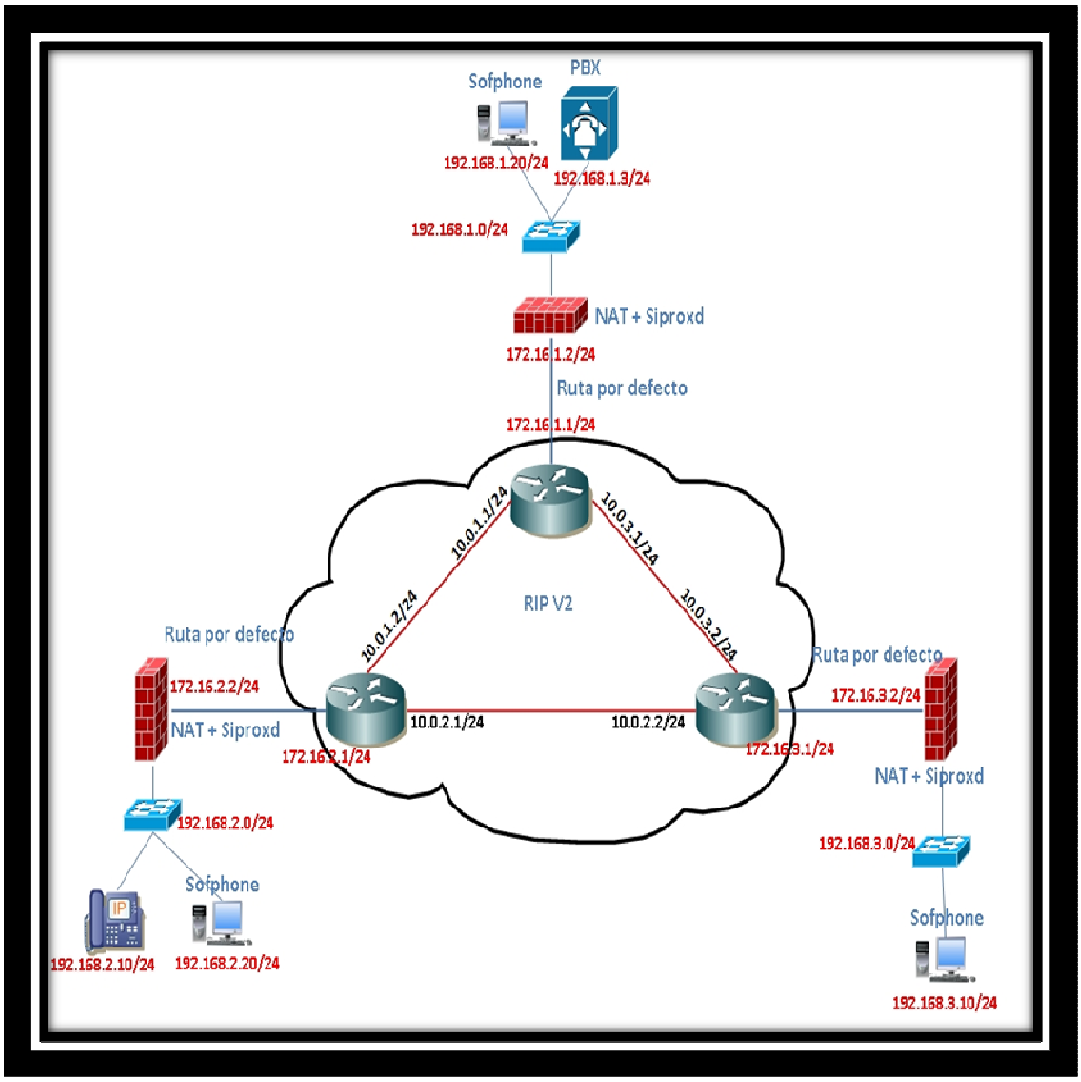


Figura IV.22. Diagrama de Pruebas

Como se muestra en la figura anterior se tiene tres oficinas la A, B y C

- La oficina A con dirección de red 192.168.1.0/24, la misma que contiene la PBX que es un servidor instalado Elastix (192.168.1.3) y también consta de un sofphone (192.168.1.20). Toda la red se encuentra protegida por un Firewall (eth0: 192.168.1.1 y la eth1:172.16.1.2) que se encuentra configurado sobre un Ubuntu Server, el mismo que hace NAT y tiene una ruta por defecto a la IP

172.16.1.1. El firewall tiene instalado Siproxd para que los equipos internos puedan llamar a las estaciones remotas en este caso con la Oficina B y C.

- La Oficina B con dirección de red 192.168.2.0/24, la oficina consta de un sofphone (192.168.2.20) y un teléfono IP (192.168.2.10). Toda la red se encuentra protegida por un Firewall (eth0: 192.168.2.1 y la eth1:172.16.2.2) que se encuentra configurado sobre un Ubuntu Server, el mismo que hace NAT y tiene una ruta por defecto a la IP 172.16.2.1. El firewall tiene instalado Siproxd para que los equipos internos puedan llamar a las estaciones remotas en este caso con la Oficina A y C.

- La oficina C con dirección de red 192.168.3.0/24, la oficina consta de un sofphone (192.168.3.10). Toda la red se encuentra protegida por un Firewall (eth0: 192.168.3.1 y la eth1:172.16.3.2) que se encuentra configurado sobre un Ubuntu Server, el mismo que hace NAT y tiene una ruta por defecto a la IP 172.16.3.1. El firewall tiene instalado Siproxd para que los equipos internos puedan llamar a las estaciones remotas en este caso con la Oficina A y B

4.2. Escenario 1: Total Operatividad

Las pruebas realizadas se hicieron en un total funcionamiento, ósea con todos los Siproxd funcionando en cada uno de los firewalls.

4.2.1 Registro de las Extensiones

Para mostrar las extensiones registradas cuando la red se encuentra en una Total Operatividad se debe ingresar al servidor Elastix, donde se debe ingresar los siguientes comandos como se muestra en la tabla IV.XL:

```
[root@elastix ~]# asterisk -r
Asterisk 1.4.26.1, Copyright (C) 1999 - 2008 Digium, Inc. and others.
Created by Mark Spencer <markster@digium.com>
Asterisk comes with ABSOLUTELY NO WARRANTY; type 'core show warranty'
for details.
This is free software, with components licensed under the GNU General
Public
License version 2 and other licenses; you are welcome to redistribute
it under certain conditions. Type 'core show license' for details.
=========================================================================
===
Connected to Asterisk 1.4.26.1 currently running on elastix (pid =
2558)
Verbosity is at least 3
elastix*CLI> sip show peers
Name/username         Host            Dyn Nat ACL Port     Status
2503/2503         172.16.2.2           D   N   A  1024     OK (5 ms)
2502/2502         192.168.1.20         D   N   A  44924    OK (102 ms)
2501/2501         172.16.2.2           D   N   A  1024     OK (29 ms)
2500/2500         172.16.3.2           D   N   A  1024     OK (14 ms)
4 sip peers [Monitored: 4 online, 0 offline Unmonitored: 0 online, 0
offline]
```

Tabla IV.XL Extensiones Registrada cuando la red se encuentra en Total operatividad.

Como se observa existen 4 Extensiones registradas, las mismas que se encuentran con sus respectivas IPs. Si la extensión se encuentra en la misma red que la PBX muestra su IP original, pero si la extensión está en otra estación remota lo que muestra es su IP pública.

Asterisk CLI es el nombre que recibe la consola de Asterisk. Es decir, una línea de comandos para controlar Asterisk directamente, el comando utilizado para ingresar a la consola es *asterisk –r.*

El comando que muestra las extensiones registradas es *sip show peers*.

4.2.2. Establecimiento de una llamada

Para establecer una llamada entre estaciones remotas debe estar en ejecución el demonio Siproxd, el mismo que permite modificar las cabeceras del mensaje SIP. Se ha tomado como ejemplo una llamada entre la extensión 2500 (sofphone) que se encuentra en la red 172.16.3.2 que es la oficina B y la extensión 2501 (Teléfono IP) que se encuentra en la red 172.16.2.2 que es la oficina C.

En la consola de Asterisk el momento de realizar la llamada se puede observar cómo se establece la misma, ver tabla IV.XLI:

```
Verbosity is at least 3

    -- Remote UNIX connection

    -- Remote UNIX connection disconnected

    -- Executing [2501@from-internal:1] Set("SIP/2500-0852df18",
"__RINGTIMER=15") in new stack

    --    Executing    [2501@from-internal:2]    Macro("SIP/2500-
0852df18", "exten-vm|2501|2501") in new stack

    -- Executing [s@macro-exten-vm:1] Macro("SIP/2500-0852df18",
"user-callerid") in new stack

    --    Executing    [s@macro-user-callerid:1]    Set("SIP/2500-
0852df18", "AMPUSER=2500") in new stack

    --   Executing   [s@macro-user-callerid:2]   GotoIf("SIP/2500-
0852df18", "0?report") in new stack

    --   Executing   [s@macro-user-callerid:3]   ExecIf("SIP/2500-
0852df18", "1|Set|REALCALLERIDNUM=2500") in new stack

    --    Executing    [s@macro-user-callerid:4]    Set("SIP/2500-
0852df18", "AMPUSER=2500") in new stack

    --    Executing    [s@macro-user-callerid:5]    Set("SIP/2500-
0852df18", "AMPUSERCIDNAME=Daniel Morales") in new stack
```

```
    --   Executing   [s@macro-user-callerid:6]   GotoIf("SIP/2500-
0852df18", "0?report") in new stack

    --    Executing    [s@macro-user-callerid:7]    Set("SIP/2500-
0852df18", "AMPUSERCID=2500") in new stack

    --    Executing    [s@macro-user-callerid:8]    Set("SIP/2500-
0852df18", "CALLERID(all)="Daniel Morales" <2500>") in new stack

    --   Executing   [s@macro-user-callerid:9]   ExecIf("SIP/2500-
0852df18", "1|Set|CHANNEL(language)=es") in new stack

    --  Executing  [s@macro-user-callerid:10]  GotoIf("SIP/2500-
0852df18", "0?continue") in new stack

    --    Executing    [s@macro-user-callerid:11]    Set("SIP/2500-
0852df18", "__TTL=64") in new stack

    --  Executing  [s@macro-user-callerid:12]  GotoIf("SIP/2500-
0852df18", "1?continue") in new stack

    -- Goto (macro-user-callerid,s,19)

    --   Executing   [s@macro-user-callerid:19]   NoOp("SIP/2500-
0852df18",  "Using  CallerID  "Daniel  Morales"  <2500>")  in  new
stack

   --  Executing  [s@macro-exten-vm:2]  Set("SIP/2500-0852df18",
"RingGroupMethod=none") in new stack

    --  Executing  [s@macro-exten-vm:3]  Set("SIP/2500-0852df18",
"VMBOX=2501") in new stack

    --  Executing  [s@macro-exten-vm:4]  Set("SIP/2500-0852df18",
"EXTTOCALL=2501") in new stack

    --  Executing  [s@macro-exten-vm:5]  Set("SIP/2500-0852df18",
"CFUEXT=") in new stack

    --  Executing  [s@macro-exten-vm:6]  Set("SIP/2500-0852df18",
"CFBEXT=") in new stack

    --  Executing  [s@macro-exten-vm:7]  Set("SIP/2500-0852df18",
"RT=15") in new stack

    -- Executing [s@macro-exten-vm:8] Macro("SIP/2500-0852df18",
"record-enable|2501|IN") in new stack

    --   Executing   [s@macro-record-enable:1]   GotoIf("SIP/2500-
0852df18", "1?check") in new stack

    -- Goto (macro-record-enable,s,4)

    --    Executing    [s@macro-record-enable:4]    AGI("SIP/2500-
```

```
0852df18", "recordingcheck|20110222-120130|1298394090.4") in new
stack

    --     Launched     AGI     Script     /var/lib/asterisk/agi-
bin/recordingcheck

  recordingcheck|20110222-120130|1298394090.4: Inbound recording
not enabled

    -- AGI Script recordingcheck completed, returning 0

    -- Executing [s@macro-record-enable:5] MacroExit("SIP/2500-
0852df18", "") in new stack

    -- Executing [s@macro-exten-vm:9] Macro("SIP/2500-0852df18",
"dial|15|tr|2501") in new stack

    --  Executing  [s@macro-dial:1]  GotoIf("SIP/2500-0852df18",
"1?dial") in new stack

    -- Goto (macro-dial,s,3)

    --   Executing   [s@macro-dial:3]   AGI("SIP/2500-0852df18",
"dialparties.agi") in new stack

    --     Launched     AGI     Script     /var/lib/asterisk/agi-
bin/dialparties.agi

  dialparties.agi: Starting New Dialparties.agi

  == Parsing '/etc/asterisk/manager.conf': Found

  == Parsing '/etc/asterisk/manager_additional.conf': Found

  == Parsing '/etc/asterisk/manager_custom.conf': Found

  == Manager 'admin' logged on from 127.0.0.1

  dialparties.agi: Caller ID name is 'Daniel Morales' number is
'2500'

  dialparties.agi: Methodology of ring is  'none'

    --  dialparties.agi: Added extension 2501 to extension map

    --  dialparties.agi: Extension 2501 cf is disabled

    --   dialparties.agi:  Extension  2501  do  not  disturb  is
disabled

  dialparties.agi: ExtensionState: 0

    --  dialparties.agi: dbset CALLTRACE/2501 to 2500
```

```
    --  dialparties.agi: Filtered ARG3: 2501

  == Manager 'admin' logged off from 127.0.0.1

    -- AGI Script dialparties.agi completed, returning 0

    --   Executing   [s@macro-dial:7]   Dial("SIP/2500-0852df18",
"SIP/2501|15|tr") in new stack

    -- Called 2501

    -- SIP/2501-08526320 is ringing

    -- SIP/2501-08526320 answered SIP/2500-0852df18

    --   Executing   [h@macro-dial:1]   Macro("SIP/2500-0852df18",
"hangupcall") in new stack

    --    Executing    [s@macro-hangupcall:1]    GotoIf("SIP/2500-
0852df18", "1?skiprg") in new stack

    -- Goto (macro-hangupcall,s,4)

    --    Executing    [s@macro-hangupcall:4]    GotoIf("SIP/2500-
0852df18", "1?skipblkvm") in new stack

    -- Goto (macro-hangupcall,s,7)

    --    Executing    [s@macro-hangupcall:7]    GotoIf("SIP/2500-
0852df18", "1?theend") in new stack

    -- Goto (macro-hangupcall,s,9)

    --    Executing    [s@macro-hangupcall:9]    Hangup("SIP/2500-
0852df18", "") in new stack

  == Spawn extension (macro-hangupcall, s, 9) exited non-zero on
'SIP/2500-0852df18' in macro 'hangupcall'

  == Spawn h extension (macro-dial, h, 1) exited non-zero on
'SIP/2500-0852df18'

  == Spawn extension (macro-dial, s, 7) exited non-zero on
'SIP/2500-0852df18' in macro 'dial'

  == Spawn extension (macro-exten-vm, s, 9) exited non-zero on
'SIP/2500-0852df18' in macro 'exten-vm'

  == Spawn extension (from-internal, 2501, 2) exited non-zero on
'SIP/2500-0852df18'
```

Tabla IV.XLI. Establecimiento de una llamada en la consola de Asterisk

Para visualizar de una mejor manera el establecimiento de una llamada se ha utilizado el programa Wireshark el cual permite observar gráficamente la conexión. Todo el establecimiento de la llamada desde la extensión 2500 a la extensión 2501 ver la figura IV.23:

