

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL



Facultad de Ingeniería en Electricidad y Computación

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

INFORME DE MATERIA DE GRADUACIÓN

Previa a la obtención del título de:

INGENIERO EN TELEMÁTICA

Presentada por:

ALEJANDRO FERNANDO AZÚ CAMPOVERDE

RAFAEL KARÍM JIMÉNEZ FERREROSA

GUAYAQUIL - ECUADOR

Año: 2010

AGRADECIMIENTO

A Dios

A nuestros profesores

A nuestros padres

A nuestros hermanos

DEDICATORIA

A Dios, guía y luz de mi camino. A la memoria de mi abuelita Francisca que me inculcó siempre excelentes valores para ser una persona de bien, a mis padres y hermanos, por su amor y apoyo incondicional, siendo una parte fundamental para la elaboración de este trabajo.

Alejandro Azú Campoverde

A mis Padres que me supieron guiar y que gracias a sus consejos me formaron para la vida. A mis Hermanos que siempre estuvieron prestos a ayudarme. A Daniela que ha estado a mi lado y me ha apoyado en todo lo que me he propuesto lograr, y a todas aquellas personas que de una u otra manera intervinieron para lograr este objetivo.

Rafael Jiménez Ferrerosa

DECLARACION EXPRESA

“La responsabilidad del contenido de este Trabajo de Grado, nos corresponde exclusivamente; y el patrimonio intelectual de la misma a la Escuela Superior Politécnica del Litoral”.

(Reglamento de Graduación de la ESPOL)

Alejandro Fernando Azú Campoverde

Rafael Karím Jiménez Ferrerosa

TRIBUNAL DE SUSTENTACIÓN

Ing. Gabriel Astudillo Brocel

PROFESOR DE LA MATERIA DE GRADUACIÓN

Ing. Patricia Chávez Burbano

PROFESOR DELEGADO POR EL DECANO DE LA FACULTAD

RESUMEN

El proyecto a realizar consiste en la implementación del protocolo SS7 (Sistema de Señalización No.7) 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.

Con la realización de este proyecto se busca 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 presenta tres partes esenciales para la realización del proyecto: equipos SDH, hardware y software. En cuanto a los equipos SDH, se va a trabajar 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.

INDICE GENERAL

RESUMEN.....	VI
INDICE GENERAL.....	VII
INDICE DE FIGURAS.....	X
INTRODUCCION.....	XII
CAPITULO 1	
1. ANTECEDENTES Y JUSTIFICACION	
1.1. Antecedentes.....	2
1.2. Justificación.....	3
1.3. Descripción del Proyecto.....	3
1.3.1. Objetivo general.....	4
1.3.2. Objetivo específico.....	5
1.4. Metodología.....	5
1.5. Perfil de la Tesis.....	6
CAPITULO 2	
2. FUNDAMENTOS TEORICOS	
2.1. Asterisk.....	9
2.1.1. Funcionalidades de Asterisk.....	11
2.2. Sistema de Señalización No. 7 (SS7).....	12
2.2.1. Ventajas de la señalización SS7.....	13
2.2.2. Arquitectura.....	14
2.2.3. Capas del modelo SS7.....	15
2.2.3.1. MTP-1.....	16
2.2.3.2. MTP-2.....	16
2.2.3.3. MTP-3.....	17
2.2.3.4. SCCP.....	17
2.2.3.5. ISUP.....	18
2.2.3.6. PACT.....	18
2.2.3.7. OMAP.....	19
2.2.4. Apoyo SS7 en Asterisk.....	19
2.2.4.1. LIBISUP.....	19
2.2.4.2. SS7 Box.....	19
2.2.4.3. Chan_ss7.....	20
2.2.4.4. Libss7.....	20

2.3. Generalidades SDH	21
2.3.1. Ventajas.....	22
2.3.2. Desventajas.....	23
2.4. OPTIX 1500B.....	23
2.4.1. Funciones.....	24
2.5. Tarjeta PQ1.....	25
2.6. Generalidades E1.....	26
2.6.1 Estructura de E1.....	28
2.7. Tarjeta digital TE205P.....	29
2.7.1. Identificación de la tarjeta TE205P.....	31
2.7.2. Selección E1/T1.....	31

CAPITULO 3

3. IMPLEMENTACION DEL PROYECTO

3.1. Asterisk.....	34
3.1.1. Requerimiento para iniciar el proyecto en Asterisk.....	34
3.1.1.1. Hardware.....	34
3.1.1.2. Software.....	34
3.2. Generalidades DAHDI.....	37
3.3. Configuración de Archivos del proyecto Asterisk.....	38
3.3.1. Configuración SIP.CONF.....	38
3.3.1.1. Configuración general.....	39
3.3.1.2. Configuración extensiones.....	40
3.3.2. Configuración EXTENSIONS.CONF.....	42
3.3.2.1. Contexto general.....	42
3.3.2.2. Contexto internal.....	42
3.3.3. Configuración CHAN_SS7.CONF.....	46
3.3.3.1. Contexto Linkset-Siuc.....	46
3.3.3.2. Contextos Link-L1 y Link-L2.....	47
3.3.3.3. Contextos host-wrks129-213fiec y host-wrks129-213fiec.....	48
3.3.4. Configuración SYSTEM.CONF.....	51

