

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL
FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN

INFORME DE MATERIA DE GRADUACIÓN
“SISTEMAS VOIP USANDO SOFTWARE LIBRE”

**“Implementación del protocolo Dundi para conectar dos o
más PBX basadas en Asterisk remotamente ubicadas”**

Previa a la obtención del Título de:

INGENIERO EN TELEMATICA
INGENIERO EN COMPUTACION ESPECIALIZACION SISTEMAS
TECNOLOGICOS

Presentada por:

ANDRES ORLANDO FLORES SOTO
CHRISTIAN XAVIER YEPEZ CASTILLO

GUAYAQUIL – ECUADOR

AÑO

2010

AGRADECIMIENTO

Nuestro sincero agradecimiento a todas las personas que de una u otra forma brindaron su colaboración para la realización de este trabajo, especialmente a la Ing. Rebeca Estrada directora de la materia de graduación y al Ing. Gabriel Astudillo.

DEDICATORIA

A NUESTROS PADRES

DECLARACIÓN EXPRESA

“La responsabilidad del contenido de esta Tesis de Grado, nos corresponde exclusivamente; y el patrimonio intelectual de la misma a la Escuela Superior Politécnica del Litoral”.

(Reglamento de Graduación de la ESPOL)

Andrés Orlando Flores Soto

Christian Xavier Yépez Castillo

TRIBUNAL DE GRADUACIÓN

Ing. Rebeca Estrada Pico

PROFESOR DE MATERIA DE GRADUACIÓN

Ing. Patricia Chávez

PROFESOR DELEGADO DE LA FACULTAD

RESUMEN

Implementar el protocolo DUNDI para comunicar 2 o más PBX basadas en Asterisk ubicados remotamente para facilitar la escalabilidad en una red de servidores y así permitir la comunicación entre dos terminales Voip unificando el plan de marcado de cada servidor PBX independiente mediante el desarrollo de una interfaz web desde el framework Elastix que facilite la configuración y comunicación entre los servidores PBX sin necesidad que el usuario manipule directamente los archivos de Asterisk para obtener eficiencia en el proceso. Para esto el servidor Norte tendrá una conexión directa con el servidor Centro. A la vez el servidor Centro a su vez tendrá conexión directa con el servidor Sur; aunque los servidores Norte y Sur no estén conectados directamente igual podrán llamarse entre ellos ya que forman parte de la red DUNDI.

INDICE GENERAL

| | PÁG. |
|--------------------------------|-------------|
| RESUMEN | VI |
| INDICE GENERAL | VII |
| INDICE DE FIGURAS | X |
| INDICE DE TABLAS | XI |
| INTRODUCCION | 12 |
| | |
| CAPITULO I | |
| ANTECEDENTES Y JUSTIFICACIONES | |
| 1.1 Antecedentes | 15 |
| 1.2 Justificación | 16 |
| 1.3 Descripción del proyecto | 17 |
| 1.4 Metodología | 19 |
| 1.5 Ventajas y Beneficios | 20 |
| | |
| CAPITULO II | |
| FUNDAMENTOS TEORICOS | |
| 2.1 Asterisk | 23 |
| 2.2 Software | 25 |
| 2.2.1 Elastix | 25 |
| 2.2.2 Softphone | 25 |

| | |
|--|----|
| 2.3 Hardware | 26 |
| 2.4 Dundi | 27 |
| CAPITULO III | |
| DESCRIPCIÓN DEL PROYECTO | |
| 3.1 Descripción Detallada | 33 |
| 3.2 Creación del Plan de Mercado | 34 |
| 3.3 Encriptación RSA | 35 |
| 3.3.1 Encriptación RSA en Dundi | 35 |
| 3.4 Configuración de archivos Asterisk | 36 |
| 3.4.1 Dundi.conf | 37 |
| 3.4.1.1 Contexto General | 37 |
| 3.4.1.2 Contexto Mappings | 39 |
| 3.4.1.3 Contexto Dirección MAC | 40 |
| 3.4.2 lax_custom.conf | 41 |
| 3.4.3 Extension.conf | 42 |
| 3.4.3.1 Dundi-priv-lookup | 43 |
| 3.4.3.2 Macro-dundi-priv | 43 |
| 3.4.4 Extensions_custom.conf | 43 |
| CAPITULO IV | |
| FUNCIONAMIENTO Y PRUEBAS | |
| 4.1 Procedimiento para conectar dos servidores de comunicación | 46 |
| 4.2 Intercambio de Claves | 49 |

| | |
|---|----|
| 4.3 Comandos Asterisk | 50 |
| 4.3.1 Dundi show peers | 51 |
| 4.4 Diagrama de conexiones | 53 |
| 4.5 Pruebas de Llamadas entre extensiones | 54 |
| | |
| Conclusiones y Recomendaciones | 56 |
| GLOSARIO | 59 |
| BIBLIOGRAFIA | 61 |

INDICE DE FIGURAS

| No. | PÁG. |
|---|-------------|
| 2.1 Softphone X-Lite | 26 |
| 2.2 Softphone Zoiper | 26 |
| 2.3 Interconexión de servidores remotos como una red E.164 | 30 |
| 3.1 Red DUNDi a implementar | 33 |
| 4.1 Creación de una extensión mediante el framework Elastix | 46 |
| 4.2 Interfaz Web que permite la configuración de cada servidor | 48 |
| 4.3 Interfaz Web para enviar una solicitud de conexión | 49 |
| 4.4 Creación automática de las claves públicas y privadas | 50 |
| 4.5 Intercambio de claves entre servidores | 50 |
| 4.6 Servidor NORTE enlazado con Servidor CENTRO | 52 |
| 4.7 Servidor CENTRO enlazado con Servidor CENTRO y SUR | 52 |
| 4.8 Servidor SUR enlazado con Servidor CENTRO | 53 |
| 4.9 Diagrama de Pruebas | 53 |
| 4.10 Líneas de asterisk indicando llamada del Norte hacia el Centro | 54 |
| 4.11 Líneas de asterisk indicando llamada del Norte hacia el Sur | 55 |
| 4.12 Líneas de asterisk indicando llamada del Centro hacia el Sur | 55 |

INDICE DE TABLAS

| No. | | PÁG. |
|------------|--|-------------|
| I | Características de requerimiento de hardware | 28 |
| II | Descripción de los servidores IP-PBX | 47 |

INTRODUCCION

La Voz sobre IP (VoIP, Voice over IP, siglas en ingles) es una tecnología que permite la transmisión de la voz a través de redes IP en forma de paquetes de datos.

La Telefonía IP es una aplicación inmediata de esta tecnología, de forma que permita la realización de llamadas telefónicas ordinarias sobre redes IP u otras redes de paquetes utilizando un PC, gateways y teléfonos estándares. En general, servicios de comunicación - voz, fax, aplicaciones de mensajes de voz - que son transportada vía redes IP, Internet normalmente, en lugar de ser transportados vía la red telefónica convencional.

Los pasos básicos que tienen lugar en una llamada a través de Internet son: conversión de la señal de voz analógica a formato digital y compresión de la señal a protocolo de Internet (IP) para su transmisión. En recepción se realiza el proceso inverso para poder recuperar de nuevo la señal de voz analógica.

Cuando hacemos una llamada telefónica por IP, nuestra voz se digitaliza, se comprime y se envía en paquetes de datos IP. Estos paquetes se envían a través de Internet a la persona con la que estamos hablando. Cuando

alcanzan su destino, son ensamblados de nuevo, descomprimidos y convertidos en la señal de voz original.