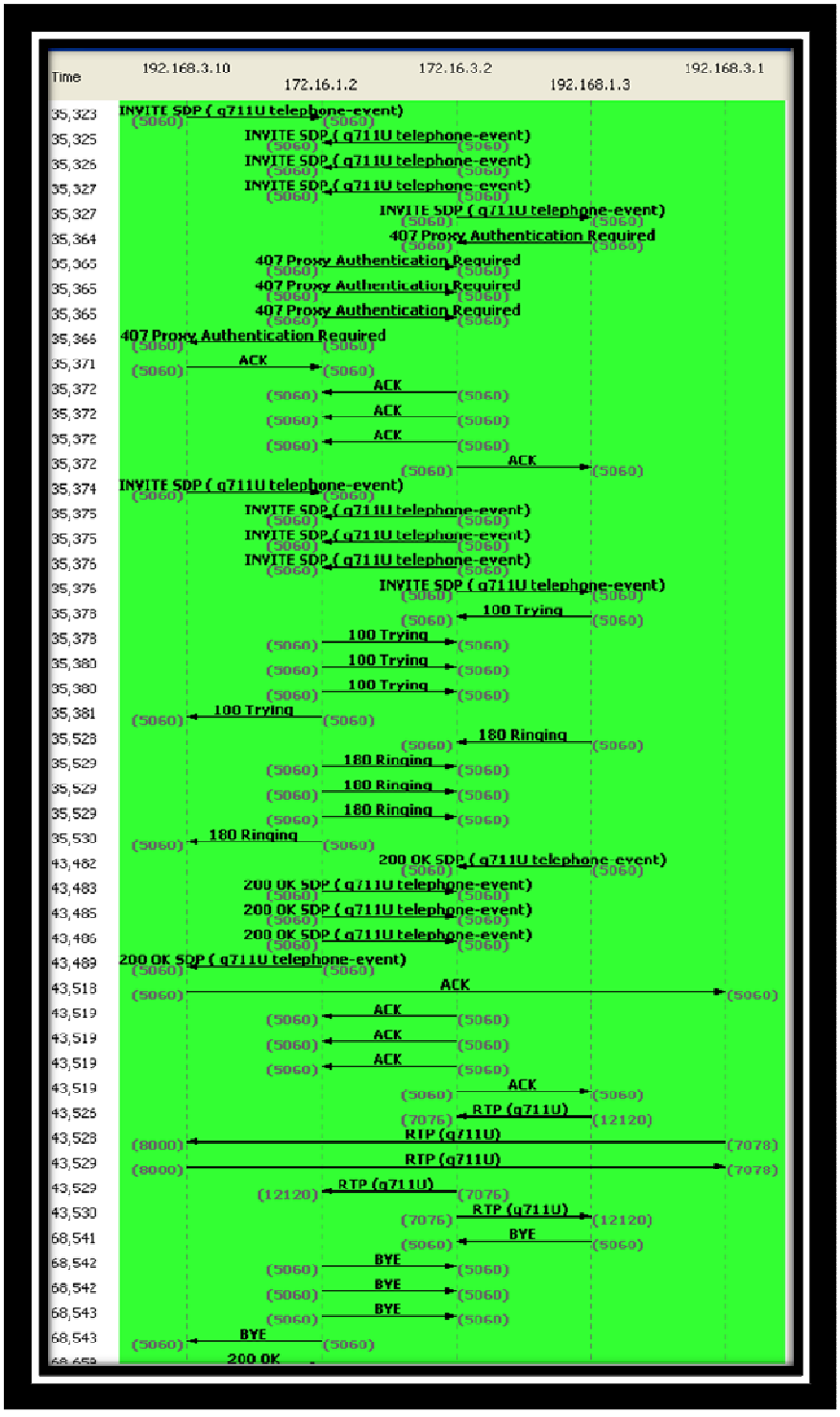


Figura IV.23: Llamada desde la extensión 2500 a Asterisk

El mensaje SIP generado muestra como se establece la comunicación entre la extensión 2500 y el Asterisk, las cuales tienen NAT y firewall, observar tabla IV.XLII:

```
Session Initiation Protocol
Request-Line: ACK sip:2501@172.16.1.2 SIP/2.0
Method: ACK
Request-URI: sip:2501@172.16.1.2
Request-URI User Part: 2501
Request-URI Host Part: 172.16.1.2
Resent Packet: False
Message Header
Via: SIP/2.0/UDP 192.168.3.10:5060;branch=z9hG4bK-d8754z-
644a3cbec448e054-1---d8754z-
Transport: UDP
Sent-by Address: 192.168.3.10
Sent-by port: 5060
Branch: z9hG4bK-d8754z-644a3cbec448e054-1---d8754z-
Max-Forwards: 70
Route: <sip:siproxd@192.168.3.1:5060;lr>
Contact: <sip:2500@192.168.3.10:5060;transport=UDP>
Contact Binding: <sip:2500@192.168.3.10:5060;transport=UDP>
URI: <sip:2500@192.168.3.10:5060;transport=UDP>
To: <sip:2501@172.16.1.2;transport=UDP>;tag=as5ea4f087
SIP to address: sip:2501@172.16.1.2
SIP to address User Part: 2501
SIP to address Host Part: 172.16.1.2
SIP tag: as5ea4f087
From: <sip:2500@172.16.1.2;transport=UDP>;tag=d2604443
SIP from address: sip:2500@172.16.1.2
SIP from address User Part: 2500
SIP from address Host Part: 172.16.1.2
SIP tag: d2604443
Call-ID: NjJjZjFmZGFkMWRlZWUyNDA5YzQzZjJkZTgxNmVlMTc.
CSeq: 2 ACK
Sequence Number: 2
Method: ACK
Proxy-Authorization: Digest
username="2500",realm="asterisk",nonce="65dc1371",uri="sip:2501@172.16
.1.2;transport=UDP",response="16cb7d523f57104ba3d3ff8128b430ca",algori
thm=MD5
Authentication Scheme: Digest
Username: "2500"
Realm: "asterisk"
Nonce Value: "65dc1371"
Authentication URI: "sip:2501@172.16.1.2;transport=UDP"
Digest Authentication Response: "16cb7d523f57104ba3d3ff8128b430ca"
Algorithm: MD5
User-Agent: Zoiper rev.9749
Content-Length: 0
```

Tabla IV.XLII. Mensaje SIP desde la extensión 2500 al Asterisk

Cuando se trata de llamadas remotas en este caso el sofphone (2500) de la oficina C con el teléfono IP (2501) de la oficina B, asterisk lo toma como si fuera dos llamas por separado. La primera llamada seria de la extensión 2500 al Asterisk y el otro desde el Asterisk a la extensión 2501, por esta razón se encuentra el siguiente diagrama, ver figura IV.24:

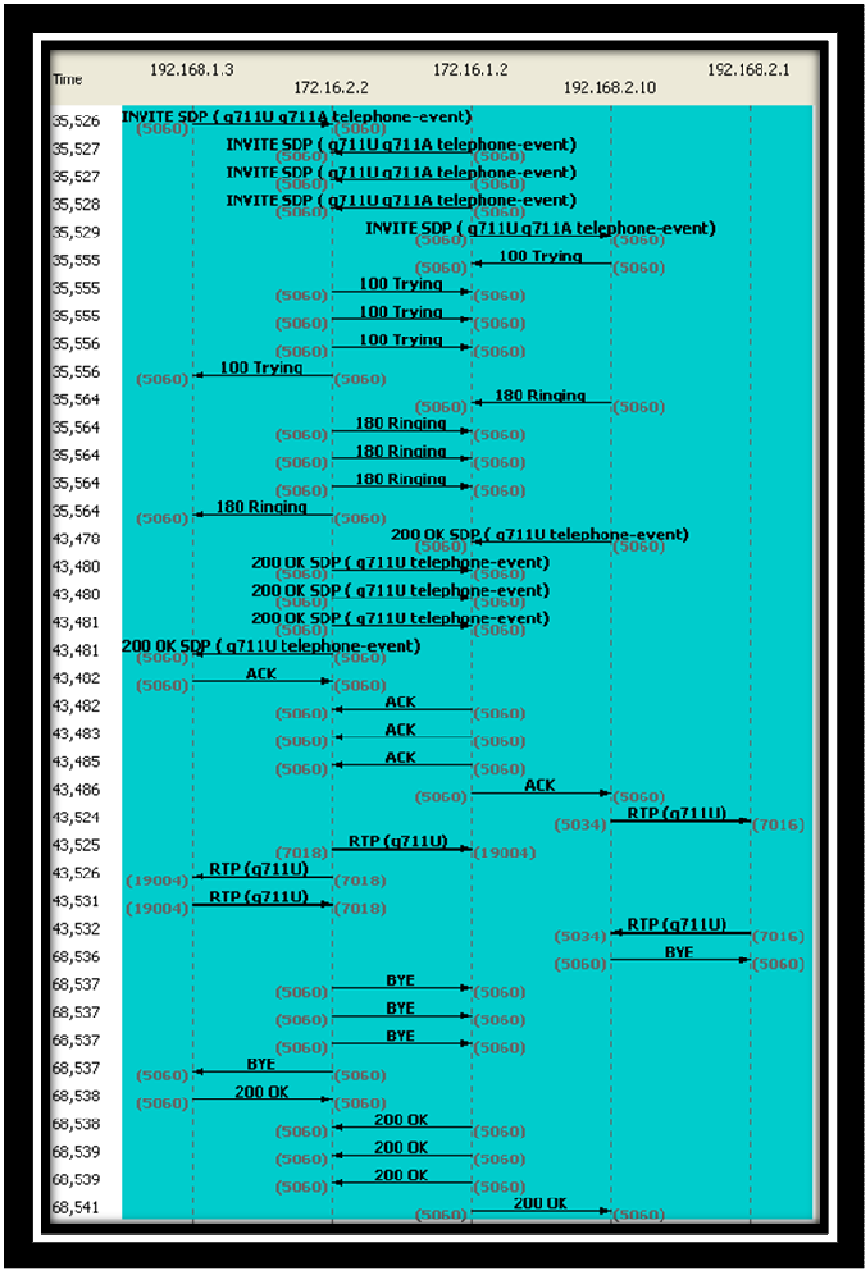


Figura IV.24. Llamada desde Asterisk a la extensión 2501

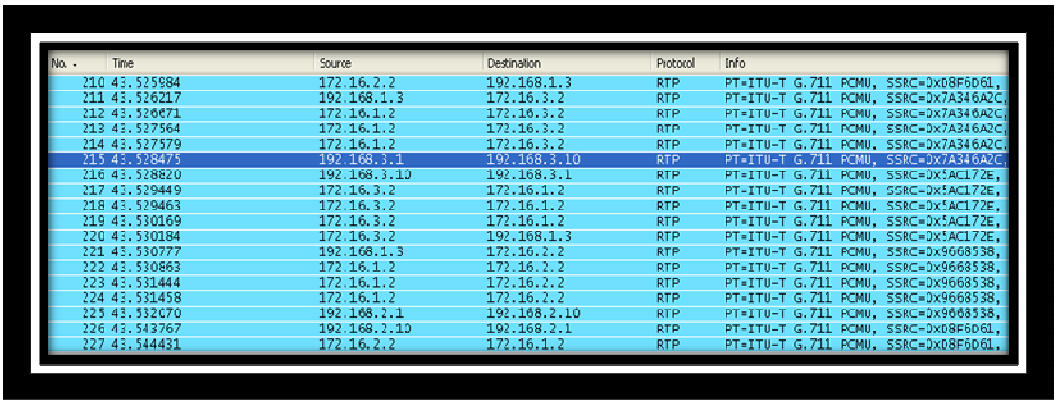
El mensaje SIP generado muestra como se establece la comunicación entre Asterisk y la extensión 2501, observar tabla IV.XLIII:

```
Request-Line: ACK sip:2501@192.168.2.10:5060 SIP/2.0
Method: ACK
Request-URI: sip:2501@192.168.2.10:5060
Request-URI User Part: 2501
Request-URI Host Part: 192.168.2.10
Request-URI Host Port: 5060
Resent Packet: False
Request Frame: 160
Response Time (ms): 7957
Message Header
Via: SIP/2.0/UDP
192.168.2.1:5060;branch=z9hG4bK5fcde80d8324c3f2f6397722a37e2ce9
Transport: UDP
Sent-by Address: 192.168.2.1
Sent-by port: 5060
Branch: z9hG4bK5fcde80d8324c3f2f6397722a37e2ce9
Via: SIP/2.0/UDP 172.16.1.2:5060;branch=z9hG4bK22bc06fa;rport
Transport: UDP
Sent-by Address: 172.16.1.2
Sent-by port: 5060
Branch: z9hG4bK22bc06fa
RPort: rport
Record-Route: <sip:siproxd@192.168.2.1:5060;lr>
From: "Daniel Morales" <sip:2500@172.16.1.2>;tag=as090e0d9c
SIP Display info: "Daniel Morales"
SIP from address: sip:2500@172.16.1.2
SIP from address User Part: 2500
SIP from address Host Part: 172.16.1.2
SIP tag: as090e0d9c
To: <sip:2501@172.16.2.2>;tag=df216fe6990e964f
SIP to address: sip:2501@172.16.2.2
SIP to address User Part: 2501
SIP to address Host Part: 172.16.2.2
SIP tag: df216fe6990e964f
Call-ID: 5852418e67aa63a429394f521dab0793@172.16.1.2
CSeq: 102 ACK
Sequence Number: 102
Method: ACK
Contact: <sip:2500@172.16.1.2>
Contact Binding: <sip:2500@172.16.1.2>
URI: <sip:2500@172.16.1.2>
SIP contact address: sip:2500@172.16.1.2
User-agent: Asterisk PBX
Max-Forwards: 69
Content-Length: 0
```

Tabla IV.XLIII. Mensaje SIP desde Asterisk a la extensión 2501.

4.2.3. Paquetes RTP

El punto más importante es el de los paquetes RTP, puesto que un sofphone o un teléfono IP pueden registrarse mediante un Proxy SIP pero al no tener un proxy RTP el audio no puede ser escuchado en las estaciones remotas. Para comprobar lo dicho se ha capturado paquetes RTP cuando una conversación se encuentra en curso como muestra la figura IV.25:



No.	Time	Source	Destination	Protocol	Info
210	43.525984	172.16.2.2	192.168.1.3	RTP	PT=ITU-T G.711 PCMU, SSRC=0x08F6D61,
211	43.526217	192.168.1.3	172.16.3.2	RTP	PT=ITU-T G.711 PCMU, SSRC=0x7A346A2C,
212	43.526671	172.16.1.2	172.16.3.2	RTP	PT=ITU-T G.711 PCMU, SSRC=0x7A346A2C,
213	43.527564	172.16.1.2	172.16.3.2	RTP	PT=ITU-T G.711 PCMU, SSRC=0x7A346A2C,
214	43.527579	172.16.1.2	172.16.3.2	RTP	PT=ITU-T G.711 PCMU, SSRC=0x7A346A2C,
215	43.528475	192.168.3.1	192.168.3.10	RTP	PT=ITU-T G.711 PCMU, SSRC=0x7A346A2C,
216	43.528820	192.168.3.10	192.168.3.1	RTP	PT=ITU-T G.711 PCMU, SSRC=0x5AC172E,
217	43.529449	172.16.3.2	172.16.1.2	RTP	PT=ITU-T G.711 PCMU, SSRC=0x5AC172E,
218	43.529463	172.16.3.2	172.16.1.2	RTP	PT=ITU-T G.711 PCMU, SSRC=0x5AC172E,
219	43.530169	172.16.3.2	172.16.1.2	RTP	PT=ITU-T G.711 PCMU, SSRC=0x5AC172E,
220	43.530184	172.16.3.2	192.168.1.3	RTP	PT=ITU-T G.711 PCMU, SSRC=0x5AC172E,
221	43.530777	192.168.1.3	172.16.2.2	RTP	PT=ITU-T G.711 PCMU, SSRC=0x9668538,
222	43.530863	172.16.1.2	172.16.2.2	RTP	PT=ITU-T G.711 PCMU, SSRC=0x9668538,
223	43.531444	172.16.1.2	172.16.2.2	RTP	PT=ITU-T G.711 PCMU, SSRC=0x9668538,
224	43.531458	172.16.1.2	172.16.2.2	RTP	PT=ITU-T G.711 PCMU, SSRC=0x9668538,
225	43.532670	192.168.2.1	192.168.2.10	RTP	PT=ITU-T G.711 PCMU, SSRC=0x9668538,
226	43.543767	192.168.2.10	192.168.2.1	RTP	PT=ITU-T G.711 PCMU, SSRC=0x08F6D61,
227	43.544431	172.16.2.2	172.16.1.2	RTP	PT=ITU-T G.711 PCMU, SSRC=0x08F6D61,

Figura IV.25. Paquetes RTP en una llamada telefónica en curso.

El tráfico de los paquetes RTP capturado es el siguiente, ver tabla IV.XLIV:

```
Internet Protocol, Src: 192.168.3.1 (192.168.3.1), Dst:
192.168.3.10 (192.168.3.10)
Version: 4
Header length: 20 bytes
Source: 192.168.3.1 (192.168.3.1)
Destination: 192.168.3.10 (192.168.3.10)
Source port: 7078 (7078)
Destination port: irdmi (8000)
Length: 180
Checksum: 0x91f7 [validation disabled]
Real-Time Transport Protocol
Stream setup by SDP (frame 96)
10.. .... = Version: RFC 1889 Version (2)
..0. .... = Padding: False
...0 .... = Extension: False
.... 0000 = Contributing source identifiers count: 0
1... .... = Marker: True
Payload type: ITU-T G.711 PCMU (0)
Sequence number: 49170
Extended sequence number: 49170
Timestamp: 1829186536
Synchronization Source identifier: 0x7a346a2c (2050255404)
Payload: FDFEFEFCFBFE7B7A7A7A7B7CFCF6F0EFEFEFF2F7FE7A7775...
```

Tabla IV.XLIV: Paquetes RTP en una conversación establecida.