CAPITULO 4

4. CONFIGURACION DE EQUIPOS SDH

4.1. Introducción.....	55
4.2. Creación de Servicio.....	56

CAPITULO 5

5. FUNCIONAMIENTO Y PRUEBAS DEL PROYECTO

5.1. Etapa de pruebas entre servidores.....	60
5.2. Inicializando con Asterisk.....	61
5.3. Etapas de pruebas con equipos SDH.....	64

CONCLUSIONES Y RECOMENDACIONES
GLOSARIO
BIBLIOGRAFÍA

INDICE DE FIGURAS

	Pag.
Figura 1.1 Esquema de la implementación del Proyecto.....	4
Figura 2.1 Software Asterisk para central telefónica.....	10
Figura 2.2 Arquitectura de SS7.....	15
Figura 2.3 Capas del modelo SS7.....	16
Figura 2.4 Apariencia y Estructura del OptiX OSN 1500B.....	26
Figura 2.5 Estructura de la trama E1.....	28
Figura 2.6 Tarjeta TE205P.....	31
Figura 2.7 Jumpers E1/T1.....	31
Figura 3.1 Configuración del Archivo sip.conf en Servidor A (a) y B (b).....	41
Figura 3.2 Configuración del Archivo extensions.conf Servidor A (a) y B (b).....	45
Figura 3.3 Configuración del Archivo chan_ss7.conf Servidor A (a) y B (b).....	50
Figura 3.4 Configuración del Archivo system.conf Servidor A (a) y B (b).....	53
Figura 4.1 Topología anillo del laboratorio de Telecomunicaciones.....	55
Figura 4.2 Selección del enlace sobre el cual se va a trabajar.....	56
Figura 4.3 Servicios que ya han sido creados.....	56
Figura 4.4 Tarjetas que posee el equipo SDH.....	57
Figura 4.5 Puertos que posee la PQ1.....	57
Figura 4.6 Nuevo servicio creado entre FIEC1 y FIEC3.....	58
Figura 5.1 Extremo de cable E1 crossover (Conector RJ-45).....	60
Figura 5.2 Color verde encendido en la tarjeta TE205P.....	61
Figura 5.3 Comando service dahdi restart.....	62
Figura 5.4 Canales habilitados por Asterisk.....	63
Figura 5.5 Servidor A, el cual acepta la llamada proveniente del servidor B.....	64
Figura 5.6. Gestión de los equipos SDH Huawei.....	65
Figura 5.7 Conexión de los servidores con los equipos.....	65
Figura 5.8 Recepción de la llamada en el servidor A, proveniente de B.....	66

Figura 5.9 Script para generar 30 llamadas simultaneas.....	66
Figura 5.10 Ejecución del script desde el servidor B.....	67
Figura 5.11 Pantalla del servidor A, recibiendo llamadas desde el servidor B.....	68
Figura 5.12 Cierre de la llamadas pasado los 90 segundos, en el Servidor A.....	69
Figura T.1 Conexión de 3 servidores Asterisk usando plataforma web.....	73

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 consta de dos servidores Asterisk, en cada uno va a estar instalada una tarjeta TE205P, la misma que va ser conectada con el equipo SDH de Huawei OptiX OSN 1500B, el cual se conecta 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 desea 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.

CAPÍTULO I

ANTECEDENTES Y JUSTIFICACIÓN

1.1. ANTECEDENTES

Toda llamada telefónica requiere de un modelo de señalización para que ésta sea establecida y mantenida. El envío del número telefónico, el tono de llamado o de ocupado y la información del número del que se llama son algunos ejemplos de señalización. La señalización, por lo tanto, permite el intercambio de información entre los componentes de una red telefónica para permitir la provisión y el mantenimiento de dicho servicio.

Debido a estas razones, SS7 se ha convertido en uno de los estándares internacionales más importantes de señalización en telefonía. Hacemos énfasis en la importancia de la señalización telefónica porque todo aquel que desee implementar servicios de VOIP en redes de cable, debería conocer el SS7, para garantizar la compatibilidad con la red telefónica tradicional.

La importancia de SS7 reside en la amplia gama de servicios que el usuario puede recibir y en la robusta arquitectura que lo respalda de la cual se hablará en capítulos posteriores, y que en conjunto, con los beneficios que nos brindan las redes SDH, las cuales actualmente soportan mayor cantidad de tráfico telefónico y de datos, que cuentan además, con un alto grado de estandarización garantizando calidad de servicio entre otras cualidades, con el fin de evolucionar para permitir esta demanda creciente de integración de servicios, operación y gestión; ayudándonos así entonces a tener grandes soluciones, tales como realizar llamadas telefónicas, proporcionar mensajería celular y lo más esencial el suministro de convergencia de servicios de voz y datos.

1.2 JUSTIFICACIÓN

El proyecto tiene como principal contribución, proveer de 30 canales de voz, los cuales nos permiten ejecutar llamadas entre ellos, por medio de la tecnología digital E1 conjuntamente implementados con el protocolo SS7 y así poder establecer la conexión con los equipos SDH, que se comunican por medio de fibra óptica, con el cual se podrá obtener la compatibilidad máxima entre diferentes suministradores y operadoras, brindándonos una red de una mayor flexibilidad, mejor aprovechamiento del ancho de banda potencial de la fibra óptica, y más capacidad de monitorización de la calidad y gestión centralizada.

Además de lo expuesto anteriormente, se debe añadir que la reducción de costos de los equipos de transmisión es altamente ventajosa para las empresas, las razones principales son la posibilidad de integrar las funciones de transmisión, multiplexación e interconexión en un solo equipo; y la alta competencia entre proveedores de equipos debida a la alta estandarización de SDH.

