

Unidad 6



Servicio de VoIP

Servicios de
red e internet



Índice



- 6.1. Funcionamiento básico de VoIP
- 6.2. Protocolos de VoIP
 - 6.2.1. Señalización SIP
 - 6.2.2. Transporte de medios: protocolo RTP/RTCP
- 6.3. Requisitos previos para instalar Asterisk
- 6.4. Descargar Asterisk
- 6.5. Instalar las dependencias de Asterisk
- 6.6. Instalar Asterisk
- 6.7. Crear un usuario en Asterisk
- 6.8. Iniciación en Asterisk



Introducción

Esta unidad va a tratar sobre la compartición de Voz por internet, o VoIP.

Para poder enseñar de manera eficiente cómo funciona este servicio veremos a lo largo de los distintos apartados las principales características que debe de tener una red para poder transmitir voz correctamente. En cuanto a esto nos referimos analizaremos los protocolos de transporte, señalización y los codificadores que son necesarios para poder transmitir la voz,

Por último, en la unidad además de contemplar Asterisk como principal software de VoIP para Linux, analizaremos las ventajas e inconvenientes que caracterizan a este servicio.

Al finalizar esta unidad

- + Sabremos cómo funciona realmente VoIP.
- + Podremos crear usuarios que sean capaces de iniciar sesión con Asterisk.
- + Conoceremos los protocolos más usados en la voz por IP.
- + Conoceremos cómo funciona Asterisk en su interfaz más básica.
- + Seremos capaces de instalar Asterisk como servicio.



6.1.

Funcionamiento básico de VoIP

VoIP son las siglas de Voz sobre IP, tecnología que permite realizar llamadas telefónicas sobre redes de datos, principalmente internet. A este término también se le conoce como telefonía de internet, porque mientras que la telefonía tradicional transporta una señal analógica sobre un cable de cobre, *VoIP* transmite la voz analógica como un paquete de datos funcionando sobre los protocolos IP y UDP. Complementario a esto, se pueden usar protocolos para transmitir la voz en tiempo real, como puede ser *RTP*, *Real Time Protocol*.

Una de las principales ventajas que presenta *VoIP* es que es en su mayoría código abierto, lo que permite que se distribuya y modifique el código fuente de la mayoría de las aplicaciones.

Como principal ventaja del servicio de voz por IP tenemos su facilidad y flexibilidad a la hora de introducir nuevos servicios en una red, pero no todo son ventajas, porque la tecnología de telefonía en internet tiene como principal defecto que se pueden producir grandes retrasos en la llegada de los paquetes de la red y es más susceptible a la interrupción de la conversación que otros tipos de telefonía.

Para poder dotar de cierta calidad al servicio de *VoIP* es importante contar con un ancho de banda considerable que pueda soportar el grandísimo tráfico que exige intercambiar voz en tiempo real.





6.2.

Protocolos de VoIP

Dentro de la telefonía por IP tenemos varios protocolos, pero solo vamos a ver dos, el protocolo de señalización SIP y el protocolo de transporte RTP/RTCP.

6.2.1. Señalización SIP

SIP es el acrónimo de *Session Initiation Protocol*, y es el estándar que IETF usa para poder establecer conexiones de voz por IP.

Este protocolo funciona en la capa de aplicación y se encarga de crear, modificar y finalizar las sesiones que engloben a varios participantes.

El protocolo de transporte, SIP, crea mediante invitaciones mensajes *SDP*, *Session Description Protocol* para poder llevar a cabo el intercambio de la información mediante los distintos canales de control.

SIP se trata de un protocolo con funcionamiento punto a punto y en el que se definen dos principales elementos que permiten su funcionamiento más básico:

- > **User agents – US:** este consta de dos partes, el cliente y el servidor. Mientras que el cliente genera las peticiones SIP y recibe las respuestas, el servidor se encarga de generar las respuestas a las peticiones que recibe,
- > **Servidores:** en los servidores de SIP tenemos tres tipos distintos de servidores dependiendo de sus funciones:
 - » *Proxy server.*
 - » *Registrar server*
 - » *Redirect server.*

Además, el protocolo SIP permite definir principalmente seis tipos de solicitudes:

- > *INVITE:* se establece una sesión.
- > *ACK:* confirma la solicitud *INVITE*,
- > *BYE:* termina la sesión.
- > *CANCEL:* cancela que se establezca una sesión.
- > *REGISTER:* comunica la localización de los usuarios.
- > *OPTIONS:* indica la información acerca del estado de la red para poder enviar y recibir información.

