ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL
CENTRO DE INVESTIGACIÓN CIENTÍFICA Y TECNOLÓGICA

Implementación del protocolo SS7 sobre conexiones entre dos servidores Asterisk utilizando los equipos SDH del Laboratorio de Telecomunicaciones

Alejandro Azú Campoverde [(1)], Rafael Jiménez Ferrerosa [(2)], Gabriel Astudillo Brocel [(3)]
Facultad de Ingeniería en Electricidad y Computación (FIEC)
Escuela Superior Politécnica del Litoral (ESPOL)
Campus Gustavo Galindo, Km 30.5 vía Perimetral
Apartado 09-01-5863. Guayaquil, Ecuador
aazu@fiec.espol.edu.ec [(1)], rkjimene@fiec.espol.edu.ec [(2)]
Escuela Superior Politécnica del Litoral (ESPOL) [(3)], Ingeniero en Electrónica y Telecomunicaciones [(3)],
gastudil@fiec.espol.edu.ec [(3)]

Resumen

El proyecto consistió en la implementación del protocolo SS7 sobre conexiones E1 por medio de 2 servidores Asterisk, utilizando para su efecto los equipos SDH que existen en el laboratorio de Telecomunicaciones, realizándose hasta 30 llamadas simultaneas, que es la capacidad que tiene un E1.

La implementación de este proyecto sirvió para demostrar las ventajas que se tienen al implementar el protocolo SS7; así también como administrar y garantizar de forma eficiente tanto los recursos económicos como tecnológicos que nos ofrece un E1, para el uso empresarial.

De esta manera, se presentaron tres partes esenciales para la realización del proyecto: equipos SDH, hardware y software. En cuanto a los equipos SDH, se trabajó con los HUAWEI OPTIX 1500B, estos tienen comunicación por medio de fibra óptica y cada uno nos provee de un E1. El hardware utilizado en cada servidor es la tarjeta Digium TE205P, la cual permite la conexión con el E1 y a la vez interactúa con Asterisk. En cuanto a la parte del software, se utiliza el protocolo SS7 para la configuración correcta entre los equipos SDH, el hardware y el software, garantizando el entorno del software libre.

Palabras Claves: *Asterisk, SS7, SDH, E1, Tarjeta E1*

Abstract

The project involved the implementation of the SS7 protocol on E1 connections via two Asterisk servers, using its effect SDH equipment that exist in the Telecommunication Laboratory, performing up to 30 simultaneous calls, which is the ability of an E1.

The implementation of this project served to demonstrate the benefits that have to implement the SS7 protocol, and also how to manage efficiently and to ensure both economic and technological resources that offers an E1, for business use.

Thus, there were three essential parts to the project: SDH equipment, hardware and software. As for the SDH equipment, we worked with OPTIX HUAWEI 1500B, these are interconnected by optical fiber and each provides us with an E1. The hardware used in each server is the Digium TE205P card, which allows the connection to the E1 and simultaneously interact with Asterisk. As part of the software, SS7 protocol is used for the correct settings between SDH equipment, hardware and software, ensuring the environment of free software.

1. Introducción

La tecnología ha avanzado rápidamente a lo largo de los años innovando la comunicación entre los seres humanos. Dentro de estos grandes logros se encuentra la tecnología VOIP (Protocolo de Voz sobre IP), el cual hace posible que la señal de voz viaje a través de la red empleando el protocolo IP (Protocolo de Internet). Esto quiere decir que la señal de voz se envía en forma digital por paquetes en lugar de las formas tradicionales (analógica) por medio de una compañía telefonía convencional. El tráfico de Voz sobre IP se puede transmitir por cualquier red IP, es decir, que la voz y los datos se transmiten de forma similar que un correo electrónico.

La principal contribución del proyecto es demostrar la utilidad que tiene la utilización del protocolo SS7 sobre conexiones E1, las mismas que nos van a proveer 30 canales de voz simultáneamente por medio de los equipos SDH, aprovechando los beneficios que nos brindan los mismos.

El proyecto constó de dos servidores Asterisk, en cada uno fué instalada una tarjeta TE205P, la misma que se conectó con el equipo SDH de Huawei OptiX OSN 1500B, el cual se conectaba con el otro equipo SDH por medio de fibra óptica y el mismo que va a ser administrado por el programa de gestión de los equipos.

Como objetivo principal se deseó lograr conectar estos equipos para establecer la conexión entre ellos, de tal manera que se pueda ejecutar y avanzar con éxito las pruebas del proyecto.

2. Metodología

Para el íntegro funcionamiento de la comunicación que se efectuó entre el protocolo SS7 sobre los enlaces E1 que tienen los servidores Asterisk, se debía realizar la siguiente configuración en los equipos a utilizar, a continuación los detalles:

1. Instalación del sistema operativo Centos en las PCs.
2. Instalación de Asterisk y de cada uno de los paquetes adicionales requeridos para la implementación del proyecto los cuales son:
    - Libpri
    - DahdiLinux
    - Dahditools
    - AsteriskAddons
    - Chan_ss7
3. Instalación de la tarjeta digital TE205P.
4. Configuración del archivo ss7.conf
5. Configuración del archivo system.conf
6. Configuración del archivo extensions.conf
7. Configuración del archivo sip.conf
8. Ejecución de Asterisk.
9. Configuración los equipos SDH del laboratorio de Telecomunicaciones.
10. Etapa de pruebas de llamadas sobre enlace SDH.

