

# IMPLEMENTACION DEL PROTOCOLO SS7 SOBRE CONEXIONES ENTRE DOS SERVIDORES ASTERISK

Freddy Huayamave<sup>(1)</sup>, Pedro Rivadeneira<sup>(2)</sup>, Washington Medina<sup>(3)</sup>  
Facultad de Ingeniería en Electricidad y Computación<sup>(1) (2) (3)</sup>  
Escuela Superior Politécnica del Litoral (ESPOL)<sup>(1) (2) (3)</sup>  
Campus Gustavo Galindo, Km 30.5 Vía Perimetral, Apartado 09-01-5863. Guayaquil, Ecuador<sup>(1) (2) (3)</sup>  
fahuayam@espol.edu.ec<sup>(1)</sup>, pedrivad@espol.edu.ec<sup>(2)</sup>, wmedina@espol.edu.ec<sup>(3)</sup>

## Resumen

*Este proyecto tiene como propósito general la implementación de 3 protocolos (SIP, IAX, SS7) sobre Asterisk, indicando sus ventajas y desventajas entre sí; esto con la finalidad de otorgar una guía acerca de cuál protocolo utilizar en la creación de una central telefónica con base en Asterisk.*

*La implementación de este proyecto busca demostrar las ventajas que se tienen al implementar el protocolo SS7, comparándolo con los protocolos SIP e IAX; así también como administrar y garantizar de forma eficiente tanto los recursos económicos como tecnológicos que nos ofrece un EI.*

*Por medio de este proyecto daremos una guía para el desarrollo e implementación del servicio de telefonía a través de Voz sobre IP; con esto se reducirán los costos en la adquisición, mantenimiento y monitoreo de equipos de comunicación evitando gastos innecesarios en centrales telefónicas que representen costos excesivos, considerando las opciones existentes en el mercado.*

## Abstract

*This project aims generally three implementing protocols (SIP, IAX, and SS7) on Asterisk, indicating its advantages and disadvantages between each other, in order to provide a guide for user who wants to build a telephone exchange based on Asterisk*

*The implementation of this project seeks to demonstrate the advantages that have to implement the SS7 protocol, comparing SIP and IAX protocols, as well as manage and efficiently ensure both economic and technological resources that offer an EI.*

*Through this project will give a guide to the development and implementation of telephone service through Voice over IP, with this we will reduce costs in the acquisition, maintenance and monitoring of communication equipment avoiding unnecessary costs in telephone implementation, considering the options available on the market.*

## 1. Introducción de Asterisk

Asterisk es una aplicación que permite el control y gestión de comunicaciones, ya sean analógicas, digitales o VoIP, mediante los diversos protocolos que se necesiten para su implementación. Su implementación bajo OpenSource presenta muchas ventajas para los desarrolladores dándoles la posibilidad de crear sistemas de comunicaciones de excelente calidad, seguridad y versatilidad. Existen diferentes versiones de Asterisk en la actualidad, la versión a utilizar en este proyecto será Asterisk 1.8 que presenta una buena funcionalidad y compatibilidad con los drivers necesarios para la identificación de la tarjeta E1/T1 a utilizar. [1]

### 1.1. Descripción de la problemática

El proyecto a realizar consiste en la implementación del protocolo SS7 (Sistema de Señalización No.7) sobre 2 servidores Asterisk. Asterisk es un software que permite la construcción de soluciones de Voz sobre IP (Central Telefónica VoIP).

Utilizando Asterisk en su entorno es posible olvidar las limitaciones tradicionales de las centrales telefónicas; esto es, no más problemas de alcanzar el máximo de extensiones posibles, evitando de esta forma el tener problemas de saturación que afectan actualmente a las centrales telefónicas convencionales. Con la realización de este proyecto se busca demostrar las ventajas que se tienen al implementar el protocolo SS7; así como también administrar y garantizar de forma eficiente tanto los recursos económicos como tecnológicos, a la vez se realizará una comparación con los protocolos SIP (*Session Initiation Protocol*) e IAX (*Inter-Asterisk eXchange protocol*), para satisfacer el conjunto de necesidades del usuario final en relación al costo-beneficio. Por medio de este proyecto daremos una guía para el desarrollo e implementación del servicio de telefonía a través de Voz sobre IP; con esto se reducirán los costos en la adquisición, mantenimiento y monitoreo de equipos de comunicación evitando gastos innecesarios en centrales telefónicas que representen costos excesivos, considerando las opciones existentes en el mercado. Al implementar telefonía IP en esta red se pueden dar los servicios de 1-800, correo de voz, llamada e espera entre otras.

