Curso de Asterisk
15
Práctico

Capa Tres Soluciones Tecnológicas S.L.
Instructor: Juan Carlos Valero

Licencia

¿Quién da la clase?

CapaTres
Soluciones Tecnológicas S.L.

Juan Carlos Valero

jcvalero@capatres.com

Avda Primero de Mayo s/n Local 20
(Locales Plaza Pedro IV exteriores)
08120 La Llagosta - Barcelona

Tel: +34 935605424
Fax: +34 935748256

www.capatres.com

dCAA Digium®-certified Asterisk® Administrator

dCAP Digium Certified Asterisk Professional

redhat. CERTIFIED ENGINEER

ELASTIX CERTIFIED ENGINEER
ECE# 200222451

CapaTres
Soluciones Tecnológicas S.L.

Consultoría Tecnológica especializada en Asterisk

http://www.capatres.com

- Operando desde el año 2005
- Amplia experiencia, 100% Asterisk y VOIP
- Especializados en soluciones a medida
- Instalaciones de todo tamaño (simple sede, multisede, etc)
- Sistemas Clusterizados, Virtualización, integraciones, etc.
- Operador IP desde 2012.
- Formadores de Asterisk independientes

¿Quién asiste al curso?

Ronda de presentaciones:

Permiten que el instructor sepa el nivel de los asistentes

-Nombre de cada asistente
-Conocimientos de Linux (experiencia a nivel de línea de comando, instalaciones, versiones o distribuciones de Linux tocadas, nivel de soltura, si se ha compilado aplicaciones,etc).
-Conocimientos de Asterisk (instalado o no, si fue mediante paquetes o compilando, si se han modificado ficheros de configuración o realizado algún despliegue, etc).

Objetivo del curso

- Sentar bases sólidas con Asterisk que faciliten el desarrollo posterior de conocimientos avanzados.
- Familiarizar a los asistentes con los sistemas de Voz sobre IP basados en Asterisk.
- Enseñar lo preciso para que los asistentes sean capaces de desarrollar tareas de instalación y configuración.

Perder el miedo a Asterisk y conocer sus posibilidades es el principal objetivo de este curso.

¡ Preguntar al profesor y pedir aclaraciones es obligatorio !

¿Cuál es el mejor Linux para Asterisk ?

En pocas palabras:

EL QUE MEJOR CONOZCAS

La capacidad de resolver los problemas en la base Linux es mas importante que los posibles incidentes al usar Asterisk. Si se conoce bien el sistema base, el resto de factores no tienen mayor importancia.

El curso se imparte sobre CentOS, un clon gratuíto de RedHat Enterprise Linux, pero los conocimientos son utilizables sin problemas en Debian ya que el curso se plantea de forma lo más neutral posible.

¿Qué es Asterisk?

Una aplicación de centralita Open Source para:

-Usuarios domésticos
-Pequeñas y Medianas Empresas
-Grandes Empresas
-Proveedores de servicios VoIP
-Compañías telefónicas

¿Qué es un B2BUA?

- Asterisk es un Back to Back User Agent. B2BUA es como se denomina a una aplicación que controle llamadas entre usuarios SIP y a diferencia de un Proxy SIP (en que este únicamente gestiona el estado de una llamada cuando se realiza), el B2BUA mantiene el estado de las llamadas para conseguir información valiosa en determinados entornos como facturación, redireccionamiento de llamadas en caso de caída de un proveedor SIP, etc.

- Asterisk es mucho más que un B2BUA ya que no únicamente controla todo esto, si no que incluso puede llegar a realizar acciones que ni un Proxy SIP ni un B2BUA pueden realizar (de un modo simple y sin recurrir a terceras aplicaciones) como: grabaciones de llamadas, sistemas de buzón de voz, reproducción de locuciones, ofrecer menús IVR, reproducir música en espera, y un larguísimo etc..

¿Que no es Asterisk?

- Asterisk NO es un proxy SIP. Aunque posea funcionalidades de estos, carece de la mayor parte de la implementación SIP necesaria (por ejemplo SIP MESSAGE, PRESENCIA, etc). Para proveer funcionalidades de Proxy SIP existen otros productos, como por ejemplo openSER, Kamailio,etc.

- Asterisk NO es un tarificador de llamadas. Puede usar herramientas externas para tarificarlas.

- Asterisk NO es una solución de MultiVideoConferencia. Al menos a corto plazo, no se espera transcoding de video en tiempo real.

- Asterisk NO es un servidor de faxes. Puede interactuar con productos de terceros para enviar y recibir faxes, pero no incluye esa funcionalidad de serie. Para eso existe Hylafax

Breve historia de Asterisk

* Mark Spencer crea en 1999 la empresa Linux Support Services (LSS) pero no tiene PBX.
* Versión 0.1.0 de Asterisk el 5 de Diciembre de 1999
* Jim Dixon crea independientemente el proyecto Zapata Telephony. La unión de los dos proyectos da lugar a Asterisk como lo conocemos hoy día.
* LSS cambia su nombre a Digium en 2002.

Versiones de Asterisk

Release Series	Release Type	Release Date	Secutiry iFx only	EOL
1.2.x		2005-11-21	2007-08-07	2010-11-21
1.4.x	LTS	2006-12-23	2011-04-21	2012-04-21
1.6.0.x	STANDARD	2008-10-01	2010-05-01	2010-10-01
1.6.1.x	STANDARD	2009-04-27	2010-05-01	2011-04-27
1.6.2.x	STANDARD	2009-12-18	2011-04-21	2012-04-21
1.8.x	LTS	2010-10-21	2014-10-21	2015-10-21
10.x	STANDARD	2011-12-15	2012-12-15	2013-12-15
11.x	LTS	2012-10-25	2016-10-25	2017-10-25
12.x	STANDARD	2013-12-20	2014-12-20	2015-12-20
13.x	LTS	2014-10-24	2019-10-24	2020-10-24
14.x	STANDARD	2016-09-26	2017-09-26	2018-09-26
15.x	STANDARD	2017-10-03	2018-10-03	2019-10-03
16.x	LTS	2018-10-11	2022-10-11	2023-10-11

Instalación Linux CentOS (1)

* El curso está preparado sobre CentOS Linux 7.x. pero es directamente transportable las instrucciones para usar Debian 9.
* Hacer una instalación mínima del entorno
* Indicar y apuntar la dirección IP del servidor y la contraseña del usuario root, que necesitaremos después.
* Tras la instalación reiniciar.

Instalación Linux CentOS(2)

* Una vez reiniciado, actualizar equipo: `yum update / apt-get update`
* Deshabilitar selinux y cortafuegos:
  - `systemctl disable firewalld`
  - `systemctl stop firewalld`
  - `nano /etc/sysconfig/selinux (en Debian esta deshabilitado por defecto)`
* Reiniciar de nuevo.

Instalación Linux CentOS(y 3)

* Instalar un entorno gráfico (solo si no se dispone de un portatil o navegador para configurar los telefonos. No usar en sistemas en producción) :
  * `yum groupinstall "Escritorio Gnome"`
  * `apt-get install gnome  (desde Debian)`
  * Para arrancar `startx`
  * Usuario normal `versus` usuario privilegiado...

Circuitos y Redes (1)

* Redes orientadas a Circuitos (ejemplo línea analógica)
* Se establece un circuito dedicado o exclusivo para cada abonado.
* Una vez establecido el circuito, éste ya no puede ser usado por otros.
* Este tipo de redes es costoso.
* En cada circuito el retardo es constante, lo cual de cierto modo es una ventaja pues no hay jitter.
* Es el tipo de redes típico de las empresas de telefonía fija para con los abonados analógicos.

Circuitos y Redes (2)

* Redes Orientadas a Paquetes (ejemplo RDSI o ethernet)
* Por un mismo medio se puede transmitir simultáneamente diferentes flujos de información.
* La información de los diferentes nodos se divide en paquetes, se intercalan y se envían por el mismo medio.
* Internet es un ejemplo de red de paquetes.
* En Internet y redes IP en general los paquetes pueden llegar desordenados. Esto puede ocasionar problemas cuando se transmite voz.

Tipos de líneas PSTN

* Loop-Start: se solicita el tono de marcado cerrando el bucle.
* Ground-Start: se solicita el tono de marcado poniendo el par a tierra.
* Kewlstart: combina loop-start con supervisión de desconexión remota, junto con que el bucle se abre cuando el otro extremo de la llamada cuelga al final de la llamada. En uso en España.

Señalización PSTN

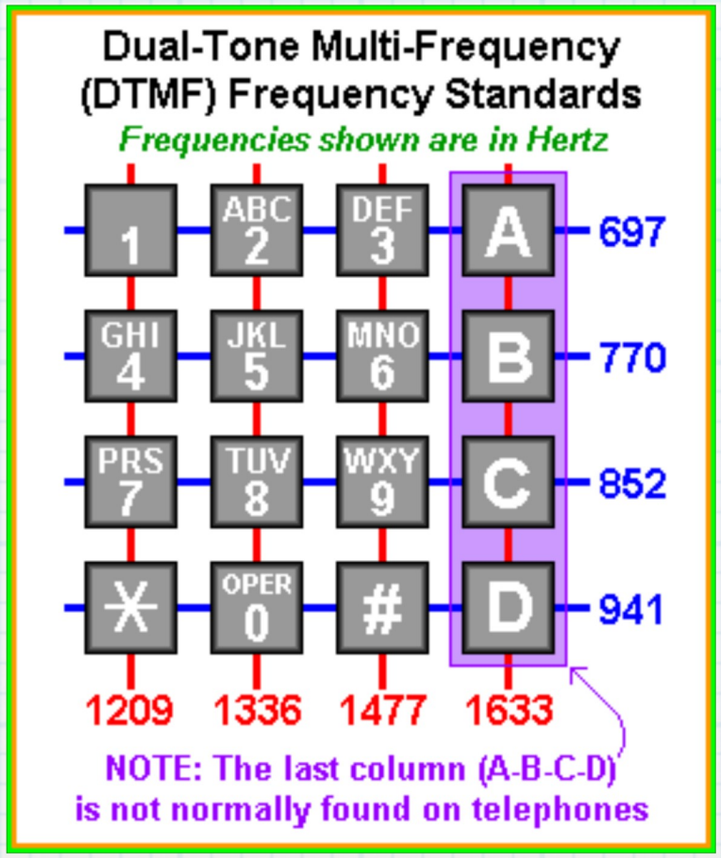
* Marcado por pulsos: se señalizan los dígitos mediante rapidas aberturas y cierres del bucle (<100ms).

* DTMF: se envían los números como secuencias de dos tonos.

* Supervisión de la llamada: se señaliza el inicio y fin de la llamada a las partes. En España se usan las inversiones de polaridad para ello (ver documento de Interfaces de Telefónica).

* Valor REN: aunque no es un parámetro de señalización como tal, indica el número de dispositivos que como máximo pueden estar conectados a un par de cobre. En España el REN es de 5 para las líneas de telefónica.

DTMF



* Importante: la transmisión de tonos DTMF con una compresión de audio elevada DESTRUYE los tonos. Cuando se usen codecs como G729 se debe usar un medio de señalización para los DTMF distinto del audio.