1.3 DESCRIPCION DEL PROYECTO

El proyecto va a ser implementado con dos servidores Asterisk, que tienen 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 1.1**, se puede observar la manera cómo van a estar colocados cada uno de los equipos a utilizar para la implementación del proyecto.

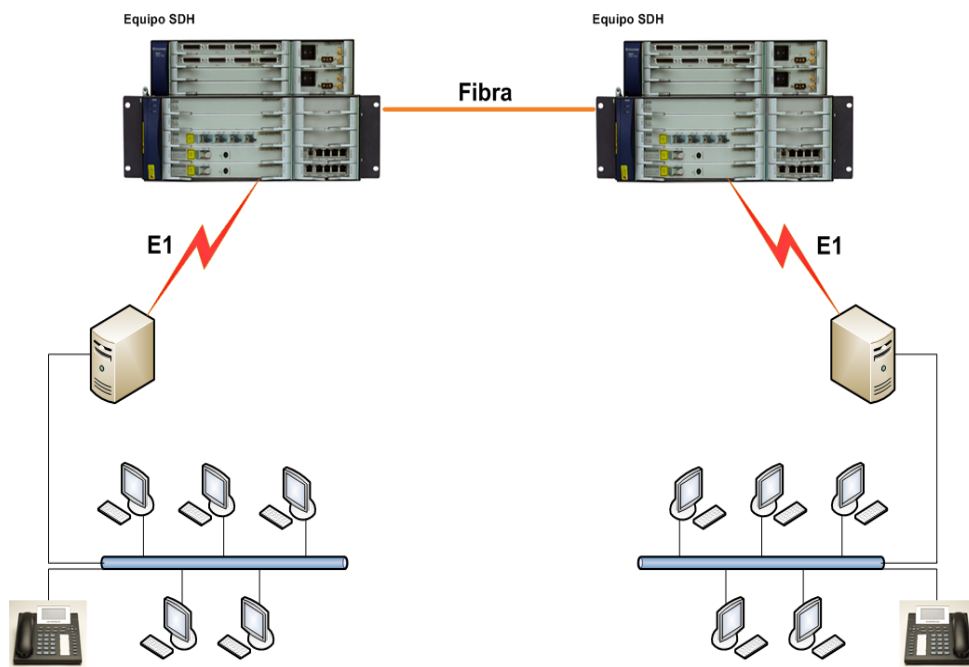


Figura 1.1 Esquema de la implementación del Proyecto

1.3.1 OBJETIVO GENERAL

Implementar el protocolo SS7 sobre conexiones E1 entre 2 servidores Asterisk, de tal manera que exista comunicación entre ellos, utilizando para su efecto los equipos SDH del laboratorio de telecomunicaciones de la FIEC.

1.3.2 OBJETIVOS ESPECIFICOS

Para lograr el objetivo general del proyecto se deben cumplir con los objetivos específicos que se presentan a continuación:

- Familiarizarse y conocer el correcto funcionamiento de los equipos a utilizar, para poder trabajar y configurar los mismos y así lograr con éxito el objetivo general del proyecto.
- Comprobar de que en efecto se está realizando la señalización SS7 entre los dos servidores Asterisk.
- Demostrar los beneficios que posee SS7.
- Implementar eficazmente cada uno de los archivos de configuración a utilizar en el software Asterisk para que exista la correcta comunicación entre los servidores y los equipos SDH.
- Demostrar la viabilidad técnica y económica sobre el uso de tecnología VoIP y SDH junto con SS7 y los E1 por medio del uso de software Asterisk.

1.4 METODOLOGÍA

Para el íntegro funcionamiento de la comunicación que se va a efectuar entre el protocolo SS7 sobre los enlaces E1 que tienen los servidores Asterisk, se debe 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
 - Asterisk
 - AsteriskAddons
 - Chan_ss7

Cabe recalcar que cada uno de los paquetes fueron descargados desde la red (Internet).

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.

1.5 PERFIL DE LA TESIS

Este proyecto de materia de graduación tiene como objetivo principal implementar conexiones (E1) utilizando para su efecto el protocolo SS7, para luego realizar la

comunicación con el equipo SDH que se encuentra disponible en el laboratorio de telecomunicaciones de la FIEC.

En el capítulo 2, se revisa los fundamentos teóricos, para así poder comprender el salto tecnológico seguido con VOIP, sus características, mecanismos de implementación, aplicaciones y servicios que proporciona actualmente, además de las proyecciones que tiene la tecnología para su futuro.

En el capítulo 3, se detallará las especificaciones técnicas de la solución, el análisis, diseño y la implementación del proyecto.

En el capítulo 4, se indicará la correcta configuración que se debe realizar en los equipos SDH para que estos puedan trabajar conjuntamente con Asterisk.

Posteriormente, en el capítulo 5, se realizará las pruebas de conexión, y el establecimiento de las llamadas telefónicas.

CAPÍTULO II

FUNDAMENTOS TEÓRICOS

2.1 ASTERISK

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.

Para funcionar con voz sobre IP no necesita de ningún hardware adicional, ahora para interconectar con la telefonía tradicional requiere de tarjetas especiales que se instalan en el computador y que son de muy bajo costo como las conocidas tarjetas FXO y FXS.