4.3. Escenario 2: PBX sin Siproxd

En esta sección se procede a detener el servicio de Siproxd en el firewall de la oficina A.

4.3.1. Registro de las extensiones

Ya con el servicio detenido se procede a ingresar a la consola de Asterisk para verificar que usuarios se encuentran registrados, ver tabla IV.XLV:

```
elastix*CLI> sip show peers

Name/username                Host              Dyn  Nat  ACL  Port
Status

2503/2503                    172.16.2.2        D    N    A    5060
OK (4 ms)

2502/2502                    192.168.1.20      D    N    A    5060
OK (2 ms)

2501/2501                    172.16.2.2        D    N    A    5060
OK (24 ms)

2500/2500                    172.16.3.2        D    N    A    5060
OK (13 ms)

4 sip peers [Monitored: 4 online, 0 offline Unmonitored: 0
online, 0 offline]
```

Tabla IV.XLV: Extensiones registradas cuando Siproxd es detenido en el Firewall A.

Como se observa en la tabla anterior, si se detiene el Siproxd en el firewall de la oficina A, las extensiones 2503, 2501 y 2500 se registran debido a que Siproxd se encuentra ejecutándose en sus respectivos firewalls.

4.3.2. Establecimiento de la llamada

Para una llamada entre estaciones remotas no existe ningún problema, es como en el caso anterior cuando Siproxd se ejecuta en todos los firewalls.

Se realiza la llamada entre la extensión 2502 (sofphone) que se encuentra en la oficina A y la extensión 2501 (teléfono IP) que se encuentra en la Oficina B. Cuando la llamada se realiza desde la extensión 2502 de la oficina B a la extensión 2502 de la oficina A se encuentra en normal funcionamiento la llamada, ver imagen figura IV.26:

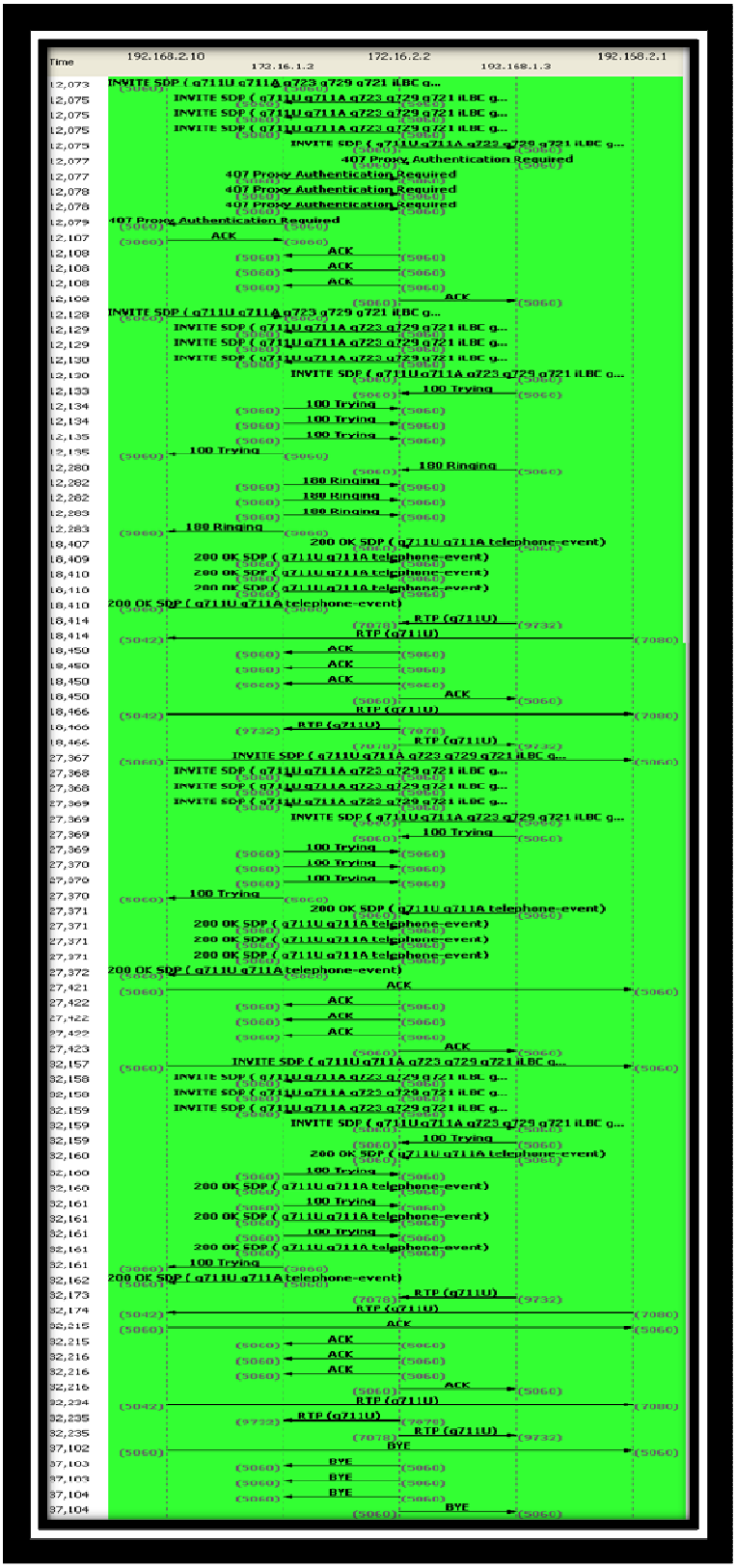


Figura IV.26. Llamada desde la extensión 2501 a 2502

Como se puede observar la llamada se establece, pero el numero de procesos para la comunicación aumenta lo cual disminuye el rendimiento de la red.

Esto es con una llamada, si existieran llamadas simultáneas se disminuye considerablemente el ancho de banda. Si la llamada se realiza desde la oficina A en este caso de la extensión 2502 a la 2501 en la oficina B, la llamada se establece pero el flujo RTP se pierde en este caso no existe audio entre las extensiones.

Esto no sucede en todos los casos, ciertas ocasiones si se establece el flujo RTP, es debido a que el firewall de la oficina A tiene el servicio Siproxd detenido y no sabe cómo resolver a los usuarios externos, ya que el Proxy SIP y RTP no se encuentra en funcionamiento. Ver figura IV.27:

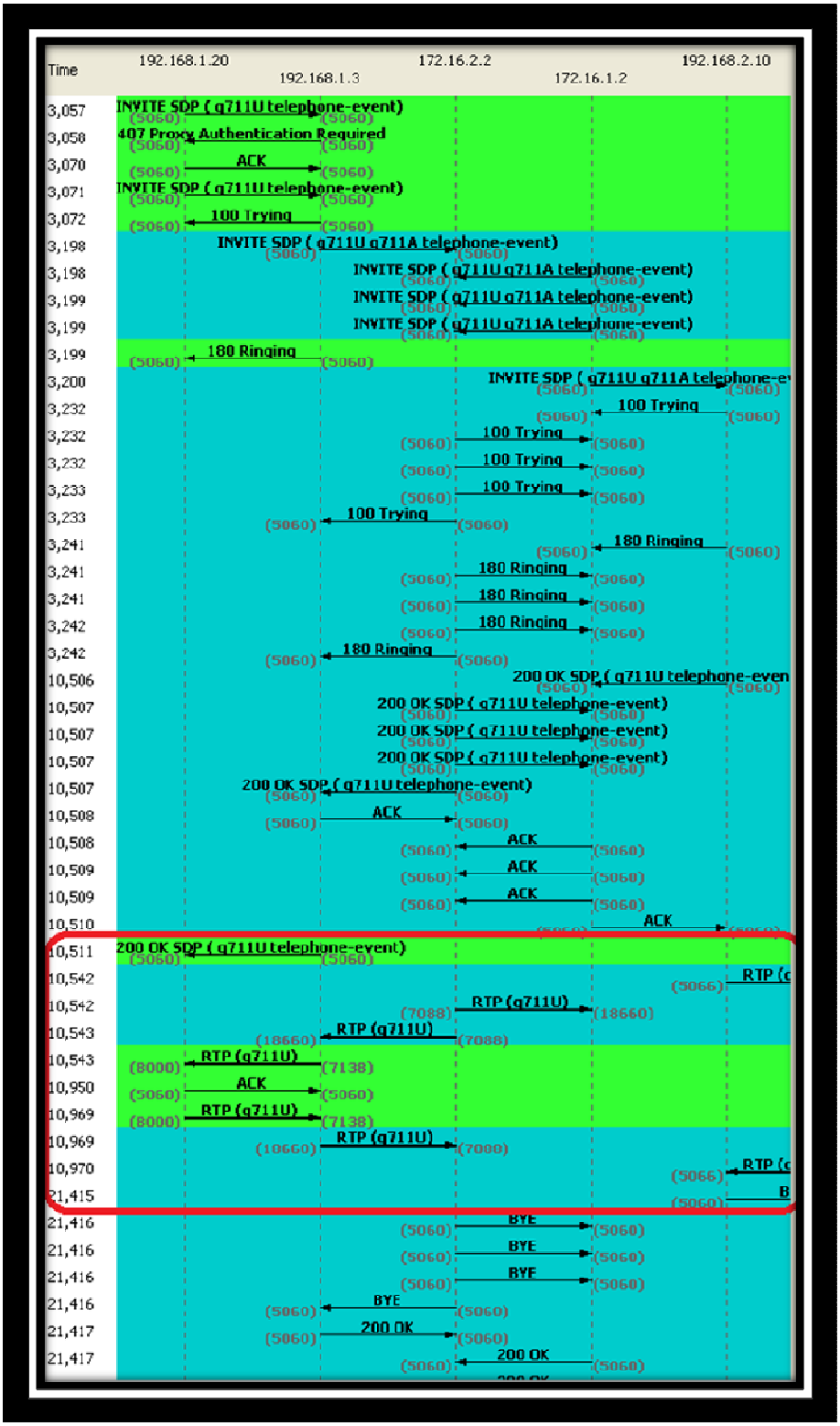


Figura IV.27: Llamada desde la extensión 2502 a 2501

Como se observa a parte de la gran cantidad de tráfico que se genera, en la sección que se encuentra encerrada en un recuadro rojo se observa que hay paquetes RTP que se dirigen hacia algún otro lugar y de igual manera viene tráfico RTP de algún lugar.

4.3.3. Paquetes RTP

Como se pudo observar existe tráfico RTP, pero en ocasiones los paquetes no van dirigidos a quien deba llegar. Al no ser dirigidos los paquetes RTP a las extensiones que iniciaron la llamada no existe audio entre las mismas. Ver figura IV.28:

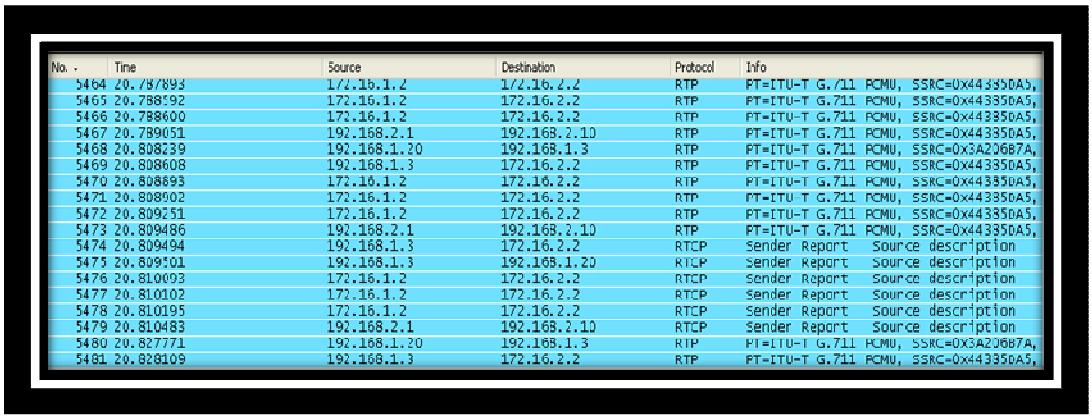


Figura IV.28: Tráfico RTP cuando Siproxd se encuentra detenido en la oficina A.

Se puede observar que el flujo de datos RTP se transmite cuando Siproxd en el firewall de la oficina A se encuentra detenido, en este caso los paquetes RTP son transmitidos en mayor cantidad, ya que pueden ser enviados a su destino correcto o pueden ser enviados a otros dispositivos. Esto se debe a que el firewall de la oficina A no ejecuta Siproxd y no sabe como enviar ciertos paquetes.

4.4. Escenario 3: En el transcurso de la llamada se detiene el servicio

4.4.1. Registro de las extensiones

Se procede ingresar a Asterisk para observar que extensiones se encuentran registradas. Ver tabla IV.XLVI:

```
elastix*CLI> sip show peers
Name/username         Host            Dyn Nat ACL Port
Status
2503/2503         172.16.2.2           D   N   A  1024     OK (5
ms)
2502/2502         192.168.1.20         D   N   A  44924    OK
(102 ms)
2501/2501         172.16.2.2           D   N   A  1024     OK
(29 ms)
2500/2500         172.16.3.2           D   N   A  1024     OK
(14 ms)
4 sip peers [Monitored: 4 online, 0 offline Unmonitored: 0
online, 0 offline]
```

Tabla IV.XLVI: Registro de extensiones.

En este caso se registran todas las extensiones, debido a que el servicio de Siproxd se detiene cuando se establece la llamada telefónica.

4.4.2. Establecimiento de la llamada

Por el momento el establecimiento de la llamada se realizará normalmente ya que Siproxd se encuentra ejecutándose en cada uno de los firewall y no existiría problema. Para la llamada se ha tomado las extensiones 2501 (teléfono IP) de la oficina B y la extensión 2500 (Softphones) de la oficina C, en la comunicación se detiene el servicio del firewall de la oficina C (192.168.3.1) con lo cual nunca se termina la llamada y queda inconclusa ya que se pierde tráfico SIP y RTP. Ver figura IV.29:

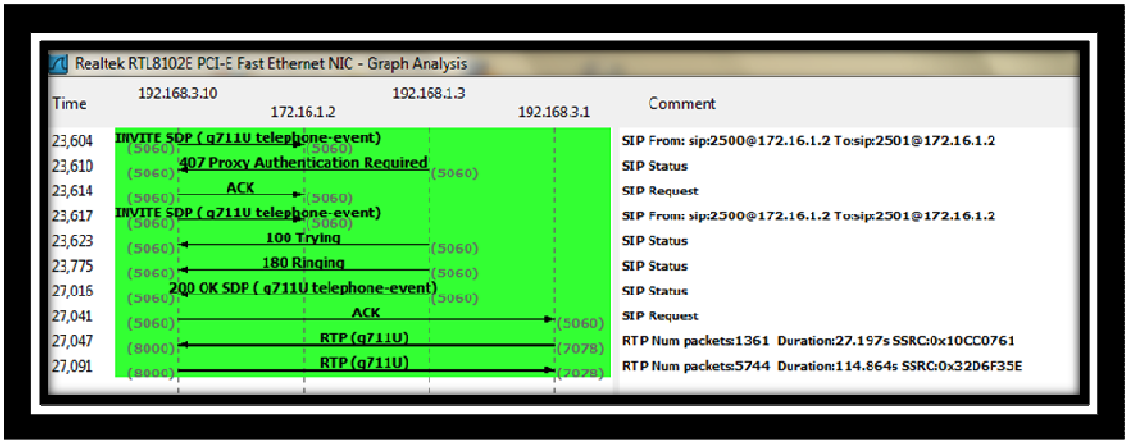


Figura IV.29. Siproxd detenido en el transcurso de una llamada

Como se observa la llamada se establece, existiendo en el transcurso de tiempo paquetes RTP, pero al detener el servicio nunca se finaliza la llamada.