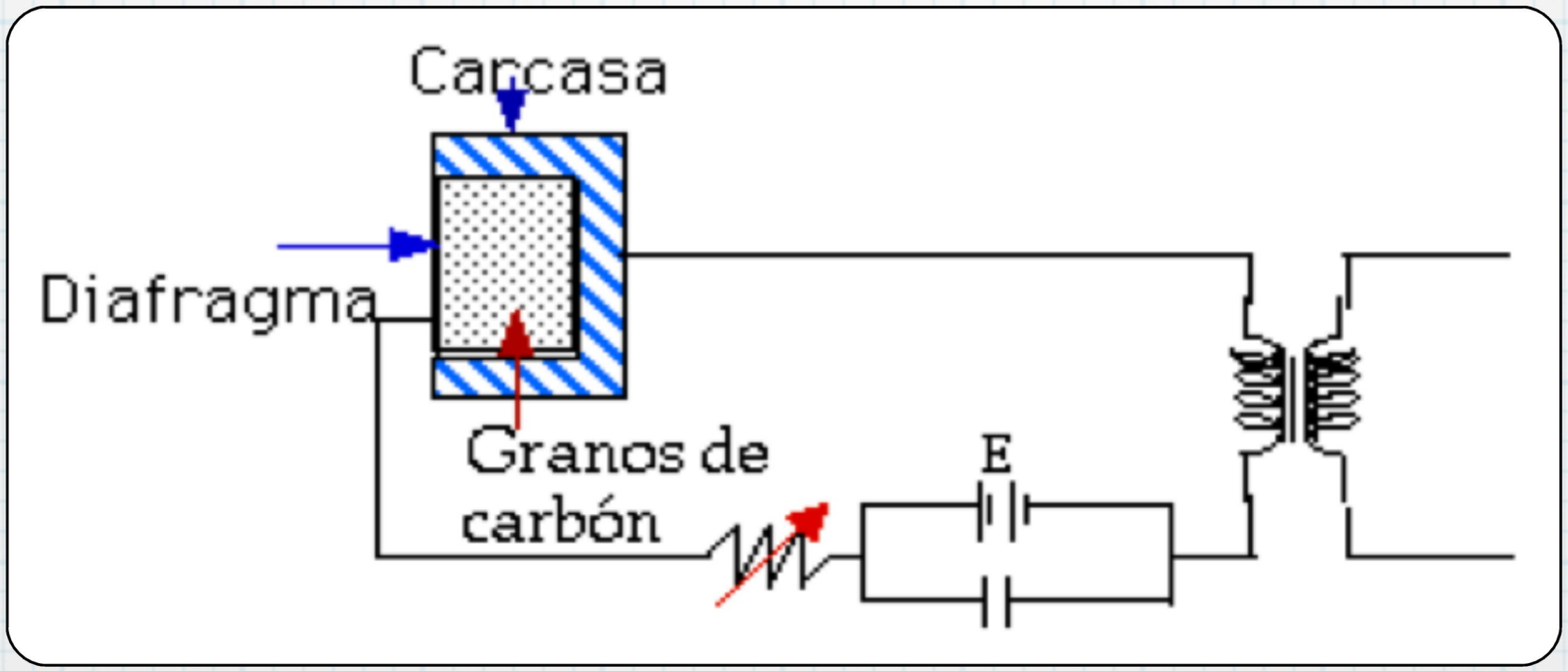
Señalización BRI/PRI

* Las líneas digitales, a diferencia de las analógicas, tienen un canal dedicado a la señalización. Por él viaja el inicio de llamada, establecimiento, mensajes y el cuelgue. Por los canales de voz, a diferencia de los de señalización, viaja solo la voz.

* En una red digital, hay un iniciador de red y terminadores de esta. Los iniciadores se dice que están en modo NT, y los puntos finales en modo TE.

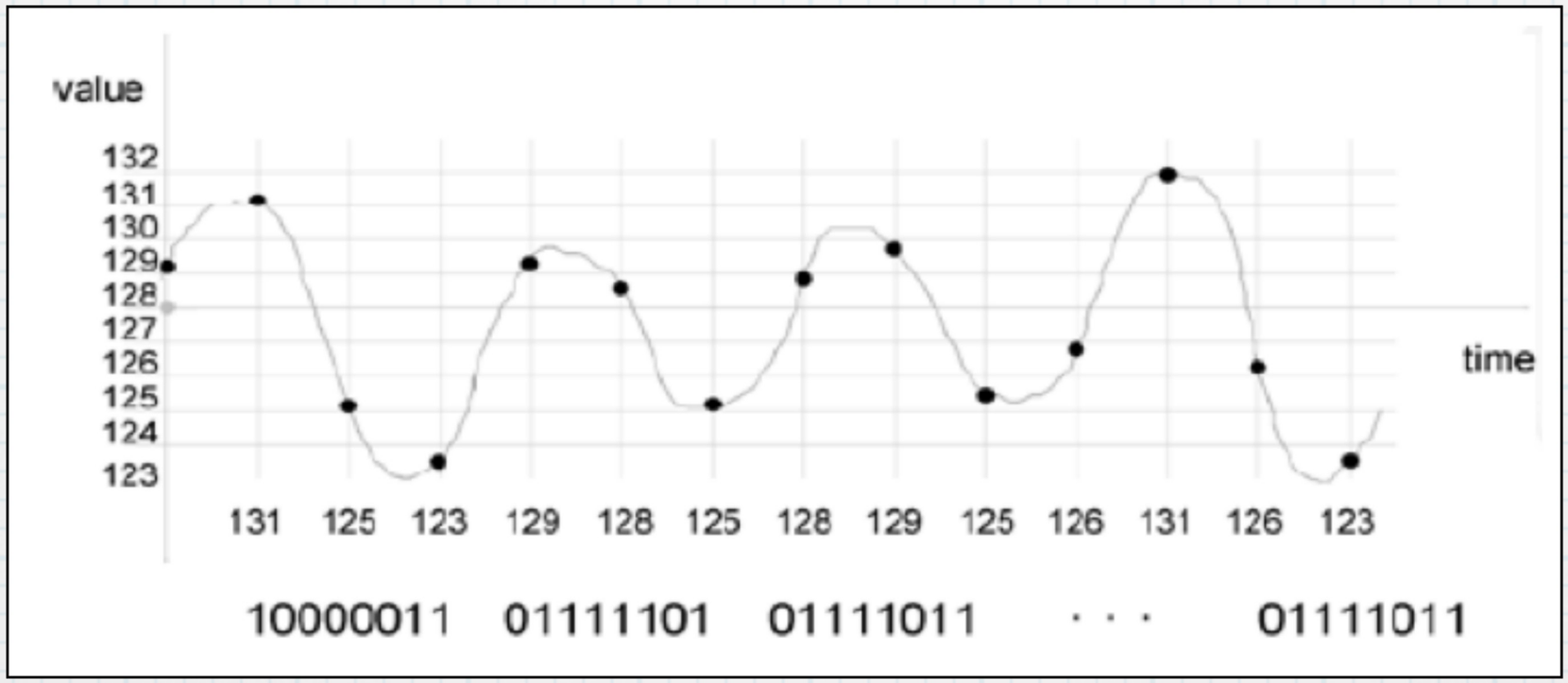
* Adicionalmente se dice que un punto de conexión puede estar en modo punto a punto (PTP) cuando solo hay dos pares en la línea, o en modo Punto a Multipunto (PTMP) cuando hay un origen y múltiples destinos.

¿Como funciona el audio digital ? (1)



- En el origen era la ley de ohm: I=e/r
- Formas de onda moduladas en amplitud.

¿Como funciona el audio digital ? (2)



- Toma de muestras a intervalos predeterminados.
- La información se convierte en un flujo de datos.

El teorema de Nyquist

* Establece la mínima frecuencia de muestreo para que la onda se pueda reconstruir en destino igual a la original.

* Nyquist sólo determina una frecuencia mínima. Teóricamente los valores muestreados deben ser exactos, pero en la práctica esto se redondea a un número finito de bits.

* Esta frecuencia mínima es 2 veces el ancho de banda que se quiere muestrear: $fm \geq 2\ BW$

* Por ejemplo, si en el teléfono se transmite voz de 400Hz a 4,000Hz se necesitará mínimo el doble, es decir 8,000Hz para muestrear esa señal.

* 8.000 Hz es la velocidad de muestreo usada en las tarjetas de telefonia en Asterisk.

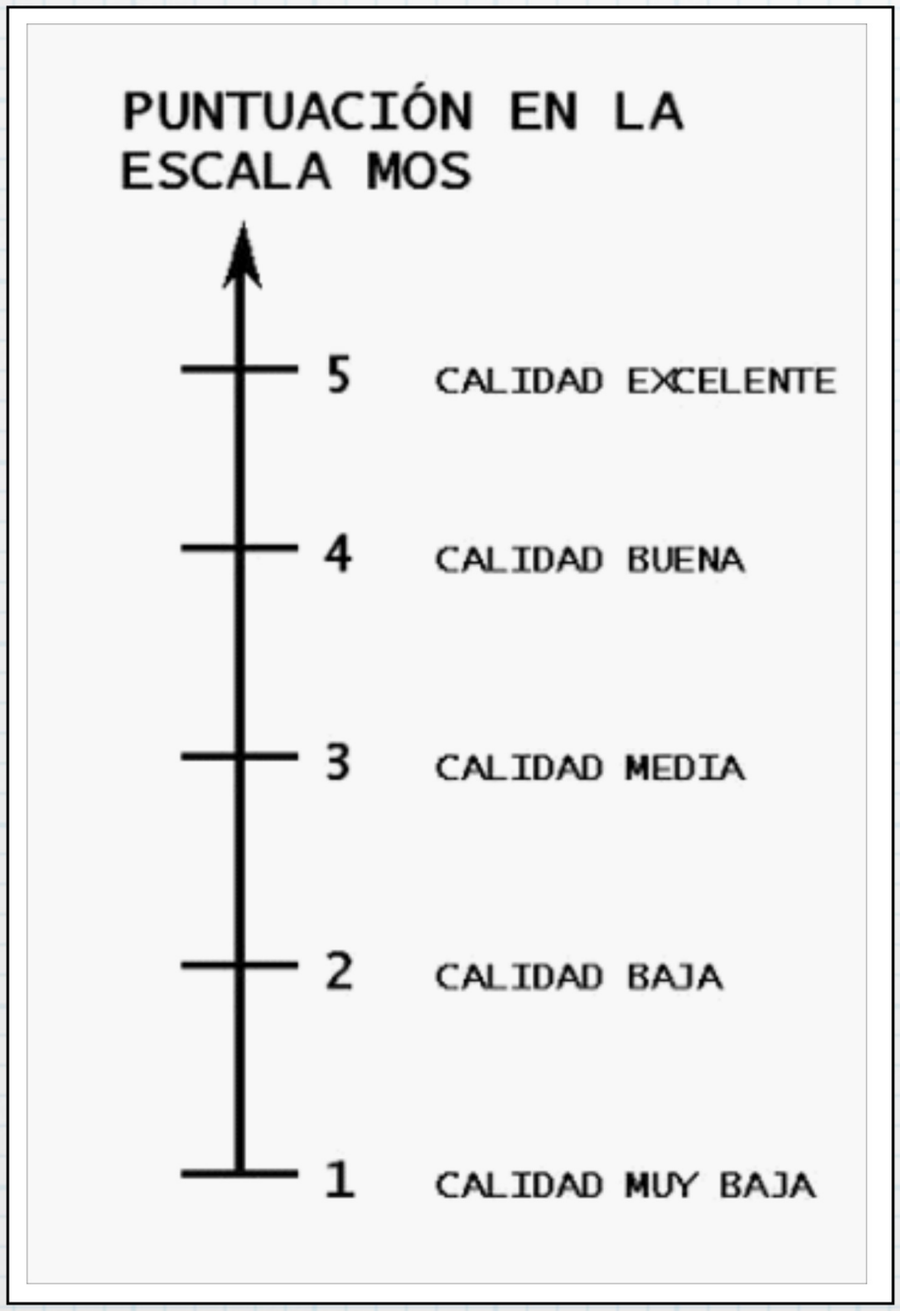
Códecs (1)

* Una voz digitalizada en PCM a 8 Khz, 16 bits por muestra, representa un flujo de datos de 128 Kb/Seg.
* Podemos reducir el ancho de banda necesario gracias al uso de códecs (compresores en tiempo real del audio).
* El ancho de banda preciso SIEMPRE será simétrico.
* El uso de un códec implicará en aquellos casos de reducción de ancho de banda una pérdida de calidad. G729=calida linea analógica.
* Tener en cuenta la sobrecarga de paquete TCP, una transmisión de 64 Kb/seg puede llegar a 80 Kb/seg.

Códecs (2)

Códec	Consumo	Ethernet	MOS	Notas
G711a	64 Kb/seg	87,2 Kbps	4,1	RDSI/PRI/Excelente calidad
G722	64Kb/seg	87,2 Kbps	4,13	Recomendado para LAN
ILBC	13.3 Kb/seg	38,4 Kbps	NA	Buena adaptación a pérdida de paquetes
G729	8 Kb/seg	31,2 Kbps	3,92	Comercial, con coste de licencia
G723	6.3 Kb/seg	21,9 Kbps	3,9	Comercial, con coste de licencia

Escala MOS



La escala MOS (Mean Opinion Score) es un modo de cuantificar la calidad del audio basándonos en la opinión del usuario. No es un método objetivo, que los hay, sino uno subjetivo que depende de la interpretación del audio que haga el usuario.