Es bueno señalar que el empleo de sistemas PBX evita conectar todos los teléfonos de una oficina de manera separada a la red de telefonía local pública (RTC), ya que funciona como un switch de red, con ello te ahorras el empleo de una línea propia con salidas de llamadas y cargos mensuales hacia la central telefónica que regresan nuevamente para establecer comunicación interna. Si posee una oficina pequeña y desea emplear software PBX tradicional, sus costos serían altos, pero con Asterisk

se reducen los costos de instalación ya que emula este tipo de centrales con lo que puede aprovechar las funciones de este tipo de sistemas. La solución de código abierto de telefonía basada en el software Asterisk desarrollado por Digium, Inc, va a proveer un vasto conjunto de funciones. Asterisk ofrece las funciones propias de las centralitas clásicas y además características avanzadas, pudiendo trabajar tanto con sistemas de telefonía estándar tradicionales como con sistemas de Voz sobre IP. A pesar de su distribución libre y gratuita Asterisk está dotado con características que sólo puedes encontrar en grandes sistemas PBX de alto costo, así posees las funcionalidades antes descritas e inclusive buzón de voz, conferencia por medio de voz y registros de llamada detallados.

Como se puede visualizar en la **Figura 2.1**, un usuario está teniendo una conversación utilizando un teléfono VOIP el mismo que va a tener configurada una dirección IP y con la cual va a poder conectarse a través de la red para realizar llamadas telefónicas, video llamadas (si el teléfono es el adecuado) entre otras aplicaciones.



Figura 2.1 Software Asterisk para central telefónica

2.1.1 FUNCIONALIDADES DE ASTERIX.

- Sistema de menú en Pantalla ADSI (Interfaz Analógico para presentación de Servicios).
- Receptor de alarmas Agregar Mensaje.
- Autenticación de llamadas con respuesta automatizada.
- Opciones de transferencia de llamadas no supervisada (automatizada).
- Opciones de registros de llamada detallados.
- Desvío de llamadas al interno en el caso que la extensión está ocupada o no responde.
- Desvío de llamada variable.
- Monitorización de llamadas, con opciones de aparcamiento de llamadas.
- Sistema de grabación de llamadas.
- Recuperación de llamadas (DID y ANI).
- Sistema de escucha de llamadas.
- Opciones de transferencia de llamadas.
- Configuración de llamada en espera.
- Identificación de llamadas con opciones de bloqueo, este sistema también se aplica a las llamadas en espera.
- Configurable para trabajar con conferencia de voz.
- Almacenamiento y recuperación en base de datos.
- Integración con Base de Datos.
- Opciones de marcado por nombre.
- Acceso directo al sistema interno.
- Tonos de llamada distintivos.

- Empleo de agentes locales y remotos.
- Configuración de música a tu elección para el proceso de espera, con un sistema de reproducción aleatoria y control de volumen.
- Marcación predictiva.
- Opciones de privacidad.
- Protocolo de establecimiento abierto (OSP).

2.2 SISTEMA DE SEÑALIZACIÓN No. 7 (SS7)

Durante mucho tiempo, la señalización en las redes de conmutación de circuitos se llevaba a cabo sobre el mismo canal en que se transmitía la señal de voz. Este acercamiento es conocido como “señalización asociada” o “señalización en banda” y, de hecho, continúa utilizándose en la actualidad. El modelo norteamericano de señalización R1 de multifrecuencia y el modelo europeo R2 de multifrecuencia forzada son ejemplos de tecnologías de señalización asociada. Este acercamiento funciona adecuadamente mientras los requerimientos de señalización de un conmutador telefónico sean entre sí mismo y otros conmutadores a los que esté conectado. Si la administración y el establecimiento de llamadas fuera la única aplicación de SS7, la señalización asociada sería suficiente para satisfacer dichas necesidades. El sistema de señalización no. 7 surge como un modelo que permite la señalización entre cualquiera de los nodos de la red.

SS7 es el estándar de la tecnología conocida como señalización de canal común (CCS, por 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.

Un modelo de señalización de canal común como lo es SS7, permite que algunos de los nodos de la red puedan analizar las señales y, con base a éstas, llevar a cabo alguna acción determinada. Gracias a esto, SS7 ofrece importantes servicios para el usuario final, entre los que se destacan el identificador de llamadas, los números gratuitos 1-800 y características de portabilidad del número telefónico. Además de todo lo que se ha comentado, SS7 permite que la señalización se lleve a cabo en todo momento, aún cuando no existan llamadas establecidas. Lo anterior pudiera ser útil para activar funciones especiales mediante algún código, sin la necesidad de establecer propiamente una llamada telefónica.

2.2.1 Ventajas de la señalización SS7

- Mejorar la flexibilidad y velocidad en el establecimiento de llamada.
- Reduce el tiempo de uso del costo de voz.
- Mejorar el control de las llamadas y la gestión (tasación).
- Señalización bidireccional.
- Eliminación de equipos por troncal señal.
- Permite cambios de información de señalización en tiempo real, entre redes de conmutación.

- Económica utilización de los circuitos de conversación.
- Admite procedimientos de transmisión de datos tales como: métodos de detección de errores y corrección de errores.