4.4.3. Paquetes RTP

Se procedió a parar el demonio Siproxd en firewall de la oficina C, el mismo que deshabilita el Proxy SIP y el Proxy RTP, con esto los Paquetes RTP no saben a dónde dirigirse, con lo cual se pierde toda comunicación en llamada previamente establecida, ver figura IV.30:

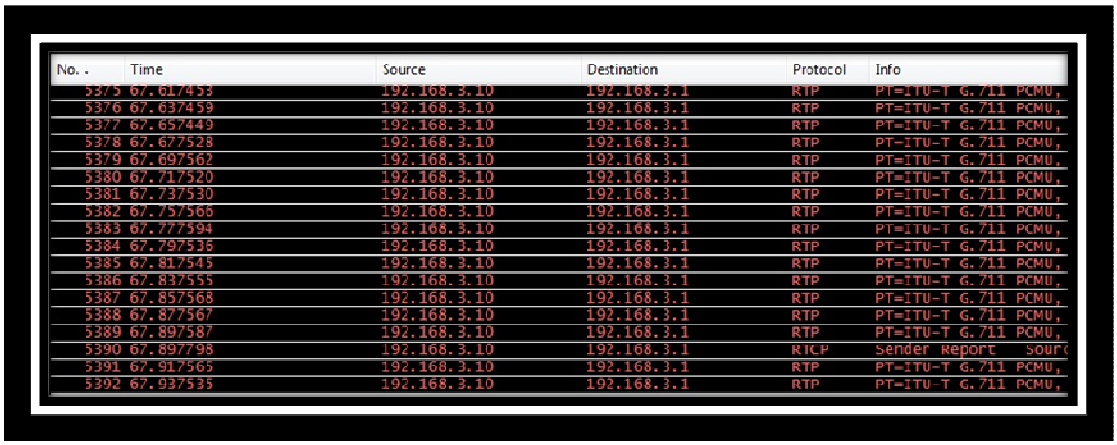


Figura IV.30: Paquetes RTP en una llamada telefónica cuando se detuvo Siproxd.

Los paquetes RTP tienen un error en la transmisión debido a que ya no saben a dónde dirigirse, solo buscan tráfico entre las IPs originales es decir a 192.168.2.20 (2501) y 192.168.3.10 (2500), como son redes inalcanzables puesto que disponen de NAT y firewall se pierde todo el audio. Ver tabla IV.XLVII:

```
Internet Protocol, Src: 192.168.3.10 (192.168.3.10), Dst:
192.168.3.1 (192.168.3.1)
Version: 4
Header length: 20 bytes
Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN:
0x00)
Total Length: 200
Identification: 0x2df0 (11760)
Flags: 0x00
Fragment offset: 0
Time to live: 128
Protocol: UDP (0x11)
Good: False
Expert Info (Error/Checksum): Bad checksum
Message: Bad checksum
Severity level: Error
Group: Checksum
Source: 192.168.3.10 (192.168.3.10)
Destination: 192.168.3.1 (192.168.3.1)
User Datagram Protocol, Src Port: irdmi (8000), Dst Port: arcp
(7070)
Stream setup by SDP (frame 5)
10.. .... = Version: RFC 1889 Version (2)
..0. .... = Padding: False
...0 .... = Extension: False
.... 0000 = Contributing source identifiers count: 0
0... .... = Marker: False
Payload type: ITU-T G.711 PCMU (0)
Sequence number: 3670
Extended sequence number: 69206
Timestamp: 2075998346
Synchronization Source identifier: 0xd38e1ab0 (3549305520)
Payload: 5F626B7EEBE2E0E5EE7B6C6663646A76F4EBE6E7EBF6776E...
```

Tabla IV.XLVII: Mensaje RTP en una llamada telefónica cuando se detuvo Siproxd**.**

4.5. Escenario 4: Siproxd detenido en uno de los Firewall

En este punto se realizó las pruebas deteniendo Siproxd en el firewall de la oficina B (192.168.2.1), pero antes de iniciar una llamada.

4.5.1. Registro de extensiones

Ya con el Siproxd detenido, se ingresa a la consola de Asterisk y se observa que las extensiones se encuentran registradas. Ver tabla IV.XLVIII:

```
elastix*CLI> sip show peers

Name/username              Host            Dyn Nat ACL Port
Status

2503/2503                  172.16.2.2       D   N   A  1024
UNREACHABLE

2502/2502                  192.168.1.20     D   N   A  5060
OK (2 ms)

2501/2501                  172.16.2.2       D   N   A  1024
UNREACHABLE

2500/2500                  172.16.3.2       D   N   A  1024
OK (14 ms)

4 sip peers [Monitored: 2 online, 2 offline Unmonitored: 0
online, 0 offline]
```

Tabla IV.XLVIII. Extensiones registradas al detener Siproxd en la oficina B.

Se puede observar que en el momento de detener el servicio en el firewall B (192.168.2.1), las extensiones pertenecientes a la red no se registran.

4.5.2. Establecimiento de la llamada

Al ser inalcanzables las extensiones 2501 y 2503 no se puede realizar llamadas.

4.5.3. Paquetes RTP

No se puede capturar tráfico RTP debido a que no hay una conexión previa establecida.

4.6. Comprobación de la hipótesis de la investigación realizada

Población: Todas las Personas que conozcan sobre el servicio VoIP.

Muestra: Se tomó 10 Personas que conocen sobre la tecnología de VoIP; esta muestra fue tomada en forma dirigida, es decir no aleatoria.

Pasos para verificar la hipótesis:

4.6.1. Planteamiento de la hipótesis

Ho: "Las técnicas de Gateways SIP aplicadas a la transmisión de tráfico de Voz IP a través de firewalls, no mejora la disponibilidad de este servicio."

Hi: "Las técnicas de Gateways SIP aplicadas a la transmisión de tráfico de Voz IP a través de firewalls, mejora la disponibilidad de este servicio."

4.6.2. Nivel de significancia

Una vez establecida la hipótesis nula y alternativa, se debe determinar el nivel de significancia, que para el caso del presente análisis se utilizara un nivel de significación estadística de **α= 0,05**

4.6.3. Criterio

De acuerdo al análisis desarrollado en la presente investigación, se ha seleccionado como estadístico de prueba de hipótesis la técnica **"chi-cuadrado"**. La fórmula que da el estadístico es la siguiente:

$$X^2 = \sum_i \frac{(observada_i - esperada_i)^2}{esperada_i}$$

Para conocer las frecuencias teóricas o esperadas, se calculan a través del producto de los totales marginales (*total del renglón x total de columna*), dividido por el número total de casos (*gran total*):

$$\boldsymbol{fe} = \frac{(\boldsymbol{total\ del\ renglón}) * (\boldsymbol{total\ de\ la\ columna})}{\boldsymbol{gran\ total}}$$

Ahora es necesario determinar el **criterio de decisión.** Entonces se acepta **Ho** cuando: $X^2_{calculado} < X^2_{tabla}$, en caso contrario se rechaza **Ho.**

Donde el valor de X^2_{tabla} representa el valor proporcionado por la tabla de "distribución X^2", según el nivel de significación elegido y los grados de libertad.

Como se mencionó anteriormente, el nivel de significancia aportado para esta investigación es de **α = 0,05.**

Para la determinación de los grados de libertad **(gl)** se debe aplicar la siguiente fórmula:

$$\boldsymbol{gl} = (r-1) * (k-1)$$

Donde **r** es el número de filas o renglones y **k** el de columnas. La investigación generó una matriz de **2r x 2k.**

Entonces:

$$\boldsymbol{gl} = (2-1) * (2-1)$$

$$\boldsymbol{gl} = (1) * (1)$$

$$\boldsymbol{gl = 1\ grado\ de\ libertad}$$

De acuerdo a la estadística de distribución de chi-cuadrado, con un nivel de significancia 0,05 a 1 grado de libertad, genera un valor de $X^2_{tabla} = 3,841$; ver tabla IV.XLIX:

Grados de libertad	Posibilidad de casualidad en el porcentaje (5% o menos considerados significativos)								
	90%	80%	70%	50%	30%	20%	10%	5%	1%
1	0.016	0.064	0.148	0.455	1.074	1.642	2.706	3.841	6.635
2	0.211	0.446	0.713	1.386	2.408	3.219	4.605	5.991	9.210
3	0.584	1.005	1.424	2.366	3.665	4.642	6.251	7.815	11.341
4	1.064	1.649	2.195	3.357	4.878	5.989	7.779	9.488	13.277
5	1.610	2.343	3.000	4.351	6.064	7.289	9.236	11.070	15.086
6	2.204	3.070	3.828	5.348	7.231	8.558	10.645	12.592	16.812
7	2.833	3.822	4.671	6.346	8.383	9.083	12.017	14.067	18.475
8	3.490	4.594	5.527	7.344	9.524	11.030	13.362	15.507	20.090
9	4.168	5.380	6.393	8.343	10.656	12.242	17.684	16.919	21.666

Tabla IV.XLIX: Distribución de X chi-cuadrado

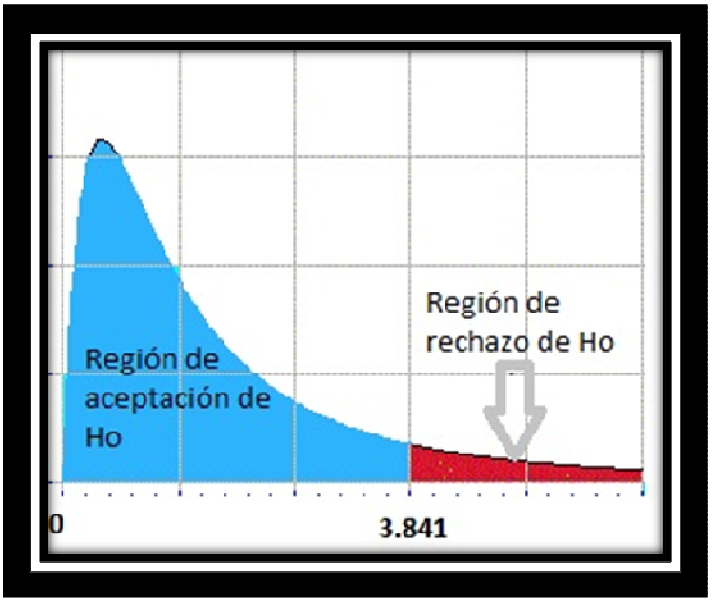


Figura IV.31. Demostración de la hipótesis

La regla de decisión es entonces: No rechazar **Ho** si el valor que se encuentre para $X^2_{calculado} < 3.841$. Si el valor calculado es igual o mayor al valor crítico, se rechaza **Ho** y se acepta **Hi**.

4.6.4. Cálculos

Los resultados que arrojó la investigación realizada (***Ver Anexo D***) se resume en la tabla mostrada a continuación, ver tabla IV.L:

FACTIBILIDAD		
Preguntas	**Mejora**	**No Mejora**
Preg. 5	0	10
Preg. 6	10	0
Total	**10**	**10**

Tabla IV.L. Resultados de encuesta sobre “Factibilidad”

Las matrices de resultados quedan conformadas de la siguiente manera, ver tablas IV.LI y IV.LII:

Disponibilidad del servicio de VoIP / Las técnicas de Gateway SIP	MEJORA	NO MEJORA	TOTAL
Utilizando Siproxd	10	0	10
No utilizando Siproxd	0	10	10
Total	10	10	20

Tabla IV.LI. Matriz de Datos Observados

Las técnicas de Gateway SIP \ Disponibilidad del servicio de VoIP	MEJORA	NO MEJORA	TOTAL
Utilizando Siproxd	5	5	10
No utilizando Siproxd	5	5	10
Total	10	10	20

Tabla IV.LII: Matriz de Datos Esperados

Ahora se diseña la tabla para aplicar la fórmula del chi-cuadrado, ver tabla IV.LIII:

	fo	fe	$(fo-fe)^2/fe$
Utilizando Siproxd mejora	10	5	5
No utilizando Siproxd mejora	0	5	5
Utilizando Siproxd no mejora	0	5	5
No utilizando Siproxd no mejora	10	5	5
Total	**20**	**20**	**20**

Tabla IV.LIII. Frecuencias Observadas / Frecuencias Esperadas

4.6.5. Decisión

Como:

$$X^2_{calculado} = 20$$

$$X^2_{tabla} = 3.841$$

Entonces:

$$X^2_{calculado} > X^2_{tabla}$$

Lo que significa que $X^2_{calculado}$ está en la zona de rechazo de la **Ho,** entonces se concluye que se rechaza la hipótesis nula y se acepta la de la investigación, esto es que:

“Las técnicas de Gateways SIP aplicadas a la transmisión de tráfico de Voz IP a través de firewalls, mejora la disponibilidad de este servicio.”

CONCLUSIONES

1. El problema del establecimiento de conexiones de VoIP a través de NAT se mantendrá mientras no exista un Proxy SIP, debido a que la señalización SIP contiene información de la dirección IP y puerto obtenido de la cabecera SIP, que corresponde con la dirección IP y puerto que la extensión tiene asignada dentro de la red nateada.

2. Los cortafuegos son un serio problema para que el servicio de VoIP funcione correctamente, debido a que la señalización SIP utiliza el puerto 5060 UDP, por lo que es necesario que los cortafuegos permitan el tráfico al puerto 5060, el flujo de datos RTP resulta más problemático, ya que los puertos se escogen dinámicamente al establecerse una conexión, debido a esto se debe utilizar un proxy RTP y no filtrar los puertos por encima de 1024.

3. Después de un estudio minucioso y detallado de los problemas que presenta el protocolo SIP a través de firewalls y NAT, se determinó que la solución más a apropiada es la utilización del software Siproxd ya que funciona como Proxy SIP, Proxy RTP, utiliza una IP pública y además soporta los 4 tipos de NAT.

4. Se demostró a través de las múltiples pruebas e implementación final detallada en la presente tesis, la viabilidad y efectividad de Siproxd bajo plataforma OpenSource, permitiendo la comunicación entre estaciones remotas aprovechando el código abierto distribuido a través de la GNU Public License.

5. Las diferentes pruebas de funcionamiento que se han realizado han verificado el correcto funcionamiento del sistema con los equipos disponibles en los 4 tipos de NAT existentes sin que el usuario tenga que modificar la configuración de traducción de direcciones de su router. Por lo tanto, no es necesario disminuir las especificaciones de seguridad de los cortafuegos gracias al uso del Proxy SIP y del Proxy RTP que facilitan la creación de unas reglas en los cortafuegos que abarca todo el tráfico de VoIP.

RECOMENDACIONES:

1. Al momento de configurar las reglas del firewall que redirige las peticiones SIP se debe tener en cuenta que el Proxy tiene que estar configurado en modo transparente, ya que tanto Asterisk y algunos Softphones no tienen la opción de Proxy SIP.

2. Se debe utilizar el Software Siproxd, para solucionar los problemas de VoIP a través de firewalls y NAT ya que el software utilizado es un Proxy SIP y RTP, además utiliza una dirección pública y soporta los 4 tipos de NAT.

3. Editar los parámetros de los archivos de configuración del Elastix, como son el sip.conf, rtp.conf y el sip_nat.conf, los cuales permitirán establecer la comunicación de la PBX que se encuentra detrás del NAT con las diferentes estaciones remotas, ya que así se encuentre correctamente configurado el Siproxd no se va a poder establecer la comunicación.

4. Tener presente que si se desea establecer mayor número de llamadas simultáneas se debe extender el rango de puertos UDP, puesto que para cada llamada se utilizan dos puertos.

5. Se debe tomar en cuenta el consumo de la memoria RAM del servidor ASTERISK, y este debe ser monitoreado constantemente ya que como se había analizado antes, este es un factor limitante en la cantidad de llamadas simultáneas.

RESUMEN

Se diseñó una propuesta de implementación de tráfico de VoIP con SIP en redes de datos a través de firewalls y NAT, para solucionar los problemas que presenta protocolo de señalización SIP cuando éste atraviesa redes nateadas y firewalls.

Para lo cual se implementó un ambiente de pruebas basado en el sistema Operativo GNU/Linux Ubuntu Server 10.10 debido a su capacidad de reconocimiento en los repositorios del software a ser usado, como por ejemplo quagga encargado del ruteo y Siproxd que se encarga de resolver los problemas de SIP a través de firewalls, mientras que para la PBX se eligió Elastix el mismo que se encuentra sobre un Centos.

Se simuló tres sucursales, la oficina A, B y C cada una con un firewall, el mismo que tiene NAT y Siproxd. Se realizaron las pruebas correspondientes en cuatro escenarios los mismos que son: Total operatividad, PBX sin Siproxd, cuando en el transcurso de la llamada se detiene el servicio y Siproxd detenido en uno de los Firewalls, obteniendo un resultado positivo con el 100% de efectividad ya que la utilización de Siproxd mejoró la disponibilidad del servicio de VoIP.