Paquetización

* El flujo contínuo de audio se divide en paquetes, en “chunks” con un tamaño prefijado. Usualmente se usan 20ms de audio por paquete. Eso son aproximadamente 320 bytes.
* Se añaden al flujo RTP las cabeceras IP, que son unos 20 bytes adicionales (sobrecarga aproximada de un 8-10 %).
* Los paquetes se envían hasta su destino, que debe ordenarlos para extraer la información.
* Importante que la MTU sea menor que el tamaño de paquete para evitar retardos adicionales. Recomposición de la información.
* Se suele usar UDP como método de transmisión en vez de TCP para ahorrar el tiempo de confirmación de entrega.

Protocolos y Asterisk

* SIP (Session Initiation Protocol). Bien soportado.
* IAX (Inter Asterisk Exchange). Nativo.
* H323 (via OpenH323). Implementación no bien soportada.
* SCCP (Cisco Call Manager). Implementación parcial.
* MGCP. No bien soportada.
* Skype (no soportada desde la compra de Skype por MicroSoft)
* XMPP/Jabber. Bien soportado desde Asterisk 1.8.
* Líneas analógicas, RDSI, móviles y primario via DAHDI.

Protocolo SIP (1)

* SIP = Session Initiation Protocol.
* Creado por el IETF como un standard para la transmisión de streamings en los años 90.
* Definición inicial en el RFC 3261, ampliada en muchos RFC según ámbitos de operación.
* SIP solo se encarga de poner a dos pares en contacto entre sí.
* La gestión y negociación del audio y/o video se deja en manos de otros protocolos como son SDP (Session Description Protocol) y RTP (Realtime Protocol). Estos se definen en los RFC 4566 y RFC 1889.

Protocolo SIP (2)

- SIP usa una serie de funciones, llamadas primitivas:
  - REGISTER
  - NOTIFY
  - INVITE
  - INFO
  - OPTIONS
  - etc...

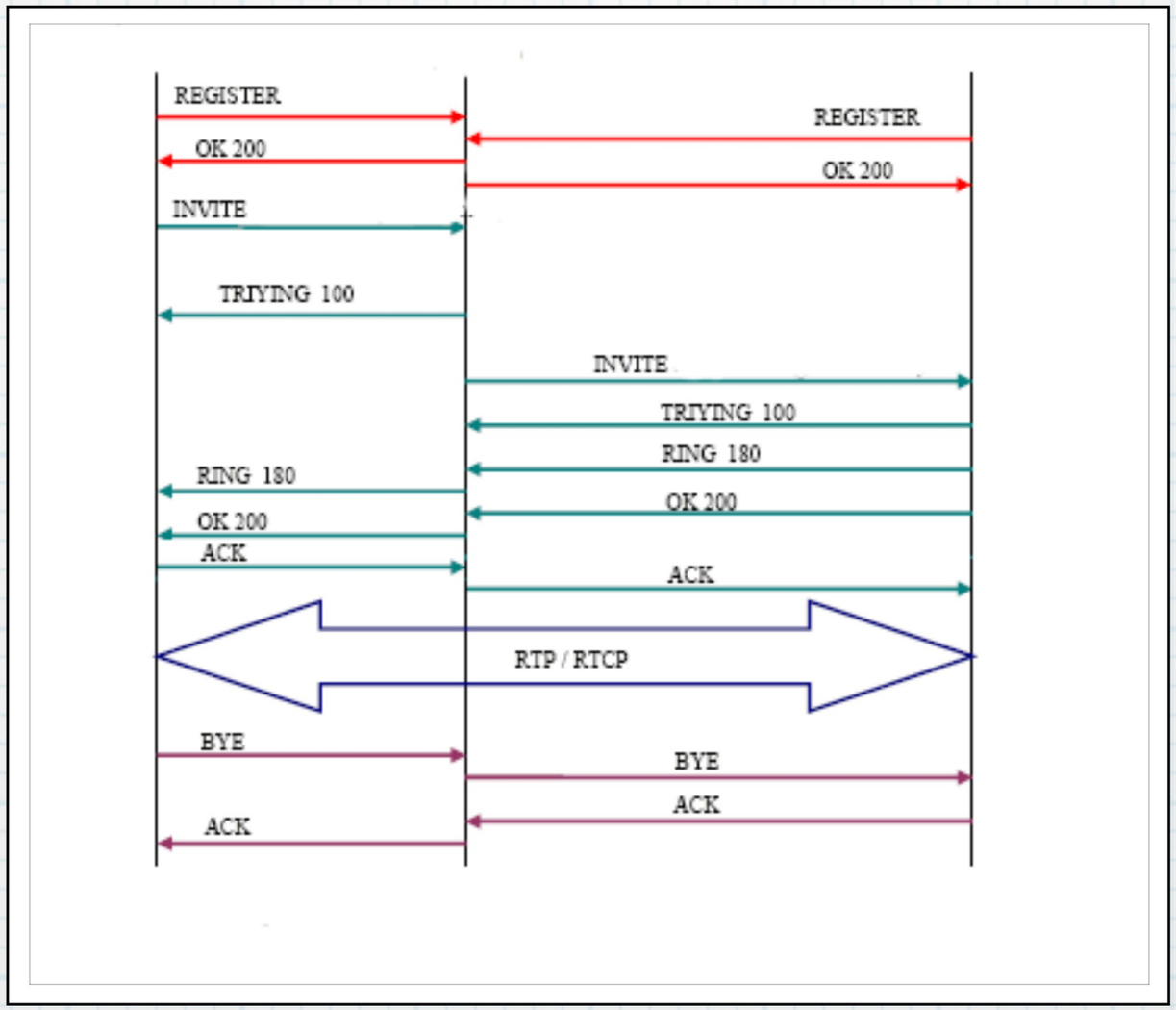
Protocolo SIP (3)

- SIP no provee servicios, solo las primitivas que pueden ser usadas para implementar servicios.
- SIP es un protocolo de transporte.
- Actualmente SIP es el standard de facto en el mercado de la Voz sobre IP.
- Usa el puerto 5060 UDP por defecto como puerto de señalización.
- Cada llamada requiere dos puertos adicionales, uno por cada sentido de la llamada, por los que se transporta mediante RTP el flujo de audio. Por tanto cada llamada precisa 3 puertos.

Protocolo SIP (4)

- Por tanto, usamos tres protocolos distintos para establecer una llamada:
  - SIP (Session Initiation Protocol): maneja la señalización, inicio, fin y establecimiento de la llamada.
  - SDP (Session Description Protocol): maneja la negociación de códecs entre los pares implicados en la llamada. Recoge la información de puertos implicados en cada extremo de la llamada.
  - RTP (Realtime Protocol): transporta el audio propiamente dicho.

Protocolo SIP (y 5)



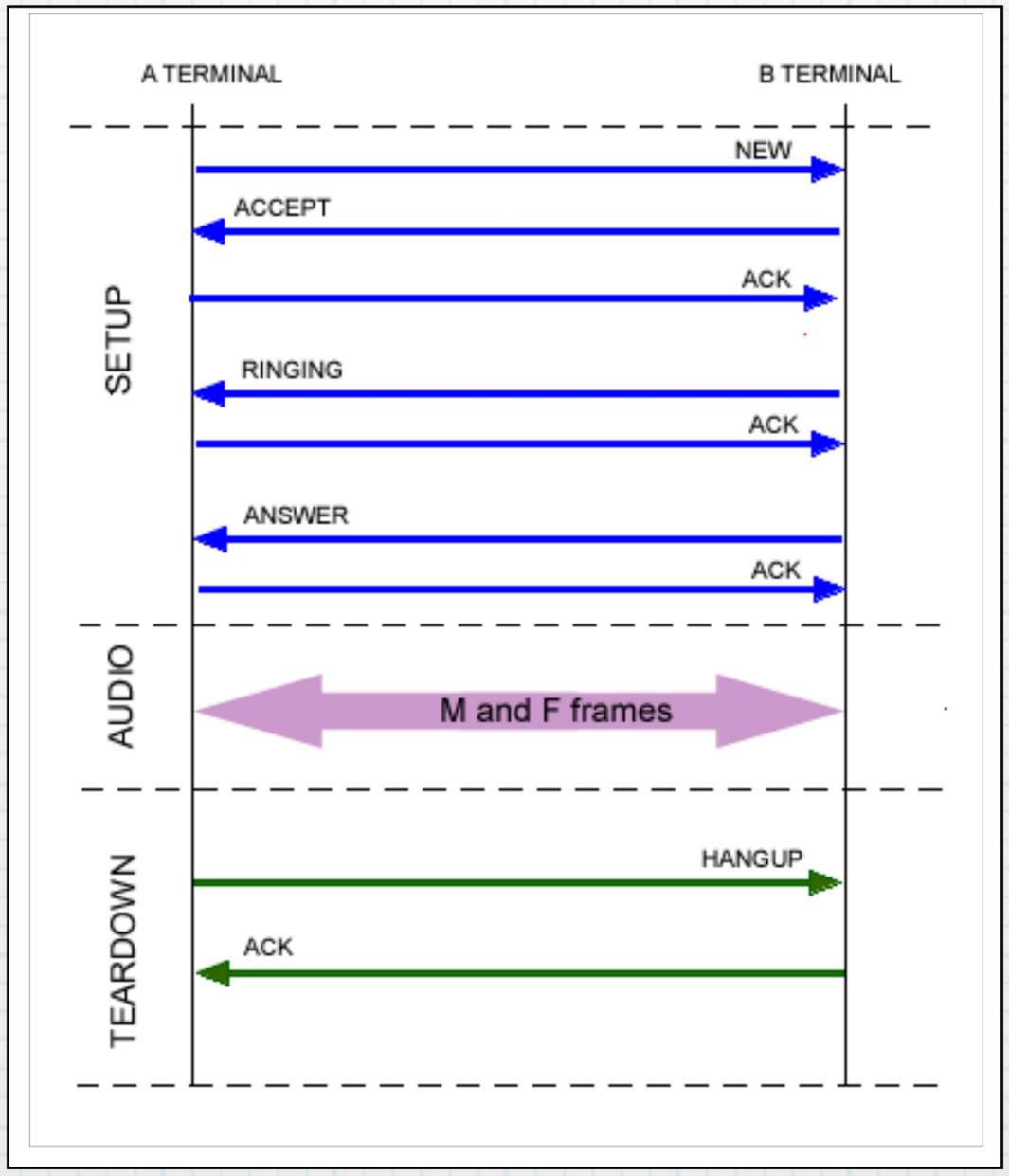
Protocolo IAX (1)

* Inter-Asterisk Exchange versión 2
* Diseñado por Mark Spencer como un protocolo de interconexión de Asterisk.
* Atraviesa mejor los NAT que SIP. ¿Pero por qué?
* Un único puerto (4569 UDP) para transportar señalización y audio.
* No es un standard como SIP, pocos dispositivos. Testing incompleto.
* Posibilidad de agrupar las llamadas en un Trunk.

Protocolo IAX (2)

* Cada llamada IAX2 en modo trunk ocupa 9.6 Kbps en su framing y sobrecarga de cabeceras TCP. Ayuda a reducir el consumo.
* Usando G711 esa reducción supone un 13% del ancho de banda.
* Usando G729 la reducción representa el 55% del ancho de banda.
* Ojo, no todos son ventajas. Menor implementación, posibles problemas de seguridad que no se han hecho manifiestos. No es un standard aprobado aunque hay un RFC solicitado.

Protocolo IAX (y 3)



Latencia

* Tiempo que tarda un paquete de información en cruzar por la red.
* RTT: Round Trip Time
* El valor máximo de RTT es de 400 ms, idealmente <200ms
* Contra menos, mejor. Idealmente 0.
* Llamadas con latencia elevada se apreciarán como el audio llegando más tarde de lo que debía.