## 2. Hardware Implementado.

En la parte de hardware contamos con los siguientes recursos

- 2 computadores con procesador Pentium IV en adelante.
- Memoria Ram superior a 512 MB

- Discos duros Sata 20 GB
- 2 tarjetas PCI con soporte E1/T1.
- 1 cable cruzado T1/E1.[2]

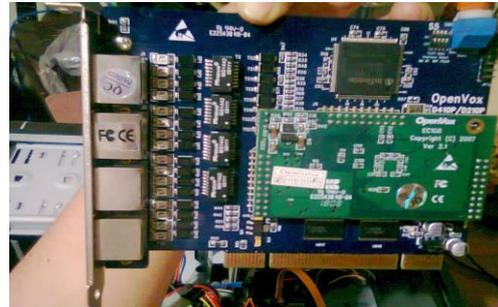


Figura 1. Tarjeta PCI OpenVox D410P



Figura 2. Tarjeta PCI OpenVox D110P

### 2.1 Software

En cuanto a software podemos utilizar:

- Linux S.O. recomendable Centos o Ubuntu Server.
- Softphone Zoiper o X-Lite.
- Wireshark analizador de paquetes.
- Putty Cliente ssh para conexión remota hacia los servidores.
- Vmware Player para la virtualización de sistemas operativos clientes.

### 2.2 Diagrama de Red

A continuación el esquemático del diagrama de red a utilizar en este proyecto.

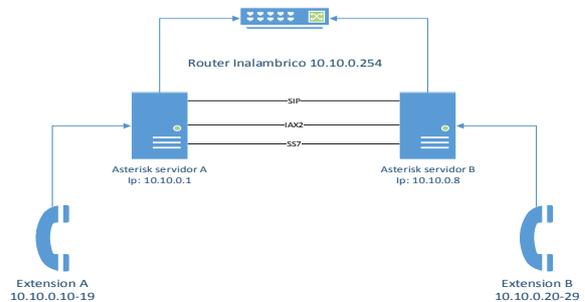


Figura 3. Diagrama de red

### 3. Configuración archivos Asterisk

En la implementación de este proyecto editaremos ciertos archivos de configuración, que permitirán la conexión entre ambos servidores utilizando SIP, IAX y SS7 respectivamente. La ubicación de estos archivos se encuentra en la ruta /etc/asterisk/.

#### 3.1 Archivo sip.conf

En este archivo se configuran las cuentas que utilizan el protocolo SIP para la comunicación entre usuarios. A continuación se muestran los parámetros básicos de configuración para este fichero, teniendo en cuenta los comentarios que están precedidos por ‘; (punto y coma) ’.

En la consola del servidor digitamos:

```
[root@asterisk-]#vim etc/asterisk/sip.conf
```

Pulsamos la tecla ‘I’ para proceder a editar el archivo. [3]

##### 3.1.1 Parámetros generales.

###### [nombre\_de\_cuenta]

Indica el nombre de la cuenta que se debe utilizar en el teléfono IP o softphone a registrar.

###### Type

Representa el tipo de extensión. Sus valores pueden ser friend el cual permite hacer y recibir llamadas, user solo recibir y peer solo puede hacer llamadas.

###### Secret

Clave de autenticación.

###### Context

Contexto o grupo de extensiones en donde será usada dicha cuenta. Ver archivo extensions.conf.

###### Language

Define las señales para un país, debe estar presente en el archivo indications.conf.

###### Disallow

Deshabilita la entrega o recepción de todos los audios de codecs para esta cuenta, esto se lo hace para que no elija cualquier códec aleatoriamente.

###### Allow

Especifica el codec de audio que puede entregar o recibir esta cuenta.

###### Nat

Asume que el dispositivo está en una red privada, y debe utilizar nat para poder salir a una red externa. Puedes ser yes o no.

##### 3.1.2 Ejemplo cuenta SIP

```
[TelefonoA1]  
type=friend  
secret=1234
```

```
host=dynamic  
context=users  
;nat=yes  
disallow=all  
allow=gsm  
allow=ulaw  
allow=alaw
```

#### 3.2 Archivo iax.conf

Para la creación de cuentas se pueden utilizar los mismos parámetros que el archivo sip.conf exceptuando por el parámetro nat que no es necesario en este protocolo. [4]

##### 3.2.1 Parámetros generales

###### Trunk

Se establece para utilizar el trunking ofrecido por IAX2, puede ser yes o no.

###### Deny

Denegar llamadas entrantes desde cualquier IP, va acompañado del parámetro permit para habilitar la única IP, por la que asterisk escucha llamadas entrantes IAX.

###### Permit

Dirección IP permitida para las llamadas entrantes.

##### 3.2.2 Ejemplo cuenta IAX

```
[TiaxB2]  
type=friend  
host=dynamic  
secret=4321iaxB  
context=iaxgroupB  
qualify=yes  
callerid=iax2B
```

#### 3.3 Archivo extensions.conf

Contiene el plan de marcado de la central telefónica para cada contexto y por lo tanto para cada cuenta.

##### 3.3.1 Parámetros generales.

###### Static

Indica si se graba o no el plan de marcado desde la consola Asterisk ejecutando save dialplan.

###### Autofallthrough

Si se observa yes es porque alguna llamada por algún índice si la salida del plan y se terminará.