De acuerdo los resultados obtenidos en el ambiente de pruebas y tomando en cuenta que para solucionar el problema del NAT se necesita de un Proxy SIP, y que para solucionar la problemática del firewall se debe utilizar un Proxy RTP, logramos concluir que el software que cumple con estas características es el Siproxd, ya que permite alrededor de 30000 llamadas simultáneas, utiliza una IP pública y soporta cuatro tipos de NAT existentes.

Por lo tanto, se recomienda la utilización del software Siproxd para solucionar los problemas que presenta el protocolo SIP cuando éste atraviesa Firewalls y NAT ya que permite mejorar en su totalidad la disponibilidad del servicio de VoIP.

SUMMARY

We designed a proposal to implement VoIP with SIP traffic in data networks through firewalls and NAT to solve the problems with SIP signaling protocol when it crosses NAT boxes networks and firewalls.

For which we implemented a test environment based on GNU / Linux Ubuntu Server 10.10 due to its ability of recognition in the software repositories to be used, such as Quagga routing and Siproxd manager in charge of solving SIP problems through firewalls, while for the PBX Elastix chose the same found on a Centos.

Was simulated three branch office A, B and C each with a firewall, it has NAT and Siproxd. Was tested in four scenarios for them are: Total operation, PBX without Siproxd, when in the course of the call service is stopped and arrested Siproxd one of the firewalls, obtaining a positive result with 100% effectiveness since the use of Siproxd improved the availability of VoIP service.

According the results of the testing environment and taking into account that to solve the problem of NAT you need a SIP Proxy, and that to solve the problem of firewall to use a Proxy RTP, we conclude that the software meets with these characteristics is Siproxd, allowing about 30000 simultaneous calls, using a public IP and NAT supports five types of existing.

Therefore, we recommend the use of Siproxd software to solve problems in the SIP when it passes through firewalls and NAT as a whole can improve the availability of VoIP service.

ANEXOS

Anexo A. Instalación de Elastix:

Para la instalación de Elastix es necesario que tengamos un computador dedicado exclusivamente para estos fines. Aqui mostraremos el proceso completo de una instalación, paso a paso, juntos con algunos trucos y sugerencias que facilitarán el trabajo.

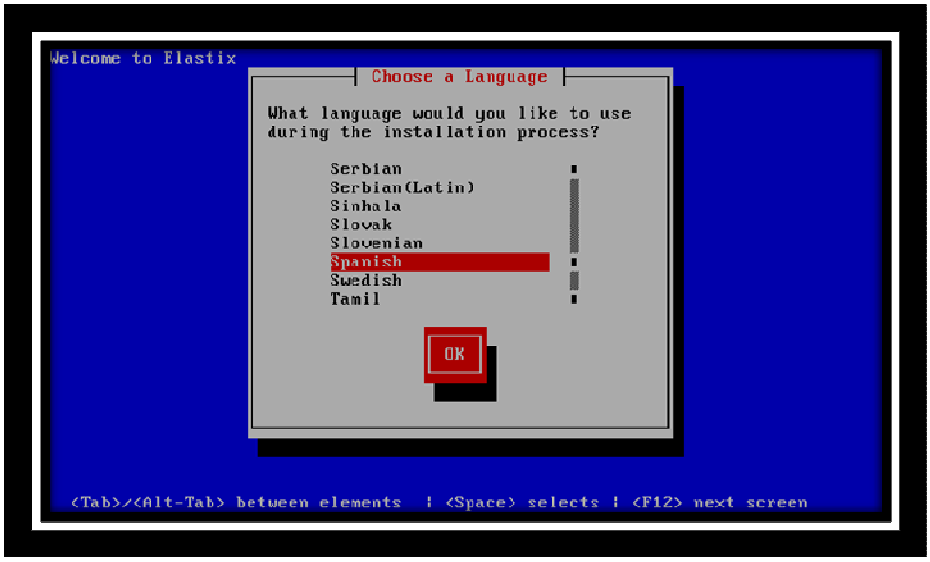
Instalación paso a paso

El siguiente procedimiento de instalación borrará todos los datos de su disco duro, razón por la cual le recomendamos hacer la instalación en un disco que no tenga información importante para usted. Luego de descargar la imagen que vamos a utilizar, necesitaremos un software que nos grabe esa imagen y a la vez convierta de ISO a formato normal para que lo podamos utilizar en un CD (hay en el mercado muchísimas herramientas que hacen esta labor).

Verificamos que nuestro computador en el BIOS tenga en el orden de arranque el CD-ROM o DVD-ROM en primer lugar. Luego, introducimos el CD y comenzamos nuestra instalación. Lo primero que veremos en la pantalla será el logo de Elastix con varias opciones para escoger, esta vez sólo le daremos a ENTER.

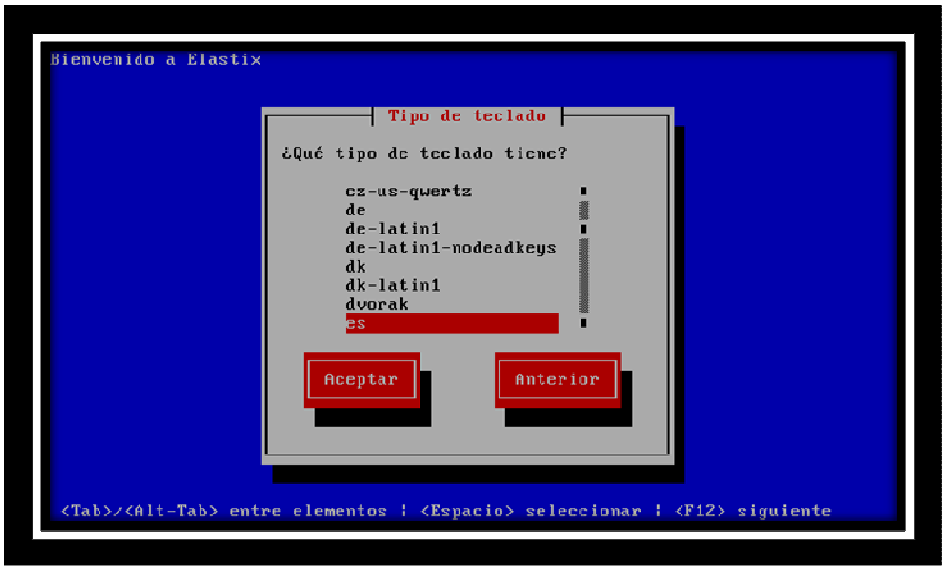
Pantalla de instalación del Elastix

Luego de esto, el sistema irá mostrando una serie de datos y parámetros hasta que llega a una pantalla donde nos pide seleccionar el lenguaje de nuestra instalación. Seleccionamos español y le damos a la tecla TAB hasta que nos coloquemos sobre el Ok.



Pantalla para la configuración del lenguaje

Luego nos va a pedir la configuración para nuestro teclado y seleccionamos a nuestro gusto para este caso la opción “es”.



Pantalla para la configuración del teclado

Posteriormente, entramos a una pantalla de recibimiento, donde se nos da la bienvenida a Elastix; damos un click en aceptar y luego nos lleva a una opción donde debemos seleccionar el tipo de la partición que queremos del disco duro y cómo queremos distribuir dichas particiones.

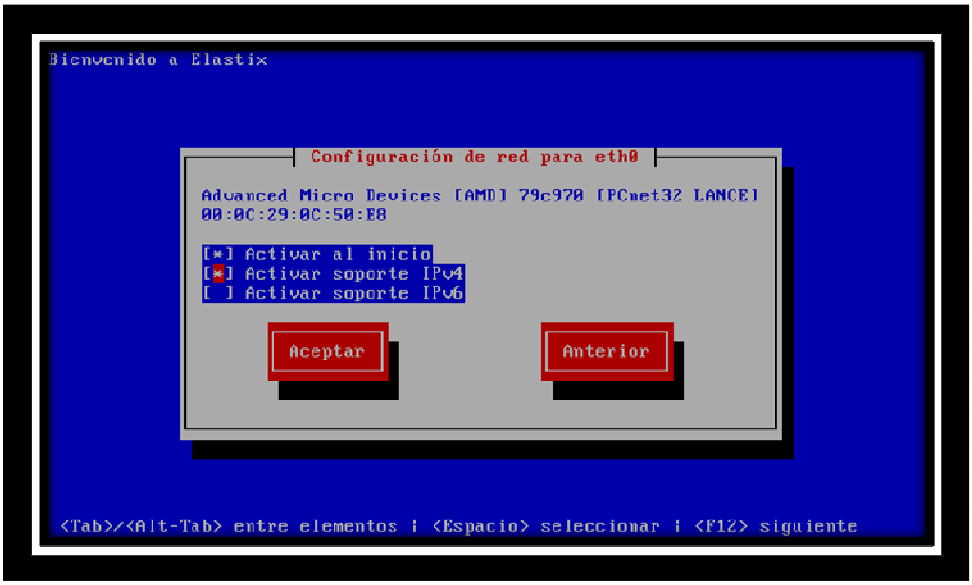
Lo recomendable es dejar que el sistema haga sus particiones automáticamente ya que viene optimizado para ello. En esta pantalla se recomienda seleccionar la primera opción que es "remover particiones en dispositivos seleccionados y crear disposición". Luego presionamos la tecla "TAB" hasta llegar a "Aceptar".

Pantalla para el particionamiento del disco duro

Cuando seleccionemos "Aceptar y presionemos "ENTER", nos saldrá un cuadro de aviso donde nos advierte sobre si estamos seguros que queremos borrar toda la información de todas las particiones, a lo que le responderemos que sí.

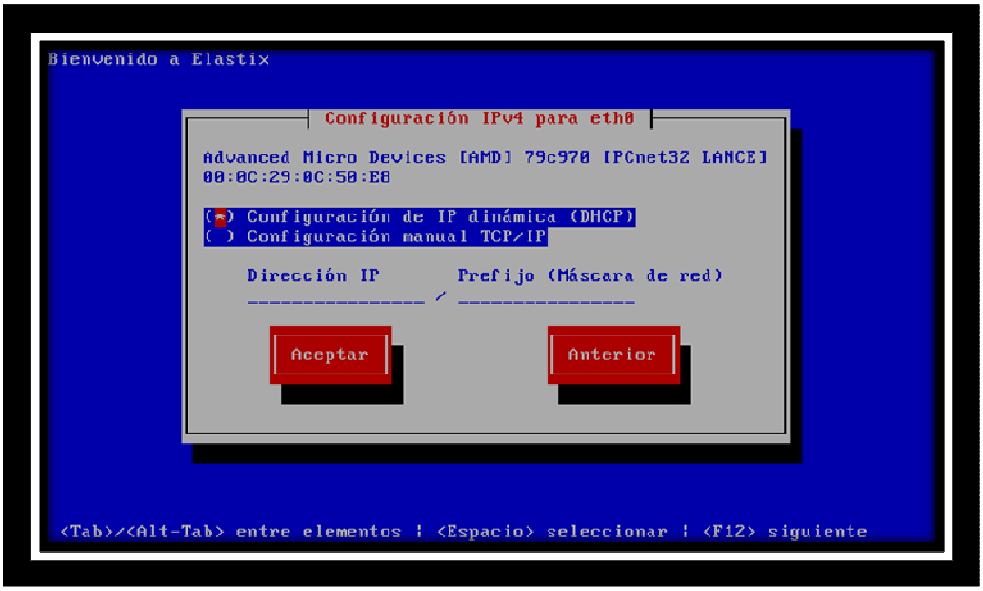
Pantalla de aviso antes de borrar todas las particiones

Finalizado esto, nos saldrá un mensaje preguntándonos si queremos revisar cómo han quedado las tablas de particiones y eso, le respondemos que no y seguiremos con la siguiente pantalla que es donde nos pide el gestor de arranque. Para la configuración de red para la eth0 seleccionamos la opción “Activar soporte IPv4” le damos "TAB" y luego "Aceptar".



Pantalla de configuración de la red para eth0

En la siguiente pantalla que viene a continuación se debe seleccionar la primera opción, la misma que viene por defecto.

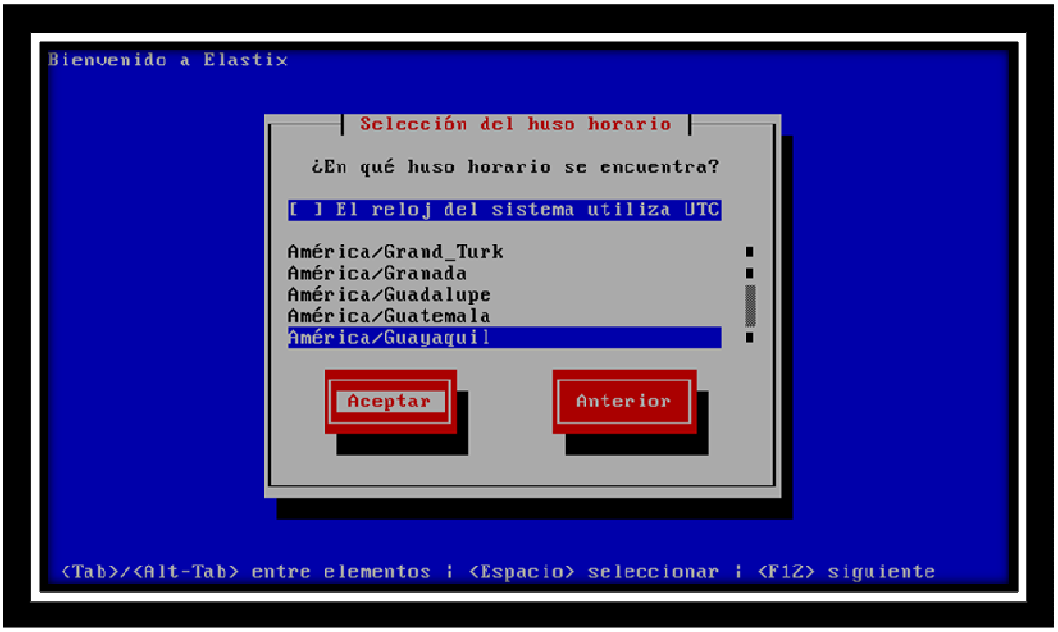


Pantalla de configuración de IPV4 para eth0

En la pantalla que tiene q ver con la configuración del nombre del host se debe escoger la opción “manualmente”, le damos "TAB" y luego "Aceptar".

Pantalla de configuración del nombre del host

Cuando terminemos de esas dos pantallas, nos saldrá la opción de seleccionar en el huso horario el país correspondiente, en nuestro caso seleccionaremos "América Guayaquil" y seguimos adelante.



Pantalla de selección del uso horario

Después de esto, nos saldrá un cuadro donde nos pide que le asignemos una contraseña al usuario root, que es el administrador del sistema. Es muy importante que no pierda esta contraseña ya que podría terminar con una reinstalación de todo el sistema si esto ocurre.

Finalizado ese cuadro, entonces llegaremos a uno donde se nos pide qué paquetes queremos instalar; este cuadro lo dejaremos intacto y solamente nos vamos a “Aceptar”. Nota: no toque ninguna opción de ese cuadro porque de ello depende el buen funcionamiento de nuestra PBX Elastix.

Pantalla para ingresar la contraseña del root

Pantalla del Pakege Group Selection

El sistema comenzará a hacer unas rutinas de preparación, verificando dependencias, paquetes, etc. Cuando esto finalice nos llevará a una ventana donde se nos dirá que

todas las actividades del proceso de instalación estarán disponibles en un archivo de log cuando el sistema lo hayamos puesto a arrancar.

Luego comenzará con el formateo de las particiones ya creadas y los sistemas de archivos. Al término de esto, veremos una pantalla donde se mostrarán las instalaciones de cada uno de los paquetes que componen a Elastix.

