

Sistema avanzado de telefonía IP con Asterisk

BUENO, BONITO Y BARATÓ



Si tu centralita no da para más, sólo necesitas un ordenador para montar una nueva centralita IP con servicios avanzados de telefonía. **POR MEHDI CHOUIKH**

Actualmente existen varios motivos por los que las empresas están planteándose la integración de sus redes de voz y datos. En primer lugar está la reducción de costes y personal. En segundo lugar, la mejora en las comunicaciones y la productividad de los empleados. Y como tercera razón de peso encontramos las aplicaciones novedosas que pueden implementarse y que, sin duda, mejoran la comunicación con los clientes.

ASTERISK: la solución para la convergencia

Hace 4 años un amigo y yo decidimos montar una empresa de telefonía IP. Yo tenía experiencia en sistemas IP usando el protocolo h323, incluso había desarrollado sistemas de tarificación con dicho protocolo basándome en el GNUgatekeeper (un servidor de señalización para sistemas de telefonía IP usando el protocolo h323), que era una de las alternativas más fiables al software propietario de aquel entonces. Montamos una solución de telefonía IP y sólo nos faltaba el hardware para interconectar con la red de telefonía tradi-

cional, llamado gateway o pasarela de VoIP. Miramos el mercado, y el equipo que necesitábamos para 60 llamadas simultáneas costaba unos 8000 euros.

En aquel momento comenzaba a sonar el nombre de Asterisk: una plataforma de telefonía IP que lo hacía todo. Asterisk es un servidor de telefonía IP multiprotocolo con todas las funcionalidades de una centralita tradicional, que además puede actuar como gateway o pasarela entre las redes de VoIP y las redes de telefonía tradicional, o PSTN, usando hardware compatible tipo Digium o Sangoma. Tras investigar un poco sobre el proyecto y realizar los cálculos económicos decidimos arriesgarnos y usar Asterisk como nuestra pasarela principal, que en una primera fase fue diseñada para soportar 30 llamadas simultáneas a la red de telefonía tradicional. Los resultados finales fueron mejores de lo esperado y hoy no sólo lo usamos como gateway, sino que además lo usamos como servidor SIP para el registro de clientes. En este artículo explicamos cómo se puede usar Asterisk como gateway, para dar salida a las llamadas que provienen de una red IP en la red de telefonía tradicional (PSTN).

¿Por dónde empezar?

El nombre de Asterisk proviene del carácter (*) usado como comodín en los sistemas Linux. Es el nombre perfecto para esta PBX por varias razones, la más importante de las cuales es la variedad de interfaces con las que Asterisk puede conectar, incluyendo:

- Interfaces analógicas, como líneas RTB, llamadas FXO, o teléfonos analógicos, llamadas FXS
- Circuitos digitales, tipo E1 ó T1 (véase Figura 1)
- Protocolos VoIP, como SIP, h323, MGCP, IAX

Así que lo primero que haremos será definir los requisitos del sistema Asterisk que queremos montar. Dicho sistema debe ser capaz de interconectar una red VOIP formada por teléfonos IP que usan el servidor Asterisk como servidor de registro y enrutamiento hacia la red PSTN.

En este tipo de sistemas el parámetro más significativo es el número de llamadas simultáneas que ha de soportar el gateway en hora punta. En nuestro caso el número máximo de llamadas simultáneas será 30.

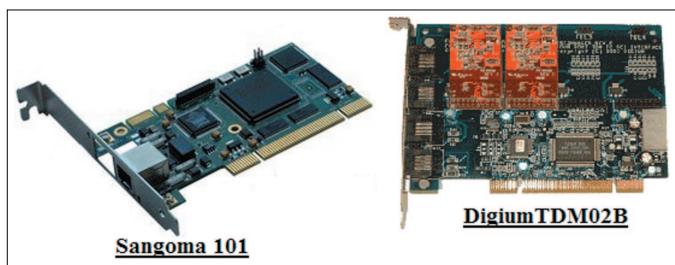


Figura 1: Tarjeta Sangoma para Interfaz E1 y una Digium para líneas analógicas.

1- **Elección del Hardware:** empezaremos eligiendo el hardware de nuestra máquina teniendo en cuenta que nuestro Gateway ha de ser capaz de soportar en hora punta la conversión de telefonía tradicional a IP de 30 llamadas simultáneas, la tarea principal de este equipo es la conversión entre dos protocolos, es decir, el recurso máspreciado es la CPU. Para nuestro caso hemos elegido un Pentium 4 a 3Ghz con 1MB de CACHE y 512 MB de RAM. El resto de hardware del equipo no es tan crítico.