Pérdida de Paquetes

* Se dice que tenemos pérdida de paquetes cuando uno o más no llegan a su destino.
* Protocolo TCP asegura la entrega a costa de un mayor consumo de ancho de banda. Puede dar lugar a escenarios de bloqueo.
* Protocolo UDP no asegura la entrega. Si no llega se deduce al reordenar paquetes.
* La pérdida de paquetes ha de ser cero o la cifra más próxima posible.
* Perder paquetes = perder información = perder audio.

Jitter

* El jitter es la oscilación de la latencia. La variación en el tiempo de entrega del paquete en la red.
* Una red estable, con la misma latencia siempre, es usable.
* Una red con jitter destroza las conversaciones.
* Jitter < 100ms. Una cifra mayor hace imposible la conversación.
* Causas de jitter: redes al límite de su capacidad, ausencia de QoS, traficos con una prioridad superior ocupando el ancho de banda.

Descargando Asterisk

* Descargar al servidor, preferiblemente a /usr/src:
* `wget http://downloads.asterisk.org/pub/telephony/dahdi-linux-complete/dahdi-linux-complete-current.tar.gz`
* `wget http://downloads.asterisk.org/pub/telephony/libpri/libpri-current.tar.gz`
* `wget http://downloads.asterisk.org/pub/telephony/asterisk/asterisk-15-current.tar.gz`

Descomprimir

* `for n in `ls *.tar.gz`; do tar xfvz $n; done`

Compilar

* Por orden: primero dahdi, después libpri y por último Asterisk.
* Prerequisito: `yum install kernel-devel`
* dahdi: `make & make install & make config`
* libpri: `make & make install`
* asterisk:
  * `contrib/scripts/get_mp3_source.sh`
  * `contrib/scripts/install_prereq install`
  * `contrib/scripts/install_prereq install-unpackaged`
  * `./configure --with-pjproject-bundled & make menuselect & make & make install & make samples & make config & make progdocs`

Configurando DAHDI (1)

* La configuración de DAHDI está en `/etc/dahdi`
* Si una tarjeta no aparece en un `lsusb`, vamos por el mal camino...
* `modules.conf` indica los controladores a cargar.
* `system.conf` indica la configuración de la tarjeta
* `dahdi_cfg` aplica la configuración.

Configurando DAHDI (2)

* Ejemplo system.conf analógica:

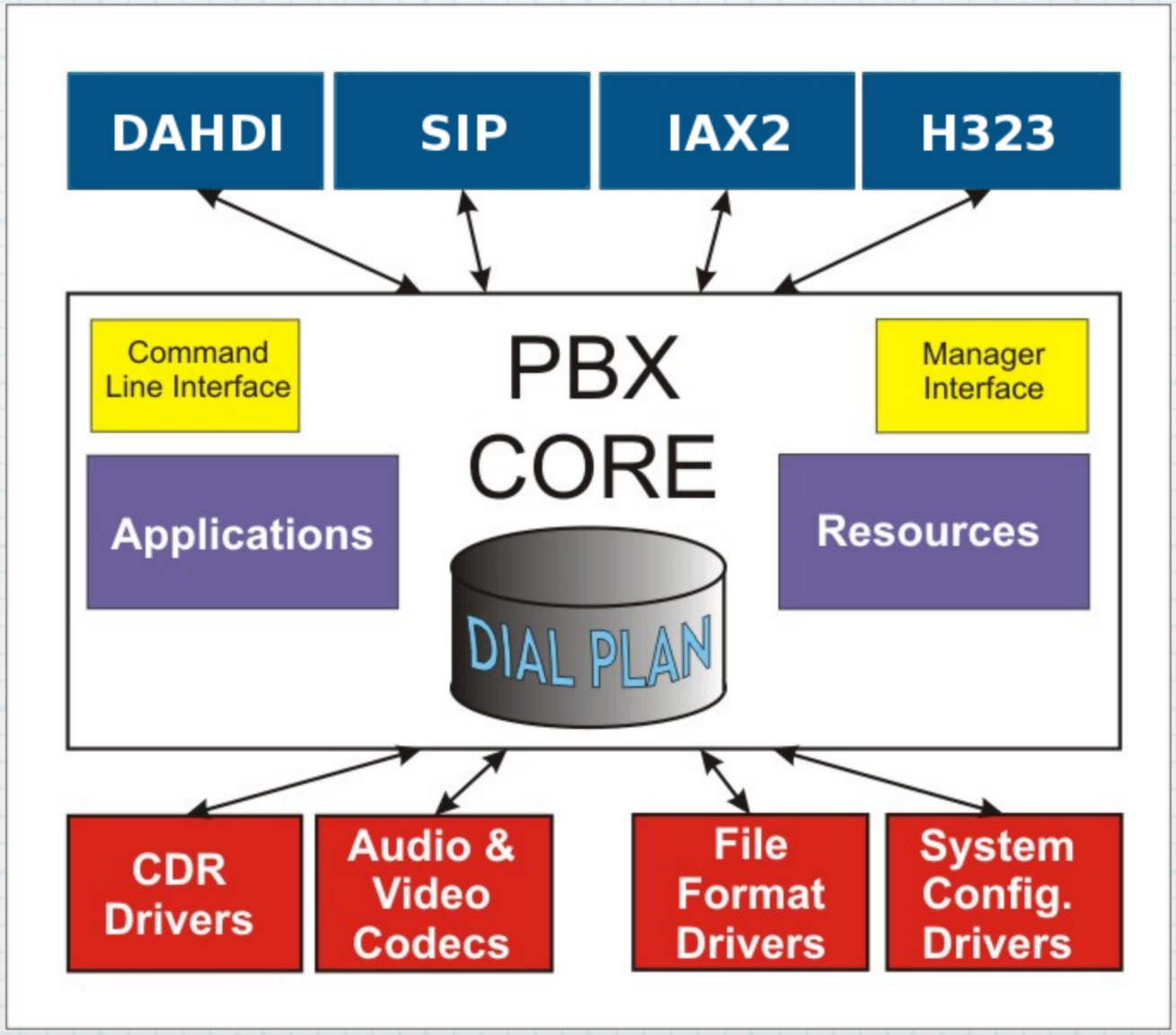
```
loadzone=es

defaultzone=es

fxoks=1-2

fxsks=3-4

echocanceller=mg2,1-4
```

Configurando DAHDI (3)

* Ejemplo system.conf RDSI/Primario:

```
loadzone=es

defaultzone=es

span=1,1,0,ccs,ami

bchan=1,2

dchan=3

span=2,1,0,ccs,hdb3,crc4

bchan=4,5

hardhdlc=6

echocanceller=mg2,1-4
```

Arquitectura de Asterisk



Asterisk Core

* Lee y procesa la configuración del sistema
* Carga los módulos dinámicos
* Ejecuta las Aplicaciones
* Provee la base de tiempos general
* Convierte entre formatos, códecs y protocolos.
* Procesa las peticiones del dialplan
* Crea las instancias de canal

Módulos / Modules

* Asterisk puede configurarse para cargar solo determinados módulos: `/etc/asterisk/modules.conf`.
* Cada módulo provee una funcionalidad al sistema
  * Recursos
  * Aplicaciones
  * Comandos CLI

Canales / Channels

- Los canales controlan la entrada y salida de Asterisk
- Cada canal opera como un traductor entre Asterisk y su propio sistema.
- Cada tipo de canal tiene su propio módulo:
  - `chan_dahdi.so`, `chan_sip.so`, etc.
- Cada tipo de tecnología tiene su propio canal:
  - SIP, IAX, DAHDI, PJSIP,SCCP, H323, etc.

Applications & Functions

* Ofrecen soluciones completas, por ejemplo el buzón de voz reside en el módulo Voicemail (app_voicemail.so)
* Las Aplicaciones permiten hacer cosas.
* Las Funciones permiten trabajar con esas cosas.
* Ej: `exten => 123,1,Dial(SIP/${FILTER(0-9,${EXTEN})})`
* Ej: `exten => 123,1,Set(${CALLERID(num)}=935605424)`
* `core show applications / core show functions`

Resources

* Parecidos a las aplicaciones
* Se cargan de forma estática
* Pueden ser usados por múltiples aplicaciones de forma simultánea
* Ej: Musica en Espera (`res_musiconhold.so`)

Codecs

- Compresor / Descompresor de audio o video.
- Usado por Asterisk para interpretar y codificar las señales de audio entrantes y salientes.
- Cargados dinámicamente
- Cada códec tiene su modulo:
  - `codec_alaw.so`
  - `codec_gsm.so`

Formats

* Los Formats son a los ficheros de audio lo que los códecs son a los streams de audio.

* Se usan para leer y escribir ficheros en distintos formatos.

* Tener `format_g729.so` no implica que se pueda escribir un fichero a g729 haciendo transcoding. Implica solo que se puede leer un stream en ese codec y grabarlo, siempre y cuando venga del mismo codec.

CDR

* Call Detail Report: informe de llamadas
* Se registra toda la actividad, no solo las llamadas de audio.
* Los datos por defecto van a un fichero CSV
* Se puede enviar a una base de datos via ODBC
* Imprescindibles para la contabilidad de consumos y usos de los sistemas de telefonía.

Los sonidos

- Asterisk provee locuciones en Inglés, Francés, Español y Ruso por defecto.
- Las locuciones se pueden seleccionar en distintas codificaciones:
  - Asterisk escogerá siempre la locución que menos carga de conversión tenga, basándose en la extensión del fichero.
  - Como mínimo ha de existir las locuciones en inglés.
  - Si no se encuentra un sonido, se revierte al inglés por defecto.
  - Si no se encuentra el inglés, la llamada se corta.

Arrancando Asterisk

- Asterisk puede arrancarse de varias maneras:
  - usualmente `/etc/init.d/asterisk start`
  - Opcionalmente `asterisk -vvvvvvvvvvc`
  - Si se usa un panel tipo FreePBX con `amportal start`
  - ¿Que hay en la consola 9 (TTY9) ?
  - Entrar a un asterisk arrancado con `asterisk -vvvvvvvvvvr`
  - Ver la diferencia segun `verbose` y `debug`

Asterisk CLI (1)

* Herramienta de configuración, testing y desarrollo.
* Permite hacer debug de distintos protocolos desde la misma cónsola, lanzar llamadas y ejecutar bloques de dialplan.
* Pone a nuestra disposición una poderosa herramienta:
  * `core show applications`
  * `core show functions`

Asterisk CLI (2)

* Recargar el sistema: `reload`
* Rearrancar: `core restart when convenient/gratefully/now`
* Parar: `core stop when convenient/gratefully/now`
* El tabulador es vuestro amigo
* Ver información: `show... (Tab)`
* Recargar una parte: `module reload chan_sip.so`
* Cargar en caliente: `module load codec_g729.so`

SIP General

* El fichero sip.conf empieza con una sección general. Esta sección permite indicar aquellos parámetros genéricos para todas las extensiones, así como aquellos que precisa la aplicación.

```
[general]
language=es
context=invalido
bindport=5060
bindaddr=0.0.0.0
disallow=all
allow=alaw
directmedia=no
nat=no
dtmfmode=rfc2833
transport=udp ; puede ser TCP,UDP o TLS
```

Configurando SIP(1)

* Toda la configuración de SIP reside en el fichero sip.conf
* Podemos configurar terminales IP, conversores externos o proveedores de servicio, y todos se definen en este fichero.
* Ejemplo básico de una definición:

```
[500]
secret=nuw7r5b3t
type=peer
host=dynamic
context=mis-numeros
disallow=all
allow=alaw
allow=g729
qualify=300
```

Configurando SIP(2)

* `type= user/friend/peer`
* **`user`**: busca para una llamada entrante coincidencia usando el usuario En el campo FROM de la petición SIP. Ha de ser el nombre de una Sección en el fichero sip.conf.
* **`peer`**: busca para una llamada entrante coincidencia usando la dirección IP y el puerto.
* **`friend`**: primero miramos el from, y después la IP y el puerto.
* Lo que hablamos aquí es el matching entre paquete y definición. Si es una definición con usuario y contraseña, se llevará a cabo la autentificación. Pero a la hora de localizar quién es el destino, es importante este punto. No influye en las llamadas salientes.