Pantalla de instalación del Elastix

Cuando la barra de progresión de la parte de abajo llegue al 100%, entonces ya tendremos nuestro sistema instalado completamente. El sistema se reiniciará y cuando vuelva a subir nos mostrará una pantalla similar a la pantalla inicial que vimos cuando introdujimos el CD de instalación. En esta fase del proceso de instalación, lo único que se nos mostrará son dos opciones para el arranque. Debemos siempre entrar en la opción que viene por defecto que es la "Elastix-base", la otra opción del "Kernel Xen" no la vamos a necesitar y su alcance está fuera de lo expuesto en este documento

Pantalla del boot de la PBX

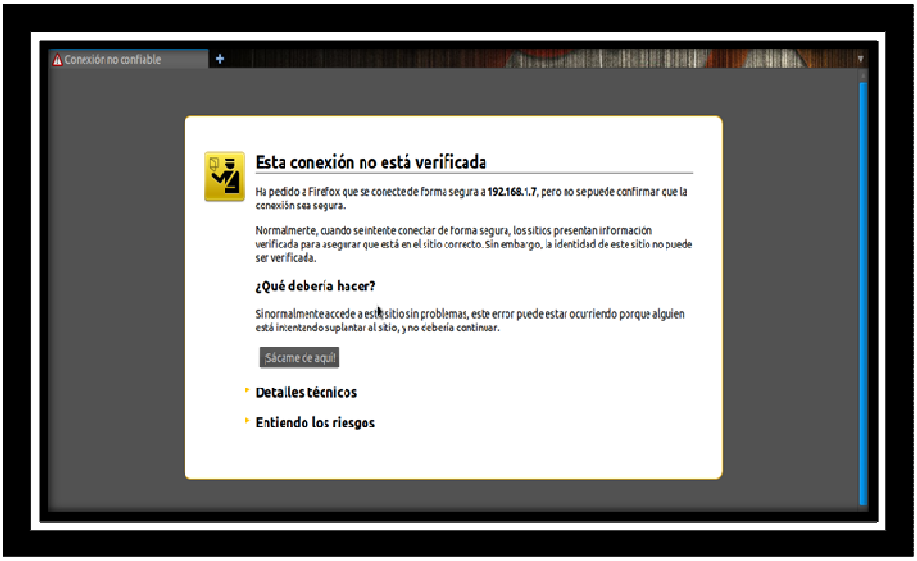
Luego de haber entrado en la opción "Elastix-base" (nota: él siempre arrancará en esta versión sin que sea necesaria nuestra intervención, por lo cual deberá sentirse tranquilo), nuestra PBX ejecutará una serie de procesos de arranque y scripts de inicio hasta que finalmente arribemos a la pantalla de bienvenida.

Anexo B. Configuración de Elastix

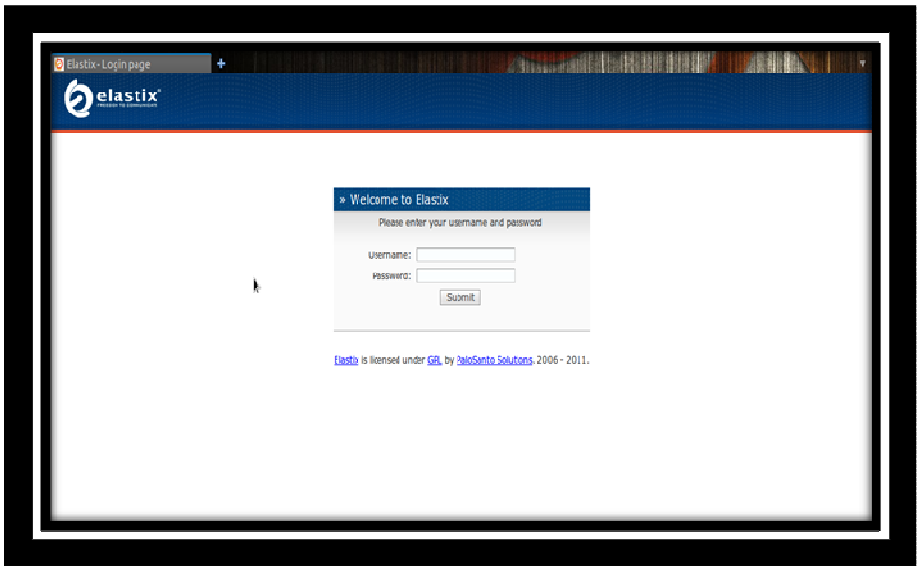
Antes de dar el primer paso, debemos estar seguros si nuestra central y el computador que estamos utilizando para acceder a la misma se pueden comunicar entre ellos a nivel de red.

Asumiendo que sí, lo que tenemos que hacer entonces es abrir un explorador y en el mismo colocar la dirección IP que le hayamos asignado a nuestra central. Inmediatamente nos saldrá una advertencia donde nos dice que no conoce esa

entidad emisora de certificados (lo que sucede es que Elastix se comunica por SSL, que es la conexión segura y emite un certificado), le damos que sí a todas las advertencias que nos hace acerca de seguridad y luego nos debe llevar a la página de inicio de Elastix, donde nos pregunta por usuario y password.



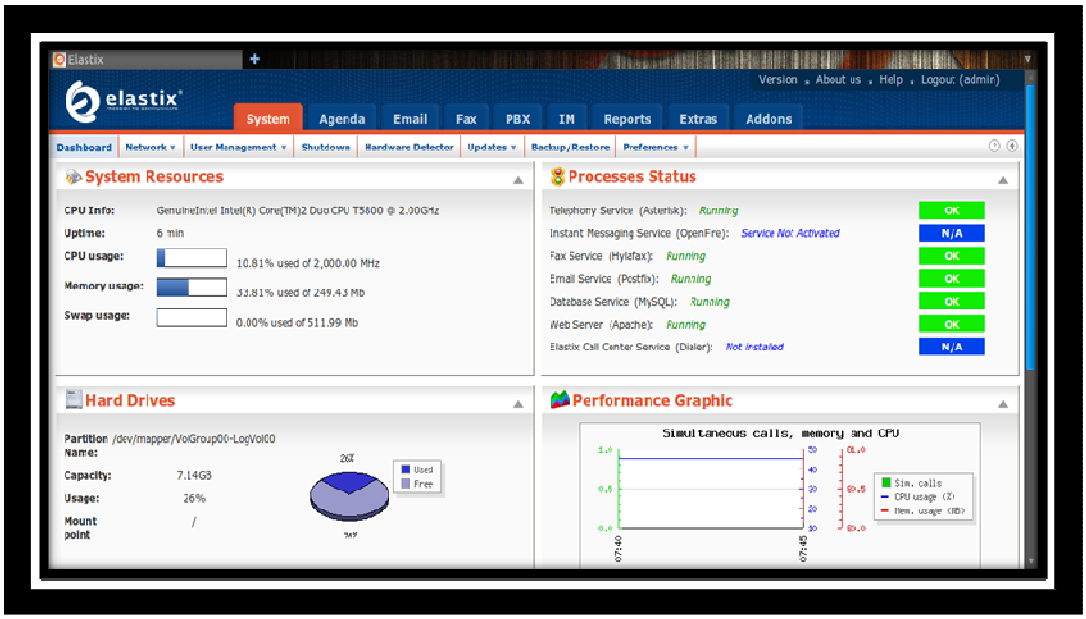
Pantalla de la página del Elastix



Pantalla de logeo del Elastix

Dashboard

Introducimos el username "admin" y el password "240124". Luego de esto, nos aparece la ventana Dashboard, desde donde guiamos a la PBX. En este dashboard podemos tener un resumen de las actividades principales de nuestra PBX, como lo son Llamadas, Emails, Faxes, Voicemails, Eventos del calendario y Emails del sistema.



Pantalla del dashboard

Preferences

Al inicio la administración Web de Elastix sigue en inglés, aunque hayamos instalado el sistema en español. Para corregir esto nos vamos a la pestaña preferences, que está ahí mismo bajo el menú de System y seleccionamos el idioma español del listado de Idiomas.

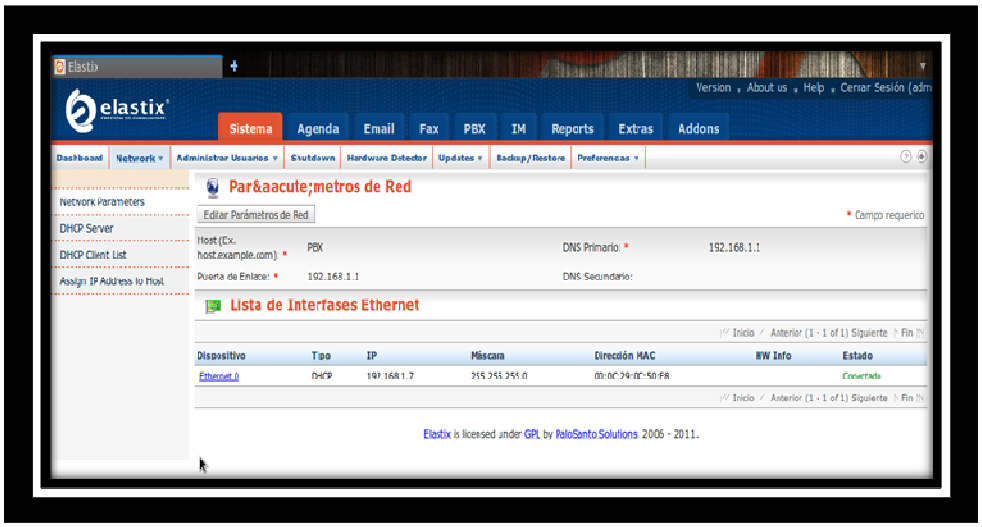
En esta misma pestaña, podemos hacer varios cambios importantes como son: Fecha y Hora, apariencia del sistema, y Currency, el cual se refiere al tipo de moneda que queremos que el sistema utilice. La fecha y la hora es importante que estén bien

ajustadas y configuradas ya que hay muchos eventos que la PBX utiliza basándose en la disponibilidad de esa información.

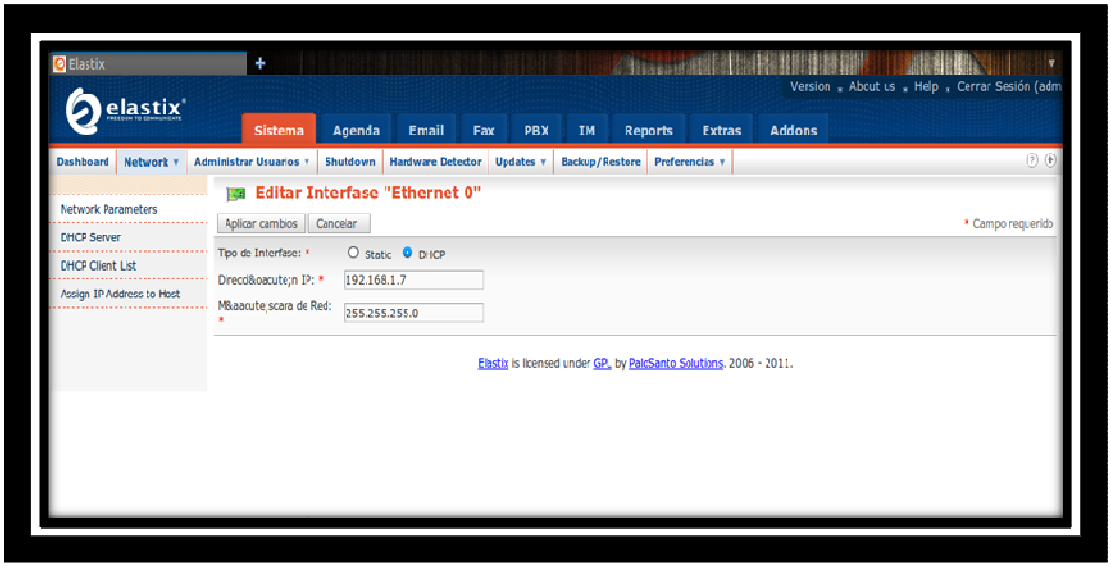
Pantalla de la opción System de Elastix

Network

Aquí podemos hacer cambios de nuestros parámetros de red por medio de la interfaz gráfica, si queremos cambiar los valores de nombre de equipo (hostname), servidores DNS, puerta de enlace, sólo debemos dar click al botón de "Editar parámetros de Red". Para cambiar parámetros como dirección IP y máscara de red, se debe dar click sobre " Ethernet 0 ", el cual está debajo de "Lista de Interfases Ethernet".



Pantalla de la opción network de Elastix

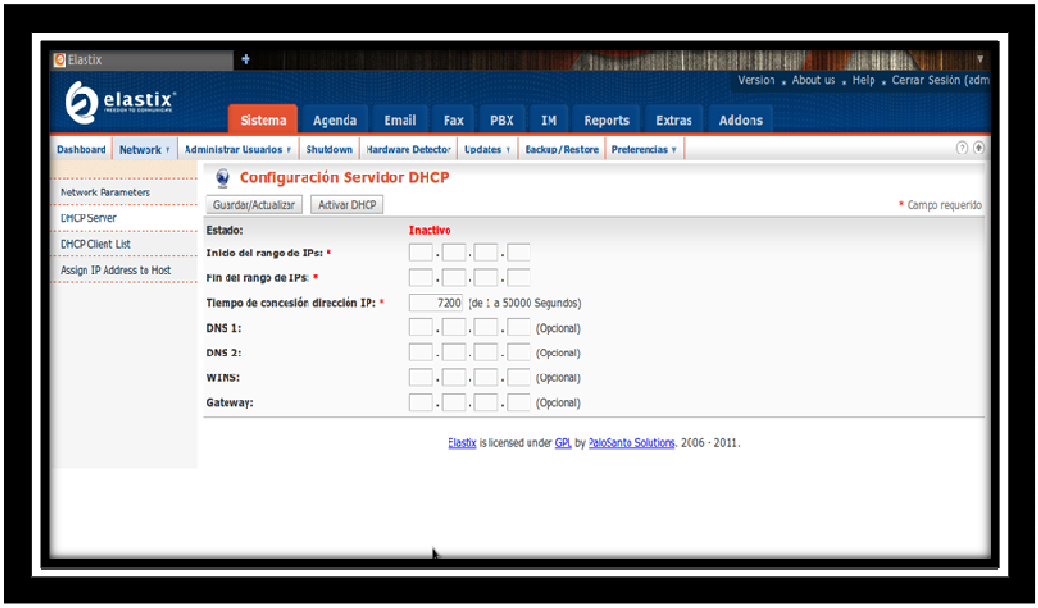


Pantalla para editar la interfase “Ethernet 0”

Servidor DHCP

Este servicio es de suma importancia si queremos asignar de forma automática direcciones a los demás equipos de nuestra red como son: Teléfonos IP, ATAs, etc. Sólo debemos ver qué rango es que queremos asignar, el tiempo que deseamos que los clientes mantengan esas IP antes de hacer una nueva petición al servidor, servidores DNS externos o de nuestra propia red, servidores WINS, y la puerta de enlace predeterminada.

Una vez hayamos llenado todos estos valores, sólo es cuestión de presionar el botón de "iniciar servicio" y listo: ya tenemos un servidor DHCP corriendo en nuestra red, ¿quién dijo que Linux no era fácil?



Pantalla de configuración del Servidor DHCP

Cargar Menú

Esta parte la explicaremos más adelante cuando carguemos el módulo de callcenter.

Apagar

Esta es una forma fácil de apagar y reiniciar el sistema, debemos tener cuidado con esta parte cuando estemos trabajando con sistemas en producción.

Detección de hardware

Hablaremos de esto en detalle en la parte de creación de troncos Zap (actualmente Dahdi).

Actualizaciones

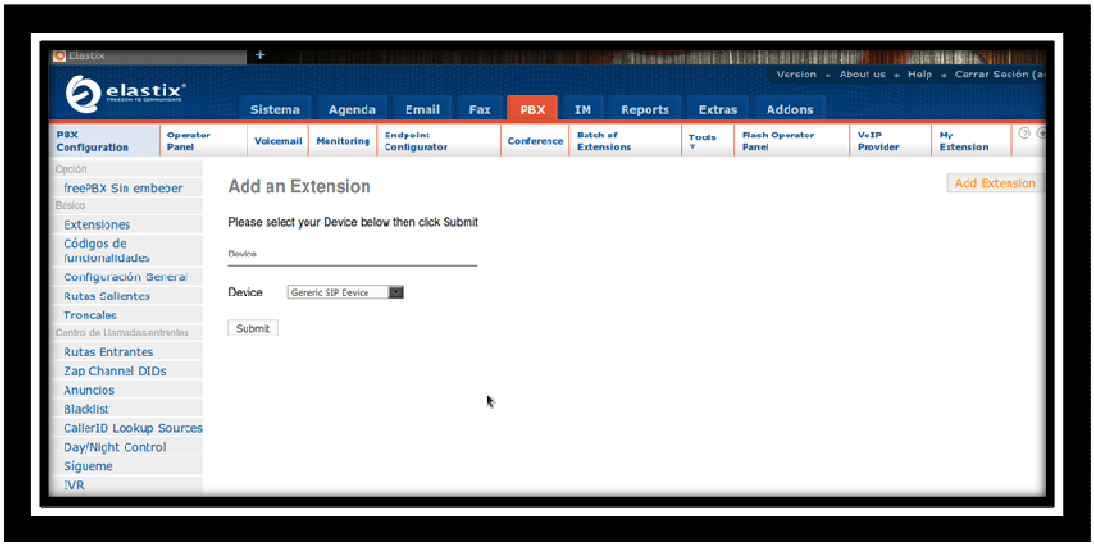
Esta parte es muy importante ya que nos presenta todos los paquetes instalados del sistema

Configuración PBX

Opción y Básico

Lo primero que haremos es ir a la pestaña que dice PBX y daremos click sobre ella. Ahí nos encontraremos con un amplio e intimidante menú que estaremos detallando a continuación.