2- **Elección de la tarjeta de comunicaciones:** para poder realizar 30 llamadas simultáneas, existen dos

que más nos guste: en nuestro caso instalamos una DEBIAN SARGE con un kernel 2.6.8. Después de la instalación del sistema operativo, debemos instalar algunos paquetes necesarios para la compilación de Asterisk y de los drivers Zaptel para tarjeta de comunicaciones:

- kernel-headers.
- bison.
- ncurses-dev.
- libssl-dev.
- libnewt-dev.
- zlib1g-dev.
- initrd-tools.



Figura 2: Hardware de telefonía IP [5].

- **Zaptel.conf:** Aquí irá la configuración de bajo nivel para la interfaz de hardware. Configuraremos los 30 canales de la E1.
- **Zapata.conf:** Asterisk usa este fichero para configurar y determinar cómo se va a comunicar con el hardware. Desde este fichero podemos controlar varios parámetros como identificación de llamadas entrantes, desvío de llamadas, cancelación de eco y muchos más.
- **Extensions.conf:** El plan de marcado será simple, pero demostrará que nuestro sistema funciona.
- **Sip.conf:** Donde configuraremos todo lo relacionado con el protocolo SIP, desde su comportamiento global hasta los usuarios autorizados y sus parámetros.

Vamos a comenzar configurando el canal Zaptel. Para ello modificaremos el fichero `/etc/zaptel.conf` en función del tipo de interfaz telefónico que tengamos. Vamos a hablar de dos de ellos: los SPAN para accesos primarios y los fxskts para líneas analógicas RTB.

La configuración de los SPAN tiene el siguiente formato de configuración:

```
span=<spannum>, <timing>,<
<LBO>, <framing>, <coding>
```

donde spanum es el número del conector de entrada, timing es el parámetro que especifica si la señal de reloj la proporciona el proveedor (valor 1) o no (valor 0). Y LBO es el valor que nos indica el nivel de potencia de la señal, que varía en función de la distancia a la central: por defecto es 0 salvo que el proveedor indique lo contrario. framing y coding definen cómo nos comunicamos con el hardware para un acceso E1. framing toma el valor hdb3 y coding el valor crc4.

Listado 1: Instalación

```
01 # cd /usr/src
02 # svn checkout http://svn.digium.com/svn/asterisk/branches/1.2
   asterisk-1.2
03 # svn checkout http://svn.digium.com/svn/zaptel/branches/1.2
   zaptel-1.2
04 # svn checkout http://svn.digium.com/svn/libpri/branches/1.2
   libpri-1.2
05 # cd ../libpri-1.2
06 # make clean; make install
07 # cd zaptel-1.2
08 # make clean; make install
09 # cd ../asterisk-1.2
10 # make clean; make install; make samples
```

posibilidades. La primera es contratar 30 líneas analógicas RTB. Esto es poco viable: la placa base de un ordenador de gama media-alta tiene como máximo seis slots PCI, por lo que sólo podemos usar 6 tarjetas Digium PCI de 4 puertos FXO, es decir 24 líneas como máximo. Además el coste de dichas tarjetas es elevado, cada una supone el 40% del precio de una tarjeta E1. La segunda posibilidad es mediante un acceso Digital E1 usando una tarjeta Digium TE110P.

3- **Instalación del sistema operativo:** una vez tengamos todo el hardware montado, instalaremos la distro Linux

Instalación

Antes de comenzar la instalación, descargamos el software necesario desde [1] para instalar Asterisk y Zaptel, y seguimos los pasos que aparecen en el Listado 1.

Configuración

Aquí nos metemos en faena de verdad, ya que comenzaremos a construir nuestro sistema Asterisk. Para ello tendremos que modificar los siguientes archivos de configuración de la carpeta `/etc/asterisk` y el fichero Zaptel que se encuentra en `/etc`:

Listado 2: Ejemplo de Configuración de zaptel.conf para un E1.

```
01 span=1,1,0,ccs,hdb3,crc4
02 # Span 1 --es un comentario
03 bchan=1-15
04 dchan=16
05 bchan=17-31
```

En caso de haber elegido línea analógica RTB en lugar de span usamos *fxsks = número-interfaces* para las líneas telefónicas [2].