Configurando SIP(3)

* El campo `host` nos indica o bien si es dinámica (`dynamic`), en cuyo caso ha de haber un proceso de registro contra el Asterisk, o si es una IP determinada (`host=10.100.20.20`).
* Cuando `host` contiene una IP, puede trabajarse sin mecanismo de registro. `insecure=port,invite`
* El campo `port` es opcional. Si no se indica es el puerto 5060.
* El campo `transport` nos sirve para indicar si el tipo de paquete es `udp`, `tcp` o `tls`. Si no se indica es `udp`.
* `username` y `secret` son descriptivos por si mismos. Importancia del uso de contraseñas complejas.

Configurando SIP(4)

* `context`: indica en que contexto del fichero `extensions.conf` se parseará una llamada entrante con destino a esa extensión.

* Los contextos son entidades aisladas. Lo que ponga en un contexto no podrá acceder ni ser accesible desde otro distinto salvo que yo lo permita explicitamente.

* `disallow= all` ; deshabilito explícitamente todos los códecs para esta extensión o trunk.

* `allow=nombredelcodec` ; habilito de forma explícita uno o más códecs para esa extensión o trunk. Puedo tener varias líneas aunque se recomienda ser muy conciso: solo aquellas que se precisen usar.

Configurando SIP(5)

* Las plantillas nos permiten reducir el código a escribir:

```
[extension](!)
label=extension
type=peer
host=dynamic
dtmfmode=rfc2833
qualify=no
disallow=all
allow=g729
allow=alaw
directmedia=no
nat=no
context=internas

[101](extension)
username=101
callerid="Eugenia_Echeverria_Rubio" <101>
secret=101hfgd7
```

Extensions.conf (1)

```
[general]
language=es
autofallthrough=no
clearglobalvars=no
priorityjumping=no

[globals]
CONSOLE=Console/dsp
FIJO=DAHDI/g0
MOVIL=DAHDI/g0

[invalido]

exten => s,1,NoOP(Llamada perdida, colgando....)
exten => s,n,Hangup()
```

Extensions.conf (2)

```
[entrantes]

exten => s,1,NoOP(Llamada entrante)
same => n,Dial(SIP/101,30,tTwW)
same => n,Hangup()

[internas]
include = salientes

exten => _5XX,1,NoOp(Llamada entre extensiones)
same => n,Dial(SIP/${EXTEN},60,tTWw)
same => n,Hangup()

[salientes]

exten => _9XXXXXXXX,1,NoOP(Llamada saliente)
same => n,Dial(DAHDI/1/${EXTEN},60,tTwW)
same => n,Hangup()
```

Plan de numeración

- Numeración 1XX colisiona con numeración de emergencia (112)
- ¿De verdad necesitamos el cero para marcar al exterior?
- Pensar que es lo que necesitamos:
  - Numeración integrada 2XXX para móviles XXX para fijos...
  - Numeraciones para “hacer cosas” con Asterisk: *22,*97
  - No usar “#” en las numeraciones
  - No contar justo... hoy sois 90, mañana 200... cambiar la numeración puede ser un trabajo tedioso el día de mañana.

E164 y ENUM

* E164 define los planes de numeración de todos los países. Estamos usando E164 desde el primer momento que llamamos a un teléfono.
* E164 limita los números a 15 dígitos, excluídos los prefijos.
* ENUM no es mas que un servicio de DNS que permite obtener el destino de una llamada via Internet.
* Poco avanzado, motivos políticos lo limitan. España ni siquiera figura en enumdata.org como pais que siquiera prevea el cambio.
* Posibilidad de llamar a pepe@suempresa.com via DNS.

SIP URI

* ¿Porqué no aceptar llamadas a pepe@sudominio.com?
* Es gratis, es simple de implementar y queda guay en las tarjetas de visita...
* Solo necesitamos dos cosas, un servicio SIP escuchando en Internet (para recibir la llamada) y un registro DNS especialmente formateado.

```
[root@localhost ~]# dig SRV _sip._udp.keynockers.com;
<<>> DiG 9.9.2-P2 <<>> SRV _sip._udp.keynockers.com;; QUESTION
SECTION:;_sip._udp.keynockers.com.      IN     SRV;; ANSWER
SECTION:_sip._udp.keynockers.com. 21600 IN     SRV    0 0 5060 sip.keynockers.com.
```

Configurar el teléfono IP

* Podemos usar cualquier terminal IP para este curso, todas precisan de unos datos mínimos de configuración, que suelen ser comunes a todas ellas. Estos son:

* El número de la extensión, el usuario de autentificación a usar (muchas veces el mismo que la extensión) y su contraseña.

* La dirección IP o nombre del servidor de registro, y del servidor Proxy saliente (normalmente el mismo que el de registro), y el puerto en el que escucha.

* Los códecs a utilizar y el modo de paso de DTMF.

Ejemplos: GrandStream

Primeras llamadas

* Observar....
* Que sea posible cursar una llamada entre dos extensiones y que tenga audio.
* Ver en la cónsola de Asterisk (`asterisk -vvvvvvvvr`)como podemos seguir la ejecución del código que hemos escrito. Depurar primeros problemas.
* Usar `sip set debug on` y hacer una llamada, observar el debug de protocolo sip que se nos presenta, para el rápido diagnóstico de problemas.
* `apt-get install sngrep` y ver herramienta y ventajas.

Enlazar centrales (1)

* Establecer un trunk sip, como veremos, es extraordinariamente simple:

Fichero sip.conf

```
[trunk-uno]
type=peer
host=ip del otro server
context=entradas
disallow=all
allow=g722
```

Fichero sip.conf

```
[trunk-dos]
type=peer
host=ip del otro server
context=entradas
disallow=all
allow=g722
```

Ficheros de configuración básicos

asterisk.conf

* Define rutas a directorios, donde está la configuración, donde los módulos, donde los binarios, etc.
* Permite indicar los parámetros en tiempo de ejecución (coloreado de cónsola, prioridad de ejecución, máximo número de llamadas, etc).
* Permite indicar usuario y contraseña con que correrá el proceso de asterisk.
* Es el único fichero que es obligatorio que esté en /etc/asterisk, el resto pueden estar en otras ubicaciones, ya que este fichero es el primero que se lee en el arranque.

modules.conf

* Es el responsable de indicar al proceso core que módulos se van a cargar y en que orden:

```
[modules]

autoload=yes

preload=> res_odbc.so

preload-required => pbx_config.so

noload => pbx_gtkconsole.so

load => cdr_addon_mysql.so
```

features.conf

* definimos algunas caracteristicas adicionales del sistema (transferencias gestionadas por Asterisk, parking de llamadas, captura general, etc).

```
[general]
transferdigittimeout => 3
xfersound = beep

xferfailsound = beeperr
pickupexten = *8

featuredigittimeout = 1000

atxfernoanswertimeout = 15


[featuremap]
blindxfer => #1
disconnect => *0
automon => *1
atxfer => *2
```

confbridge.conf

* sustituye al venerable Meetme. Aplicación de conferencias.
* Para conferencias simples, no es necesario editar el fichero, pero al mismo tiempo es interesante darle un vistazo para ver las posibilidades de configuración que tiene.
* Se invoca con la aplicación Confbridge.

```
exten => 77,1,ConfBridge(${EXTEN})
same => n,Hangup()
```

musiconhold.conf

* controla las “clases” de música en espera disponibles en el sistema

```
[default]
mode=files
directory=/var/lib/asterisk/moh

[native-random]
mode=files
directory=/var/lib/asterisk/moh
random=yes      ; Play the files in a random order

[ulawstream]
mode=custom
application=/usr/bin/streamplayer 192.168.100.52 888
format=ulaw
```

queues.conf (1)

* controla las distintas colas y su comportamiento con los agentes.

```
[general]
persistentmembers = yes
autofill = yes
monitor-type = MixMonitor

[nombredelacola]
musicclass = default
strategy = ringall
timeout = 15
retry = 5
wrapuptime=15
autofill=yes
maxlen = 0
periodic-announce-frequency=60
announce-holdtime = yes
```

queues.conf (2)

```
announce-round-seconds = 10
monitor-format = gsm|wav|wav49
monitor-type = MixMonitor
joinempty = yes
leavewhenempty = yes
ringinuse = no

member => DAHDI/1
member => Agent/1001
member => SIP/101
```

voicemail.conf (1)

* controla el sistema de buzones de voz, qué se envía y a donde.

```
[general]
format=wav49|gsm|wav
serveremail=asterisk
attach=yes
maxmsg=100
maxmessage=180    ; segundos
minmessage=3      ; segundos
maxsilence=10
silencethreshold=128
attachfmt=wav49
saycid=yes
sayduration=no
```

voicemail.conf (2)

```
[default]

1234 => 4242,Example Mailbox,root@localhost

4200 => 9855,Mark Spencer,markster@linux-
support.net,,attach=no|serveremail=myaddy@digium.com|
tz=central|maxmsg=10

4069 => 6522,Matt Brooks,matt@marko.net,,|tz=central|
attach=yes|saycid=yes|dialout=fromvm|callback=fromvm|
review=yes|operator=yes|envelope=yes|sayduration$
```

chan_dahdi.conf

```
[trunkgroups]
[channels]
relaxdtmf=no
echocancel=yes
busydetect=yes
busycount=5
overlapdial=yes
usecallerid=yes
rxgain=9
txgain=5
language=es
signalling=fxs_ks
callerid=asreceived
group=0
context=from-pstn
channel => 1
```

```
signalling=bri_cpe
switchtype=euroisdn
group=1
context=from-pstn
channel=> 1,2

signalling=pri_cpe
switchtype=euroisdn
group=2
context=from-pstn
channel=> 1-15,17-31

signalling=fxo_ks
echocancel=yes
rxgain=-1
txgain=-2
context=extensiones
channel=33-48
```

Profundizando en el dialplan

Sintaxis de las extensiones

```
exten => 500,1,Answer()
same => n,Wait(2)
same => n,Playback(bienvenido)
same => n,Hangup()
```

```
exten => extension, prioridad, aplicación
```

Las prioridades pueden numerarse de forma estricta o usar la letra n (de next) para indicar el valor siguiente. El uso de labels (etiquetas) para disponer de puntos de salto está permitido siempre y cuando se invoque desde la misma extension. Por ejemplo:

```
exten => 500,n(etiqueta),Dial....
```

Funciones y aplicaciones

Nos valdremos de las funciones y aplicaciones para diseñar nuestro dialplan. Són las encargadas de realizar acciones sobre canales, variables, bases de datos, y las que realmente ejecutarán las llamadas a través del canal correspondiente.

Podemos listar las aplicaciones disponibles en nuestro sistema usando el comando "`core show applications`" desde la cónsola de Asterisk, y ver la ayuda detallada de cada una de las aplicaciones con "`core show applications nombredeaplicacion`".

Las funciones nos permiten manipular y trabajar con la información de las aplicaciones, y accederemos del mismo modo a la ayuda contextual usando "`core show functions`" y "`core show function nombredelafuncion`".

Answer / Hangup

* Es buena idea contestar la llamada si tenemos, por ejemplo, que reproducir un audio. De no hacerlo así, la llamada no progresará convenientemente.
* Playback responde la llamada si no estaba previamente contestada. Read no.
* La aplicación para contestar se llama “Answer”.
* La aplicación para colgar se llama “Hangup”.
* Es una buena práctica cerrar todos los flujos de llamadas con Hangup o con otras funciones que eviten comportamientos indeseados.

Llamar: Dial

- Posiblemente Dial es la aplicación más compleja de todo el dialplan. Permite llamar a una tecnología/extension.
- Ver “core show application Dial”

```
exten=> 501,1,Dial(SIP/501)
same => n,Hangup()

exten => 502,1,Dial(DAHDI/1)
same => n,Hangup()

exten => 503,1,Dial(local/pepe@contexto/n)
same => n,Hangup()
```

Extensiones especiales

a	Al pulsar * desde el buzón de voz
h	Hangup
i	Inválido
o	Operador – al pulsar 0 en el buzón
s	Start - Inicio
t	Timeout alcanzado
T	Timeout absoluto - absolutetimeout()
fax	Detección de fax de dahdi
hint	Ver sección hint mas adelante

Ejemplo:

```
exten => h,1,ResetCDR()
exten => h,n,Hangup()

exten => fax,1,Dial(IAX2/999,60,tT)
```

Patterns

Los patterns es la primera herramienta a explotar en nuestro dialplan, a fin de evitar la repeticion de codigo:

X – Cualquier dígito entre 0 y 9
Z – Cualquier dígito entre 1 y 9
N – Cualquier dígito entre 2 y 9

. - Un digito + “n” caracteres
! - Cero o un digito + “n” caracteres

[01] – Dígitos 0 o 1 (Ej. 9[12]0 puede ser 910 o 920).
[3-7] – Dígitos entre 3 y 7 (3,4,5,6,7)
[237-9] – Dígitos 2 / 3 / 7 / 8 / 9

Siempre se resuelve la mejor coincidencia posible.