CAPÍTULO 1

ANTECEDENTES Y JUSTIFICACIÓN

1.1 Antecedentes

Desde sus inicios, las telecomunicaciones se han caracterizado por ser muy costosas debido a la inversión en infraestructura y a los dispositivos necesarios para permitir el acceso a la comunicación en distintas áreas, sin mencionar las altas tasas por interconexión que muchas telefónicas cobran de acuerdo a sus intereses de mercado. Por este motivo las empresas siempre han buscado métodos y nuevas tecnologías para reducir éstas tarifas sin comprometer la fiabilidad.

El fenómeno de la Internet ha sido pieza clave para que podamos alcanzar esta meta, ya que gracias al constante desarrollo de las redes IP combinado con técnicas avanzadas de digitalización de voz y los protocolos de control y transmisión en tiempo real, han hecho que la telefonía sobre IP se convierta en un tema estratégico, ya que permite la calidad de servicio a bajo costo.

Las versiones anteriores de VoIP estaban plagadas de problemas en el servicio, resultaban ruidosas y se perdían las llamadas. En los últimos años, Asterisk se ha convertido en una herramienta que ha promovido la masificación de esta tecnología principalmente dentro del campo empresarial, ya que además de ser de código abierto, hace que cualquier computadora estándar pueda funcionar como una central telefónica privada de acuerdo a las necesidades de sus usuarios, sin costos adicionales de implementación de hardware.

Uno de los mayores atractivos de Asterisk es que soporta múltiples protocolos de VoIP, entre ellos DUNDI que permite a las empresas conectar varias PBX adyacentes o remotas a través de una red punto a punto.

DUNDI es un protocolo de enrutamiento mediante el cual se puede crear una red de servidores de VoIP, que a futuro podría revolucionar el campo de la telefonía convencional.

El siguiente proyecto de graduación busca establecer la comunicación entre servidores Asterisk usando el protocolo DUNDI y presentar al usuario final una interfaz web que facilite la configuración y manipulación de la centralita.

1.2 Justificación

Debido a que en la actualidad las centralitas que utilizan Asterisk han ido ganando territorio en el sector de la telefonía privada a tal punto de que es muy común encontrarlas en muchos tipos de negocios, oficinas e instituciones surge la idea de establecer comunicación entre estas centralitas a través de una red de datos como la Internet. DUNDI nos permite realizar esto ya que es un protocolo de enrutamiento de código abierto diseñado para mantener tablas de enrutamiento dinámico entre sistemas compatibles.

1.3 Descripción del proyecto

Actualmente el proceso de configuración para aplicar el protocolo DUNDI se lo realiza manualmente y de una manera no optima ya que se manipulan los archivos directamente. Debido a esto, la implementación propuesta se basa en la utilización de herramientas disponibles y de libre acceso (Open Source).

Utilizaremos SQLITE como motor de base de datos, para nuestro caso nos brinda propiedades suficientes para automatizar la información de cada “servidor” cuando deseen comunicarse.

El siguiente paso es la creación de módulos dentro del Framework Elastix, este Framework nos permite utilizar sus bondades embebidas como lo es el FreePbx la cual es la más importante por la estabilidad. Cabe recalcar que la plataforma antes mencionada está desarrollada en Php como lenguaje de programación y utiliza el modelo Vista-Controlador en su estructura, por lo que para esta implementación necesitaremos desarrollar una interface web que nos permitan ingresar información de los “servidores VoIP”, y otra que nos permita el envío solicitud de conexión siendo esta la más importante logrando así que el usuario no manipule el archivo de configuración “dundi.conf”. sino que creamos nuestros propios archivos como resultado de

la extracción de la información de nuestra base de datos y los incluimos en el archivo principal.

Ahora cada extensión que creamos desde freePBX se añadirá al contexto [ext-local] el cual está incluido en dundi-priv-canonical que va a usar DUNDI para indicar que extensiones tenemos localmente.

Si nosotros tenemos la extensión responderemos con los datos para la comunicación hacia esa extensión; esos datos a su vez los interpreta el servidor que está preguntando y usa la función switch que pusimos en el archivo extensions_custom.conf para comunicarse con la extensión local.

El objetivo principal de nuestro proyecto de graduación es implementar el protocolo DUNDI para comunicar 2 o más PBX basadas en Asterisk y así compartir el plan de marcado de cada servidor independiente ubicado remotamente.

Como objetivos específicos hemos definido los siguientes:

- Facilitar la escalabilidad en una red de servidores PBX basados en Asterisk

- Permitir la comunicación entre dos terminales Voip unificando el plan de marcado de cada servidor PBX independiente.
- Habilitar una interfaz web desde el framework Elastix que facilite la configuración y comunicación entre los servidores PBX.

1.4 Metodología

Haremos uso de tres computadores personales que se comportaran como servidores para implementar el protocolo DUNDI, los cuales llevaran instalado una distribución del sistema operativo Linux configurado para soportar el framework ELASTIX. Para efectos de prueba es necesario crear una red LAN, la cual incluya a los tres servidores. Luego procederemos a crear las extensiones respectivas de cada servidor y a su vez instalar los softphones que utilizaran dichas extensiones. También es necesario manipular los archivos de configuración de Asterisk, lo cual explicamos más adelante, para finalmente poder realizar llamadas de un Servidor PBX basado en Asterisk hacia otro.

1.5 Ventajas y beneficios

Una telefonía de código abierto como lo es Asterisk es capaz de crear una cantidad ilimitada de oportunidades y beneficios. Entre algunas de ellas podemos decir que:

- Permite crear redes de telecomunicaciones a la medida. A esto se le atribuye en gran parte el éxito de Asterisk ya que permite construir un sistema de telefonía totalmente personalizado hasta el más mínimo detalle.
- Posibilita implementar un sistema a bajo costo. Asterisk permite que cualquier PC, incluyendo a aquellas de no tan alto rendimiento sirvan eficazmente como un servidor de comunicaciones. De esta manera una PC básica tiene la inteligencia suficiente para competir contra cualquier sistema propietario, el cual resulta ser en la mayoría de los casos mucho más costoso.
- Proporciona una integración adecuada de las tecnologías de comunicación. Asterisk acopla fácilmente distintos servicios tales como: integración de correo electrónico, voz, video y todo lo que una red de datos permite transmitir a la telefonía. Además según sean

nuestras necesidades podemos escoger el medio a través el cual efectuamos la comunicación, ya sea este cableado por cobre, fibra o inalámbrico.

Asterisk tiene el potencial de incrementar estas ventajas mediante el uso del protocolo DUNDI. Este protocolo nos hace posible la comunicación entre dos o más PBX, lo que permite generar una verdadera escalabilidad en la red. Además DUNDI es un protocolo de enrutamiento inteligente el cual es capaz de aprender rutas cada vez que efectúan las llamadas lo que en efecto causa que a medida que aprende rutas nuevas disminuirán los tiempos de respuesta. Otra de las ventajas es que DUNDI no tiene o no se basa en un sistema centralizado de control, lo que ayuda a que la red no se sature de información y proporciona un sistema más confiable ya que si hubiese una falla en un PBX el sistema trataría de buscar una ruta alterna para efectuar la llamada.

CAPÍTULO 2

FUNDAMENTOS TEORICOS

2.1 ASTERISK

Asterisk es una centralita software (PBX) de código abierto. Como cualquier centralita PBX permite interconectar teléfonos y conectar dichos teléfonos a la red telefónica convencional.

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

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

Para poder utilizar teléfonos convencionales en un servidor Linux corriendo Asterisk o para conectar a una línea de teléfono analógica se suele necesitar hardware especial (no vale con un modem ordinario). Digium y otras compañías venden tarjetas para este fin.

Pero quizás lo más interesante es que Asterisk soporta numerosos protocolos de VoIP como SIP y H.323. Asterisk puede operar con muchos teléfonos SIP, actuando como "registrar" o como "Gateway" o entre teléfonos IP y la red telefónica convencional. Los desarrolladores de Asterisk han diseñado un nuevo protocolo llamado IAX para una correcta optimización de las conexiones entre centralitas Asterisk.