Lo primero que veremos es una opción que nos dice freePBX sin embeber, esta es el alma de Elastix, mientras Freepbx es el motor de gestión de la central completa. Los creadores de Elastix, desarrollaron una versión resumida de Freepbx. Aunque a la vez, nos dejaron una versión normal y sin modificar para que podamos hacer cosas que no se pueden hacer con la versión de Elastix. Esta parte la veremos más adelante y también mostraremos muchas opciones que se pueden hacer por Freepbx.



Pantalla de configuración de la PBX

Creando Extensiones

Ahora vamos a tratar una parte muy interesante: la creación de extensiones, o como dicen en gran parte de centro y sur América: “creación de anexos”. Lo primero que veremos será la opción de crear extensiones tipo SIP.

Nota: “SIP es un protocolo desarrollado por el IETF MMUSIC Working Group con la intención de ser el estándar para la iniciación, modificación y finalización de sesiones interactivas de usuario, donde intervienen elementos multimedia como el video, voz, mensajería instantánea, juegos online y realidad virtual”. No es la mejor de las opciones pero es la más difundida y estandarizada.

En la parte de extensiones nos saldrá la opción de crear “Generic Sip Device”, sólo presionaremos el botón “submit” y nos presentará una serie de campos para ser llenados por nosotros.

User Extensions: es el número de la extensión que vamos a asignar, por ejemplo: 2500, etc.

Display Name: es el nombre que aparece en una extensión vecina cuando marcamos hacia ella, por ejemplo: Daniel Morales

CID Num Alias: este es una máscara para el número que tenemos, por ejemplo: si tenemos un grupo de Timbrado o el departamento técnico tiene 5 usuarios, pero cada vez que alguien llama al departamento de soporte nos interesa que se marque la extensión 2020, procederemos a colocarles a los cinco usuarios de nuestro

departamento el número 2020 en este campo y las personas que reciban las llamadas creerán que todas vienen de esa extensión.

SIP Alias: si usted desea asignar un nombre a una extensión para que otras extensiones SIP puedan marcarle de esta forma, aquí es que debe ser colocado. SIP soporta el marcado por nombre, además de la marcación numérica, es decir, que en vez de SIP/2500 podemos utilizar SIP/Daniel y funciona de la misma manera.

Outbound CID: en este campo podemos colocar un caller-id (identificador de número) diferente al de nuestra central cuando estemos marcando fuera de nuestra central. Es decir, que aquí puedo sobrescribir el caller id de mi central con el que tenga puesto aquí. Para esto el proveedor debe soportar este procedimiento para que funcione correctamente.

Ring Time: tiempo que debe timbrar una extensión antes de entrar al buzón de voz, por lo general, esta opción no se configura sino que se toma del valor que ya está expresado en general settings.

Call Waiting: se usa para llamadas en espera. Es de suma importancia que esta opción esté habilitada (enable), porque de aquí depende que nuestro teléfono pueda recibir otra llamada cuando tengamos la línea ocupada.

Call Screening: esta función permite que cuando un usuario nos llama desde fuera a nuestra extensión, se le requiera grabar su nombre para luego la central transferirnos dicha grabación, dándonos la opción de aceptar o rechazar la llamada. Existe también el Call Screening con memoria (Memory). Lo que este último hace es, poner al sistema a requerir la grabación del nombre de la persona que nos llama por primera vez. Ya

con su nombre y número registrados, cuando aquella vuelva a marcar desde ese mismo número, la PBX simplemente verificará su caller id y no le requerirá que grabe su nombre sino que a nosotros nos pondrá la última grabación que se haya hecho desde ese número.

Emergency CID: este es un Caller Id que se utilizará solamente cuando hagamos una llamada de emergencia como al 911, por ejemplo. Aquí podemos especificar otro número diferente.

DID Description: este es un campo solamente descriptivo, se utiliza para hacer una descripción del DID. Hagamos un paréntesis para definir lo que es DID. DID: Direct Inward Dialing (también llamado DDI en Europa), es un servicio ofrecido por las compañías telefónicas para ser usado con los sistemas de central telefónica de los clientes, en donde la compañía telefónica (telco) asigna un rango de números asociados con una o más líneas telefónicas.

Su propósito es permitir a una empresa asignar un número personal a cada empleado, sin requerir una línea telefónica separada por cada empleado. De esta manera, el tráfico telefónico puede ser segmentado y administrado más fácilmente. DID requiere que se compre una línea RDSI (ISDN) o Digital y que se pida a la compañía telefónica que asigne un rango de números. Luego se necesitará en sus instalaciones el equipo respectivo, el cual consiste de tarjetas BRI, T1 o E1.

Add Inbound DID: este campo sirve para agregar un DID directamente a esta extensión cuando estemos marcando hacia afuera.

Add Inbound CID: se usa en conjunto con “Add Inbound CID”.

This device uses sip technology: aquí es que se define el tipo de tecnología que estamos usando, esto es de vital importancia, ya que más adelante veremos que este es el único campo que cambia cuando estemos creando otro tipo de extensión.

Secret: esta es la contraseña que debemos asignar a la extensión que creemos. Debe ser una clave recordable ya que la utilizaremos posteriormente cuando configuremos una extensión. Por lo general, caemos en el error de asignar el mismo número de extensión como clave. Para un entorno de pruebas esto no sería problemas, pero debemos tener cuidado de incurrir en esta práctica en sistemas en producción.

Dtmfmode: (Dual Tone Multifrecuency) Multifrecuencia de doble tono. Tonos en diferentes hertz que utilizan una telefonía para marcar números. Cada número u opción del teléfono tiene un tono propio que es identificado en la telefonía. Este campo puede tener cuatro opciones: inband, rfc2833, info y auto.

Le recomendamos que utilice la opción que viene por defecto. Si quiere investigar acerca de la utilidad y función particular cada método, le dejamos todas las opciones abiertas. Sólo le diremos que, cuando esté configurando un proveedor de Voz Sobre IP con troncos SIP, este modo debe estar preferiblemente en info(dtmfmode=info).

Language Code: con esta opción, si tenemos las voces instaladas en español e inglés al mismo tiempo, cuando especifiquemos “es” todos los avisos o anuncios se escucharán en español, como son los de buzón de voz, etc.

Record Incoming: esta opción sirve para grabar todas las conversaciones salientes si seleccionamos “always”, o no grabar nunca si seleccionamos “never”. Por defecto

viene “On Demand”, o sea, que podemos decidir cuándo grabar, inclusive si estamos en medio de una conversación.

Status: está dentro de Voicemail & Directory, sirve para habilitar el uso de buzón de voz a la extensión, por defecto viene deshabilitado.

Voicemail Password: se trata de la contraseña del buzón de voz, la que el usuario debe utilizar para recoger sus mensajes. Esta clave sólo puede ser numérica y el usuario puede cambiarla cuando entra al menú de su buzón de voz.

Email Address: es el correo donde los mensajes de voz serán enviados una vez recibidos, los mensajes son anexados en formato Wav.

Pager Email Address: este correo sólo sirve para recibir notificaciones cortas acerca de que tiene un mensaje de voz en su buzón. Esto es ideal para cuando queremos recibir sólo una notificación en un celular o un Blackberry.

Email Attachment: esta es la opción que nos permite anexar o no el mensaje que recibamos en el buzón de voz.

Play CID: se trata de la opción que nos anuncia el teléfono o la extensión de la persona que nos dejó el mensaje de voz.

Play Envelope: tener esta opción habilitada nos permite escuchar la fecha y la hora en la que la persona nos dejó el mensaje de voz.

Delete Voicemail: si esta opción está habilitada, todos los mensajes de voz serán enviados por correo y después serán automáticamente borrados. Debe tener cuidado

porque una vez que han sido enviados ya no se pueden recuperar ni desde la interfaz web ni marcando desde una extensión.

VM Options: sirve para pasar parámetros a las opciones de buzón de voz como cantidad máxima de mensajes, zona horaria, etc., por ejemplo: maxmessage=60|maxlogins=3. Etc.

VmX Locater™: cuando esta opción es habilitada el usuario tiene control sobre sus mensajes de voz y de su buzón, mediante el portal Web ARI (Asterisk Recording Interface). Con este portal el usuario puede ver sus grabaciones de voz, reenviar sus mensajes de voz, etc. Para esto necesita tener creado un usuario, cosa que veremos más adelante.

Voicemail Instructions: cuando no está habilitada, la persona que nos va a dejar un mensaje de voz sólo escuchará un pito (beep). Cuando está seleccionada utilizamos los avisos o anuncios por defecto que trae el sistema.

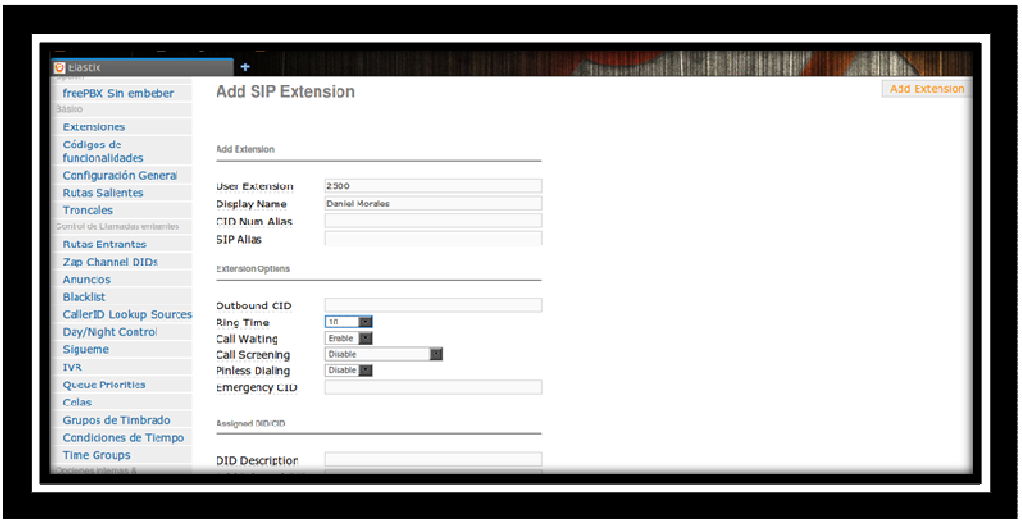
Opción Alfio©: Press 0: esta opción se usa para que la persona que llama pueda presionar el cero y ser redirigida a la recepción, sin tener que escuchar completo el saludo de bienvenida de nuestro buzón de voz. Esto se puede costumizar con la extensión que queramos. Press 1: hace la misma función, pero por lo general, podemos colocar aquí nuestro celular u otro número externo. Press 2: se refiere a lo mismo que las anteriores opciones.

Creando una extensión SIP

Ya que hemos explicado casi todas las funciones y opciones de una extensión SIP en Elastix, vamos a crear una extensión SIP tal como se muestra en la pantalla. Primero vamos a crear la extensión SIP 2500, para esto sólo debemos agregar este número en el campo “User Extensions”, luego en el “Display name” ponemos Daniel Morales.

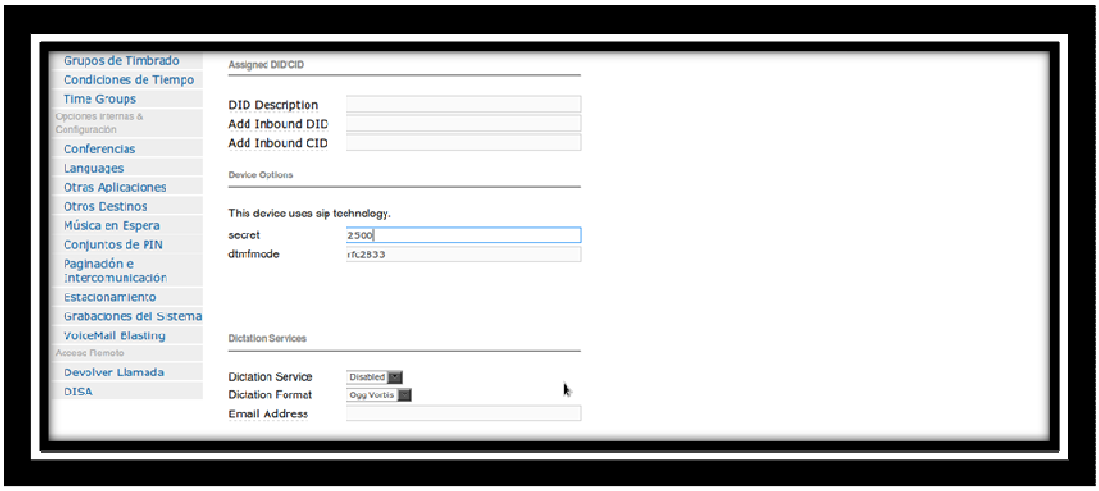
Después de esto, nos vamos al “secret” y colocamos 2500 como clave. Seguimos hacia abajo y habilitamos la opción de buzón de voz y le agregamos como clave el número de la extensión. Con estas opciones es más que suficiente por ahora.

Vamos a la parte del fondo y le damos a “Submit”. Luego de esto, nos aparece en la parte superior de la página un cintillo o banda de color rosado claro que dice: “Apply Configuration Changes Here”, damos click sobre dicha banda (la cual debe desaparecer después de haber dado click) y listo.



Pantalla para agregar una extensión SIP

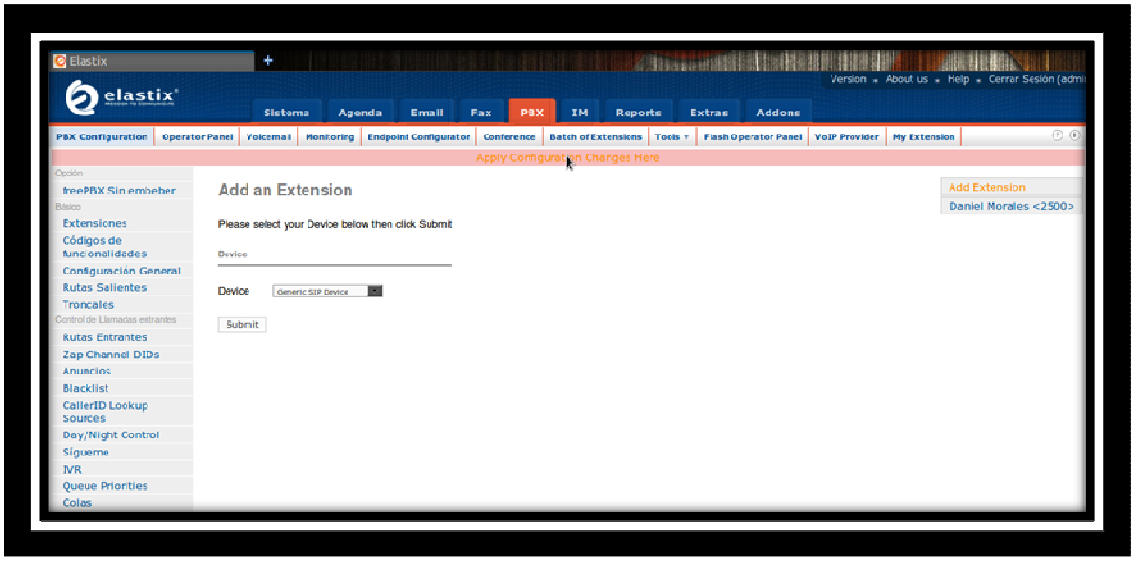
En las siguientes pantallas se detalla cómo deben llenarse cada uno de los campos para así poder crear una extensión SIP:



Pantalla para agregar una extensión SIP

Pantalla para agregar una extensión SIP

Con esta pantalla podemos comprobar la correcta creación de la extensión SIP:



Pantalla para verificar la creación de la extensión SIP

Nota: es un error muy común entre los usuarios nuevos de Elastix olvidarse de darle click a la banda de "Apply. Configuration Changes Here" Hasta que no demos click sobre esta banda, los cambios no se van a reflejar en nuestro sistema.