¡¡¡¡¡¡ OJO A LOS PATTERNS !!!!!!!
Patterns mal formados permiten inyección de cadenas de marcado.

Includes

* Un contexto puede ser incluído dentro de otro usando includes:
  * `include => nombre del contexto`
* Un fichero puede ser incluido dentro de otro usando includes:
  * `#include fichero.loquesea`
* Un fichero incluído no hace que sus contextos lo sean también.
* Ojo a la herencia de includes.

Saltos

* Es posible hacer saltos en el dialplan:
  * `Goto(contexto,extension,prioridad)`
  * `GotoIF(condicion?salto si cierta:salto si falsa)`
  * `Gosub(contexto,extensión,prioridad)  / Return`
  * `GosubIf(condicion?salto si cierta:salto si falsa)`
  * `GotoIFTime(condicion?salto si cierta)`
  * `While(condicion) / EndWhile()`
  * `ExecIf / ExecIfTime`

Variables y corte

- ${nombre-de-variable} ; variables definibles por el usuario. Pueden ser mayúsculas o minúsculas.
- ${EXTEN} ; variable definida por el sistema. Siempre en MAYUSCULAS. EXTEN contiene, para cada hilo de ejecución, el valor de la extensión siendo procesada.
- Corte de cadenas : ${nombre:desplazamiento:longitud}
  - Si el desplazamiento es negativo, comienza a contar desde la derecha.
  - Si longitud se omite o es negativa, se devuelve el resto de la cadena.

${VARIABLE:3} - Elimina los tres primeros dígitos
${VARIABLE:-3} - Elimina todo menos los últimos tres dígitos
${VARIABLE:1:4} - Elimina 1 carácter del principio y muestra los 4 dígitos siguientes únicamente.
${VARIABLE:-4:3} - Elimina todo menos los cuatro últimos dígitos y muestra los tres primeros.

Manipulando variables (1)

Las expresiones son combinaciones de variables, operadores y valores que se hacen interactuar para producir un resultado.

Por ejemplo, para sumar a la variable NUMERO un numero, haríamos:

$[${NUMERO}+1]

Para aplicar por ejemplo una suma dentro de una ejecución de dialplan haríamos:

```
exten => 123,1,Set(VARIABLE_EJEMPLO=1)
exten => 123,2,Set(NUEVA_VARIABLE=$[${VARIABLE_EJEMPLO}+1])
exten => 123,3,SayNumber(${NUEVA_VARIABLE})
```

Manipulando variables (2)

Los Operadores Booleanos nos permitirán manipular las variables.

expre1|expre2	OR: evalúa ambas, devuelve 1 si una de las dos es cierta o 0 si ambas.
expre1&expre2	AND: devuelve 1 si ambas son ciertas o 0 si no.
expre1=expre2	EQ: devuelve 1 si ambas son iguales o 0 si no
expre1>expre2	GT: devuelve 1 si expre1 es mayor que expre2
expre1<expre2	LT: devuelve 1 si expre1 es menor que expre2
expre1>=expre2	devuelve 1 si expre1 es mayor o igual que expre2
expre1<=expre2	devuelve 1 si expre1 es menor o igual que expre2
expre1!=expre2	devuelve 1 si expre1 es distinta de expre2

Manipulando variables (3)

Operadores matemáticos

expre1+expre2	Suma de expresiones
expre1-expre2	Resta de expresiones
expre1*expre2	Multiplicación de expresiones
expre1/expre2	División de expresiones
expre1%expre2	Resto de una operación de división

Hints

Asterisk nos provee de herramientas para la monitorización del estado de extensiones, mediante el parámetro especial HINT.

En el SIP.CONF debe establecerse `notifyringing=yes` y `limitonpeers=yes`, y las extensiones deben tener el parámetro `call-limit` establecido a un valor numérico.

```
exten => 101,hint,SIP/100
```

Observar que no tiene prioridad. Se puede monitorizar el estado con un show hints, que nos mostrará cuantas extensiones estan usando esa monitorización.

```
localhost*CLI> core show hints     -= Registered Asterisk Dial Plan Hints =-
_10X@internas  : SIP/${EXTEN}    State:Unavailable     Watchers  0
```

Macros (1)

- Las macros se consideran deprecated y van a dejar de ser usadas en breve. De todos modos hay que explicarlas, pero se recomienda sustituirlas por `Gosub`.
- Una macro no es mas que un contexto que comienza por la palabra macro, como `[macro-pepe]`
- El primer punto de ejecución dentro de una macro será la extensión especial “`s`” obligatoriamente.
- Una macro se ejecuta en un espacio de memoria separado, se le deben de pasar los argumentos de forma posicional.
- Aplicación macro: `macro(pepe,arg1,arg2,arg3....)`

Macros (2)

```
[macro-pepe]

;recibiremos dos argumentos que se recuperan como ${ARG1}...

exten => s,1,NoOP(-- Entrada a la macro pepe--)
same => n,NoOP(Hemos recibido el primer argumento: ${ARG1})
same => n,NoOp(y como segundo argumento: ${ARG2})
same => n,Goto(manolo,1)

exten => manolo,1,Dial(SIP/${ARG1},${ARG2},tw)
same => n,Hangup()
```

Aplicaciones Avanzadas

Calendarios (1)

Crear una cuenta en Google para poder acceder al servicio de Google Calendar, que usaremos como base de las pruebas con calendarios y Su integración con Asterisk.

Acceder a las propiedades del calendario y buscar este valor:

Configuración de Calendar → Calendarios → seleccionar calendario.

En la sección “Dirección Privada” hacer click sobre el icono de ICAL Y tomar nota del valor que devuelve, que usaremos al configurar Las propiedades del acceso al calendario.

Calendarios (2)

Configurar el acceso a un calendario en red (usaremos en el Aula un acceso a Google Calendar para hacer la demostración) Y en base al estado del calendario tomaremos decisiones.

`/etc/asterisk/calendar.conf`

```
[calendar_juancarlos]
type = ical
url = http://www.google.com/calendar/ical/.........
user = usuarioi@gmail.com
secret = contraseñasuperguay
refresh = 2
timeframe = 600
```

Calendarios (3)

Comprobamos que el calendario se ve desde asterisk:

```
localhost*CLI> calendar show calendars
Calendar                Type         Status
--------                ----         ------
calendar_juancarlos     ical         busy
```

Definir en /etc/asterisk/extensionsconf un codigo similar a este:

```
exten => 102,1,GotoIf(${CALENDAR_BUSY(calendar_juancarlos)}?cerrado)
exten => 102,n,Dial(SIP/102,30,tw)
exten => 102,n,Hangup()

exten => 102,n(cerrado),NoOP(evento del calendario de cerrado)
exten => 102,n,Hangup()
```

Calendarios... (y 4)

Funciones que nos permiten trabajar con los calendarios:

```
${CALENDAR_BUSY(calendario)} ← para recuperar estado ocupado o disponible
${CALENDAR_QUERY(calendario,inicio,fin} ← Consulta estado, RETORNA ID
${CALENDAR_QUERY_RESULT(id,campo} ← procesa el query anterior
```

CALENDAR_QUERY_RESULT remite recuperar:

```
Summary ← sumario del evento
Description ← descripcion completa del evento
Organizer ← quien organiza la cita
Location ← ubicación de la cita
Calendar ← nombre del calendario implicado
Uid ← identificador único del evento
Start ← inicio del evento en formato EPOCH (Unix Time)
End ← fin del evento en formato EPOCH
Busystate ← estado (0) libre (1) tentative (2) ocupado
Attendeed ← lista separada por comas de los participantes en el evento

${CALENDAR_WRITE(calendario,campo)} ← crea una cita. Lista de campos.
```

CCSS

Call Completion Supplementary Services (CCSS)

El CCSS permite disparar un evento de forma automática al final de Una llamada. Por ejemplo es util para crear una retrollamada como
Existen en otros sistemas.

/etc/asterisk/sip.conf

```
[108]
...
call-limit=1
cc_agent_policy=generic
cc_monitor_policy=generic
```

/etc/asterisk/extensions.conf

```
exten => _10X,1,Dial(SIP/${EXTEN},20,tw)
exten => _10X,n,Hangup()

exten => *22,1,CallCompletionRequest()
exten => *22,n,Hangup()
exten => *23,1,CallCompletionCancel()
exten => *23,n,Hangup()
```

Base de Datos interna (1)

* Asterisk incorpora un motor SQLite para gestionar su base de datos interna.

```
localhost*CLI> database show/SIP/Registry/102                     :
10.100.20.19:5060:300:102:sip:102@10.100.20.19:5060/SIP/Registry/103                  :
10.100.20.15:5060:180:103:sip:103@10.100.20.15:5060/SIP/Registry/200                  :
10.100.20.11:5062:3600:200:sip:200@10.100.20.11:5062/SIP/Registry/201                  :
10.100.20.12:5062:3600:201:sip:201@10.100.20.12:5062/dundi/secret                  :
UIQcJ+aZ4iKBMI0dpDw+ZQ==;tEqLEZlzsxVAQdAxuvXtKQ==/dundi/secretexpiry                :
13674238846 results found.
```

* Esta base de datos puede ser usada para lo que precisemos, siempre teniendo en cuenta que no es una base de datos para poner cientos de miles de registros, o penalizaremos sensiblemente el rendimiento de asterisk.

Base de Datos interna (2)

```
exten => s,n,GotoIf($[${DB_EXISTS(FOFICINA/FUERA)}]?7:200)

; act. fuera de la oficina
exten => *21,1,Answer()
exten => *21,n,noOP(${DB_DELETE(FOFICINA/FUERA)})
exten => *21,n,Playback(activated)
exten => *21,n,Hangup()

; desactivacion fuera de la oficina
exten => *22,1,Answer()
exten => *22,n,Set(${DB(FOFICINA/FUERA)}=YES)
exten => *22,n,Playback(de-activated)
exten => *22,n,Hangup()
```

ODBC (1)

* Asterisk puede, via ODBC, hablar con distintos motores de base de datos, a fín de recuperar información y usarla en su dialplan.
* Para poder usarlo, primero precisamos tener la base de ODBC instalada en el sistema operativo.
* `yum install unixODBC unixODBC-devel libtool-ltdl libtool-ltdl-devel mysql-connector-ODBC`
* Si es preciso, recompilar asterisk para que dispongamos de los módulos de acceso a ODBC. Mirar el make menuconfig para ello.

ODBC (2)

* Primero habilitaremos la parte de Linux, para ello editaremos el fichero /etc/odbcinst.ini.

```
[MySQL]
Description= ODBC para MySQL
Driver = /usr/lib/libmyodbc3.so
Setup = /usr/lib/libodbcmyS.so
Fileusage = 1
```

* Verificaremos con "`odbcinst -q -d`" que ve los cambios que hemos hecho, mostrándonos la etiqueta "`[MySQL]`" que hemos escrito.

ODBC (3)

* A continuación editaremos /etc/odbc.ini que es donde proporcionaremos un identificador que usaremos después desde Asterisk para acceder al motor ODBC.

```
[conector-asterisk]
Description = Conexión MySQL con Asterisk
Driver = MySQL   ; el nombre de la seccion que creamos
Database = asterisk
Server = localhost
UserName = asterisk
Password = klwuebt3487r6
Port = 3306
Socket = /var/lib/mysql/mysql.sock
```

ODBC (4)

* Completados los pasos de configuración de ODBC, pasaremos, después de haber recompilado si fuese preciso, a configurar Asterisk para poder acceder a los datos. Esta configuración se realizará en dos pasos, uno para el acceso a los datos, en el fichero `res_odbc.conf` y otro con las querys que precisemos en el fichero `func_odbc.conf`.

```
[asterisk]
enabled=> yes
dsn => conector-asterisk
username => asterisk
password=> elquesea
pooling=> no
pre-connect => yes
```

* Verificar desde asterisk tras un reload con “odbc show”

ODBC (5)

* Por último, editamos func_odbc.conf para definir la consulta.

```
[CONSULTA]
dsn=asterisk
readsql=SELECT * FROM PEPE WHERE NUMBER='${ARG1}'
```

* Y desde el dialplan usaremos la funcion ODBC_nombre para hacer la consulta.

```
exten => 22,1,GotoIf($[${ODBC_CONSULTA(${CALLERID(num)}=935605424]?llamar)
```

* Consultar con “core show applications” las distintas que contienen ODBC para ver sus opciones.