3. Descripción del Proyecto

El proyecto se implementó con dos servidores Asterisk, que tenían instalado el sistema operativo Centos, cada uno de ellos con una tarjeta digital Digium TE205P, utilizando un puerto de dicha tarjeta, del cual nosotros aprovecharemos una conexión para enlace E1 que nos brinda, proporcionándonos hasta un máximo de 30 canales, ya sea para voz o datos y así poder comunicarnos con el mundo exterior por medio de SS7. Cada E1 lo obtenemos de los equipos SDH Huawei OptiX OSN 1500B, el cual tiene comunicación por medio de fibra siendo habilitados por la gestión de sistema que el mismo laboratorio de telecomunicaciones posee.

En la **Figura 1**, se puede observar la manera cómo se colocaron cada uno de los equipos a utilizados para la implementación del proyecto.

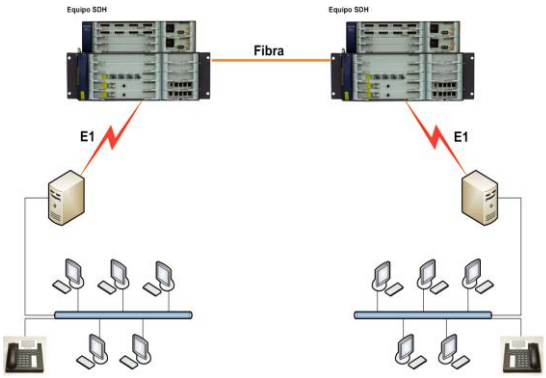


Figura 1. Esquema de la implementación

4. Asterisk y SS7

Asterisk es un software tipo PBX, lo que significa que funciona como una central secundaria privada automática, se puede obtener a partir de su empleo

una central telefónica conectada directamente a la red pública de teléfono por medio de líneas troncales para gestionar, además de las llamadas internas, las entrantes y/o salientes con autonomía sobre cualquier otra central telefónica. Está diseñado originalmente para que funcione con Linux, pero trabaja muy bien con BSD, Windows (emulado) y OS X. Funciona a partir del protocolo IP y puede interactuar con casi todo el equipo de telefonía basado en los estándares usando un hardware relativamente económico. Provee servicios voicemail (correo de voz), comunicación directa, identificación de llamadas, respuesta de voz interactiva y llamada en espera. Para ello emplea servicio de llamadas ID con los protocolos SIP, H323, ADSI, IAX, SS7, entre otros.

SS7 es el estándar de la tecnología conocida como señalización de canal común (CCS, pos sus siglas en inglés), que consiste en el uso de un canal diferente al canal de voz, destinado únicamente a la señalización. Esta separación permite que, por un lado, se tenga un canal que lleve la conversación y, por otro, un canal que contenga la señalización, de manera que ambos se lleven a cabo de manera independiente. Por este motivo, SS7 es un sistema de señalización fuera de banda que se caracteriza por la transmisión de paquetes de datos a alta velocidad y por la posibilidad de permitir la señalización entre diferentes elementos de la red, entre los cuales no se tiene una comunicación directa.

5. Implementación del proyecto

La solución presentada se basó en el software libre Asterisk que permite implementar centrales telefónicas a pequeña, mediana y gran escala, que por ser de libre acceso lo utilizamos en forma gratuita. Además tiene muchos addons que nos sirven para ampliar los usos de dicho software y encontrar recursos para solucionar toda clase de problemas como lo resolverían las centrales telefónicas privadas, con la diferencia que ahorramos el costo del equipo y tenemos un mayor control en el manejo de llamadas.

5.1 Requerimientos para iniciar el proyecto en Asterisk

5.1.1 Hardware

Las características al momento de seleccionar un servidor puede ser considerada fácil o complicada, esto depende del presupuesto del usuario. De tal manera se debe tener en cuenta el diseño general del sistema a implementar y las funcionalidades que requerirá, esto ayudará a determinar la marca, modelo del CPU, tarjeta madre, y fuente de energía.

Los requisitos de hardware que se configuraron en cada uno de los servidores para el desarrollo del proyecto, son los siguientes:

- Procesador arquitectura x86 de 2.6 GHz con 800 MHz FSB
- 1 GB RAM DDR400
- 80 GB en disco duro
- Tarjeta de red 10/100 Mbps
- Tarjeta Digium Dual E1/T1 card

5.1.2 Software

Para poder gozar de todo lo que nos brinda Asterisk tuvimos que bajar, extraer, compilar e instalar los paquetes nombrados a continuación:

- Libpri
- Dahdilinux
- Dahditools
- Asterisk
- AsteriskAddons
- Chan_ss7

Todos estos paquetes se los guardó en /usr/scr
Luego de esto se comenzó a extraer por medio de tar –xvzf.

Una vez hecho esto se procedió a compilar cada archivo que está en cada carpeta.

5.1.3 Generalidades DAHDI

DAHDI es compatible con versiones de Asterisk mayores a 1.4.22 reemplaza a los módulos del kernel Zaptel. El propósito principal de la versión 2.0.0 es la incorporación de apoyo de BRI.