###### Clearglobalvars

Si está activado se liberan las variables globales cuando se recargan las extensiones o se reinicia Asterisk.

###### Priorityjump

Si tiene valor yes, la aplicación soporta jumping o salto a diferentes prioridades.

### 3.4 Archivo ss7.conf

En este archivo se configuran los parámetros de conexión SS7 para cada servidor. Este archivo no viene incluido al instalar Asterisk, por lo que no lo encontraremos sin antes haber instalado chan\_ss7.

Estos dos parámetros son muy importante estén bien configurados en ambos servidores

```
[host-B]  
enabled => yes  
opc => 0x2  
dpc => siuc:0x1  
links => 12:1
```

### 3.5 Archivo system.conf

Este archivo sirve para la configuración de señalización en las tarjetas E1/T1 que vamos a utilizar para la conexión de los dos servidores por medio de ss7. [5]

### 3.6 Configuración dahdi\_channels.conf

Este archivo debe ser modificado ya que si no lo hacemos puede generar problemas en la señalización en el span o conector a utilizar para la conexión por medio de ss7.

## 4. Configuración del Equipo de Interconexión

El equipo de interconexión puede ser un switch o un router, dependiendo de los requerimientos para la implementación, a veces resulta útil tener un equipo inalámbrico, ya que en la actualidad existen diversos softwares y aplicaciones que pueden ejecutarse en dispositivos inteligentes, laptop o tablets, al realizar las pruebas se implementó el router ya que puede funcionar como switch y a la vez brinda la conexión inalámbrica.

### 4.1 Creación del Servicio

Asterisk puede implementar múltiples servicios tales como centrales telefónicas, correo de voz, llamadas en espera, entre otras. En este proyecto abarcaremos un tipo de servicio básico y común usado en algunas instituciones.

Este servicio está basado en la implementación de un IVR, donde el usuario llama y escucha una grabación que le presente una serie de opciones para comunicarse con el departamento o persona que se indica en la grabación

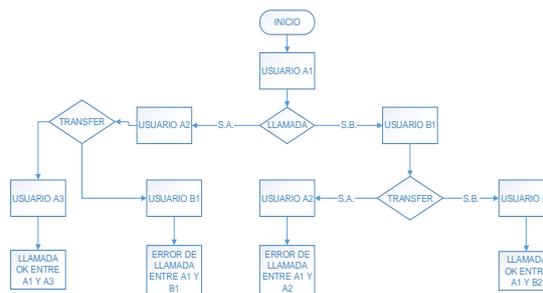


Figura 4 Diagrama de flujo del servicio

## 5. Funcionamiento y pruebas del proyecto

En este capítulo se detalla la parte técnica utilizada para la implementación del proyecto, las instalaciones, configuraciones, pruebas y funcionamiento se muestran a continuación.

### 5.1 Etapa de pruebas entre servidores

Para poder programar en Asterisk primero debemos conocer algunos comandos de Linux, que serán de mucha ayuda para la configuración, adicionalmente cada servidor debe contar con.

- Una instalación básica de Instalación de UBUNTU SERVER 12.04, debido a que nuestra prioridad es la configuración de Asterisk.
- Una vez instalado Ubuntu Server 12.04 procedemos con la instalación de Asterisk. Antes de iniciar el proceso de instalación hay que asegurarse que el sistema operativo esté actualizado.
- Instalamos las dependencias básicas que necesitaremos para Asterisk

Una vez cumplida la correcta instalación se debe comprobar que los archivos sip.conf, iax.conf, ss7.conf estén correctamente configurados en cada uno de los servidores para empezar con las pruebas.

### 5.2 Capturas con Wireshark

Para tomar las capturas con Wireshark debemos tener instalado en el servidor la herramienta Tcpdump que permite capturar paquetes recibidos y transmitidos, en tiempo real, en la red en que se encuentra nuestro servidor.

Ingresamos a la consola de nuestro servidor como usuario root y creamos un nuevo directorio en donde guardaremos las capturas.

```
[root@asterisk -]# mkdir capturas
```

Entramos al directorio creado

```
[root@asterisk -]# cd capturas
```

Para proceder con la captura utilizamos el comando:

[root@asterisk -]# tcpdump -i any -s 1500 -w capturaXX.pcap

Donde "XX" es el número de captura, las que se grabaran con extensión .pcap que puede ser leída por Wireshark.

Una vez ejecutado el comando comenzará a capturar paquetes de todas las interfaces en el servidor, si queremos que solo tome capturas de una interface agregamos el nombre de la interfaz (eth0,eth1,eth2...) al lado del comando -i.

Para terminar con la captura tecleamos CTRL+C.

Ahora ya podemos copiar las capturas creadas para analizar con wireshark.

### 5.2.1 Mensajes y pruebas

Mensajes SIP al momento de registrar un usuario SIP a través del softphone Zoiper en Windows.