AMI (1)

El Asterisk Manager Interface permite controlar y monitorizar el estado de un servidor Asterisk. Es un servicio accesible mediante Red (escucha por defecto en el puerto 5038 TCP) que nos permite ejecutar órdenes y recibir información de eventos en ejecución.

Para activarlo, lo primero que hay que hacer es crear un fichero llamado `/etc/asterisk/manager.conf` que contendrá:

```
[general]
enabled=yes
port=5038
bindaddr=0.0.0.0

[usuario]
secret=password
read=system,call,log,verbose,command,agent,user
write=system,call,log,verbose,command,agent,user
```

AMI (2)

Adicionalmente podremos limitar la entrada al manager por IP, dentro de la sección del usuario:

```
deny= 0.0.0.0/0.0.0.0
permit= 10.13.13.0/255.255.255.0
```

Una vez rearranquemos Asterisk el interface del Manager estará disponible para conectar.

```
[root@centralita es]# telnet 127.0.0.1 5038
Trying 127.0.0.1...
Connected to localhost.localdomain (127.0.0.1).
Escape character is '^]'.
Asterisk Call Manager/1.0

Action: Login
Username: usuario
Secret: password

Response: Success
Message: Authentication accepted
```

AMI (3)

Para ver una lista de comandos posibles en el manager:

```
manager show commands
```

Normalmente operaremos contra el Manager mediante interfaces que nos permitan cierta comodidad. Algunos de ellos son:

- Flash Operator Panel http://www.asternic.org
- ASTTapi http://sourceforge.net/projects/asttapi/
- Activa TSP http://activa.sourceforge.net/

Los dos últimos son los que nos permitirán, por ejemplo, originar el marcado desde MicroSoft Outlook. Para ello lo que proporcionan estas aplicaciones es, por un lado, una conectividad TAPI para Windows, y por el otro una conexión Telnet a la máquina Asterisk.

BEST PRACTICES (1)

Exploración sistemática de puertos SIP via Internet, ataques con pérdidas millonarias de dinero en clientes. La seguridad HAY que tomársela en serio.

Buenas prácticas:

** Filtrar datos: evitar ataques de rellamada

** Correcto uso de los números de Dispositivo: O porqué no usar números para identificar a los dispositivos.

** Passwords Seguros: una política de contraseñas es imprescindible.

** Reducir los errores de parseado: Usar Goto() o Same.

** Evitar la exposición a Internet: uso de puertos distintos, VPN, etc.

BEST PRACTICES (2)

Filtrado de datos:

```
[incoming]
exten => _X.,1,Verbose(2,Incoming call to ${EXTEN})
exten => _X.,n,Dial(SIP/${EXTEN})
exten => _X.,n,Hangup()
```

Es posible inyectar cadenas: 500&SIP/itsp/14165551212

```
exten => _X.,n,Dial(SIP/500&SIP/itsp/14165551212)
```

- No usar el punto “.”. Patterns estrictos: _XXX
-Uso de FILTER:

```
[incoming]
exten => _X.,1,Verbose(2,Incoming call to ${EXTEN})
exten => _X.,n,Dial(SIP/${FILTER(0-9,${EXTEN})})
exten => _X.,n,Hangup()
```

BEST PRACTICES (3)

Correcto nombrado de dispositivos:

```
[1000]
type=friend
context=international_dialing
Secret=1000

[0004f2040001]
type=friend
context=international_dialing
secret=aE3%B8*$jk^G

Exten => 101,1,Dial(SIP/0004f2040001,30,tw)
```

No es muy práctico, complicado para grandes entornos salvo que sean realtime o similares. Eso si, ayuda sobremanera a mejorar la seguridad.

BEST PRACTICES (4)

Evitar la exposición de los sistemas Asterisk a Internet

```
[general]
language=es
srvlookup=yes
context=invalido
bindport=5060   ;← cambiar a 5089, 6063... 10125...
bindaddr=0.0.0.0
```

La exploración de vulnerabilidades se hace de forma automática, todo lo que disminuya la visibilidad de nuestra central en Internet es muy recomendable. Idealmente no usar accesos públicos, limitar con VPN la visibilidad. Reglas estrictas de cortafuegos pueden ser útiles, pero no debemos de confiar a pies juntillas en ellas.

SRTP: Audio encriptado

Definir en `/etc/asterisk/sip.conf` la directiva:

```
encryption=yes
```

Puede definirse en una plantilla o individualmente en las extensiones.
Habilitar en las terminales el soporte de RTP encriptado, por ejemplo en Snom:

TLS : codificando (1)

Definir en `/etc/asterisk/sip.conf` la directiva:

```
tlsenable=yes
tlsbindaddr=0.0.0.0
tlscertfile=ruta_al_certificado  (key and certificate)
tlscafile=ruta_al_fichero_ca (si es autofirmado)
tlscapath=ruta_al_directorio_de_ca
tlsdontverifyserver=yes  (por defecto no. Solo para
autofirmados)
tlscipher= valor
```

TLS : codificando (y 2)

Ejemplo TLS entre dos servidores:

`servidor A`

```
[general]
tlsenable=yes
tlscertfile=/etc/asterisk/asterisk.pem
tlscafile=/etc/ssl/ca.pem
register => tls://100:test@192.168.0.100:5061

[101]
type=friend
context=internal
host=192.168.0.100
secret=test
dtmfmode=rfc2833
disallow=all
allow=ulaw
transport=tls
port=5061
```

`servidor B`

```
[general]
tlsenable=yes
tlscertfile=/etc/asterisk/asterisk.pem
tlscafile=/etc/ssl/ca.pem

[101]
type=friend
context=internal
host=dynamic
secret=test
dtmfmode=rfc2833
disallow=all
allow=ulaw
transport=tls
port=5061
```

SIP y los NAT (1)

* El cambio en las cabeceras IP al pasar a través de un NAT, no modifica el payload SIP que contiene el paquete.
* Podemos indicar, si nuestra IP pública es fija, su valor con el parámetro “`externaddr=88.77.99.11`”, y deberemos de complementarlo con “`localnet=192.168.1.0/255.255.255.0`” donde indicaremos nuestra red privada, para que Asterisk sepa como distinguirlas. Por último, activaremos “`nat=yes`” para que haga NAT.
* Esto funciona razonablemente, pero... ¿ que ocurre si nuestra ip es dinámica ?

SIP y los NAT (2)

* En aquellos casos que nuestra IP es dinámica podemos usar un nombre de máquina tipo Dyndns y apoyarnos en "`externhost=miasterisk.dyndns.org`". Para hacer mas granular el refresco de ip, usaremos "`externrefresh=100`" con un valor en segundos.

* Si no tenemos un nombre de máquina, podemos usar STUN para ayudarnos en proceso de averiguar cual es nuestra IP Pública.

* Para ello configuraremos el fichero `res_stun_monitor.conf`.

SIP y los NAT (3)

* Con esta configuración cada “n” segundos se verificará nuestra dirección IP Pública, a través del servicio STUN indicado. Esto refrescará la información que maneja `chan_sip.so`.

```
[general]
stunaddr=stun.xten.com
stunrefresh=30
```

WebRTC

* WebRTC es un modo que permite que los desarrolladores puedan escribir aplicaciones Javascript para comunicarse con Asterisk. Se puede simplificar diciendo que de este modo se dispondrá de un softphone simplemente cargando una página web.
* Actualmente solo está soportado en las versiones modernas de Chrome.
* Es un requisito para usarlo tener el soporte de SRTP. Si no está compilado, hay que hacer un paso atrás y compilarlo.
* Se ha de tener compilado también `res_http_websocket`
* Ejemplo librerías en www.jssip.net.

Asterisk y Google (1)

* Vamos a ver como usar los servicios de Google, en concreto mensajería instantánea y llamadas por Gtalk, a través de Asterisk.
* Precisamos tener compilado el soporte para chan_motif y para res_xmpp.
* Configuraremos en primer lugar nuestra cuenta de Google en el fichero xmpp.conf

Asterisk y Google (2)

```
[general]
autoregister=yes
[asterisk]
type=client
serverhost=talk.google.com
username=asterisk.jcs@gmail.com
secret=msydf6i2fsdf
port=5222
usetls=yes
usesasl=yes
buddy=sus.1972.tp@gmail.com
buddy=eb3dgz@gmail.com
status=available
statusmessage="Asterisk en casita"
timeout=100
```

Asterisk y Google (3)

* Tras un reload podremos ver el estado de la conexión:

```
localhost*CLI> xmpp show connectionsJabber Users and their
status:       [asterisk] asterisk.jcs@gmail.com     - Connected
```

* Y tendremos disponibles las herramientas que empiezan con Jabber para poder enviar por ejemplo mensajes:

```
same => n,JabberSend(asterisk,eb3dgz@gmail.com,Llamada recibida del ${CALLERID(num)})
```

* Una vez completado este paso, iremos a la parte del audio. Para ello editaremos chan_motif.conf.

Asterisk y Google (4)

```
[default]
(!)disallow=allallow=ulawallow=h264context=entran
tesconnection=asterisk;configuracion para
Jingle[jingle-endpoint](default)transport=ice-udp

allow=g722
allow=alawallow=ulawconnection=asterisk
; configuracion para Google Talk[gtalk-endpoint]
(default)transport=google
connection=asterisk; configuracion para  Google
Voice[gvoice](default)transport=google-v1
connection=asterisk
```

El futuro: de chan_sip a chan_pjsip

Migrando a PJSIP (1)

* Aunque pueden coexistir, para este ejemplo deshabilitaremos el uso de `chan_sip`. Para ello, en el fichero modules.conf le haremos un `noload => chan_sip.so`.

* Si deseamos que coexistan ambas pilas de protocolo SIP, entonces tenemos que marcar que cada una de ellas use un puerto distinto. De no ser asi, colisionarían.

* Editaremos en el fichero `pjsip.conf` (partiendo de un fichero en blanco) las opciones siguientes, para por ejemplo definir una extensión en nuestra centralita.

Migrando a PJSIP (2)

Primero definiremos el transporte, que puede tener NAT o no...

```
[transporte-udp]
type=transport
protocol=udp
bind=0.0.0.0
local_net=10.13.13.0/255.255.255.0
external_media_address=<mi ip publica>
external_signalling_address=<mi ip publica y puerto>
```

A continuación definiremos el endpoint (extension o trunk) que usará este transporte que hemos definido:

Migrando a PJSIP (3)

Definimos el endpoint:

```
[500]
type=endpoint
context=internas
disallow=all
allow=g729
transport=transporte-udp
direct_media=no
force_rport=yes
rtp_simmetric=yes
mailboxes=500@default
auth=500
aors=500
```

Tendremos que definir ahora a donde apunta AUTH y AORS...

Migrando a PJSIP (4)

Definimos el auth (datos de autentificacion):

```
[500]
type=auth
auth_type=userpass
password=urybwfk3748736ftekr
username=500
```

Y definimos el Address of record (AOR):

```
[500]
type=aor
max_contacts=2
qualify_frequency=30
```

Y con esto tenemos definida una extension... ¿Como la llamaremos?

Migrando a PJSIP (5)

En el fichero extensions.conf usaremos la orden Dial para llamar a la extension definida:

exten => _5XX,1,Dial(PJSIP/${EXTEN},30,tw)

Y podremos ver desde la consola el estado del registro usando:

```
pjsip show endpoints
```

Explicar que es el forking y como se usa aprovechando el parametro max_contacts definido en el registro AOR.

Migrando a PJSIP (6)

¿ Y si necesitamos deshabilitar res_pjsip en vez de chan_sip?