Entre las características de DAHDI, tenemos:

- Supresores de eco, pueden ser aplicados ahora por canales y seleccionados en la configuración del tiempo.
- Cambios en la asignación de memoria del canal de un gran bloque dentro de un bloque pequeño en orden a reducir errores de memoria a un sistema que ha estado corriendo para algún tiempo.
- Cambios en el diseño para soporte de paquetes binarios.

5.2 Configuración de Archivos del Proyecto Asterisk

En el servidor A y B se procedió a realizar cambios en los archivos de configuración sip.conf, extensions.conf, ss7.conf y system.conf, cabe recalcar que el servidor A es aquel que va a recibir las llamadas que se van a realizar desde el servidor B.

5.2.1 Configuración SIP.CONF

/etc/asterisk/sip.conf
El protocolo SIP se utiliza para configurar las extensiones que se van a requerir en el proyecto.

El archivo sip.conf tiene tres estructuras:
General: donde hay que definir la configuración general de nuestras extensiones.

Central: donde configuraremos el registro a nuestros proveedores VoIP (y, si queremos, los datos para conectar entre ellos distintos servidores Asterisk).

Final: donde se configura las extensiones internas y externas.

5.2.1.1 Configuración general

[general]
Etiqueta que introduce la parte general de la configuración

context=default
Permite hacer búsquedas de registros DNS SRV para llamadas SIP salientes basadas en los nombres de dominio
svrlookup=yes
Permite hacer búsquedas de registros DNS SRV para llamadas SIP salientes basadas en los nombres de dominio.

language=es
Si hemos instalado locuciones en más de un idioma, aquí podemos definir cual idioma usará la extensión. En este caso se instalaron las locuciones en español, y se especifica con el prefijo es.

5.2.1.2 Configuración extensiones

[xxxx]
Número de la extensión

type=friend
Tipo de extensión. Puede ser friend, user o peer. Friend puede hacer y recibir llamadas, user solo recibir y peer solo puede hacer (como en el caso de proveedores VoIP que usamos solo para hacer llamadas)

secret=xxxx
Define la contraseña de la extensión

qualify=yes
Determina el tiempo de respuesta de una extensión y si está alcanzable o no

nat=no
Si la extensión se conecta al servidor asterisk detrás de un firewall hay que poner yes, *caso contrario no.*

host=dynamic
si la extensión se conecta remotamente cambiando continuamente su dirección IP se pone este parámetro

canreinvite=no
Yes si queremos que la extensión intente conectarse directamente con la extensión llamada. No si queremos que Asterisk haga de puente entre las dos extensiones.

context=internal
El contexto que usará la extensión

A continuación se muestra la configuración final del archivo sip.conf, tanto en el servidor A como en el servidor B.

```
[general]

context=default
srvlookup=yes
language=es

[1001]
type=friend
secret=1001
qualify=yes
nat=no
host=dynamic
canreinvite=no
context=internal
```

(a)

```
[general]

context = default
srvlookup = yes
languaje = es

[2001]
type = friend
secret = 2001
qualify = yes
nat = no
host = dynamic
canreinvite = no
context = internal
```

(b)

Figura 2. Configuración del archivo sip.conf en servidor A (a) y B (b)

5.2.2 Configuración EXTENSIONS.CONF

/etc/asterisk/extensions.conf

Este archivo contiene el plan de marcado de la central telefónica para cada contexto y por tanto para cada usuario. El plan de marcado tiene 4 definiciones fundamentales: contexto, extensiones, prioridades y aplicaciones.

5.2.2.1 Contexto general

[general]

Se establecen configuraciones generales. Que se aplica al resto de contexto.

static=yes

Se lo puede indicar en el archivo, como no se lo puede poner. Indica si se ha de hacer caso a un comando "save dialplan" desde la consola. Por defecto es "yes".

5.2.2.2 Contexto internal

Es una instrucción que asterisk seguirá como consecuencia de una llamada entrante o por dígitos marcados en un canal activo, en este contexto se establece el dial plan para las extensiones internas que se conectan mediante el protocolo SIP.

exten => 1001,1,Dial(SIP1001,10,r)

Define el canal de salida SIP para las extensiones, con su propio nombre. De esta manera se tiene configurado la extensión 1001 y 1002. Luego del nombre tenemos la prioridad y por último la aplicación que se va a ejecutar, es decir una acción en la llamada.

exten => 1003,1,Answer()

Forma para que conteste.

exten => 1003,2,Wait()

Para que espere una minima cantidad de tiempo, en caso de que este ocupado.

exten => 1003,3,Hangup()

Se colgara, así terminado la llamada.

Luego se agregan 30 extensiones más, con el mismo formato que la 1003, de manera que tendríamos los 30 canales disponibles en un E1.

exten => _2XXX, 1, Dial(DAHDI/g1/${EXTEN},10,r)