Ahora pasaremos al */etc/asterisk/zapata.conf*. Este fichero es el más complejo, ya que encontramos muchos parámetros (véase Listado 3). El fichero

```
helena:~# ztcfg -v
Zaptel Configuration
=====
SPAN 1: CCS/HDB3 Build-out: 0 db (CSU)/0-133 feet (DSX-1)
31 channels configured.
helena:~#
```

Figura 3: Resultado mostrado tras configurar Zaptel.

tiene una sola sección [channels], en donde especificamos el modo de señalización y los distintos parámetros para el uso de la tarjeta Zaptel. En el Listado 3 hay un ejemplo con los parámetros más usados, el primero es *context* y se refiere a la sección del plan de marcado (extensiones.conf) donde definimos las acciones a realizar con las llamadas recibidas. *transfer* es el parámetro que nos permitirá transferir las llamadas recibidas por este canal. *signalling* define el modo de señalización que usa la tarjeta Zaptel y tiene que coincidir con lo definido en el *zaptel.conf*. En nuestro caso el valor es "pri_cpe", que corresponde a los accesos E1. La siguiente opción es *SWITCHTYPE*, que nos define el tipo de marcado de un canal E1, y el valor más común es "euroisdn", aunque depende del proveedor del servicio. Para poder habilitar los parámetros de identificación de llamada usamos los parámetros *usecallerid*, igual a "yes", y *callerid*, igual a "asreceived"; para eliminar el eco creado por las líneas analógicas jugaremos con las opciones *echocancel* y *echocancelwhenbridged*. Los pondremos a "yes", pero aún así el cancelador requiere un cierto tiempo para

aprender a corregir el eco: para esto añadimos el parámetro *echotraining* con el valor "yes". Esto le indica a Asterisk que envíe un tono al inicio de cada llamada para medir el eco de la línea, y aprender así con mayor velocidad el comportamiento del eco. Si aún así seguimos teniendo algo de eco podemos terminar de cancelarlo jugando con las ganancias de recepción *rxgain* y transmisión *txgain*: sus valores pueden variar de -100% a 100%. Finalmente definimos el número de canales que tenemos, e incluso podemos agruparlos para ser tratados como uno solo. En nuestro caso hemos agrupado los 30 canales en un solo grupo. Para ver más opciones de configuración de *zapata.conf* véase [4].

Una vez modificados los ficheros *zaptel.conf* y *zapata.conf*, tendremos que comprobar si funciona la configuración que hicimos. Para ello vamos a cargar los drivers de

zapatel con los órdenes *#modprobe zapatel* y *#modprobe wctw11xp*. Tras esto, ejecutamos el configurador del *zaptel* con *#ztcfg -v*, que nos mostrará el número de canales configurados (véase Figura 3).

Pasamos a ver el fichero *sip.conf*, donde decidimos quién puede registrarse en nuestro sistema y con quién nos podemos registrar. Vemos que está estructurado en secciones. La primera es [general] donde se configura el comportamiento global del protocolo y el canal SIP:

```
[general]
port=5060
;Puerto de registro
bindaddr=0.0.0.0
;dirección ip donde escuchar
;la ip 0.0.0.0 = todas
context=default
;Contexto por defecto
disallow=all
;Deshabilita todos los codecs
allow=gsm
;Permite sólo el códec gsm
```

A continuación encontramos una sección por usuario tipo [usuario] y luego los parámetros del usuario como sigue:

```
[100]
type=friend
secret=suclave
context=salientes
callerid="Usuario 1" <100>
host=dynamic
canreinvite=yes
disallow=all
allow=gsm
```

Después de crear los usuarios SIP, sólo nos falta diseñar nuestro plan de marcado para poder llamar y recibir llamadas.

Listado 3: Ejemplo de Configuración de zapata.conf para un E1.

```
01 [channels]
02 context=entrantes
;Contexto donde entran las
;llamadas en plan de marcado
03 transfer=yes ;Nos
;permitirá transferir llamadas
04 signalling=pri_cpe
05 switchtype=euroisdn
06 usecallerid=yes
;Habilita y deshabilita el uso
;del número llamante
07 echocancel=yes
;parámetros para la
;cancelación del eco
08 echocancelwhenbridged=yes
09 echotraining=yes
10 group=1
11 rxgain=0.0 ;ganancia
;en db de las señales de
;entrada y salida.
12 txgain=-5.0
13 channel => 1-15
14 channel => 17-31
```

Para ello modificamos el fichero *extensions.conf*, considerado por muchos el fichero más importante de Asterisk. Este fichero a su vez tiene varias secciones, la primera de las cuales es *general*, y comienza siempre así:

```
[general]
static=yes
writeprotect=yes
```

Las siguientes secciones son los contextos. Cada contexto tiene al menos una extensión (una extensión es un número de teléfono que puede estar compuesto por números, letras o una combinación de ellos). Cada extensión tiene priori-



dades y aplicaciones para ejecutar. Gracias a los contextos podemos organizar nuestro plan de marcado y restringir llamadas a algunos destinos a los usuarios que queramos.