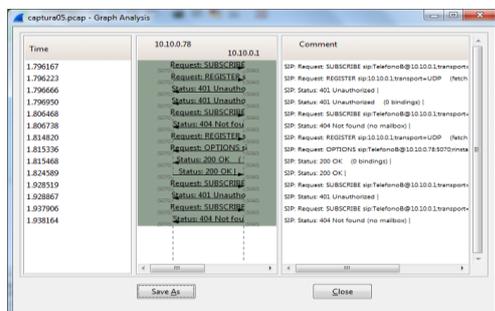


Figura 5 Mensajes SIP, Registro de usuario

Las figura 5 nos indica los pasos para el registro de una cuenta SIP en Asterisk, vemos dos estados de OK 200 lo cual significa una correcta asociación con el servidor.

En las figura 6 se describe una llamada SIP, se utiliza un filtro RTP, ya que por este protocolo es por el que viaja el audio perteneciente a la llamada.

Con esta captura de paquetes es posible volver a reproducir la llamada, esto sería muy útil para cuestiones de seguridad.

No.	Time	Source	Destination	Protocol	Length	Info
18	0.000000	10.10.0.1	10.10.0.1	RTP	24	PP-TTU-T 6.711 PCM, SSRC=0x15884870, Seq=413304, T1=0x4353804, Mark
39	16.397940	10.10.0.1	10.10.0.78	RTP	87	PP-GDP 06.10, SSRC=0x77295206, Seq=43330, T1=0x55016448, Mark
41	16.336972	10.10.0.1	10.10.0.1	RTP	87	PP-GDP 06.10, SSRC=0x58848470, Seq=47354, T1=0x4353804, Mark
42	16.332128	10.10.0.1	10.10.0.58	RTP	214	PP-TTU-T 6.711 PCM, SSRC=0x20523885, Seq=41659, T1=0x4353804, Mark
43	16.332913	10.10.0.78	10.10.0.1	RTP	87	PP-GDP 06.10, SSRC=0x58848470, Seq=47355, T1=0x4353804, Mark
44	16.332005	10.10.0.1	10.10.0.58	RTP	214	PP-TTU-T 6.711 PCM, SSRC=0x20523885, Seq=41660, T1=0x4353804, Mark
45	16.379204	10.10.0.78	10.10.0.1	RTP	87	PP-GDP 06.10, SSRC=0x58848470, Seq=47356, T1=0x4353804, Mark
46	16.372020	10.10.0.1	10.10.0.58	RTP	214	PP-TTU-T 6.711 PCM, SSRC=0x20523885, Seq=41661, T1=0x4353804, Mark
47	16.369398	10.10.0.78	10.10.0.1	RTP	87	PP-GDP 06.10, SSRC=0x58848470, Seq=47357, T1=0x4353804, Mark
48	16.380442	10.10.0.1	10.10.0.58	RTP	214	PP-TTU-T 6.711 PCM, SSRC=0x20523885, Seq=41662, T1=0x4353804, Mark
49	16.421945	10.10.0.78	10.10.0.1	RTP	87	PP-GDP 06.10, SSRC=0x58848470, Seq=47358, T1=0x4353804, Mark
50	16.422029	10.10.0.1	10.10.0.58	RTP	214	PP-TTU-T 6.711 PCM, SSRC=0x20523885, Seq=41663, T1=0x4353804, Mark
51	16.429204	10.10.0.78	10.10.0.1	RTP	87	PP-GDP 06.10, SSRC=0x58848470, Seq=47359, T1=0x4353804, Mark
52	16.432021	10.10.0.1	10.10.0.58	RTP	214	PP-TTU-T 6.711 PCM, SSRC=0x20523885, Seq=41664, T1=0x4353804, Mark
53	16.451897	10.10.0.78	10.10.0.1	RTP	87	PP-GDP 06.10, SSRC=0x58848470, Seq=47360, T1=0x4353804, Mark
54	16.451984	10.10.0.1	10.10.0.58	RTP	214	PP-TTU-T 6.711 PCM, SSRC=0x20523885, Seq=41665, T1=0x4353804, Mark
57	16.467919	10.10.0.58	10.10.0.1	RTP	214	PP-TTU-T 6.711 PCM, SSRC=0x00200044, Seq=48806, T1=0x2570043020, Mark
58	16.468622	10.10.0.1	10.10.0.78	RTP	87	PP-GDP 06.10, SSRC=0x58848470, Seq=47361, T1=0x4353804, Mark
59	16.472922	10.10.0.78	10.10.0.1	RTP	87	PP-GDP 06.10, SSRC=0x58848470, Seq=47362, T1=0x4353804, Mark
60	16.472023	10.10.0.1	10.10.0.58	RTP	214	PP-TTU-T 6.711 PCM, SSRC=0x20523885, Seq=41666, T1=0x4353804, Mark
61	16.487494	10.10.0.58	10.10.0.1	RTP	214	PP-TTU-T 6.711 PCM, SSRC=0x00200044, Seq=48807, T1=0x2570043020, Mark
62	16.487632	10.10.0.1	10.10.0.78	RTP	87	PP-GDP 06.10, SSRC=0x58848470, Seq=47363, T1=0x4353804, Mark

Figura 6 Mensajes SIP, Llamada entre usuarios

La figura 7 nos muestra los paquetes que se transmiten al momento de suscribir o registrar un usuario IAX en el servidor Asterisk, observamos cómo son enviados los mensajes ACK, los que indican que se ha establecido la conexión.