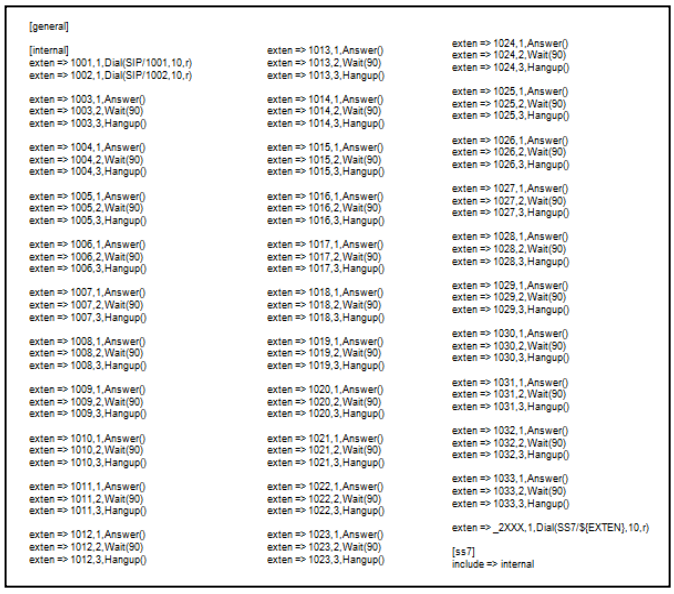
Se configuran las extensiones de destino al cual queremos llamar conformado con la tecnología DAHDI, seguido de la fuente remota, que en nuestro caso es g1 al cual se refiere a los 30 canales que están disponibles, luego tenemos ${EXTEN} para que Asterisk guarde la variable que hemos marcado, le sigue el tiempo de espera, que van a ser 10 segundos y el último argumento es una función para modificar el comportamiento de Dial a lo que ponemos la letra r para que el llamante escuche el tono de timbrado.

[from_B]

De esta manera acepta las llamadas que vienes del servidor B.

include => internal

Usara el contexto internal.



```
[general]

[internal]
exten => 1001,1,Dial(SIP/1001,10,r)
exten => 1002,1,Dial(SIP/1002,10,r)

exten => 1003,1,Answer()
exten => 1003,2,Wait(90)
exten => 1003,3,Hangup()

exten => 1004,1,Answer()
exten => 1004,2,Wait(90)
exten => 1004,3,Hangup()

exten => 1005,1,Answer()
exten => 1005,2,Wait(90)
exten => 1005,3,Hangup()

exten => 1006,1,Answer()
exten => 1006,2,Wait(90)
exten => 1006,3,Hangup()

exten => 1007,1,Answer()
exten => 1007,2,Wait(90)
exten => 1007,3,Hangup()

exten => 1008,1,Answer()
exten => 1008,2,Wait(90)
exten => 1008,3,Hangup()

exten => 1009,1,Answer()
exten => 1009,2,Wait(90)
exten => 1009,3,Hangup()

exten => 1010,1,Answer()
exten => 1010,2,Wait(90)
exten => 1010,3,Hangup()

exten => 1011,1,Answer()
exten => 1011,2,Wait(90)
exten => 1011,3,Hangup()

exten => 1012,1,Answer()
exten => 1012,2,Wait(90)
exten => 1012,3,Hangup()

exten => 1013,1,Answer()
exten => 1013,2,Wait(90)
exten => 1013,3,Hangup()

exten => 1014,1,Answer()
exten => 1014,2,Wait(90)
exten => 1014,3,Hangup()

exten => 1015,1,Answer()
exten => 1015,2,Wait(90)
exten => 1015,3,Hangup()

exten => 1016,1,Answer()
exten => 1016,2,Wait(90)
exten => 1016,3,Hangup()

exten => 1017,1,Answer()
exten => 1017,2,Wait(90)
exten => 1017,3,Hangup()

exten => 1018,1,Answer()
exten => 1018,2,Wait(90)
exten => 1018,3,Hangup()

exten => 1019,1,Answer()
exten => 1019,2,Wait(90)
exten => 1019,3,Hangup()

exten => 1020,1,Answer()
exten => 1020,2,Wait(90)
exten => 1020,3,Hangup()

exten => 1021,1,Answer()
exten => 1021,2,Wait(90)
exten => 1021,3,Hangup()

exten => 1022,1,Answer()
exten => 1022,2,Wait(90)
exten => 1022,3,Hangup()

exten => 1023,1,Answer()
exten => 1023,2,Wait(90)
exten => 1023,3,Hangup()

exten => 1024,1,Answer()
exten => 1024,2,Wait(90)
exten => 1024,3,Hangup()

exten => 1025,1,Answer()
exten => 1025,2,Wait(90)
exten => 1025,3,Hangup()

exten => 1026,1,Answer()
exten => 1026,2,Wait(90)
exten => 1026,3,Hangup()

exten => 1027,1,Answer()
exten => 1027,2,Wait(90)
exten => 1027,3,Hangup()

exten => 1028,1,Answer()
exten => 1028,2,Wait(90)
exten => 1028,3,Hangup()

exten => 1029,1,Answer()
exten => 1029,2,Wait(90)
exten => 1029,3,Hangup()

exten => 1030,1,Answer()
exten => 1030,2,Wait(90)
exten => 1030,3,Hangup()

exten => 1031,1,Answer()
exten => 1031,2,Wait(90)
exten => 1031,3,Hangup()

exten => 1032,1,Answer()
exten => 1032,2,Wait(90)
exten => 1032,3,Hangup()

exten => 1033,1,Answer()
exten => 1033,2,Wait(90)
exten => 1033,3,Hangup()

exten => _2XXX,1,Dial(SS7/${EXTEN},10,r)

[ss7]
include => internal
```

(a)

```
[internal]

exten => 2001,1,Dial(SIP/2001,10,r)
exten => 2002,1,Dial(SIP/2002,10,r)

exten => _1XXX,1,Dial(SS7/${EXTEN},10,r)
exten => 123,1,System(/etc/asterisk/script)


[ss7]

include => internal
```

(b)

Figura 3. Configuración del Archivo extensions.conf

Servidor A (a) y B (b)

5.2.3. Configuración SS7.CONF

En cada equipo, Asterisk se debe instalar y configurar apropiadamente. Es una buena idea conectar a por lo menos uno de los equipos un teléfono (SIP), para hacer la prueba más fácil, pero es también posible utilizar otros medios como softphones para realizar las pruebas correspondientes. Lo importante es que haya una manera de generar llamadas en uno de los equipos (o en ambos).