6.2.2. Transporte de medios: protocolo RTP/RTCP

Para poder enviar y recibir voz y vídeo sobre IP, esta tecnología se basa en el protocolo IP y UDP, pero no es suficiente para asegurar que se transmita correctamente la voz. Además, si recordamos, UDP se trata de un protocolo que no tienen corrección de errores, por lo que no sabemos nunca si han llegado correctamente los paquetes. Por eso, para poder transportar en tiempo real la información de audio y vídeo, existen dos protocolos que se deben implementar. Estos son: *RTP*, *Real-Time Transport Protocol* y *RTCP*, *RTP Control Protocol*.

Ambos protocolos se encuentran en el nivel de aplicación y son usados junto al protocolo de transporte UDP. Ambos funcionan tanto en *unicast* como en *multicast*.

RTP y RTCP no usan los mismos puertos, el primero usa normalmente un puerto número par y el segundo el siguiente puerto impar. Al abrir una sesión RTP, se abre automáticamente una sesión RTCP implícita.

La principal función de ambos protocolos es la de poder proporcionar un medio uniforme de transmisión de datos que aguarde las limitaciones que crea el envío en tiempo real.

RTP, sobre todo permite lo siguiente:

- > Identificar el tipo de información transportada.
- > Añadir marcadores temporales que ayudan a identificar el instante en el que se envía un paquete. Así, la aplicación de destino puede sincronizar los flujos y medir los retardos y la fluctuación.
- > Incluir números de secuencia a la información que se transporta y así poder determinar si ha habido alguna pérdida de paquetes o si no han llegado en el orden correcto.



6.3.

Requisitos previos para instalar Asterisk

Para poder configurar el servicio de voz por IP en Ubuntu nos vamos a basar en *Asterisk*, distribución de *software* libre.

Para poder instalar este servicio debemos de tener instalados los siguientes paquetes, primeramente:

- > wget
- > build-essential
- > subversion

6.4.

Descargar Asterisk

Para poder usar *Asterisk* tenemos dos modos, el primero es instalar el apropiado paquete, y el segundo, que es el que vamos a explicar aquí, es instalarlo desde la página web e ir configurando sus dependencias.

Lo primero que vamos a hacer es movernos al directorio `/usr/src/`, que es donde vamos a descargar el paquete de *Asterisk*.

Lanzamos el siguiente comando para descargar *Asterisk*:

`wget ruta_descarga`

```
root@ubuntu:/usr/src# wget http://downloads.asterisk.org/pub/telephony/asterisk/asterisk-18-current.tar.gz
--2022-09-07 13:44:58-- http://downloads.asterisk.org/pub/telephony/asterisk/asterisk-18-current.tar.gz
Resolviendo downloads.asterisk.org (downloads.asterisk.org)... 170.249.154.172
Conectando con downloads.asterisk.org (downloads.asterisk.org)[170.249.154.172]:80... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 28193615 (27M) [application/x-gzip]
Guardando como: 'asterisk-18-current.tar.gz'

asterisk-18-current 100%[=====] 26,89M 4,75MB/s en 17s

2022-09-07 13:45:15 (1,55 MB/s) - 'asterisk-18-current.tar.gz' guardado [28193615/28193615]
```

Imagen 1. Descarga de *asterisk* 1

Con el comando `tar`, descomprimos el fichero de instalación y podemos ver que se crea un nuevo directorio con todos los ficheros necesarios.

```
root@ubuntu:/usr/src# tar xzf asterisk-18-current.tar.gz
root@ubuntu:/usr/src# cd asterisk-18.14.0/
root@ubuntu:/usr/src/asterisk-18.14.0# ls
addons          images
agi             include
apps            install-sh
asterisk-18.14.0-summary.html LICENSE
asterisk-18.14.0-summary.txt main
autoconf        Makefile
bootstrap.sh    Makefile.moddir_rules
bridges          Makefile.rules
BSDMakefile     makeopts.in
BUGS             menuselect
build_tools     missing
cdr              mkinstalldirs
cel              pbx
ChangeLog        phoneprov
CHANGES         README-addons.txt
channels         README.md
codecs           README-SERIOUSLY.bestpractices.md
config.guess     res
configs          rest-api
config.sub       rest-api-templates
configure        sample.call
configure.ac     sounds
contrib          static-http
COPYING          tests
```

Imagen 2. Descarga de *asterisk* 2



6.5.