No.	Time	Source	Destination	Protocol	Length	Info
4	1.328426	10.10.0.78	10.10.0.1	IAX2	106	IAX, source callID 30, timestamp 2ms REGREQ
5	1.328793	10.10.0.1	10.10.0.78	IAX2	107	IAX, source callID 1, timestamp 2ms CALLTOEN
6	1.368569	10.10.0.78	10.10.0.1	IAX2	157	IAX, source callID 30, timestamp 3ms REGREQ
7	1.368900	10.10.0.1	10.10.0.78	IAX2	83	IAX, source callID 9279, timestamp 3ms REGAUTH
8	1.408551	10.10.0.78	10.10.0.1	IAX2	138	IAX, source callID 30, timestamp 4ms REGREQ
9	1.408861	10.10.0.1	10.10.0.78	IAX2	56	IAX, source callID 4697, timestamp 3ms PONG
10	1.409023	10.10.0.1	10.10.0.78	IAX2	104	IAX, source callID 9279, timestamp 4ms REGACK
11	1.428545	10.10.0.78	10.10.0.1	IAX2	60	IAX, source callID 1, timestamp 3ms ACK
12	1.428565	10.10.0.78	10.10.0.1	IAX2	60	IAX, source callID 1, timestamp 3ms PONG
13	1.428715	10.10.0.1	10.10.0.78	IAX2	54	IAX, source callID 4697, timestamp 3ms ACK
14	1.449826	10.10.0.78	10.10.0.1	IAX2	60	IAX, source callID 30, timestamp 3ms ACK

Figura 7 Mensajes IAX Registro de usuario

En la figura 8 nos muestra el transcurso de la llamada, aquí se puede observar las tramas o frames que utiliza el protocolo IAX para transportar el audio, a su vez se ve el codec de audio utilizado en esta llamada (G.711)

No.	Time	Source	Destination	Protocol	Length	Info
8127	55.338767	10.10.0.1	10.10.0.99	IAX2	208	208-byte packet, source callID 163, timestamp 48370ms, Raw mu-law data (6.711)
8128	55.338990	10.10.0.1	10.10.0.9	IAX2	218	218-byte packet with 1 media frame for 1 call
8129	55.339239	10.10.0.99	10.10.0.1	IAX2	208	208-byte packet, source callID 5, timestamp 48370ms, Raw mu-law data (6.711)
8130	55.339499	10.10.0.1	10.10.0.9	IAX2	218	218-byte packet with 1 media frame for 1 call
8131	55.339773	10.10.0.1	10.10.0.99	IAX2	208	208-byte packet, source callID 163, timestamp 48370ms, Raw mu-law data (6.711)
8132	55.339999	10.10.0.1	10.10.0.9	IAX2	218	218-byte packet with 1 media frame for 1 call
8133	55.339823	10.10.0.9	10.10.0.1	IAX2	218	218-byte packet with 1 media frame for 1 call
8134	55.339781	10.10.0.1	10.10.0.99	IAX2	208	208-byte packet, source callID 163, timestamp 48370ms, Raw mu-law data (6.711)
8135	55.339847	10.10.0.99	10.10.0.1	IAX2	208	208-byte packet, source callID 5, timestamp 48370ms, Raw mu-law data (6.711)
8136	55.337137	10.10.0.1	10.10.0.99	IAX2	208	208-byte packet, source callID 163, timestamp 48370ms, Raw mu-law data (6.711)
8137	55.337634	10.10.0.99	10.10.0.1	IAX2	218	218-byte packet with 1 media frame for 1 call
8138	55.337700	10.10.0.1	10.10.0.99	IAX2	208	208-byte packet, source callID 163, timestamp 48370ms, Raw mu-law data (6.711)
8139	55.337418	10.10.0.99	10.10.0.1	IAX2	208	208-byte packet with 1 media frame for 1 call
8140	55.337418	10.10.0.99	10.10.0.1	IAX2	208	208-byte packet, source callID 5, timestamp 48370ms, Raw mu-law data (6.711)
8141	55.339843	10.10.0.1	10.10.0.99	IAX2	208	208-byte packet with 1 media frame for 1 call
8142	55.339815	10.10.0.1	10.10.0.99	IAX2	208	208-byte packet, source callID 163, timestamp 48370ms, Raw mu-law data (6.711)
8143	55.339815	10.10.0.1	10.10.0.99	IAX2	208	208-byte packet, source callID 163, timestamp 48370ms, Raw mu-law data (6.711)
8144	55.339894	10.10.0.1	10.10.0.9	IAX2	218	218-byte packet with 1 media frame for 1 call
8145	55.339894	10.10.0.1	10.10.0.9	IAX2	218	218-byte packet with 1 media frame for 1 call
8146	55.418813	10.10.0.9	10.10.0.1	IAX2	218	218-byte packet with 1 media frame for 1 call
8147	55.421900	10.10.0.1	10.10.0.9	IAX2	208	208-byte packet, source callID 163, timestamp 48370ms, Raw mu-law data (6.711)
8148	55.421904	10.10.0.1	10.10.0.9	IAX2	218	218-byte packet with 1 media frame for 1 call
8149	55.421904	10.10.0.1	10.10.0.9	IAX2	218	218-byte packet with 1 media frame for 1 call
8150	55.421906	10.10.0.1	10.10.0.9	IAX2	218	218-byte packet with 1 media frame for 1 call
8151	55.421908	10.10.0.9	10.10.0.1	IAX2	218	218-byte packet, source callID 5, timestamp 48370ms, Raw mu-law data (6.711)