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

2.2 Software

2.2.1 Elastix

Elastix fue creado y actualmente es mantenido por la compañía ecuatoriana Palosanto Solutions. Elastix fue liberado por primera vez en marzo de 2006 pero no se trataba de una distribución sino más bien de una interfaz para mostrar registros de detalles de llamadas para Asterisk, fue recién a finales de diciembre de 2006 cuando se lo lanzó como una distribución que contenía muchas herramientas interesantes administrables bajo una misma interfaz Web que llamó la atención por su usabilidad.

Elastix es una interfaz libre de Servidor de Comunicaciones Unificadas que integra en un solo paquete: VoIP PBX, fax, mensajería instantánea, correo electrónico, colaboración.

Desde entonces hasta la fecha esta distribución no ha parado de crecer en popularidad y actualmente es una de las preferidas del mercado.

2.2.2 Softphone

Para nuestro objetivo de comunicar terminales de servidores IP-PBX diferentes buscamos herramientas de libre acceso para lograrlo, optamos sin

ninguna particularidad por el softphone X-LITE para simular un teléfono IP físico tipo SIP y para los tipo IAX utilizamos el softphone ZOIPER.



Fig. 2.1 Softphone X-LITE



Fig. 2.2 Softphone ZOIPER

2.3 Hardware: Requerimientos del Sistema

La configuración óptima de hardware que necesite nuestro servidor de comunicaciones dependerá en gran parte de la cantidad de canales que

deseemos que estén operando simultáneamente. Asterisk nos proporciona una guía que permite que el sistema funcione correctamente sin perder la calidad en la comunicación.

Asterisk hace uso intensivo del procesador, debido a que lo usa para hacer el procesamiento de los canales de voz. Para construir un Asterisk PBX es necesario un procesador compatible con Intel mínimo un Pentium 300Mhz y memoria RAM 256 MB. Asterisk no requiere mucho espacio en disco, cerca de 100MB compilados, luego dependiendo de las demás aplicaciones como más código fuente, buzón de voz y grabaciones, deberemos seleccionar la cantidad de disco duro que necesitemos.

2.4 DUNDI

DUNDI por sus siglas en ingles significa Distributed Universal Number Discovery, es un protocolo que nos permite buscar y compartir planes de marcación entre servidores IP-PBX, como lo es Asterisk, por medio de un sistema par a par que se utiliza para localizar gateways de Internet para servicios de telefonía. Debido a que es una red par a par no existen roles fijos de cliente-servidor. Por este motivo a diferencia de los sistemas centralizados, como el estándar ENUM, DUNDI se distribuye de manera completa a lo largo de toda la red sin ninguna autoridad centralizada.

Tabla I Características de requerimiento de hardware

| Tipo de Uso | Cantidad de Canales | Requerimientos Mínimos |
|--------------------------|----------------------------|---|
| Sistema para Pruebas | Hasta 5 | 400-MHz x86, 256 MB RAM |
| Oficina/Hogar Pequeña | De 5 a 10 | 1-GHz x86, 512 MB RAM |
| Sistema Empresa Pequeña | Hasta 15 | 3-GHz x86, 1 GB RAM |
| Sistema Mediano a Grande | Más de 15 | Dual CPUs, posiblemente requiera servidores |

El protocolo fue inventado por Mark Spencer, quién también hizo el IP-PBX Asterisk por lo que ambos sistemas son compatibles entre sí y los comandos de búsqueda se pueden hacer directamente desde la línea de comandos de Asterisk.

Funcionamiento

Antes de implementar DUNDI debemos saber cómo funciona para esto imaginemos a DUNDI como un gran directorio telefónico que le permite a preguntar a los demás equipos si conoce alguna ruta alterna para realizar la llamada ya sea esta hacia una extensión o hacia algún teléfono de red telefónica publica conmutada (PSTN por sus siglas en ingles). Por ejemplo, supongamos que estamos dentro de un red DUNDI y deseamos comunicarnos con la extensión 555 la cual no pertenece de manera local,

entonces nuestro servidor "A" preguntara a los otros equipos que dentro de la red si conocen a la extensión 555, suponiendo que el servidor "B" es uno de los equipos conectados con nuestro servidor, este también preguntara a el resto de equipos si conoce la extensión. Así directa o indirectamente esta consulta se hará a todos los equipos pertenecientes a la red DUNDI. La centralita que contenga esa extensión será la encargada de emitir una respuesta y será algo parecido a "IAX2/usuario:clave@1.2.3.4/555".

Finalmente el equipo que hizo la consulta utilizará esa información para llamar efectivamente a la extensión 555. Además la respuesta es guardada por todos los equipos para de esta manera en la siguiente oportunidad que se efectúe otra llamada a la misma extensión, los equipos seleccionen la mejor ruta. Este aprendizaje de rutas ayuda a la red a incrementar su tiempo de respuesta y a reducir la carga de información. Hay que tener en cuenta que la información guardada tiene un tiempo límite de validez, sin embargo la solución a esto es declarar el argumento switch => dentro del archivo extensions.conf.

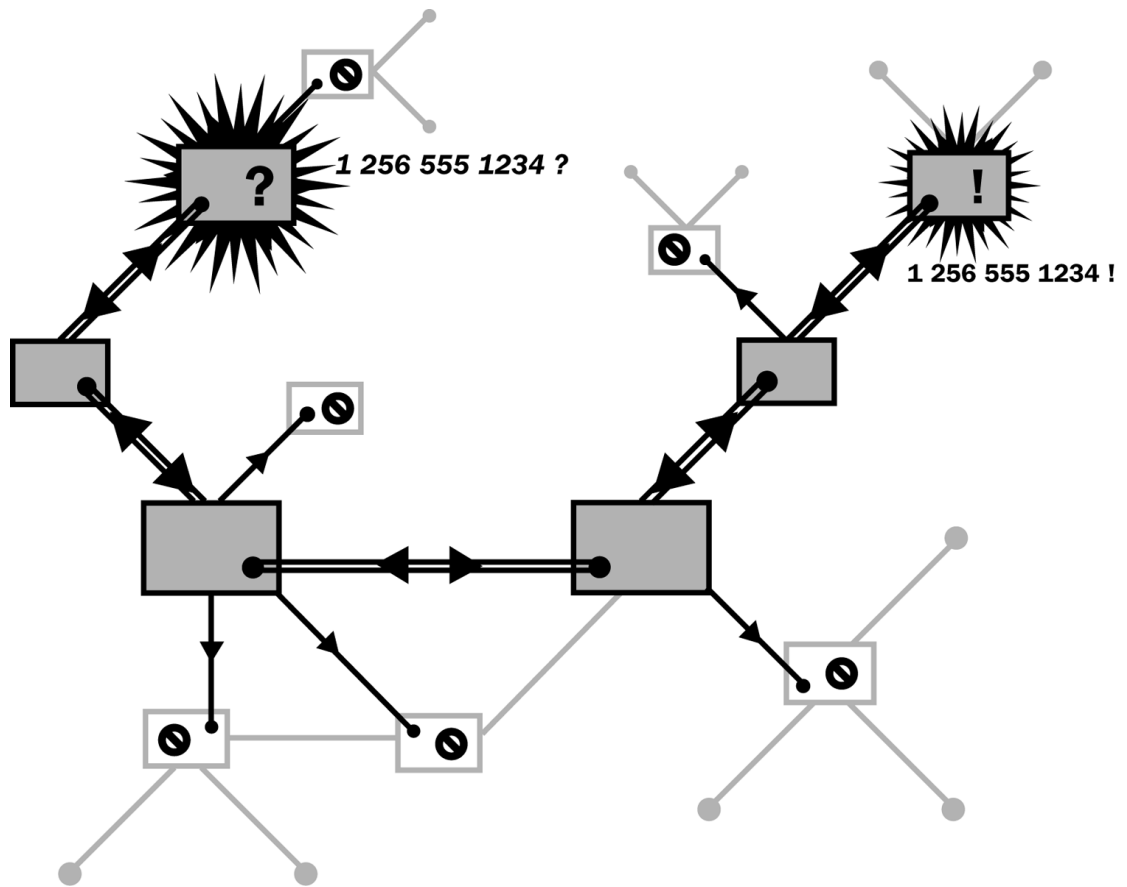


Fig. 2.3 Interconexión de servidores remotos como una red E.164

DUNDi se puede utilizar dentro de una empresa para crear una completa PBX asociada con ningún punto de falla central, también ofrece la posibilidad de agregar arbitrariamente nuevas extensiones, gateways u otros recursos en una red de confiable de los servidores de comunicación, donde cualquier creación, movimiento, cambio, fallo o nuevas rutas son absorbidas automáticamente dentro de la nube sin tener que realizar ninguna configuración adicional.