Como ya sabemos entrar en nuestra PBX en modo texto mediante la herramienta Putty, intentamos en la consola del shell de Linux el siguiente comando "asterisk -r". Este comando nos lleva a la consola de administración de asterisk en modo texto. Una vez dentro, ejecutamos el siguiente comando "sip show peers" y nos debe reflejar una entrada como*: 2500 (Unspecified) D N 0 UNKNOWN.*

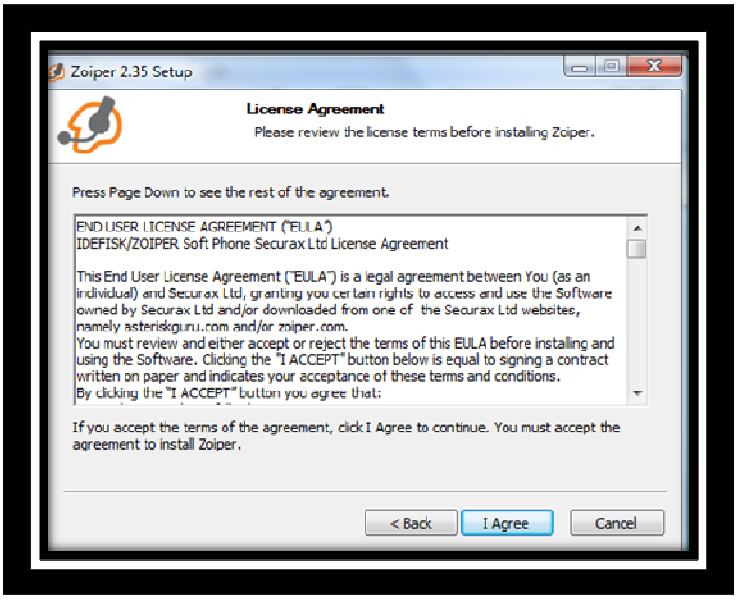
Esta presentación obedece a que tenemos la extensión ya creada pero no tenemos ningún dispositivo con dicha extensión asignada ni registrada en nuestra PBX Elastix.

Anexo C. Instalación y configuración del softphone “ Zoiper”

A continuación se presentan las pantallas para la instalación del Softphone Zoiper:

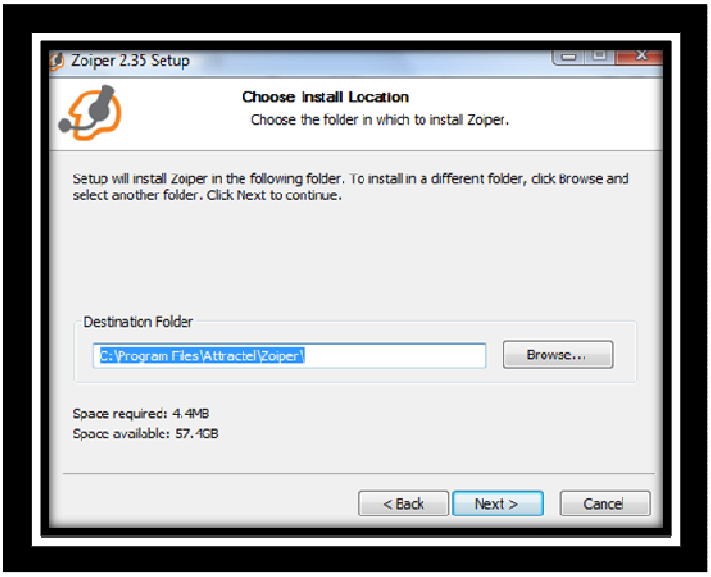
Pantalla de bienvenida a la instalación de Zoiper

En esta pantalla aceptamos los términos expuestos en la licencia y empezamos la instalación del Softphone:



Pantalla de los acuerdos de la licencia

En esta pantalla elegimos la extensión o la carpeta en donde se va a instalar el Softphone:

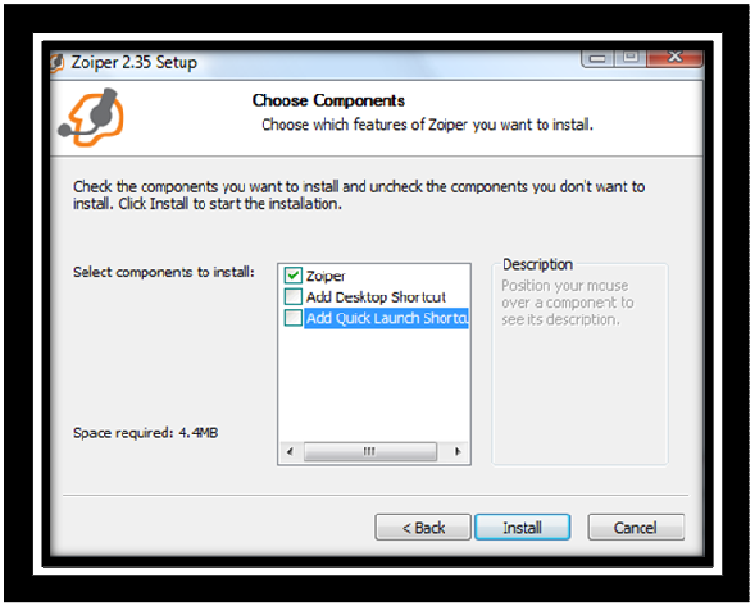


Pantalla de la extensión en donde se va a instalar Zoiper

Elegimos una carpeta en el menú de inicio para ubicar los accesos directos del Softphone Zoiper y damos click en next.

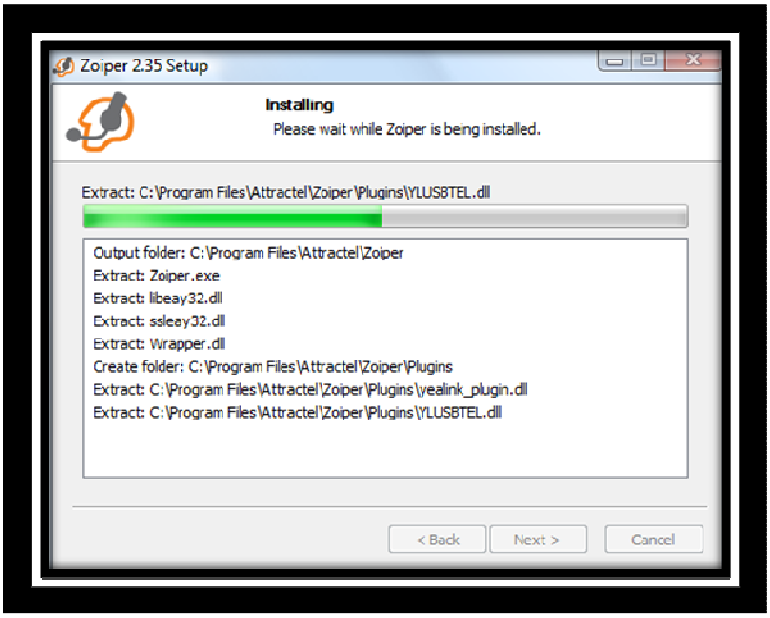
Pantalla de la carpeta del Menú Inicio

En ésta pantalla se deben seleccionar los componentes requeridos y luego damos un click en install:



Pantalla de elección de los componentes

La siguiente la pantalla es cuando el softphone ya se encuentra en proceso de instalación:



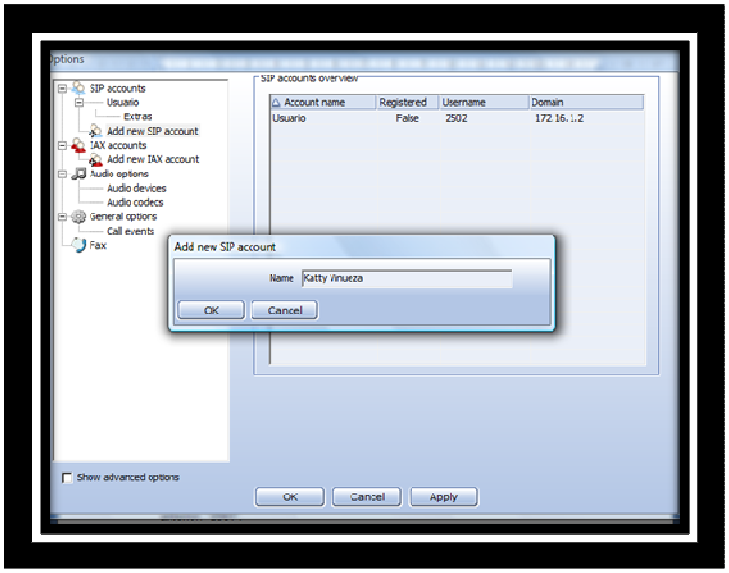
Pantalla del proceso de instalación

Esta pantalla aparecerá cuando el Softphone haya terminado su instalación.

Pantalla final de la instalación de Zoiper

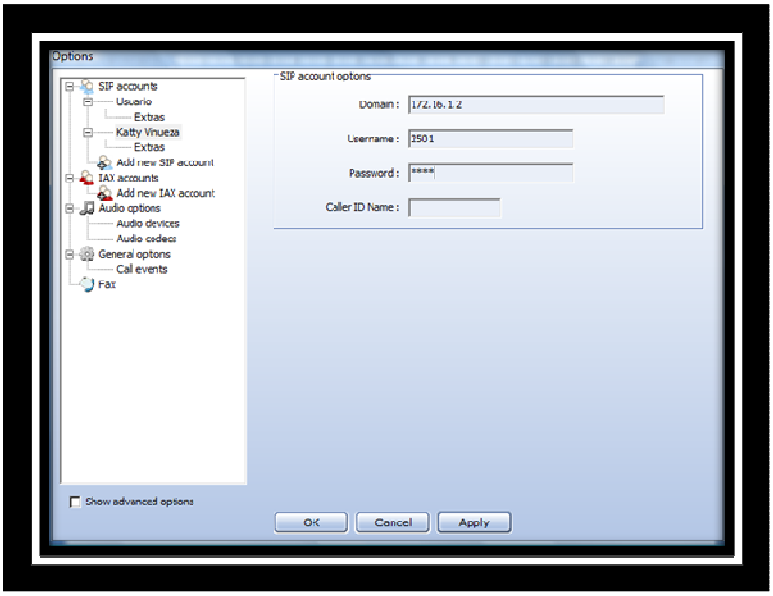
Una vez instalado, vamos a la parte superior donde hay tres opciones “Zoiper” “Contacs” y “Help”. Elegimos la primera que es “Zoiper” y ahí dentro seleccionamos “preferences”. Dentro de “preferences” nos aparecen una serie de opciones a configurar.

Si usted se fija bien, Zoiper puede soportar tanto el protocolo SIP como IAX2, a nosotros nos interesa SIP, por lo tanto, vamos a donde dice “Add new SIP account” y ahí, como por arte de magia, nos aparecerá un cuadrito donde pondremos un nombre descriptivo a la cuenta que vamos a crear en este caso Katty Vinueza.



Pantalla para la creación de una cuenta SIP

Luego, nos lleva a un cuadro de configuración que es súper sencillo, en donde nos pregunta “Server Hostname/IP”, ahí agregamos la dirección IP de nuestra central Elastix. La misma es 172.16.1.2. En el campo de abajo nos pregunta “Username”, ahí colocamos nuestro número de extensión que ya habíamos creado en el capítulo anterior: “2501”.



Pantalla de configuración de parámetros de la cuenta

ANEXO D. Encuesta

ENCUESTA

Objetivos:

La presente encuesta tiene como finalidad la demostración de la hipótesis del tema de Investigación.

Instructivo:

Señor/a encuestado tenga la bondad de responder la pregunta que usted crea conveniente marcando con una X, recuerde que de su respuesta depende la veracidad de esta investigación.

Cuestionario:

1. **¿Ha utilizado en alguna ocasión la tecnología de VoIP?**
   SI _____ NO____
2. **¿Conoce el protocolo de señalización SIP?**
   SI _____ NO____
3. **¿Conoce los problemas que presenta el protocolo SIP a través de firewalls?**
   SI _____ NO____
4. **¿Conoce los problemas que presenta el protocolo SIP cuando atraviesa NAT?**
   SI _____ NO____
5. **¿En el Ambiente de Pruebas sin la utilización del software Siproxd en los firewalls, se establece la comunicación?**
   SI _____ NO____
6. **¿En el Ambiente de Pruebas con la utilización del software Siproxd en los firewalls, se establece la comunicación?**

SI _____ NO____

Los resultados obtenidos en la encuesta son los siguientes:

PREGUNTA	SI	NO
Preg. 1	10	0
Preg. 2	8	2
Preg. 3	5	5
Preg. 4	4	6
Preg. 5	0	10
Preg. 6	10	0

Resultados de la encuesta

Para la demostración de la hipótesis se ha utilizado la pregunta cinco y seis, debido que en esas preguntas se observa si el software Siproxd resuelve o no el problema del protocolo SIP a través de Firewalls y NAT.

BIBLIOGRAFÍA

LIBROS

1. DEMSTER, B. y GOMILLION, D; Building Telephony Systems with Asterisk; 2da ed.; Manchester; Packt Publishing; 2009; pp.25-78.
2. GONCALVES, F; Building Telephony Systems with OpenSER; 2da ed.; Birmighan; Packt Publishing; 2008; pp. 7-288.
3. LANDIVAR, E; Comunicaciones Unificadas con Elastix; 2da ed.; Mexico.; CITEM.; 2009; pp. 7-125.
4. MUÑOZ, A; Elastix a Ritmo de Merengue; 2da ed.; República Dominicana; Corsami.; 2010; pp. 11-75.

BIBLIOGRAFÍA DE INTERNET

1. **NAT.**

   www.alumni.caltech.edu/~dank/peer-nat.html

   2010-11-27

2. **OPENSER**

   www.opensips.org/

   2011-01-04

3. **OPENSER**

   www.voip-info.org/wiki/view/OpenSER

   2011-01-29

4. **OPENSER**

   www.openser.com/

   2011-01-25

5. **OPENSER**

   www.kamailio.org/w/

   2011-02-27

6. **RTP**

   http://en.wikipedia.org/wiki/Real-time_Transport_Protocol

   2011-02-12

7. **RTP**

   www.voip-info.org/wiki/view/RTP

   2011-02-13

8. **RTP**

   www.ietf.org/proceedings/51/slides/avt-6/sld001.htm

2011-02-04

9. SIP

http://es.wikipedia.org/wiki/Session_Initiation_Protocol

2010-12-27

10. SIP

http://www.voip-info.org/wiki/view/SIP

2011-01-08

11. SIPROXD

http://siproxd.sourceforge.net/index.php?op=odoc

2011-01-05

12. SIP NAT

www.voipuser.org/forum_topic_7295.html

2011-01-19

13. SIP NAT

http://saghul.net/blog/2010/02/28/ice-%C2%BFla-solucion-definitiva-al-nat-en-sip/

2011-01-04

14. SIP NAT

www.linuxjournal.com/article/9399

2010-12-28

15. SIP NAT

http://freshmeat.net/articles/nat-traversal-for-the-sip-protocol

2011-01-06

16. SIP NAT

www.asteriskguru.com/tutorials/sip_nat_oneway_or_no_audio_asterisk.html

2011-01-18

17. STUN

http://es.wikipedia.org/wiki/STUN

2011-01-09

18. STUN

http://en.wikipedia.org/wiki/Session_Traversal_Utilities_for_NAT

2011-01-22

19. STUN

www.voip-info.org/wiki/view/STUN

2011-02-01

20. STUN

www.3cx.es/voip-sip/stun-server.php

2010-12-16

21. TCP CONNECTIONS

www.reports-archive.adm.cs.cmu.edu/anon/isri2005/CMU-ISRI-05-104.pdf

2010-11-14

22. TRADITIONAL IP NETWORK ADDRESS TRANSLATOR

www.faqs.org/rfcs/rfc3022.html

2010-10-24

23. TRAVERSING FIREWALLS AND NATS WITH VOICE AND VIDEO OVER IP

www.tandberg.net/collateral/white_papers/WR-trans-firewalls-nats.pdf

2010-12-08

24. VOZ SOBRE IP

http://es.wikipedia.org/wiki/Voz_sobre_IP

2010-12-26