Figura 8 Mensajes IAX Llamada entre usuarios

Una vez realizadas las configuraciones en los archivos necesarios para el enlace SS7, procedemos a verificar el establecimiento de una llamada a través de esta señalización.

Para comprobar el estado del enlace vamos a la consola de Asterisk y digitamos:

```
asterisk*CLI> ss7 status
```

Si en el parámetro idle aparece el número 30 significa que estamos listos para utilizar el enlace por medio de SS7; caso contrario aparecerá el número 30 en el parámetro resetting lo que indica una mala configuración

En este caso tendremos que revisar cual es la causa del problema, pudiendo ser estas:

- Tarjetas insertadas de forma incorrecta en ranura PCI.
- Cable utilizado para el enlace en mal estado; recordar el tipo de configuración del cable debe ser Cruzado T1/E1.
- Falla en las ranuras PCI de los servidores.
- Error en la configuración de los archivos necesarios para ss7.

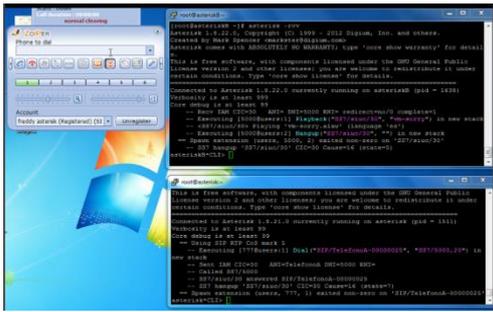


Figura 9 Proceso de llamada en SS7

### 5.3 Pruebas con Equipos

Se realizaron 4 escenarios para comprobar la versatilidad y funcionalidad de Asterisk en este proyecto, adicionalmente se uso software libre para el monitoreo (Wireshark).

#### 5.3.1 Escenario 1

En este escenario se realizaron 5 llamadas simultáneas por medio de SIP entre ambos servidores, con el wireshark se pudo comprobar la carga que genera en los servidores el protocolo SIP



Figura 10 Llamadas simultaneas SIP

#### 5.3.2 Escenario 2

Se realiza las mismas pruebas que en el escenario 1 pero ahora con el protocolo IAX.



Figura 11 Llamadas simultaneas IAX

### 5.3.3 Escenario 3

Este escenario busca comprobar la utilidad del servicio planteado en este proyecto, el cual consta de un IVR de ayuda para la ubicación de las extensiones hacia los usuarios o departamentos que correspondan.



Figura 12 Transferencia de Llamadas IVR

### 5.3.4 Escenario 4

Realizar una llamada SIP y otra llamada IAX por medio de SS7 para comprobar la estabilidad del enlace, mediante la consola de Asterisk comprobamos el correcto funcionamiento y los mensajes que se presentan en ss7.

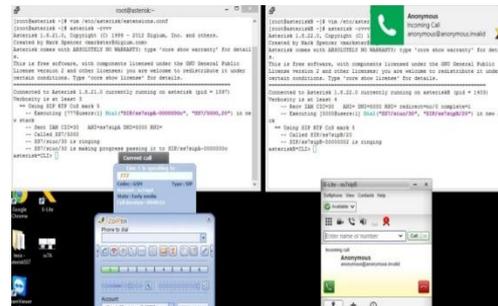


Figura 13 Proceso de llamada SS7

### 5.4 Tabla Comparativa

	SIP	IAX
NAT	La señalización y los datos se transportan independientemente en SIP, lo que hace que se produzcan problemas de NAT en el flujo de audio, cuando éste superar los routers y	La señalización y los datos viajan juntos en IAX y esto permite evitar los problemas de NAT que usualmente se crean en SIP.

	firewalls. SIP puede necesitar un servidor STUN para los problemas de NAT.	
ANCHO DE BANDA	Son mensajes de texto cuando se usa SIP.	IAX procura disminuir lo mas posible la información de las cabeceras de los mensajes, y al mismo tiempo disminuyendo el ancho de banda.
ESTANDARIZACIÓN	Hace muchos años que SIP es un protocolo estandarizado por IETF y lo implementan todos los fabricantes de equipos y software.	Aún no está estandarizado el protocolo IAX debido a que fue creado para resolver los problemas de NAT y por ello que no es utilizado en muchos dispositivos que se encuentran en el mercado.