2.2.2 Arquitectura

SS7 divide claramente los planos de señalización y circuitos de voz. Una red SS7 tiene que estar hecha de equipos capaces de soportar SS7 de terminal a terminal para proveer su funcionalidad completa. La red está hecha de muchos tipos de enlace (A, B, C, E, y F) y tres nodos de señalización: Punto de Conmutación de Servicios (SSP), Punto de Transferencia de Señal (STP), y Punto de Control de Servicio (SCP). Cada nodo es identificado en la red por un número, un código punto. Los servicios extendidos son entregados por una interfaz de base de datos a nivel SCP usando X.25.

Para que la arquitectura SS7 sea robusta, la red deberá diseñarse de tal forma que ofrezca un alto grado de redundancia. De esta forma, cualquier problema que pudiera surgir en alguno de los nodos o en alguno de los enlaces, no provocaría una catástrofe en la red y, en consecuencia, se logra una arquitectura confiable y veloz. La **Figura 2.2** muestra un sencillo ejemplo de la disposición de los elementos de la red SS7 y la manera en que estos elementos forman una red interconectada. Será importante enfatizar ciertas características a las que se hace referencia en el mismo esquema:

- Los STPs llevan a cabo las mismas funciones y, por lo tanto, son parte de la redundancia de la arquitectura.

- Cada SSP cuenta con dos enlaces, uno a cada STP. La señalización SS7 al resto del mundo se envía a través de cualquiera de estos enlaces, por lo que se observa aún más redundancia.
- Los STPs que forman un par entre sí, se unen mediante un enlace.
- Dos pares de STPs siempre se encuentran interconectados entre sí.
- Al igual que los STPs, los SCPs se implementan en pares. Sin embargo, no existe un enlace que una los puntos de control que forman un par.
- Siempre existe una señalización indirecta asociada a los elementos de ambas redes interconectadas.

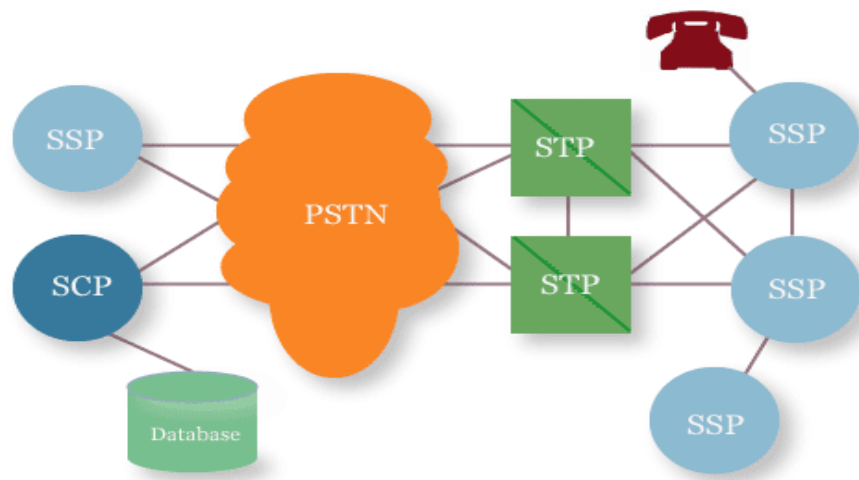


Figura 2.2 Arquitectura de SS7

2.2.3 Capas del modelo SS7

Como ya hemos visto, la arquitectura de red SS7 consiste en un conjunto interconectado de elementos que intercambian mensajes para llevar a cabo funciones de señalización. El protocolo SS7, que en realidad no es uno solo, sino una agrupación de servicios. Al igual que muchos otros

protocolos recientes, SS7 se encuentre dividido en capas y presenta similitudes con el modelo OSI. La figura 2.5 muestra cuáles son las capas que integran el modelo SS7.

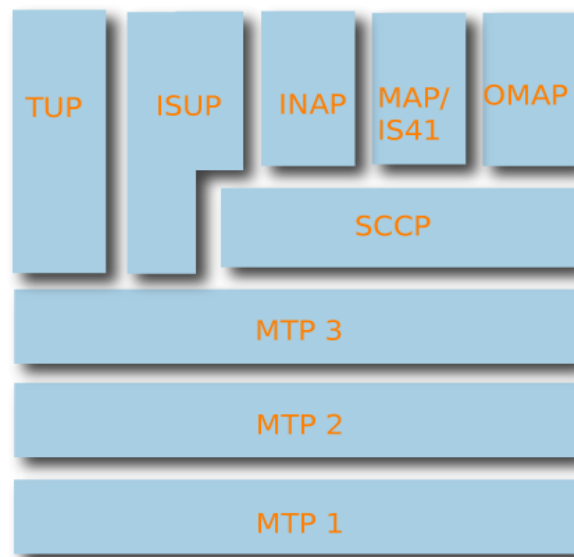


Figura 2.3 Capas del modelo SS7

2.2.3.1 La capa física (MTP-1)

Esto define las características físicas y eléctricas de los enlaces de la red de señalización SS7.

2.2.3.2 Parte de transferencia de mensajes del nivel 2 (MTP-2)

Proporciona la funcionalidad de la capa de enlace. Se asegura de que los dos puntos extremos de un enlace de señalización fiable puedan intercambiar mensajes de señalización. Incorpora capacidades tales como la comprobación de errores, control de flujo, y el control de secuencia.

2.2.3.3 Parte de transferencia de mensajes del nivel 3 (MTP-3)

Extiende la funcionalidad proporcionada por el plan de mediano plazo a nivel 2 para proporcionar la funcionalidad de capa de red. Se asegura que los mensajes puedan ser entregados entre los puntos de señalización a través de la red SS7, independientemente de si están directamente relacionados. Incluye funciones tales como nodo de direcciones, rutas alternativas, y el control de la congestión.