Ahora crear el archivo de configuración para chan_ss7, en:

/etc/asterisk/ss7.conf

La configuración de este archivo contiene:

5.2.3.1 Contexto Linkset-Siuc

[linkset -siuc]

En este contexto se establecen las configuraciones generales que se aplican al resto del mismo.

enabled =>yes
Aquí se habilita el contexto *linkset-siuc.*

enable_st => no
El final de la pulsación (ST) se utiliza para determinar cuando la dirección de entrada es completa

use_connect => yes
Respuesta de llamada entrante con la CON en lugar de la ACM y ANM.

hunting_policy => even_mru
Esto establece la política de la caza, es decir, el algoritmo utilizado para asignar un circuito para las llamadas salientes. Esto se debe configurar de forma adecuada en cada extremo del enlace SS7 para minimizar el riesgo de colisión al llamar.

context => ss7
El contexto asterisk para las llamadas entrantes en este linkset. El valor que por defecto se va a usar es ss7.

language => es
El lenguaje de asterisk para las llamadas entrantes en este linkset.

subservice => auto

5.2.3.2 Contextos Link-L1 y Link-L2

[link-l1], [link-l2]

En estos contextos se van a establecer los canales que serán usados para voz y cuál va a ser el canal a usarse para señalización.

linkset => siuc
Quiere decir que este contexto pertenece a linkset SIUC.

channels => 1-15,17-31
El conjunto de canales de audio/voz en este contexto.

schannel => 16
El canal de señalización es el 16.

firstcic => 1
Indica que el primer código de identificación del circuito (CIC) en el campo de datos del cuerpo del mensaje va a ser el 1.

enabled => yes
Aquí se habilita el contexto *link-l1 o link-l2.*

5.2.3.3 Contextos host-wrks129-213fiec y host-wrks129-214fiec

[host-wrks129-213fiec], [host-wrks129-214fiec]

chan_ss7 auto-configura haciendo coincidir el nombre de host con la sección de máquinas host <name> en el archivo de configuración, en este caso puede ser 'wrks19-228fiec' o 'wrks19-211fiec'. Por lo tanto, el archivo de configuración se puede utilizar en varios equipos.

enabled => yes
Aquí se habilita el contexto *host-wrks129-213fiec o host-wrks129-214fiec.*

opc => 0x1, 0x2
El punto de código para este punto de señalización SS7 es 0x1(para el servidor A) o 0x2 (para el servidor B).

dpc => siuc:0x2 o 0x1
El punto de destino (par) es el código de 0x2 (para el servidor A) o 0x1 (para el servidor B) linkset SIUC.

links => l1:1, l2:1

Los enlaces en el host es "l1", conectado a / span conector # 1(para el servidor A) o el host es "l2", conectado a / span conector # 1(para el servidor B).

Esto configura al equipo A con el código de punto de destino número 1, y al equipo B con el código de punto de destino número 2 (puesto que esto es una red privada cerrada SS7, éstos códigos de punto falsos son aceptables). Se configuran 30 circuitos de voz (el timeslot 16 se utiliza para señalización y no está disponible para voz). Las llamadas entrantes llegan al dialplan en el contexto "**ss7**" en ambos equipos.

En la **Figura 4**, se puede visualizar el archivo de configuración de SS7.

```
[linkset-siuc]
enabled => yes
enable_st => no
use_connect => yes
hunting_policy =>
even_mru
context => ss7
language => es
subservice => auto

[link-l1]
linkset => siuc
channels => 1-15,17-31
schannel => 16
firstcic => 1
enabled => yes

[link-l2]
linkset => siuc
channels => 1-15,17-31
schannel => 16
firstcic => 1
enabled => yes

[host-wrks129-213fiec]
enabled => yes
opc => 0x1
dpc => siuc:0x2
links => l1:1

[host-wrks129-214fiec]
enabled => yes
opc => 0x2
dpc => siuc:0x1
links => l2:1
```

Figura 4. Configuración del Archivo ss7.conf Servidor A y B

5.2.4 Configuración SYSTEM.CONF

/etc/dahdi/system.conf

En este archivo se configuran todos los parámetros requeridos para las tarjetas E1, es instalado por el paquete dahdi.

La configuración de este archivo contiene
span=(spannum),(timing),(LBO),(framing),
(coding)
spannum= Numero del span. Esto comienza en 1.

timing= Para sincronizar el tiempo de los dispositivos.
0: no usa este span como fuente de sincronizacion; envía tiempo de sincronización a otro terminal.
1: Usa como fuente primaria de sincronización.
2: establece como secundario

LBO= Line Build Out – Largo del cable entre la tarjeta y modem proveedor SmartJack/telco. Casi simpre deberia de ser 0. Esta distancia no incluye el cobre en la calle a el CO/exchange.
0: 0 dB (CSU) / 0 - 133 feet (DSX-1)
1: 133 - 266 feet (DSX-1)
2: 266 - 399 feet (DSX-1)
3: 399 - 533 feet (DSX-1)
4: 533 - 655 feet (DSX-1)
5: -7.5 dB (CSU)
6: -15 dB (CSU)
7: -22.5 dB (CSU)

Framing (tramado)= Establece como va a ser la comunicación entre el hardware con el otro terminal de la línea. Para E1, es CAS o CCS, en nuestro proyecto usamos CCS.