PUERTOS	SIP usa un puerto (5060) para señalización y dos RTP (Real-time Transport Protocol) por cada conexión de audio (tres puertos minimo).	Para el envío de la información de señalización y datos de llamada IAX usa un puerto (4569). Por este motivo usa un mecanismo de multiplexión o <i>trunking</i> .
FLUJO DE AUDIO AL USAR UN SERVIDOR	La señalización de control en SIP pasa siempre por el servidor; sin embargo, la información de audio (flujo RTP) puede transportarse de un extremo a otro sin pasar por servidor SIP.	Todo el tráfico de audio debe pasar obligatoriamente por el servidor IAX, al pasar la señalización y los datos conjuntamente, y esto ocasiona un incremento en el ancho de banda que los servidores IAX deben soportar, más aun cuando hay muchas llamadas simultáneas.

FUNCIONALIDADES	<p>Cualquier información podría ser transmitida por SIP ya que es un protocolo de propósito general, pudiendo cualquier información, audio o video, ser transmitida sin dificultad.</p>	<p>IAX es un protocolo creado para telefonía IP y transmisión de video. Puede presentar funcionalidades interesantes como la de enviar o recibir planes de marcado (<i>dialplans</i>) que son muy útiles al usarlo junto con los servidores Asterisk.</p>
-----------------	---	---

## 6. Guía de elección

Para realizar una toma de decisión correcta, al momento de querer implementar una central telefónica, haremos una comparación entre los protocolos SIP, IAX2 y SS7 en base a varias características de estos protocolos, para que de acuerdo a esto el usuario tenga un conocimiento más claro al momento de implementar una central, según sus necesidades.

Las principales diferencias entre IAX2 y SIP son las siguientes:

El protocolo IAX2 maneja un sólo puerto de comunicaciones, el UDP 4569, esto quiere decir que tanto la señalización y el flujo de audio RTP de todas las llamadas viajan multiplexadas por el mismo canal, haciéndolo un protocolo casi transparente para los firewalls. Por otra parte, el protocolo SIP utiliza 3 puertos de comunicaciones: uno para señalización 5060 y dos para el flujo de audio RTP, haciendo que este protocolo necesite de procesos adicionales para resolver los problemas de NAT, en algunos casos con la utilización de un servidor STUN.

IAX2 reduce el ancho de banda utilizado en una llamada, al codificar los mensajes de señalización de forma binaria, así mismo IAX2 también intenta reducir la cantidad de información requerida para la señalización.

SIP es un protocolo que ya es un estándar IETF, por eso es muy común encontrar hardware y dispositivos que manejan este protocolo. En cambio, IAX2 aún está a la espera de su estandarización y en la actualidad se cuenta con muy pocos dispositivos que lo utilicen.

Aunque aparentemente el protocolo IAX2 hace que la señalización y la voz viajen por el mismo canal y sea una ventaja en los procesos de NAT, elimina la posibilidad de utilizar servidores proxy para las llamadas, y esto lleva a que el flujo de audio RTP (Real-time Transport Protocol) tenga que pasar obligatoriamente por el servidor, incrementando notablemente los costos por transferencia de datos del servidor.

La utilización de un servidor Proxy SIP, en un escenario VoIP, nos da la posibilidad de manejar miles de llamadas sin tener la carga del flujo de datos RTP de la llamada, ya que el audio viaja de un punto a otro sin tener que pasar por el servidor, de ahí la ventaja de mantener ambos tipos de datos (señalización y voz) separados.

En conclusión, ninguno de los dos protocolos es mejor o peor que el otro, simplemente tienen usos en aplicaciones y situaciones diferentes; por ejemplo, si se requiere de una estructura donde los problemas de NAT son críticos se puede optar por utilizar IAX2, pero si por el contrario se requiere de un escenario donde haya que manejar muchas llamadas, y se quiera disminuir los costos por transferencia de datos, utilizando SIP y un servidor Proxy, sería una mejor alternativa.

Existen infinidad de aplicaciones en las que hay que definir qué protocolo es el más adecuado, pero con la práctica se lo podrá resolver cada vez más rápido.

Por otra parte, tenemos las ventajas y características importantes del SS7, las cuales son muy superiores a los otros dos protocolos antes comparados. SS7 tiene alta capacidad, es decir con un solo enlace de señalización soporta cientos de troncales. Además, muestra una alta velocidad, puede establecer una llamada a través de varias centrales en menos de un segundo. Los enlaces de datos son de 56 Kbps nacional y de 64 Kbps internacionales.