DUNDi también puede utilizar todo el Internet para formar una red E.164 confiable que permita a los proveedores de servicios habilitar números de teléfono reales a través de Internet. Esto permite tener el control de los números sin necesidad de un cargo adicional y transparente para el usuario. Un uso responsable y una correcta creación de políticas nos ayudarían a prevenir llamadas VoIP tipo “spam”.

DUNDi por sí mismo no es un protocolo de señalización de VoIP ni tampoco un medio de comunicación. En lugar de ello, Dundi publica rutas por las cuales se accede a través de protocolos estándar como IAX, SIP y H.323.

El protocolo DUNDi utiliza por defecto el puerto 4520/UDP. Habrá que abrirlos en los firewalls. Es independiente de IAX, es decir, la cadena de conexión devuelta en una consulta puede hacer referencia a cualquier tipo de canal (IAX2, SIP, H323, etc.), aunque normalmente se usa el IAX2.

CAPÍTULO 3

DESCRIPCION DEL PROYECTO

3.1 Descripción detallada

Nuestro proyecto consiste en lograr la comunicación entre 2 o más IP-PBX, en las cuales se encuentra instalado Asterisk, de tal manera que nos permita realizar llamadas desde las extensiones de un servidor local hacia las extensiones de uno remoto.

En nuestro caso tenemos 3 servidores Asterisk, NORTE, CENTRO y SUR, en todos debemos instalar Elastix, la cual es una herramienta que nos proporciona una interfaz que más adelante nos permitirá manipular nuestros servidores a través de una interfaz web. Esto facilitara la labor del administrador del servidor ya que todo cambio que realice usando la interfaz web se graban directamente en los archivos de configuración de DUNDI.

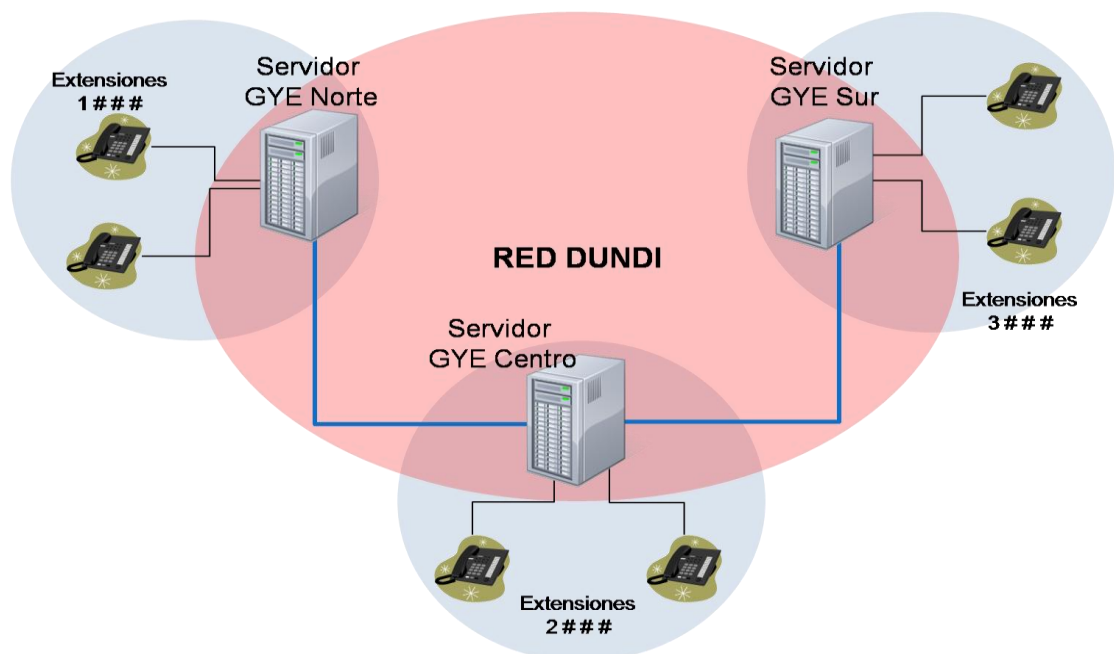


Fig. 3.1 Red DUNDi a implementar

Para implementar dicho protocolo en cada servidor es necesario:

- Crear un plan de marcación distinto para cada uno.
- Configurar los archivos: `iax.conf`, `extensions.conf` y `dundi.conf`.
- Antes de comenzar es necesario conocer las direcciones IP y direcciones mac de cada servidor.

Nuestro servidor Norte tendrá una conexión directa con el servidor Centro.

El servidor Centro a su vez tendrá conexión directa con el servidor Sur.

Aunque los servidores Norte y Sur no estén conectados directamente igual podrán llamarse entre ellos ya que forman parte de la red DUNDI.

Para que los cambios se hagan efectivos podemos ejecutar el comando:
`asterisk -rx "restart when convenient"` o si no queremos esperar:
`asterisk -rx "restart now"`.

3.2 Creación de un plan de marcado

El plan de marcación nos sirve para llevar un control ordenado de las extensiones que crearemos en nuestras centralitas. Esto nos ayuda a generar un número único para cada extensión. De esta manera por ejemplo nuestro número podría estar formado de la siguiente forma: un código para

identificar la ciudad (CC), un código de institución (CI) y el número de nuestra extensión en la centralita.

De esta manera nos aseguramos que no habrá ninguna extensión con algún número repetido.

Por otro lado, existe también la posibilidad de conectar centralitas que ya están funcionando de manera independiente y que pueden tener extensiones iguales, para estos casos se puede asignar un peso al enlace el cual evitaría que nuestra llamada falle por motivo de dos números iguales.

3.3 Encriptación RSA

El sistema criptográfico con clave pública RSA es un algoritmo asimétrico cifrador de bloques, que utiliza una clave pública, la cual se distribuye (en forma autenticada preferentemente), y otra privada, la cual es guardada en secreto por su propietario.

Encriptación RSA en DUNDi

DUNDi usa certificados de encriptación RSA para compartir sus planes de marcado, además por que las respuestas a una consulta incluyen el passwd de la troncal.

Usaremos la misma clave para la comunicación entre todos los servidores. Así que el primer paso es generar los certificados de encriptación uno de los equipos (en este ejemplo en Servidor NORTE):

```
cd /var/lib/asterisk/keys  
astgenkey -n SERVERS-DUNDI
```

Ahora necesitamos compartir los certificados entre los dos servidores; esto es para que cada uno pueda desencriptar al otro.

Elastix también permite hacer esto a través de su interfaz web, lo que se hace es que un servidor solicita a otro establecer un enlace, luego el solicitante debe esperar hasta que el otro servidor acepte el requerimiento de conexión.