Las extensiones tienen el siguiente formato:

```
exten => número, prioridad,➤
aplicación(arg1,arg2,...)
```

donde número es el número de teléfono, la prioridad es el orden en la cual se ejecuta, y finalmente aplicación, la aplicación a ejecutar. La más importante de estas últimas es Dial(), que permite realizar llamadas, y a su vez tiene el siguiente formato: Dial(canal/número, tiempo espera, opciones)

El primer contexto en nuestro plan es [default], Asterisk lo usa para manejar las llamadas no autorizadas o no definidas para el canal SIP, así que lo único que pondremos en este contexto es un mensaje que avisa al usuario de que el número al que están llamando no es correcto.

```
-- Executing Dial("SIP/100-9ad9", "zap/g2/902023156|90|t|T") in new stack
-- Requested transfer capability: 0x00 - SPEECH
-- Called g2/902023156
-- Zap/32-1 is proceeding passing it to SIP/100-9ad9
-- Zap/32-1 answered SIP/100-9ad9
-- Hungup 'Zap/32-1'
```

Figura 5: La consola de Asterisk muestra esto al realizar una llamada.

```
-- Starting simple switch on 'Zap/44-1'
-- Accepting overlap call from '680982256' to '951047200' on channel 0/13, span 2
-- Executing Dial("Zap/44-1", "sip/951047200@10.143.0.241|90|Tt") in new stack
-- Called 951047200@10.143.0.241
```

Figura 6: La consola de Asterisk muestra esto al recibir una llamada.

```
[default]
exten => s,1,Answer
exten => s,2,Playback(invalid)
```

Luego definiremos el contexto para llamadas salientes, al que llamaremos [local], que tendrá una sola extensión para las llamadas nacionales que sólo permite llamadas que empiecen con 9 y en las que el segundo dígito sea una cifra entre 1 y 9, del tercero al último entre 0 y 9 y con la longitud que sea. Esta extensión tiene dos prioridades, la primera ejecuta la aplicación Dial eligiendo el grupo 1 del canal Zaptel usando la tarjeta que configuramos llamando al número local definido por la variable \${EXTEN} que no es más que el número marcado.

```
[locales]
exten => _9ZXXXXXXX,1,➤
Dial(Zap/g1/${EXTEN},90,Tt)
exten => _9ZXXXXXXX,2,Hangup()
```

Finalmente definimos el contexto para llamadas entrantes [entrantes] que recibimos a través de nuestro acceso telefónico.

```
[entrantes]
exten => 951233900,1,➤
Dial(Sip/100,90,Tt)
exten => 951233900,2,Hangup()
```

Probando Probando, 1, 2, 3

Por fin, ejecutamos Asterisk con el comando `Asterisk -vvvc`, configuramos un teléfono IP software o hardware (véanse Figuras 1 y 2), y probamos a realizar llamadas a números locales (véase Figura 4 con el resultado de la consola de Asterisk). Finalmente realizamos una llamada a un número de nuestras líneas (véase Figura 5) y veremos cómo suena la extensión correspondiente, y al descolgar... ¡todo funciona!

Conclusión

Además de lo contado, con Asterisk se puede hacer cualquier cosa imaginable: un centro de llamadas de atención al cliente, un sistema de respuesta interactiva, un sistema de consulta de pedidos, donde al cliente le contesta una operadora digital que le presenta unas opciones, introduce su código, y le responde con el estado de su pedido... entre otra infinidad de cosas. Lo único que hace falta son ganas y algo de tiempo. Para más información recomendamos una visita a [6].

EL AUTOR

Mehdi Chouikh, es ingeniero técnico de telecomunicaciones, Responsable de VoIP de Universal Telecom Experts, y tiene más de 4 años de experiencia ofreciendo servicios de telefonía y networking.

RECURSOS

- [1] Sitio oficial de asterisk <http://www.asterisk.org>
- [2] Información adicional sobre la configuración del Zaptel: http://www.digium.com/downloads/configuring_zaptel.pdf
- [3] Cómo configurar el wanpipe de Sangoma: <http://sangoma.editme.com/wanpipe-asterisk-configure>
- [4] Información adicional sobre el fichero zapata.conf: http://www.digium.com/asterisk_handbook/zapata.conf.pdf
- [5] En este sitio podemos comprar teléfonos IP y servicios de telefonía IP <http://www.mitelefonovirtual.com>
- [6] Wiki acerca de telefonía IP, con énfasis en Asterisk <http://www.voip-info.org>