Presenta un ahorro, ya que puede ser usado por un amplio rango de servicios de telecomunicaciones y requiere menos hardware que los sistemas anteriores. El SS7 puede aplicarse a todas las redes de telecomunicaciones nacionales e internacionales, así como en redes de servicios especializados (RSE) y en las redes de servicios digitales.

SS7 utiliza líneas separadas a las de la voz de los datos para dar información a la central sobre las llamadas en tránsito. Esto ha proporcionado muchas ventajas, una de ellas, es la optimización de la utilización del tiempo en las líneas existentes actualmente para voz; esto es debido a que la conmutación de las líneas se produce al momento de completarse la llamada. Además, como SS7 se ejecuta sobre líneas digitales y sistemas computarizados, entonces la optimización del proceso es mayor; teniendo la posibilidad de ofrecer otros servicios, tales como por ejemplo: acceso a bases de datos, llamado de vuelta automático, identificación del número de llamadas y muchos otros.

SS7 utiliza un canal de señalización común para todos los circuitos (en vez de tener señalización asociada a cada canal de tráfico), en caso, que se tengan uno o más enlaces de señalización. Un enlace puede manejar cientos de circuitos. En éste las señales de control se transmiten en canales dedicados exclusivamente a señalización, agrupando la señal de varios canales de voz y permitiendo que los conmutadores permitan intercambiar mensajes, en vez de secuencia de tonos y pudiendo operar bajo dos modalidades: canal asociado y canal no asociado.

A medida que nos movemos hacia la convergencia entre la red telefónica de conmutación de circuitos y el mundo IP de conmutación de paquetes, SS7 ha sido importante para los desarrolladores que buscan integrar los dos mundos y aprovechar lo mejor de ambos.

Por estas características y ventajas en las redes podemos concluir que el protocolo SS7 tiene superioridad ante los otros dos protocolos, con el único requerimiento que se tenga conocimiento suficiente para su configuración y manejo al momento de la implementación.

En esta guía se encuentran tres protocolos muy buenos para la implementación de una central telefónica; todo de acuerdo a las necesidades que tengamos en varios escenarios.

## 7. Conclusiones

Con la implementación del proyecto y a través de múltiples pruebas, se permitió la comunicación entre los dos servidores Asterisk empleando el código abierto distribuido del software Asterisk, con esto se pudo demostrar que el proyecto tiene eficiencia y eficacia al momento de interconectar correctamente los dos servidores junto con los protocolos SIP, IAX2 y SS7.

Al momento de probar el funcionamiento nos percatamos de que pueden trabajar los 3 protocolos al mismo tiempo. El funcionamiento de un protocolo es independiente para los otros, lo que significa que mientras realizamos una llamada SIP al mismo tiempo podemos hacer una llamada IAX o SS7. El número de llamadas simultáneas dependerá del ancho de banda, tipo de códec y recursos de hardware donde se encuentra instalado Asterisk.

Habiendo realizado los respectivos cambios en los archivos de configuración para SS7, observamos en las consolas de Asterisk que existen 30 canales para la transmisión de audio o datos, esto indica una correcta configuración y conexión entre ambos servidores por medio de SS7. Además al realizar la llamada de prueba se muestran los mensajes ISUP (CIC, IAM, DNI) que forman parte de este tipo de señalización.

Si se requiere crear una central telefónica interna en una empresa, basta con los protocolos SIP e IAX para su implementación. Estos trabajan por medio de IP y los costos son muy bajos en comparación a SS7. Si requerimos de una conexión a la red pública, tanto de telefonía fija como de telefonía móvil, lo recomendable es trabajar con el protocolo SS7 aunque éste demande costos elevados para llevarse a cabo. Es importante resaltar los beneficios que brinda SS7 tales como robustez en la corrección de errores, roaming, portabilidad de número local, mayor seguridad en las llamadas, entre otros.

## 8. Bibliografía

[1] Asterisk (nd). Introducción de Asterisk. Tomado de [http://comunidad.asterisk-es.org/index.php?title=Introduccion\\_a\\_Asterisk](http://comunidad.asterisk-es.org/index.php?title=Introduccion_a_Asterisk)

[2] pt (nd). Configuración de cable cruzado E1/T1. Tomado de <http://www.voipinfo.org/storage/users/59/27059/images/2957/medium.png>

[3] Voip Foro (n.d). Ejemplo de comunicación SIP. Tomado de <http://www.voipforo.com/SIP/SIPejemplo.php>

[4] Voip Foro (n.d). Ejemplo de comunicación IAX. Tomado de <http://www.en.voipforo.com/IAX/IAX-example-messages.php>

[5] Virtualpabx (2011). Asterisk SS7 How to. Tomado de <http://virtualpabx.wordpress.com/2011/07/21/asterisk-ss7-howto/>