3.4 Configuración de archivos de Asterisk

Para que DUNDI funcione correctamente es necesario modificar algunos archivos de configuración, entre ellos:

- Dundi.conf
- Extensions.conf
- lax.conf

- Además también se deben modificar o crear todos los archivos que se incluyen en los contextos.

A continuación procederemos a revisar más detalladamente la configuración de todos los archivos.

3.4.1 Configuración dundi.conf

El archivo dundi.conf originalmente viene con todos los parámetros inhabilitados para que se pueda configurar manualmente como se lo ha descrito antes, es recomendado por expertos en asterisk que siempre se trabaje con una copia o se manipule indirectamente los archivos, por lo que optamos por la segunda alternativa para lo cual creamos tres archivos para poder configurarlo mediante la interfaz grafica desarrollada.

3.4.1.1 Contexto General

En esta sección del archivo dundi.conf definimos nuestra propia identificación en la nube DUNDi, así como las opciones globales, debemos rellenar los campos con los datos reales de nuestros equipos (Compañía, Dirección, MAC, etc.). Para esto incluimos debajo de este contexto la llamada a nuestro primer archivo de configuración:

```
[general]  
#include dundi_general_custom_elastix.conf
```

Archivo `dundi_general_custom_elastix.conf`

Al incluir el archivo debajo del contexto *general* nuestro proceso automáticamente escribe la información de cada servidor ingresada por la interfaz web como lo puede observar en las siguientes líneas.

```
department=Nombre del Servidor  
organization=AlgunaOrganizacion  
locality=Ciudad  
stateprov=Provincia  
country=Ecuador  
email=servidor@dundi.com  
phone=+593- 5555555
```

Los siguientes parámetros pueden dejarse configurados tal como están originalmente.

Se define en que red va a escuchar DUNDI.

```
bindaddr=0.0.0.0
```

El puerto en el que va a escuchar, por defecto 4520.

```
port=4520
```

Nos identificamos con nuestra MAC address.

```
entityid="mac_add"
```

El número de consultas que va a hacer DUNDI hasta alcanzar un plan de marcado.

```
ttl=12
```

Finaliza las conexiones fallidas.

```
autokill=yes
```

3.4.1.2 Contexto mappings

Básicamente aquí definimos nuestra respuesta a una consulta hacia determinado número que tengamos localmente. Tal como se hizo con el contexto *general* aquí también se incluye la llamada de nuestro segundo archivo.

```
[mappings]
#include dundi_mappings_custom_elastix.conf
```

Archivo dundi_mappings_custom_elastix.conf

Al incluir el archivo debajo del contexto *mappings* nuestro proceso automáticamente escribe la información como lo puede observar en las siguientes líneas.

Para Servidor local:

```
priv => dundi-priv-
canonical,0,IAX2,dundi:${SECRET}@"direccion_ip"/${NUMBER},nopa
rtial
priv => dundi-priv-
customers,100,IAX2,dundi:${SECRET}@"direccion_ip"/${NUMBER},n
opartial
priv => dundi-priv-via-
pstn,400,IAX2,dundi:${SECRET}@"direccion_ip"/${NUMBER},nopartia
l
```

Los valores de `#{NUMBER}` y `#{SECRET}` se reemplazaran automáticamente, según el numero que se consulte y la clave aleatoria que se haya generado en ese momento.

3.4.1.3 Contexto *peers* (Dirección MAC)

Por último, debemos definir los PEERS esto lo hacemos incluyendo la llamada a nuestro tercer archivo:

```
#include dundi_mappings_custom_elastix.conf
```

Archivo `dundi_peers_custom_elastix.conf`

En este archivo se escribirá la información automáticamente de cada *peer* que se consultaran y serán consultados (con la opción `symmetric`).

Para Servidor local:

Identificamos al servidor remoto por su `entityid`

```
[mac_address_remota]
```

Permitir y realizar conexiones

```
model=symmetric
```

IP ser servidor remoto

```
host="ip del servidor remoto"
```

Clave local y clave remota

```
inkey=SERVERS-DUNDI  
outkey=SERVERS-DUNDI
```

Incluimos el contexto antes definido en la sección `mappings`


```
include=priv
permit=priv
qualify=yes
order=primary
```

3.4.2 Configuración `iax_custom.conf`

Aquí debemos definir la configuración de la troncal; como dundi va a hacer todo el trabajo de enviar la cadena exacta de como contactarlos solo creamos un usuario que recibe las llamadas autenticadas de la nube DUNDi:

En Servidor local, en Servidor remoto 1 y Servidor remoto 2.

```
[dundi]

type=user
dbsecret=dundi/secret
context=ext-local
```

Podemos restringir ciertos codecs para mejorar la calidad de voz, o dar prioridad al ancho de banda (con gsm o g729)

```
disallow=all
allow=ulaw
```

3.4.3 Configuración `extensions.conf`

Para lograr la integración con freePBX debemos modificar un poco el contexto "from-internal" que es el contexto que usa freePBX para las extensiones.

Teniendo en cuenta que cuando un número no es encontrado en el plan de marcado local se usa el contexto [bad-number] es allí donde debemos modificar y la configuración debe ser exactamente la misma en los tres servidores.

En el archivo extensions.conf buscamos la sección:

```
[from-internal]
include => from-internal-xfer
include => bad-number
```

y comentamos la línea `include => bad-number` poniendo un punto y coma adelante así:

```
;include => bad-number
```

y agregamos la línea

```
include => dundi-priv-lookup
```

que es un contexto que crearemos luego en el archivo `extensions_custom.conf`, quedando así:

```
[from-internal]
include => from-internal-xfer
;include => bad-number
include => dundi-priv-lookup
```

3.4.3.1 Sección dundi-priv-lookup

Este contexto se encarga de hacer la búsqueda de un numero por dundi

Antes de hacer la búsqueda definimos apropiadamente nuestro caller id ya que si no tendremos un caller id como "device<0000>".

```
exten => _X.,1,Macro(user-callerid)
exten => _X.,n,Macro(dundi-priv,${EXTEN})
exten => _X.,n,GotoIf($["${DIALSTATUS}" = "BUSY"]?100)
exten => _X.,n,Goto(bad-number,${EXTEN},1)
exten => _X.,100,Playtones(congestion)
exten => _X.,101,Congestion(10)
```

3.4.3.2 Macro-dundi-priv

Esta es la macro que llamamos desde el contexto *[dundi-priv-lookup]*.

También evita que hayan loops en las consultas dundi.

```
exten => s,1,Goto(${ARG1},1)
switch => DUNDi/priv
```

3.4.4 Configuración extensions_custom.conf

Aquí definimos los contextos personalizados para que dundi mapee nuestra extensiones, así como una macro para hacer las búsquedas en otros equipos (lo que evita loops) y la sentencia para redirigir las llamadas (switch).

Debemos añadir las siguientes líneas al final del archivo extensions_custom.conf en los dos servidores:

```
[dundi-priv-canonical]
```

Aquí incluimos el contexto que contiene las extensiones.

```
include => ext-local
```

Aquí incluimos el contexto que contiene las colas de atención o queues.

```
include => ext-queues
```

Si tenemos clientes (o revendemos servicios) podemos listarlos aquí

```
[dundi-priv-customers]
```

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

Aquí podemos incluir el contexto con nuestras troncales hacia la PSTN,

```
[dundi-priv-via-pstn]
```

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

Si queremos que los demás equipos puedan usar nuestras troncales

```
include => outbound-allroutes
```

En el contexto `[dundi-priv-local]` unificamos los tres contextos, este lo podemos usar como contexto de la troncal iax de dundi.

```
[dundi-priv-local]
```

```
include => dundi-priv-canonical  
include => dundi-priv-customers  
include => dundi-priv-via-pstn
```

CAPÍTULO 4

FUNCIONAMIENTO Y PRUEBAS

4.1 Procedimiento para conectar dos Servidores de comunicación

Como habíamos mencionado anteriormente Asterisk nos permite realizar cualquier tipo de configuración a nuestro Servidor PBX a través de una interfaz web.