En `/etc/asterisk/modules.conf` pondremos:

```
noload=> res_pjsip.so
noload=> res_pjsip_pubsub.so
noload=> res_pjsip_session.so
noload=> chan_pjsip.so
noload=> res_pjsip_exten_state.so
noload=> res_pjsip_log_forwarder.so
```

Esto dejará solo a chan_SIP como proveedor de la pila de protocolo SIP.
De cara a futuro PJSIP es quien quedará. Preparar migraciones con tiempo antes de la desaparición de PJSIP.

Temas Legales de la Grabación de llamadas (y de la escucha)

Grabaciónes... (1)

La legislación actual en España determina que:

- Es ilegal grabar o escuchar una conversación en la que uno no es parte, y los demás desconocen la grabación o escucha.
- Es legal grabar una conversación en la que uno es parte, pero es ilegal facilitar esa grabación a un tercero que no haya sido autorizado por los que participaron en la conversación.

El Tribunal Constitucional lo deja bien claro en su sentencia de 29 de noviembre de 1984, STC 11/1984, cuando establece, entre otras consideraciones que:
"Quien graba una conversación de otros atenta, independientemente de toda otra consideración, al derecho reconocido en el art. 18.3 CE; por el contrario, quien graba una conversación con otro no incurre, por este solo hecho, en conducta contraria al precepto constitucional citado."

Grabaciones... (2)

Si uno no es parte en la conversación estará vulnerando un derecho fundamental, reconocido en el artículo 18.3 de la Constitución, pero quien graba las palabras que un tercero le dirige no está realizando por ese sólo hecho ilícito alguno. Cuestión diferente sería si esa conversación se divulga y la intromisión que pueda suponer en la esfera de la persona cuyas palabras se han recogido.

Para las grabaciones ajenas, el Código Penal castiga con prisión de uno a cuatro años y multa de doce a veinticuatro meses. El artículo 197 castiga a quien para descubrir los secretos o vulnerar la intimidad de otro, sin su consentimiento, utilice artificios técnicos de escucha, transmisión, grabación o reproducción del sonido o de la imagen, o de cualquier otra señal de comunicación.

Grabaciones... (3)

Se plantearon a la Agencia Española de Protección de Datos diversas cuestiones relacionadas con la recopilación por parte de una empresa de diversos registros de voz, con la finalidad de elaborar un programa de "software" de reconocimiento de voz. La recopilación tendría lugar mediante la realización de llamadas telefónicas efectuadas desde un Estado miembro de la Unión Europea.

En relación con esta cuestión, se considera que siempre que quien haya de realizar el tratamiento tenga conocimiento directo o indirecto de quién es la persona cuya voz está siendo objeto de grabación, así como de su número de teléfono, la grabación efectuada tendrá la naturaleza de dato de carácter personal y el tratamiento efectuado estará sometido a la normativa de protección de datos, al incorporarse al mismo los datos identificativos del sujeto (nombre y apellidos), su número de teléfono y su voz, conforme a lo dispuesto en el artículo 3.a) de la LOPD y el artículo 1.4 del Real Decreto 1332/1994, de 20 de junio, que indica que dichos datos podrán proceder de información acústica.

Troubleshooting (1)

* ECOS en línea telefónica:

NEAR-END: se origina en la terminal telefónica o en la línea analógica en el lado cliente.

FAR-END: se origina en el extremo remoto de la conversación, o por reflexión en un enlace hibrido.

Se puede combatir el NEAR-END con una mejor calidad de equipamiento, o usando un cancelador de eco por hardware/software de buena calidad. El FAR-END solo puede ser eliminado por un cancelador dedicado o por el operador de telefonía.

Troubleshooting (2)

* Mal ajuste de la impedancia de linea (solo analógicas)

Se puede ajustar la impedancia de la linea (en la tarjeta) con el comando fxotune. Se deberan aplicar los cambios siempre antes de cargar Asterisk.

```
fxotune -i 5 -vv
```

Los datos se escriben en `/etc/fxotune.conf` y se cargan con:

```
fxotune -s
```

La orden debe insertarse en el script de inicio de Asterisk.

Troubleshooting (3)

* Mal ajuste de los niveles de audio

Como hemos visto a lo largo del curso, los niveles de tx y rx se ajustan en `/etc/asterisk/chan_dahdi.conf`.

Hay una herramienta (`dahdi_monitor`) que permite ver los niveles de audio, con una indicación numérica a fin de calibrar.

```
dahdi_monitor <numero de canal dahdi> -vv
```

Los valores numericos no deben sobrepasar nunca el 14000.

Troubleshooting (3)

* Mal ajuste del cancelador de eco

En `/etc/asterisk/chan_dahdi.conf` se define para cada grupo de canales los parametros de cancelación de eco:

```
; valores validos 256(32ms),512(64ms),1024(128ms)

echocancel=yes

echotraining=yes;<- siempre NO con Canceladores prop.

echocancelwhenbridged=no
```

Si modificando estos ajustes no es posible eliminar el eco, habría que optar por otros cancelador de eco opcionales.

Troubleshooting (4)

* Audio entrecortado o aparición de ruidos en la línea

Si la línea es analógica, desconfiar primero de esta, pero probar con `dahdi_test` el rendimiento de la tarjeta. Interrupciones compartidas pueden causar clics en el audio o ruidos, si la compartición es con un dispositivo de gran carga (disco duro, tarjeta de red, etc).

Si la línea es digital, buscar errores CRC o HDLC en los logs. Normalmente es problema de una mala linea, mal cable o pérdida de la señal de sincronía RDSI. Ayuda el tener mas de una fuente de sincronía en el fichero `/etc/dahdi/system.conf`.

Dimensionado Servidores (1)

* La gran duda a la hora de instalar una centralita usando Asterisk es: ¿ Como de grande ha de ser el servidor ?

Algunas directrices generales:

- El principal factor limitante es el transcoding (conversión entre distintos códecs). Es importante minimizar la conversión a fin de reducir la carga de procesador.

- Generalmente se acepta que se requieren 40 Mhz de procesador por canal concurrente de voz si hay transcoding (3 Ghz = 75 conversaciones).

- Digium recomienda 2 procesadores a 2.8 Ghz y 1 Gb de RAM para 120 canales concurrentes con transcoding G729 – Alaw.

Dimensionamiento Servidores (2)

* Core show translation:

```
*CLI> core show translation
Translation times between formats (in microseconds) for one second of data
 Source Format (Rows) Destination Format (Columns)

        gsm  ulaw  alaw  g726 adpcm  slin lpc10  g729 speex speex16  ilbc g726aal2  g722 slin16 testlaw speex32 slin12 slin24 slin32 slin44 slin48 slin96 slin192
gsm       - 15000 15000 15000 15000  9000 15000 15000 15000    23000 15000    15000 17250  17000   15000   23000  17000  17000  17000  17000  17000  17000   17000
ulaw  15000     -  9150 15000 15000  9000 15000 15000 15000    23000 15000    15000 17250  17000   15000   23000  17000  17000  17000  17000  17000  17000   17000
alaw  15000  9150     - 15000 15000  9000 15000 15000 15000    23000 15000    15000 17250  17000   15000   23000  17000  17000  17000  17000  17000  17000   17000
g726  15000 15000 15000     - 15000  9000 15000 15000 15000    23000 15000    15000 17250  17000   15000   23000  17000  17000  17000  17000  17000  17000   17000
adpcm 15000 15000 15000 15000     -  9000 15000 15000 15000    23000 15000    15000 17250  17000   15000   23000  17000  17000  17000  17000  17000  17000   17000
slin   6000  6000  6000  6000  6000     -  6000  6000  6000    14000  6000     6000  8250   8000    6000   14000   8000   8000   8000   8000   8000   8000    8000
lpc10 15000 15000 15000 15000 15000  9000     - 15000 15000    23000 15000    15000 17250  17000   15000   23000  17000  17000  17000  17000  17000  17000   17000
g729  15000 15000 15000 15000 15000  9000 15000     - 15000    23000 15000    15000 17250  17000   15000   23000  17000  17000  17000  17000  17000  17000   17000
speex 15000 15000 15000 15000 15000  9000 15000 15000     -    23000 15000    15000 17250  17000   15000   23000  17000  17000  17000  17000  17000  17000   17000
speex16 23500 23500 23500 23500 23500 17500 23500 23500 23500        - 23500    23500 15000   9000   23500   23000  17500  17000  17000  17000  17000  17000   17000
ilbc  15000 15000 15000 15000 15000  9000 15000 15000 15000    23000     -    15000 17250  17000   15000   23000  17000  17000  17000  17000  17000  17000   17000
g726aal2 15000 15000 15000 15000 15000  9000 15000 15000 15000    23000 15000        - 17250  17000   15000   23000  17000  17000  17000  17000  17000  17000   17000
g722  15600 15600 15600 15600 15600  9600 15600 15600 15600    15000 15600    15600     -   9000   15600   23000  17500  17000  17000  17000  17000  17000   17000
slin16 14500 14500 14500 14500 14500  8500 14500 14500 14500     6000 14500    14500  6000      -   14500   14000   8500   8000   8000   8000   8000   8000    8000
testlaw 15000 15000 15000 15000 15000  9000 15000 15000 15000    23000 15000    15000 17250  17000       -   23000  17000  17000  17000  17000  17000  17000   17000
speex32 23500 23500 23500 23500 23500 17500 23500 23500 23500    23500 23500    23500 23500  17500   23500       -  17500  17500   9000  17000  17000  17000   17000
slin12 14500 14500 14500 14500 14500  8500 14500 14500 14500    14000 14500    14500 14000   8000   14500   14000      -   8000   8000   8000   8000   8000    8000
slin24 14500 14500 14500 14500 14500  8500 14500 14500 14500    14500 14500    14500 14500   8500   14500   14000   8500      -   8000   8000   8000   8000    8000
slin32 14500 14500 14500 14500 14500  8500 14500 14500 14500    14500 14500    14500 14500   8500   14500    6000   8500   8500      -   8000   8000   8000    8000
slin44 14500 14500 14500 14500 14500  8500 14500 14500 14500    14500 14500    14500 14500   8500   14500   14500   8500   8500   8500      -   8000   8000    8000
slin48 14500 14500 14500 14500 14500  8500 14500 14500 14500    14500 14500    14500 14500   8500   14500   14500   8500   8500   8500   8500      -   8000    8000
slin96 14500 14500 14500 14500 14500  8500 14500 14500 14500    14500 14500    14500 14500   8500   14500   14500   8500   8500   8500   8500   8500      -    8000
slin192 14500 14500 14500 14500 14500  8500 14500 14500 14500    14500 14500    14500 14500   8500   14500   14500   8500   8500   8500   8500   8500   8500       -
```

Recursos

Asterisk the Definitive Guide (4th Edition)
http://ofps.oreilly.com/titles/9781449332426/

Asterisk 11 Official Wiki
https://wiki.asterisk.org/wiki/display/AST/Asterisk+11+Documentation

VOIP-INFO
http://www.voip-info.org/

Lista de correo Asterisk-ES
https://groups.google.com/forum/#!forum/asterisk-es

Certificación dCAA

Digium ofrece la posibilidad obtener la certificación dCAA (Digium Certified Asterisk Administrator) de forma gratuíta y online mediante el formulario ubicado en:

http://www.digium.com/en/training/asterisk/certifications/dcaa

La prueba consta de 60 preguntas en inglés y se hace online, requiriéndose tan solo tener una cuenta de acceso a la pagina web de Digium. Este examen es el previo a presentarse al dCAP (Digium Certified Asterisk Professional) y es el examen que se pasa tras el curso Asterisk Fast Start.

Se requiere un mínimo de 80% de aciertos. ¡ Animo !

Fin del curso

Es imposible hacer maestros en solo cinco días. Pero si que es posible guiar al alumno por los sitios mas peliagudos, donde una persona sola ante del dilema de un problema, puede quedar atascada. Nuestro propósito por tanto ha sido en todo momento daros las herramientas para que podais ampliar el conocimiento que os hemos dado aquí, y que de ese modo, perdido el miedo inicial a Asterisk, podáis avanzar sin problemas.

Y recordar que, en la medida de mis posibilidades, yo como profesor del curso estoy a vuestra disposición para poder ampliaros este contenido y resolver las posibles dudas que os puedan salir.