2.2.3.4 Señalización de control de conexión de pieza (SCCP)

Ofrece dos funciones principales que faltan en el plan de mediano plazo. La primera de ellas es la capacidad para hacer frente a las solicitudes dentro de un punto de señalización. El plan de mediano plazo sólo puede recibir y enviar mensajes de un nodo en su conjunto, no se ocupa de las aplicaciones de software dentro de un nodo.

Si bien la red de gestión de mensajes de plan de mediano plazo y llamar a base de mensajes de configuración están dirigidas a un nodo en su conjunto, otros mensajes son utilizados por aplicaciones independientes (denominados subsistemas) dentro de un nodo. Ejemplos de los subsistemas son 800 procesamiento de llamadas, procesamiento de tarjetas de llamadas, la red inteligente avanzada (AIN), y la costumbre local de servicios de la zona de

señalización (CLASS), servicios (por ejemplo, repetir la marcación y el retorno de llamada). El SCCP permite que estos subsistemas se aborden de forma explícita

2.2.3.5 La parte de usuario RDSI (ISUP)

Define los mensajes y el protocolo utilizado en el establecimiento y ejecución de llamadas de voz y datos a través de la red pública conmutada (PSN). A pesar de su nombre, ISUP se utiliza tanto para llamadas RDSI y no RDSI. En la versión norteamericana de SS7, los mensajes ISUP se basan exclusivamente en plan de mediano plazo para el transporte de mensajes entre los nodos que se trate.

2.2.3.6 Capacidades de aplicación de la parte de transacciones (PACT)

Define los mensajes y el protocolo usado para la comunicación entre aplicaciones (desplegadas como subsistemas) en los nodos. Se utiliza para los servicios de base de datos, tales como tarjetas telefónicas, 800, y la AIN, así como de conmutador a conmutador de servicios, incluyendo la marcación de repetición y devolver la llamada.

2.2.3.7 Operaciones, Mantenimiento y Administración de pieza (OMAP)

OMAP define los mensajes y el protocolo diseñado para ayudar a los administradores de la red SS7. Hasta la fecha, el más plenamente desarrollado y desplegado de estas capacidades son los procedimientos para la validación de las tablas de enrutamiento de red y para el diagnóstico de problemas de enlace. OMAP incluye los mensajes que utilizan tanto el plan de mediano plazo y SCCP para el enrutamiento.

2.2.4 Apoyo SS7 en Asterisk

2.2.4.1 LIBISUP

Libisup fué la primera solución SS7 diseñada para su uso con Asterisk. Representa un producto comercial desarrollado por tecnologías Cosini y licenciado bajo la licencia comercial de Digium. Libisup sustituye a libpri (una biblioteca estándar RDSI PRI) en la arquitectura de Asterisk y permite hacer uso de características estándar de Asterisk y aplicaciones. El hardware que se puede instalar como una interfaz a la RTB con señalización SS7 es representado por las tarjetas Digium E1.

2.2.4.2 SS7box

Un rival de Digium, Sangoma introdujo una solución SS7 llamada ss7box que viene junto con ss7boost.

El software se utiliza para la interconexión de redes PSTN e IP en todas las redes en general y utiliza tarjetas de hardware de Sangoma.

Las funcionalidades de SS7box y SS7boost pueden ser descritas como router y Gateway de señalización para ambas variantes de SS7 (ITU y ANSI) y permite una llamada y funciones de red inteligente a través de una red IP.

2.2.4.3 Chan_ss7

Asterisk controlador del canal SS7 (chan SS7) es un software de código abierto desarrollado por la compañía danesa Sifira A / S. La licencia de chan SS7 es GPL (Licencia General Pública). No es certificado oficialmente para la interoperabilidad de SS7.

Esta solución incluye una implementación de la capa de MTP2, más esencial de MTP3 y un gran subconjunto de las funciones de ISUP. El apoyo a la linksets múltiples con CPD diferentes y múltiples hosts de configuración, con un reparto de carga y conmutación por error puede ser interesante desde el punto de vista de gestión de red.

2.2.4.4 Libss7

Libss7 es la última aplicación de SS7 apoyo para Asterisk lanzado por los desarrolladores de Asterisk en el verano de 2006. Como

libisup, representa una biblioteca de sustitución de libpri y haciendo uso de Zap Chan como controlador de canal para Asterisk y un hardware zaptel como una interfaz con la red SS7. La funcionalidad de cubiertas en esta solución es la variante de la UIT de SS7 implementingMTP2, MTP3, y los protocolos ISUP.

Se seleccionó de todas las aplicaciones disponibles a chan_ss7 puesto que es una de las soluciones más utilizadas hasta el momento facilitándonos así la búsqueda de información adecuada para poder trabajar de una manera eficiente con cada uno de los archivos necesarios para la implementación del proyecto.

2.3 GENERALIDADES SDH

La jerarquía digital síncrona (SDH), es un estándar internacional para sistemas ópticos de telecomunicaciones de altas prestaciones que se puede considerar como la revolución de los sistemas de transmisión, como consecuencia de la utilización de la fibra óptica como medio de transmisión. Esta red, por su característica sincrónica, está optimizada para manejo de anchos de banda fijos, lo que la ha convertido en el medio natural para la transmisión de telefonía tradicional. La jerarquía SDH se desarrolló en EE. UU., bajo el nombre de SONET o ANSI T1X1 y posteriormente el CCITT (Hoy UIT-T) en 1989 publicó una serie de recomendaciones donde quedaba definida con el nombre de SDH.