- Evitamos que el servidor sea manipulado de manera física.
- Facilita la tarea de agregar extensiones (Fig. 4.1.1)
- Presenta un entorno más agradable al usuario.
- Cualquier tipo de comando Asterisk puede ser ejecutado desde la interfaz web.



Fig. 4.1 Creación de una extensión mediante el framework Elastix

El primer paso es conocer la información correspondiente a cada servidor

Tabla II. Descripción de los servidores IP-PBX

| Nombre del Servidor | Dirección Ip | Mac Address |
|----------------------------|---------------------|--------------------|
| NORTE | 192.168.1.103 | 08:00:27:D6:37:19 |
| CENTRO | 192.168.1.104 | 08:00:27:71:A7:B3 |
| SUR | 192.168.1.105 | 00:0C:29:72:10:0F |

Procedemos a modificar los archivos `dundi.conf`, `extesions.conf` e `iax.conf`, de la manera en que se detalló previamente en el capítulo 3, así también lo hacemos con archivos adicionales que declaramos dentro de cada contexto.

Luego ingresamos a nuestro servidor mediante la interfaz web en el framework Elastix usando un visualizador.

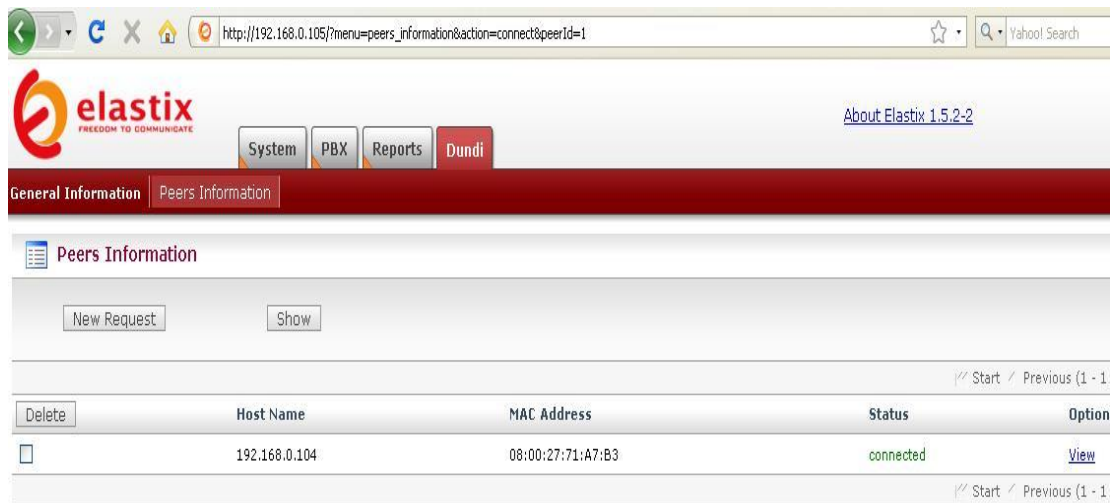
Dentro de este encontramos una pestaña titulada DUNDI, en la cual podemos ingresar los datos de nuestro servidor (fig. 4). Al momento de hacer esto y ejecutar el “upload” automáticamente se generan las claves privadas y públicas de nuestro servidor, estas nos servirán para solicitar y establecer una conexión con otro servidor.



Fig. 4.2 Interfaz Web que nos permite la configuración de cada servidor

Para poder establecer comunicación entre dos servidores es necesario generar una petición de conexión. En la pestaña “Peers Information” podemos visualizar el estado actual de nuestra red, es decir los peers que están enlazados con nuestro servidor y el estado de este enlace.

Además en esta misma sección se puede solicitar una nueva conexión. Una vez realizada esa petición se deberá esperar una confirmación por parte del PBX remoto. En el momento que el servidor remoto acepte esta solicitud se ejecuta el procedimiento para intercambiar las claves, tanto las públicas como las privadas, sin esta acción no es posible establecer la comunicación ya que DUNDI lo exige como una forma de validación para evitar intrusiones de equipos desconocidos.



Elastix is licensed under [GPL](#) by [PaloSanto Solutions](#), 2006 - 2008.

Fig. 4.3 Interfaz Web para enviar una solicitud de conexión

4.2 Intercambio de claves

Luego de haber generado las claves pública y privada de nuestro PBX podemos verificar si se crearon haciendo una revisión del siguiente directorio.

```
/var/lib/asterisk/keys
```

Si todavía no se ha hecho el intercambio de claves con el servidor remoto solo aparecerán nuestras claves locales, las reconocemos porque su nombre comienza con CER seguido de la mac address de nuestro servidor.

```

[root@elastix keys]# cd /var/lib/asterisk/keys/
[root@elastix keys]# ll
total 24
-rw-r--r-- 1 asterisk asterisk 887 Oct 28 15:30 CER080027D63719.key
-rw-r--r-- 1 asterisk asterisk 272 Oct 28 15:30 CER080027D63719.pub
-rw-rw-r-- 1 asterisk asterisk 272 Aug 12 18:33 freeworlddialup.pub
-rw-rw-r-- 1 asterisk asterisk 272 Aug 12 18:33 iaxtel.pub
[root@elastix keys]# _

```

Fig. 4.4 Creación automática de las claves publicas y privadas

Después de hacer el intercambio de claves verificamos que las claves pública y privada del PBX remoto se hayan añadido a nuestro directorio.

```

[root@elastix keys]# cd /var/lib/asterisk/keys/
[root@elastix keys]# ll
total 32
-rw-r--r-- 1 root      root      891 Oct 28 15:45 CER08002771A7B3.key
-rw-r--r-- 1 asterisk asterisk 272 Oct 28 15:38 CER08002771A7B3.pub
-rw-r--r-- 1 asterisk asterisk 887 Oct 28 15:30 CER080027D63719.key
-rw-r--r-- 1 asterisk asterisk 272 Oct 28 15:30 CER080027D63719.pub
-rw-rw-r-- 1 asterisk asterisk 272 Aug 12 18:33 freeworlddialup.pub
-rw-rw-r-- 1 asterisk asterisk 272 Aug 12 18:33 iaxtel.pub
[root@elastix keys]# _

```

Fig. 4.5 Intercambio de claves entres servidores

4.3 COMANDOS ASTERISK

El uso de comandos en Asterisk nos facilita la visualización de todas las modificaciones que realicemos. También nos facilita el monitoreo de las extensiones registradas en nuestro PBX. Además podemos verificar si el estado de nuestra comunicaciones con los demás PBX.

Tenemos dos formas de introducir nuestros comandos, una es directamente desde el CLI de Asterisk, lo cual significa que tenemos que estar manipulando directamente nuestro servidor.

La otra manera es a través del framework Elastix, el cual nos permite introducir comandos desde un cuadro de texto y luego de esto nos presenta la información requerida. Esta segunda forma es más recomendada ya que no es necesario manipular de manera física al servidor.

4.3.1 DUNDI SHOW PEERS

La primera prueba que realizaremos es la de verificar si nuestra centralita está reconociendo a sus pares y verificamos el estado de los enlaces. Para esta prueba usamos el comando:

dundi show peers

De esta manera utilizando la interfaz web que nos proporciona Elastix, comprobamos lo siguiente:

- El servidor CENTRO debe estar enlazado con el servidor NORTE y con el servidor SUR.
- El servidor NORTE debe estar enlazado únicamente con el servidor CENTRO.

- El servidor SUR debe estar enlazado únicamente con el servidor CENTRO.