Coding (codigo)= Otro parámetro de comunicación entre el terminal de línea con el hardware.

Para E1: El código es AMI or HDB3 (E1 puede necesitar crc4). En este caso utilizamos hdb3.

Bchannels= los canales b por donde se va a ir la información, por tanto los únicos canales que transmiten son 1-15 y del 17-31

Dchannels= Los canales d son para señalización, es por esta razón que el canal 16 está reservado para esta función específica.

Loadzone= es el idioma
Defaultzone= el idioma por default

```
span=1,0,0,ccs,hdb3
bchan=1-31
```

(a)

```
span=1,1,0,ccs,hdb3
bchan=1-31
```

(b)

Figura 5. Configuración del archivo system.conf Servidor A (a) y B (b)

6. CONFIGURACIÓN DE EQUIPOS SDH

Los laboratorios de Telecomunicaciones utilizan los equipos Huawei OptiX 1500 con la topología anillo, garantizando una buena operación y mantenimiento de estos.

En términos del nivel de protección de la red, el OptiX OSN 1500B tiene dos fibras de sección de protección múltiple (MSP) anillo.

Los anillos de fibra que se encuentran en el laboratorio constan de tres equipos llamados FIEC1, FIEC2 y FIEC3, los cuales se gestionan a través del servidor T2000, el cual nos informa de las novedades que se presentan en los equipos. Como se aprecia en la **Figura 6** esta es la topología anillo con la que trabaja el laboratorio.

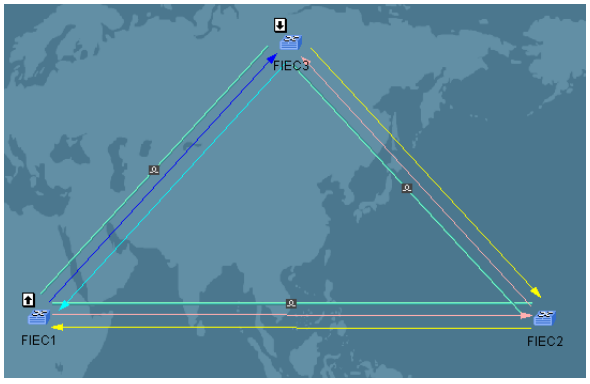


Figura 6. Topología anillo del laboratorio de Telecomunicaciones

6.1 CREACIÓN DE SERVICIO (RUTA)

Primero se determinó entre que equipos se iba a realizar el servicio es decir cuál iba a ser la fuente (source) y cuál iba a ser el destino (sink), en este caso se iba a trabajar con los equipos FIEC1 y FIEC3 como se visualiza en la **Figura 7**.

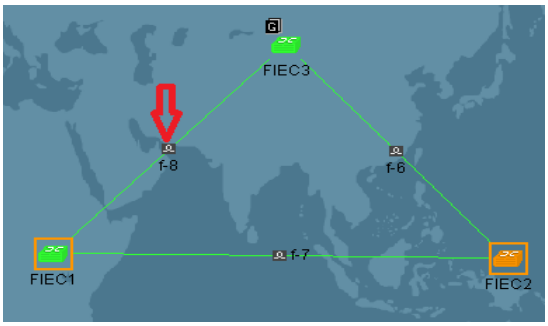


Figura 7. Selección del enlace sobre el cual se va a trabajar

Una vez seleccionado el enlace se hizo click derecho y se seleccionó la opción **Query Relevant Trails**. Aquí se muestran los servicios (caminos) que fueron creados anteriormente (véase la **Figura 8**).

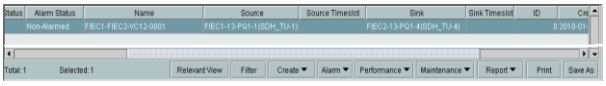


Figura 4.3 Servicios que ya han sido creados

Se procedió a hacer click en el botón **Create** y se seleccionó la opción **Create Trail.**

Tal como se muestra en la **Figura 9**, se va a mostrar una pantalla en la cual se puede visualizar cada uno de las tarjetas que tiene instaladas el equipo SDH.

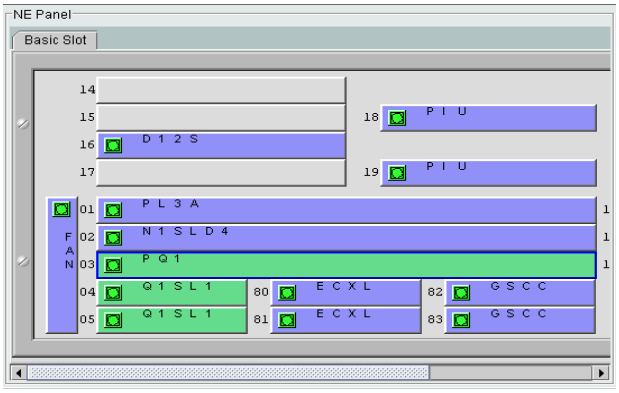


Figura 9 Tarjetas que posee el equipo SHD

Se seleccionó la Tarjeta PQ1 puesto que esta es la que nos va a permitir crear el servicio entre la fuente y el destino. Al seleccionar dicha tarjeta, se nos van a mostrar todos los puertos que la misma contiene

(**Figura 10**); en este caso, el laboratorio de telecomunicaciones cuenta sólo con cuatro puertos hábiles para trabajar, así que se tuvo que escoger únicamente los puertos habilitados para poder configurar el servicio tanto para la fuente como para el destino.