Instalar las dependencias de Asterisk

Para que *Asterisk* funcione correctamente necesita tener ciertas dependencias instaladas y funcionales, las principales e indispensables son las fuentes de MP3 para poder construir el mismo módulo y poder usar los ficheros MP3 en el servicio.

Para poder instalar estas dependencias debemos de realizar los siguientes pasos:

1. Lo primero es en el directorio que se ha creado tras la descompresión de los ficheros, asignar permisos de ejecución al *script* que instala estas dependencias. Se debe de usar el siguiente comando:

```
chmod u+x contrib/scripts/get_mp3_source.sh
```

2. Ahora instalamos las dependencias ejecutando el *script* con el comando:

```
./contrib/scripts/get_mp3_source.sh
```

```
root@Ubuntu:/usr/src/asterisk-18.14.0# chmod u+x contrib/scripts/get_mp3_source.sh
root@Ubuntu:/usr/src/asterisk-18.14.0# ./contrib/scripts/get_mp3_source.sh
A   addons/mp3
A   addons/mp3/MPGLIB_README
A   addons/mp3/common.c
A   addons/mp3/huffman.h
A   addons/mp3/tablint.c
A   addons/mp3/Makefile
A   addons/mp3/README
A   addons/mp3/decode_t386.c
A   addons/mp3/dct64_t386.c
A   addons/mp3/MPGLIB_TODO
A   addons/mp3/mpg123.h
A   addons/mp3/layer3.c
A   addons/mp3/mpglib.h
A   addons/mp3/decode_ntom.c
A   addons/mp3/interface.c
Se exportó la revisión 204.
root@Ubuntu:/usr/src/asterisk-18.14.0#
```

Imagen 3. Instalar dependencias de *asterisk* 1

3. Realizamos los mismos pasos con el fichero *install_prereq*, pero en este caso el comando de ejecución queda:

```
./contrib/scripts/install_prereq install
```

4. Dejamos las opciones por defecto.
5. Si todo ha salido bien. Al final deberemos ver un mensaje como el de la siguiente imagen:

```
#####
## install completed successfully
#####
```

Imagen 4. Instalar dependencias de *asterisk* 2



6.6.

Instalar Asterisk

Los comandos que vamos a ir enumerando a continuación, además de finalizar la instalación de *Asterisk*, realizará comprobaciones para cerciorarse de que las dependencias del sistema son las necesarias.

Vamos a seguir los siguientes pasos:

1. Lo primero que vamos a hacer es lanzar el *script configure*, para lo que habrá de darle permisos como anteriormente y ejecutarlo con el comando. `/configure`.
2. Si todo sale bien, al final de la ejecución del comando saldrá un dibujo como el de la siguiente imagen:



Imagen 5. Instalación de *asterisk* 1

3. En este momento se deben de seleccionar los módulos que se desean compilar e instalar, en este caso queremos el de MP3, para lo que lanzamos el comando siguiente:

`make menuselect`

Esto habrá que hacerlo desde una terminal gráfica, porque se necesita un tamaño mínimo para desplegar menú en la terminal.

```
root@Ubuntu:/usr/src/asterisk-18.14.0# make menuselect
CC="cc" CXX="g++" LD="" AR="" RANLIB="" CFLAGS="" LDFLAGS="" make -C menuselect
CONFIGURE_SILENT="--silent" cmenuselect
make[1]: se entra en el directorio '/usr/src/asterisk-18.14.0/menuselect'
gcc -g -D_GNU_SOURCE -Wall -Wno-deprecated-declarations -DHAVE_NCURSES -I/usr/i
nclude/libxml2 -c -o menuselect.o menuselect.c
gcc -g -D_GNU_SOURCE -Wall -Wno-deprecated-declarations -DHAVE_NCURSES -c -o
strcompat.o strcompat.c
gcc -g -D_GNU_SOURCE -Wall -Wno-deprecated-declarations -DHAVE_NCURSES -c -
o menuselect_curses.o menuselect_curses.c
gcc -o cmenuselect menuselect.o strcompat.o menuselect_curses.o -lncurses -lt
info -lxml2
make[1]: se sale del directorio '/usr/src/asterisk-18.14.0/menuselect'
CC="cc" CXX="g++" LD="" AR="" RANLIB="" CFLAGS="" LDFLAGS="" make -C menuselect
CONFIGURE_SILENT="--silent" nmenuselect
make[1]: se entra en el directorio '/usr/src/asterisk-18.14.0/menuselect'
gcc -g -D_GNU_SOURCE -Wall -Wno-deprecated-declarations -DHAVE_NCURSES -c -o
menuselect_newt.o menuselect_newt.c
gcc -o nmenuselect menuselect.o strcompat.o menuselect_newt.o -lnewt -lxml2
make[1]: se sale del directorio '/usr/src/asterisk-18.14.0/menuselect'
CC="cc" CXX="g++" LD="" AR="" RANLIB="" CFLAGS="" LDFLAGS="" make -C menuselect
CONFIGURE_SILENT="--silent" gmenuselect
make[1]: se entra en el directorio '/usr/src/asterisk-18.14.0/menuselect'
make[1]: No se hace nada para 'gmenuselect'.
make[1]: se sale del directorio '/usr/src/asterisk-18.14.0/menuselect'
make[1]: se entra en el directorio '/usr/src/asterisk-18.14.0'
Generating input for menuselect ...
```