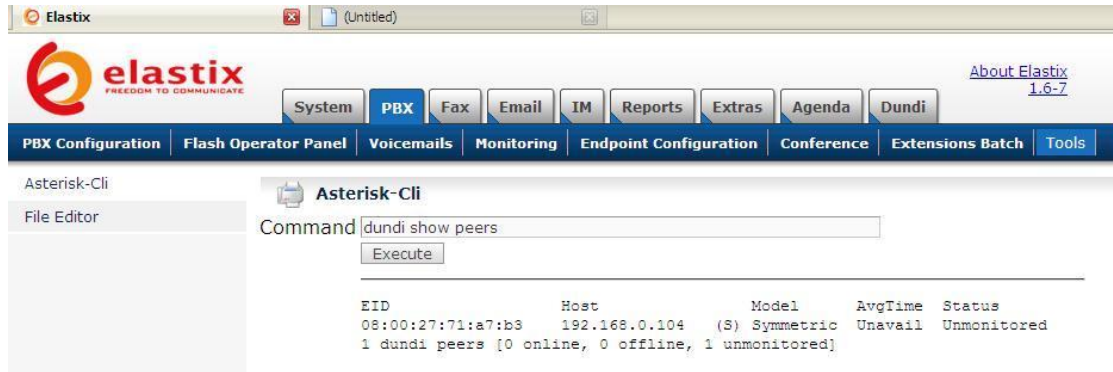


Fig. 4.6 Servidor NORTE enlazado con Servidor CENTRO

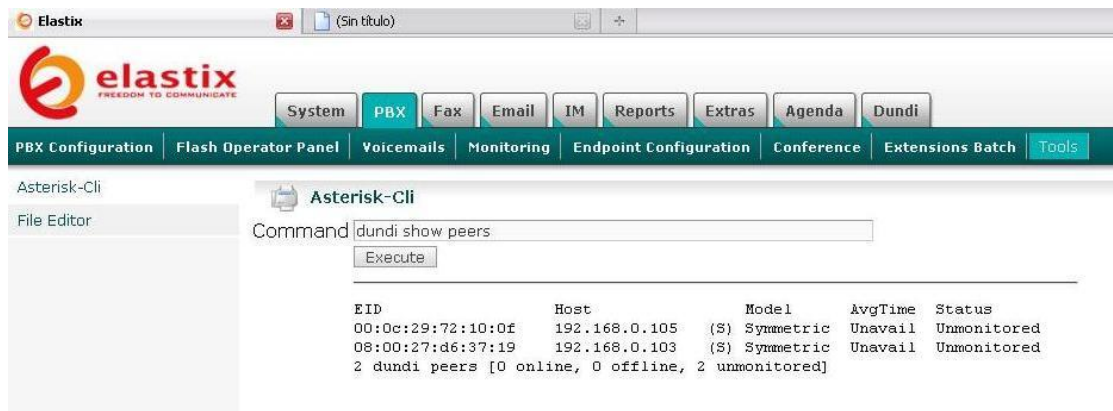


Fig. 4.7 Servidor CENTRO enlazado con Servidor CENTRO y SUR

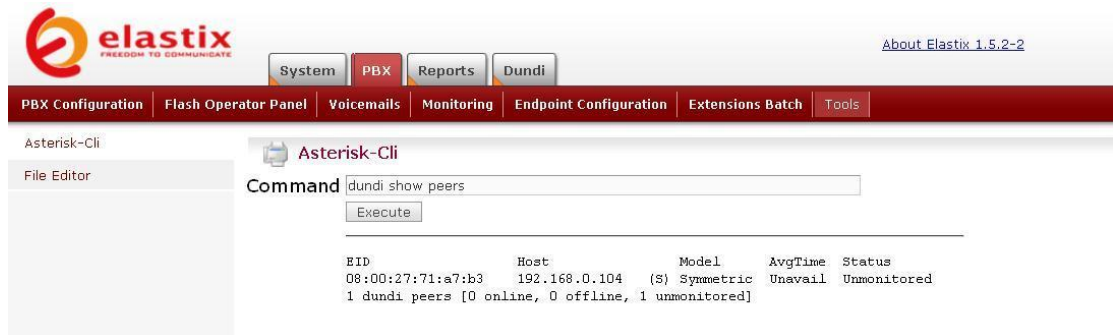


Fig. 4.8 Servidor SUR enlazado con Servidor CENTRO

4.4 Diagrama de conexiones para efecto de prueba.

En esta grafica se muestra el Diagrama que se implementó para efecto de pruebas

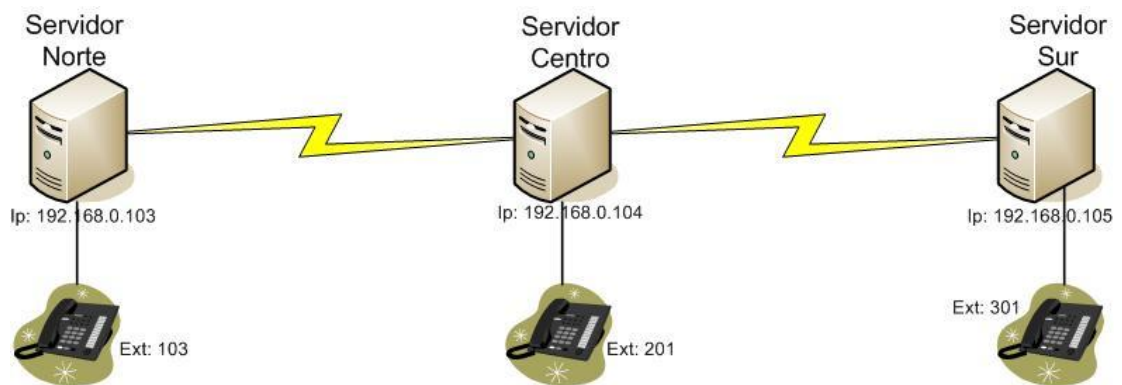


Fig. 4.9 Diagrama de Pruebas

4.5 Pruebas de llamadas entre las extensiones

En esta sección verificaremos que podamos efectuar llamadas entre las extensiones conectadas al servidor NORTE con las del servidor CENTRO y SUR. Además visualizaremos lo que ocurre en la línea de comandos de Asterisk de cada PBX al momento de realizar estas llamadas.

En las figuras. 4.10, 4.11 y 4.12 se muestra detalladamente el proceso que realiza asterisk en el preciso instante de realizar llamadas entre los servidores.

Llamadas entre ext: 1 # # y ext: 2 # # realizadas con éxito:

```
-- Goto (macro-dundi-priv,201,1)
-- Call accepted by 192.168.0.104 (format ulaw)
-- Format for call is ulaw
-- Called dundi:6PJ2XUYFjmecDxtoERYQJw==@192.168.0.104/201
-- IAX2/192.168.0.104:4569-6308 is ringing
-- IAX2/192.168.0.104:4569-6308 stopped sounds
-- IAX2/192.168.0.104:4569-6308 answered SIP/103-084f5f18
e lastix*CLI> _

-- Goto (macro-dundi-priv,103,1)
-- Called dundi:jIZyEouknkt6h3QbSKa7EA==@192.168.0.103/103
-- Call accepted by 192.168.0.103 (format ulaw)
-- Format for call is ulaw
-- IAX2/192.168.0.103:4569-880 is ringing
-- IAX2/192.168.0.103:4569-880 stopped sounds
-- IAX2/192.168.0.103:4569-880 answered SIP/201-095c9528
e lastix*CLI> _
```