Uno de los objetivos de esta jerarquía estaba en el proceso de adaptación del sistema PDH (Plesiochronous Digital Hierarchy), ya que el nuevo sistema jerárquico se implantaría paulatinamente y debía convivir con la jerarquía plesiócrona instalada. Ésta es la razón por la que la ITU-T normalizó el proceso de transportar las antiguas tramas en la nueva. La trama básica de SDH es el STM-1 (Synchronous Transport Module level 1), con una velocidad de 155 Mbps.

Cada trama va encapsulada en un tipo especial de estructura denominado *contenedor*. Una vez encapsulados se añaden cabeceras de control que identifican el contenido de la estructura (el contenedor) y el conjunto, después de un proceso de multiplexación, se integra dentro de la estructura STM-1. Los niveles superiores se forman a partir de multiplexar a nivel de Byte varias estructuras STM-1, dando lugar a los niveles STM-4, STM-16 y STM-64.

2.3.1 Ventajas

SDH presenta una serie de ventajas respecto a la jerarquía digital plesíncrona (PDH), algunas de ellas son:

- El proceso de multiplexación es mucho más directo. La utilización de punteros permite una localización sencilla y rápida de las señales tributarias de la información.
- El procesamiento de la señal se lleva a cabo a nivel de STM-1. Las señales de velocidades superiores son sincronas entre sí y están en fase por ser generadas localmente por cada nodo de la red.

- Las tramas tributarias de las señales de línea pueden ser subdivididas para acomodar cargas plesíncronas, tráfico ATM o unidades de menor orden. Esto supone mezclar tráfico de distinto tipo dando lugar a redes flexibles.
- Compatibilidad eléctrica y óptica entre los equipos de los distintos proveedores gracias a los estándares internacionales sobre interfaces eléctricos y ópticos.

2.3.2 Desventajas

- Algunas redes PDH actuales presentan ya cierta flexibilidad y no son compatibles con SDH.
- Necesidad de sincronismo entre los nodos de la red SDH, se requiere que todos los servicios trabajen bajo una misma referencia de temporización.
- El principio de compatibilidad ha estado por encima de la optimización de ancho de banda. El número de Bytes destinados a la cabecera de sección es demasiado grande, lo que lleva a perder eficiencia.

2.4 OPTIX 1500B

El OptiX OSN 1500 es una nueva generación de equipos Huawei Technologies Co., Ltd., desarrollada para mejorar el estatus y futuro de las redes de área metropolitana (MAN), integrando la tecnología de la jerarquía digital sincrónica (SDH), múltiple división de ondas (WDM), Ethernet, Modo de transferencia asincrónica (ATM),

jerarquía digital plesincrónica (PDH). Así se puede transmitir servicios de voz y datos eficientemente en la misma plataforma.

Tiene forma de caja como se puede observar en la **Figura 2.4** y puede ser instalado en un espacio de 300mm o 600mm, ya sea en la pared o en el escritorio.

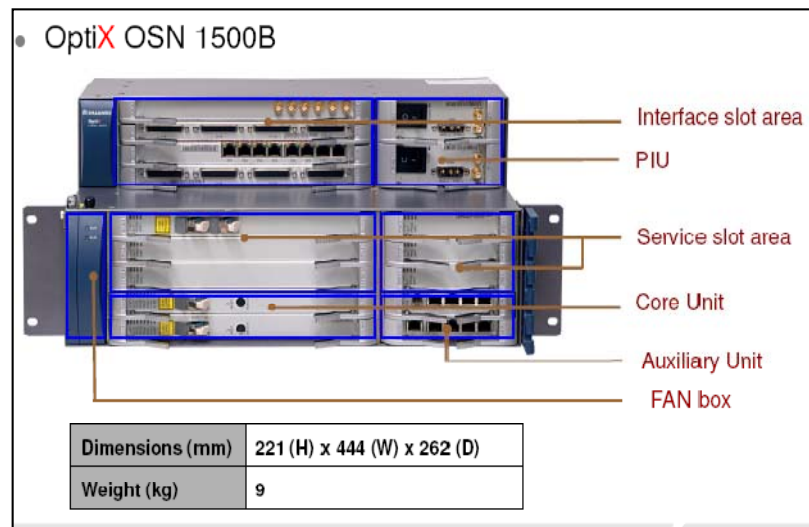


Figura 2.4 Apariencia y Estructura del OptiX OSN 1500B

2.4.1 FUNCIONES

El subrack 1500B OptiX OSN es de una estructura de dos capas, consiste en el área de ranuras para tarjetas de procesamiento, área de ranuras para tarjetas de interfaz, área de ranuras para la tarjeta de interfaz auxiliar, área de alimentación y el área del ventilador.