Imagen 6. Instalación *asterisk* 2



6.7.

Crear un usuario en Asterisk

Asterisk por defecto funciona con el usuario *root*, pero es más seguro trabajar con otro usuario, por lo que vamos a crear un nuevo usuario y configurarlo en la plataforma de *Asterisk* para que funcione con este usuario. Seguimos los siguientes pasos:

1. Creamos el usuario *asterisk* sin directorio *home*, pero con distintas opciones, como podemos ver en la siguiente imagen:

```
root@Ubuntu:/usr/src/asterisk-18.14.0# adduser --system --group --home /var/lib/asterisk --no-create-home asterisk
Añadiendo el usuario del sistema 'asterisk' (UID 131) ...
Añadiendo un nuevo grupo 'asterisk' (GID 139) ...
Añadiendo un nuevo usuario 'asterisk' (UID 131) con grupo 'asterisk' ...
No se crea el directorio personal '/var/lib/asterisk'.
root@Ubuntu:/usr/src/asterisk-18.14.0#
```

Imagen 11. Crear usuario en *asterisk* 1

2. Editamos el fichero `/etc/default/asterisk`.
3. En este fichero tenemos dos líneas comentadas que debemos descomentar, que son:

```
AST_USER="asterisk"
AST_GROUP="asterisk"
```

```
# socket, the asterisk database, etc.
AST_USER="asterisk"
AST_GROUP="asterisk"
```

Imagen 12. Crear usuario *asterisk* 2

4. Añadimos al usuario *asterisk* a los grupos *dialout* y *audio*.
5. Cambiamos los permisos de todos los ficheros y directorios de *Asterisk* para que el usuario que se ha creado pueda trabajar sobre ellos, se hacen con los comandos que se pueden ver en la siguiente imagen:

```
root@Ubuntu:/usr/src/asterisk-18.14.0# chown -R asterisk /var/{lib,log,run,spool}/asterisk /usr/lib/asterisk/ /etc/asterisk/
root@Ubuntu:/usr/src/asterisk-18.14.0# chmod -R 750 /var/{lib,log,run,spool}/asterisk /usr/lib/asterisk/ /etc/asterisk/
root@Ubuntu:/usr/src/asterisk-18.14.0#
```

Imagen 13. Crear usuario *asterisk* 3



6.8.

Iniciación en Asterisk

Una vez que todo se encuentra correctamente configurado, podemos iniciar el servicio de *Asterisk* con el siguiente comando:

```
systemctl start Asterisk
```

Si ya se encuentra iniciado el servicio, se puede acceder a la interfaz de línea de comandos de *Asterisk* usando el comando:

```
asterisk -vvvr
```

Podemos salir de esta interfaz con la opción `exit`.

```
root@Ubuntu:/usr/src/asterisk-18.14.0# asterisk -vvvr
Asterisk 18.14.0, Copyright (C) 1999 - 2022, Sangoma Technologies Corporation and others.
Created by Mark Spencer <markster@digium.com>
Asterisk comes with ABSOLUTELY NO WARRANTY; type 'core show warranty' for details.
This is free software, with components licensed under the GNU General Public
License version 2 and other licenses; you are welcome to redistribute it under
certain conditions. Type 'core show license' for details.
-----
Connected to Asterisk 18.14.0 currently running on Ubuntu (pid = 15548)
Ubuntu*CLI> exit
Asterisk cleanly ending (0).
Executing last minute cleanups
root@Ubuntu:/usr/src/asterisk-18.14.0#
```

Imagen 14. Consola de Asterisk

Se puede habilitar el servicio de modo que al iniciar el sistema se inicie por defecto si usamos el comando `systemctl` con la opción `enable`.





 www.universae.com