Fig. 4.10 Líneas de asterisk indicando llamada del NORTE hacia el Centro

Llamadas entre ext: 2 # # y ext: 3 # # realizadas con éxito:

```
-- Executing [3010from-internal:2] Macro("SIP/201-095c7a00", "dundi-priv:301") in new stack
-- Executing [s@macro-dundi-priv:1] Goto("SIP/201-095c7a00", "3011") in new stack
-- Goto (macro-dundi-priv,301,1)
-- Called dundi:kLUQbUrQty+4lz9KndChCQ==@192.168.0.105/301
-- Call accepted by 192.168.0.105 (format ulaw)
-- Format for call is ulaw
-- IAX2/192.168.0.105:4569-13858 is ringing
-- IAX2/192.168.0.105:4569-13858 stopped sounds
-- IAX2/192.168.0.105:4569-13858 answered SIP/201-095c7a00
elastix*CLI> _
```

```
-- Called 301
-- SIP/301-09337bd0 is ringing
-- SIP/301-09337bd0 answered IAX2/dundi-2211
elastix*CLI> _
```

Fig. 4.11 Líneas de asterisk indicando llamada del NORTE hacia el SUR

Llamadas entre ext: 1 # # y ext: 3 # # realizadas con éxito:

```
-- Executing [s@macro-dundi-priv:1] Goto("SIP/103-004f5f18", "3011") in new stack
-- Goto (macro-dundi-priv,301,1)
-- Called dundi:kLUQbUrQty+4lz9KndChCQ==@192.168.0.105/301
-- Call accepted by 192.168.0.105 (format ulaw)
-- Format for call is ulaw
-- IAX2/192.168.0.105:4569-8638 is ringing
-- IAX2/192.168.0.105:4569-8638 stopped sounds
-- IAX2/192.168.0.105:4569-8638 answered SIP/103-004f5f18
elastix*CLI> _
```

```
-- Executing [s@macro-dial:7] Dial("IAX2/dundi-6587", "SIP/3011|tr") in new stack
-- Called 301
-- SIP/301-09337bd0 is ringing
-- SIP/301-09337bd0 answered IAX2/dundi-6587
elastix*CLI> _
```

Fig. 4.12 Líneas de asterisk indicando llamada del CENTRO hacia el SUR

CONCLUSIONES Y RECOMENDACIONES

Quedó demostrado el gran potencial de uso que tienen Asterisk y Dundi, tomando en cuenta la tasa de crecimiento y la facilidad que se brinda para esto, se puede proyectar hacia la formación de una gran red de PBXs interconectadas entre ellas para comunicar a usuarios de todas partes del mundo, parecido a lo que es Internet hoy en día.

Hablamos de una tasa de crecimiento alto por varios motivos entre los cuales destacan que:

- **Asterisk es un software libre**, lo que hace que se expanda de forma masiva y rápida en todo el mundo.
- **Protocolos de comunicación definidos.** Dundi es el protocolo encargado de establecer la comunicación entre PBXs.
- **El sistema se adapta a nuestras necesidades.** Asterisk es sumamente flexible tratándose de los requerimientos del usuario es así que las aplicaciones cubren un rango amplio que van desde sistemas caseros de comunicación con 2 o 3 canales de comunicación hasta sistemas para grandes empresas los cuales superan los 15 canales de comunicación simultanea.

Debido a que DUNDI es un protocolo de enrutamiento nos brinda cierto nivel de inteligencia para seleccionar entre todas las rutas conocidas la mejor, o en su defecto la que menor peso tenga.

Al ser Dundi un sistema no centralizado, es decir que la información que cada uno de los equipos aprende la comparte con el resto, mejora su respuesta ante fallas de un equipo y es poco probable que el sistema colapse por dichas fallas.

1. Se recomienda seguir implementando soluciones ya sean de escritorio y/o web para el manejo de Asterisk ya que esto reduce la cantidad de errores ingresados al sistema de forma involuntaria cuando se manipula directamente los archivos nativos; así con parámetros definidos y configurables en nuestro caso desde una interfaz web podemos indicarle al usuario final que o como configurar el sistema proporcionándole alternativas u opciones que eviten el erróneo ingreso de información. Para ejemplo tenemos el framework Elastix, el cual es una interfaz que nos facilita el uso de Asterisk y además que por ser de código abierto nos permite modificarla para que se adapte a nuestros requerimientos.

2. Si se va a establecer una comunicación de PBXs Asterisk que no hayan sido configurados previamente, es decir nuevos, lo ideal es partir con

un plan de marcado predefinido para evitar repetición de números de extensiones entre nuestros servidores.

3. Si nos encontramos con PBXs que ya están funcionando. ¿Cómo implementar DUNDI en PBX que ya están funcionando? ¿Cómo resolver el problema de repetición de extensiones?

4. Si ya esta creadas las PBX y queremos conectarlas entre ellas debemos tomar en cuenta si tendremos extensiones repetidas. Para evitar que nuestra llamada falle debido una duplicación de extensiones en varias PBX, podemos configurar un peso a nuestra conexión de tal forma que al momento de realizar una llamada la ruta hacia la extensión deseada será la que tenga menor peso.

GLOSARIO

Dialplan: Un plan de marcado establece el número esperado y el patrón de dígitos de un número de teléfono. Esto incluye los códigos de país, códigos de acceso, códigos de área y todas las combinaciones de dígitos marcados.

Gateway: Es un dispositivo, con frecuencia un ordenador, que permite interconectar redes con protocolos y arquitecturas diferentes a todos los niveles de comunicación. Su propósito es traducir la información del protocolo utilizado en una red al protocolo usado en la red de destino.

Open Source: Código abierto (en inglés open source) es el término con el que se conoce al software distribuido y desarrollado libremente. El código abierto tiene un punto de vista más orientado a los beneficios prácticos de compartir el código que a las cuestiones morales y/o filosóficas las cuales destacan en el llamado software libre.

PBX: Un PBX es cualquier central telefónica conectada directamente a la red pública de teléfono por medio de líneas troncales para gestionar, además de las llamadas internas, las entrantes y/o salientes con autonomía sobre cualquier otra central telefónica.

MAC: Media Access Control (Control de acceso al medio). Identificador hexadecimal de 48 bits que corresponde de manera única a cualquier interfaz o dispositivo de red (routers, switch, tarjetas de red)

MACRO: es una serie de instrucciones que se almacenan para que se puedan ejecutar de forma secuencial mediante una sola llamada u orden de ejecución.

SOFTPHONE: Un Softphone (en inglés combinación de Software y de Telephone) es un software que hace una simulación de teléfono convencional por computadora.

TRONCAL: En lenguaje técnico de telefonía, una línea troncal es un enlace que interconecta las llamadas externas de una central telefónica, concentrando y unificando varias comunicaciones simultáneas en una sola señal para un transporte y transmisión a distancia más eficiente (generalmente digital) y poder establecer comunicaciones con otra central o una red entera de ellas.

VOIP: Voz sobre Protocolo de Internet, también llamado Voz sobre IP, VozIP, VoIP (por sus siglas en inglés), es un grupo de recursos que hacen posible que la señal de voz viaje a través de Internet empleando un protocolo IP (Internet Protocol).

BIBLIOGRAFIA

- [1] Jim Van Meggelen, Jared Smith, and Leif Madsen (2005). Asterisk The Future of Telephony.
- [2] Talking Around the world (2009). Instalación de Asterisk 1.6.x en Ubuntu 8.1.<http://www.voztovoice.org/?q=node/165>.
- [3] El Ajonjolí. Configurando DUNDI en Elastix (Asterisk + free PBX) <http://elajonjoli.org/node/11>.
- [4] VoIP-info.org. DUNDI Enterprise configuration IAX. <http://www.voip-info.org/wiki/view/DUNDI+Enterprise+Configuration+IAX>.
- [5] DUNDI - Distributed Universal Number Discovery. <http://www.dundi.com/>
- [6] Mark Spencer (2004). Distributed Universal Number Discovery (DUNDi™) and the General Peering Agreement (GPA™). <http://www.dundi.com/dundi.pdf>.
- [7] Julián J. Menéndez. Usando la red DUNDI en Asterisk. <http://www.julianmenendez.es/usando-dundi-asterisk/>