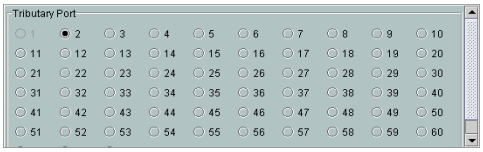


Figura 10 Puertos que posee la tarjeta PQ1

Como se visualiza en la **Figura 10** el puerto 1 ya ha sido utilizado así que se procedió a escoger un puerto disponible (puerto dos) para realizar el camino; se hizo click en **OK** y **c**on esto ya se creó la fuente (source); de la misma manera se tuvo que realizar la configuración para el destino (sink). Finalmente al tener los dos puertos configurados se hizo click en **Apply** y como se visualiza en la **Figura 11** se va a poder apreciar la nueva creada que en este caso fué:

Fuente (Source): FIEC1-13-PQ1-2(SDH_TU-2)
Destino (Sink): FIEC3-13-PQ1-1(SDH_TU-1)

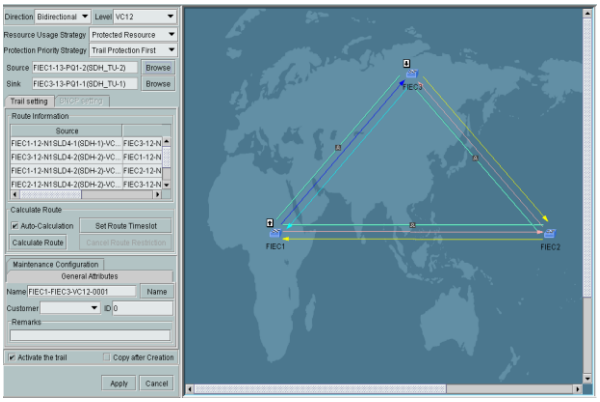


Figura 11 Nuevo servicio creado entre FIEC1 y FIEC3

7. FUNCIONAMIENTO Y PRUEBAS DEL PROYECTO

7.1 Pruebas con equipos SDH

Una vez establecida la conexión, por medio del cable T1 crossover, se procedió a establecer comunicación con los equipos SDH del laboratorio de telecomunicaciones.

Para conectar la tarjeta TE205P con los equipos SDH se utilizó un cable RJ45 directo, previamente a la conexión entre servidores y equipos, se debe de verificar con el servidor T2000, los canales que se encuentran habilitados.

Es asi, que viendo el sistema de gestion en el servidor de los equipos, se ve que estan habilitados FIEC1 en el puerto 1 y FIEC2 en el puerto 4.

Alarm Status	Name	Source	Source Timeslot	Sink
Non-Alarmed	FIEC1-FIEC2-VC12-0001	FIEC1-13-PQ1-1(SDH_TU-1)		FIEC2-13-PQ1-4(SDH_TU-4)

Figura 12 Gestión de los equipos SDH Huawei

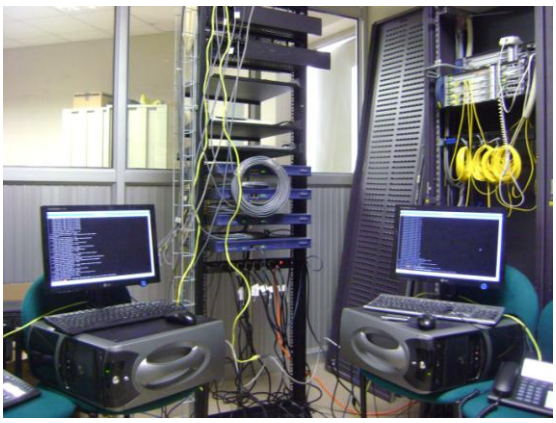


Figura 13 Conexión de los servidores con los equipos SDH

Se verificó la correcta comunicación entre los equipos SDH con los servidores, ingresando en el servidor B el siguiente comando:

originate SS7/1001 application echo

```
wrks19-228fiec*CLI>
    -- Recv IAM CIC=30   ANI=2001 DNI=1001 RNI= redirect=no/0 complete=1
    -- Executing [1001@ss7:1] Dial("SS7/siuc/30", "SIP/1001|10|r") in new stack
    -- Called 1001
    -- SIP/1001-0845b338 is ringing
    -- SIP/1001-0845b338 answered SS7/siuc/30
wrks19-228fiec*CLI>
```

Figura 14 Recepción de la llamada en el servidor A, proveniente de B.

Comprobado que existía comunicación entre servidores y equipos SDH tan solo con un canal del E1 (como se visualiza en la **Figura 14**), se procedió a

utilizar todos los canales que nos provee, de tal manera que se generen las 30 llamadas simultaneas.

Para realizar esto se tuvo que realizar un script, el cual se lo ubicó en la carpeta de Asterisk, a continuación se presenta el documento del script (**Figura 15**):

/etc/asterisk/script

```
#!/bin/sh
a=1033

for ((i=1;i<32;i+=1));do
#echo "entra" $a
asterisk -rx "originate SS7/$a
application echo"
a=`expr $a - 1`
#echo $a
done
```

Figura 15 Script para generar 30 llamadas simultaneas

Ya que no se contaba con la cantidad de teléfonos requeridos para la realización de las llamadas, se utilizó este archivo que tiene una pequeña programación que nos va a servir para poder efectuar las 30 llamadas de una manera simultánea.

Luego de la creación del script lo único que se debió hacer fué realizar las 30 llamadas simultáneas, digitando desde el servidor B en el teléfono VOIP el número registrado, el cual es **123**.

Figura 16 Pantalla del servidor A, recibiendo llamadas desde el servidor B

Luego de los 90 segundos, que es lo que se ha configurado en el archivo extensions.conf del servidor A, se procedieron a cerrar de forma automáticamente cada llamada que se estableció.

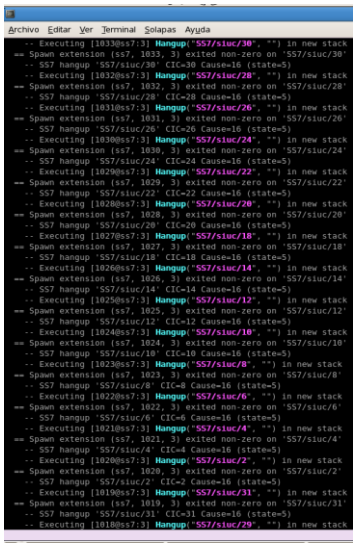


Figura 17. Cierre de la llamadas pasado los 90 segundos, en el Servidor A

8. CONCLUSIONES

- El software libre provee de herramientas útiles y adaptables al entorno, gracias a su soporte y su universo que se encuentran en continuo crecimiento y mejoramiento, sin costo alguno.

- A través de múltiples pruebas e implementación final, se pudo demostrar que el proyecto tiene la viabilidad y efectividad de servidores Asterisk junto con el protocolo SS7, los equipos SDH y las conexiones E1, permitiendo la correcta comunicación entre ellos aprovechando el código abierto distribuido a través del software Asterisk.

- Asterisk es un buen recurso a implementar en una empresa debido a que posee características fáciles de administrar y personalizar además de los costos de implementación que van a resultar mucho más económicos que los que comúnmente se suele invertir en la creación de una central telefónica genérica.

- SS7 efectivamente usa un canal diferente al canal de voz, destinado únicamente a la señalización permitiendo así que el canal que

lleva la conversación y el canal que lleva la señalización se lleven a cabo de manera independiente.

9. RECOMENDACIONES

- A la hora de instalar la tarjeta TE205P en la Pc, es necesario tomar en consideración que los jumpers estén correctamente configurados para que funcione a la conveniencia del usuario, es decir como E1 o bien como T1.

- Tomar muy en cuenta el color que nos indica el led de la tarjeta TE205P, ya que si es de color rojo quiere decir que no se están entendiendo los equipos. El color verde indica éxito en la comunicación.

- Verificar en la gestión de los equipos SDH el camino previamente creado y los puertos disponibles en el rack para luego conectar físicamente, de la manera correcta con los puertos de los servidores Asterisk.

- Comprobar que cada uno de los archivos de configuración utilizados para la realización del proyecto estén debidamente conformados para luego no tener problemas al momento de realizar las pruebas del mismo.

- Se aconseja seguir trabajando en la implementación de proyectos en los cuales se exploten aún más los beneficios que nos proporcionan Asterisk y SS7 puesto que las mismas son herramientas revolucionarias que nos van a brindar apoyo en la creación y administración de una central telefónica la misma que podría brindar múltiples tipos de servicios como por ejemplo: buzón de voz, call center, reconocimiento de llamadas; pudiéndose así implementar la misma en una empresa especializada en servicios de telefonía como Porta, Movistar, Alegro, entre otras .

10. REFERENCIAS

[1] HUAWEI, OptiX OSN 1500 Intelligent Optical Transmission System V100R008, 2007

[2] VOIP-INFO, Configuración de la tarjeta TE110P, http://www.voip.unam.mx/mediawiki/index.php/Instalaci%C3%B3n_y_Configuraci%C3%B3n_de_la_Tarjeta_TE110P, 2010

[3] VOIP-INFO, Archivo system.conf, http://www.voip-info.org/wiki/view/system.conf, 2010

[4] VOIP-INFO, Archivo ss7.conf, http://voip.megawan.com.ar/doku.php/asterisk_configuracion_ss7.conf, 2010

[5] VOIP-INFO, Archivo chan_ss7.conf, http://voip.megawan.com.ar/doku.php/asterisk_configuracion_chan_ss7_guia_rapida, 2010

[6] WIKIPEDIA, E1, http://es.wikipedia.org/wiki/E1, 2010

[7] WIKIPEDIA, Sistema de señalización No.7, http://es.wikipedia.org/wiki/Sistema_de_se%C3%B1alizaci%C3%B3n_por_canal_com%C3%BAn_n.%C2%BA_7, 2010

[8] CINIT, SS7 , http://www.cinit.org.mx/articulo.php?idArticulo=7, 2010