Las funciones de estas áreas son las siguientes:

- Espacio para tarjetas de interfaz (Interface slot area): Esta área se utiliza para alojar las tarjetas de interfaz tributaria y Ethernet para la OSN OptiX 1500B.
- Área de suministro de energía (Power Supply Area): Esta área se utiliza para albergar dos tarjetas PIU (Power Interface Unit), que se utilizan para suministrar energía para el equipo.
- Espacio para el área de procesamiento de tarjetas (Service Area Slot): Esta área se utiliza para alojar las tarjetas de procesamiento de línea, tributaria y de Ethernet para la OSN OptiX 1500B.
- Unidad de Núcleo (Core Unit): Funciona igual que el área de procesamiento de tarjeta, con la única diferencia que aquí se ubica la tarjeta central que administra los equipos.
- Área de la tarjeta de interfaz auxiliar (Auxiliary Unit): Esta zona se utiliza para alojar la tarjeta de interfaz auxiliar que proporciona interfaces de alarma, interfaz del teléfono orderwire, interfaz de gestión y mantenimiento y la interfaz de reloj.
- Área de ventilación (Fan Area): Esta área se utiliza para ubicar un módulo de ventilación, lo cual disipa el calor generado por el equipo.

2.5 Tarjeta PQ1

La PQ1 es una tarjeta de procesamiento de 63 señales E1, en términos de la versión, función, principios, panel frontal, configuración y especificaciones. La PQ1 puede ser usada para procesar señales E1 y los gastos generales, para informar de

alarmas y eventos de rendimiento, para ofrecer la función de mantenimiento y la protección del TPS.

El principio de trabajo de la tarjeta PQ1 consiste en el PPI, la cartografía E1/T1 mapping/demapping, modulo de conversión de la interfaz.

2.6 Generalidades E1

E1 es un formato de transmisión digital; su nombre fue dado por la administración de la (CEPT). Es una implementación de la portadora-E.

El formato de la señal E1 lleva datos en una tasa de 2,048 millones de bits por segundo y puede llevar 32 canales de 64 Kbps * cada uno, de los cuales treinta y uno son canales activos simultáneos para voz o datos en SS7 (Sistema de Señalización Número 7). En R2 el canal 16 se usa para señalización por lo que están disponibles 30 canales para voz o datos. E1 lleva en una tasa de datos algo más alta que el T-1 (que lleva 1,544 millones de bits por segundo) porque, a diferencia del T-1, no hace el bit-robbing y los ocho bits por canal se utilizan para cifrar la señal. E1 y el T-1 se pueden interconectar para uso internacional.

Un enlace E1 opera sobre dos juegos separados de cable, usualmente es un cable coaxial. Una señal nominal de 2.4 voltios es codificada con pulsos usando un método que evita períodos largos sin cambios de polaridad. La tasa de línea es de 2.048 Mbit/s (full duplex, ej. 2.048 Mbit/s descarga y 2.048 Mbit/s carga) el cual esta abierto en 32 segmentos de tiempo (llamados Time Slots), cada uno tiene un turno direccionado de 8 bit. De esa manera cada casilla envía y recibe un número de 8

bits muestreado 8000 veces por segundo ($8 \times 8000 \times 32 = 2'048,000$). Esto es ideal para llamadas telefónicas de voz, en donde la voz es muestreada en un número de 8 bits y esa tasa de datos es reconstruida en el otro extremo.

Una casilla de tiempo (TS0) es reservado para efectos de segmentación, y transmite alternadamente un patrón arreglado. Esto permite al receptor detectar el inicio de cada trama y encontrar cada canal en el turno. Los estándares permiten que se realice un chequeo de redundancia cíclica a través de todos los bits transmitidos en cada segmento, para detectar si el circuito está perdiendo bits (información), pero esto no siempre es usado.

Una casilla de tiempo (TS16) es usualmente reservada para propósitos de señalización, para controlar la configuración de la llamada y desmonte de acuerdo a varios protocolos estándar de telecomunicaciones. Esto incluye señalización de canales asociados (CAS) en donde un juego de bits es usado para replicar la apertura y cerrada del circuito (como si se descolgara y se marcara en un teléfono analógico). Sistemas más recientes usan señalización de canal común (CCS) como ISDN o sistema de señalización número 7 (SS7) el cual envía pequeños mensajes codificados con más información de la llamada, incluyendo Identificador de llamada (Caller ID), tipo de transmisión requerida etc.

ISDN es usado normalmente entre nodos locales de telefonía y negocios principales, mientras que SS7 es casi exclusivamente usado entre nodos y operadores. SS7 puede manejar hasta 4096 circuitos por canal de señalización, de

esa manera es levemente más eficiente en el uso total de la transmisión del ancho de banda.

A diferencia de los anteriores sistemas T-carrier desarrollados en Norteamérica, todos los 8 bits de cada muestreo están disponibles en cada llamada. Esto permite el sistema E1 ser usado igualmente bien para circuitos de llamadas de datos, sin riesgos de pérdidas de información.

Mientras que el estándar CEPT G703 especifica muchas opciones para la transmisión física, se utiliza de forma casi exclusiva el formato HDB3.

El E1 se usa en todo el mundo excepto Canadá, Estados Unidos y Japón.

2.6.1 Estructura de la Trama E1



Figura 2.5 Estructura de la trama E1

Sin una estructura de trama, en la transmisión de señales digitales multiplexadas, sería imposible recuperar la información, ya que no sabríamos dónde empiezan los bits de un canal y donde los de otro.

Dentro de la trama hay 32 intervalos de tiempo (*time slots*) ocupado cada uno por 8 bits como se muestra en la **Figura 2.5**. Entonces cada trama lleva $32 \times 8 = 256$ bits. La forma en que se acomodan estos canales se muestra a continuación:

La trama se repite cada 125 mseg., que corresponde al periodo de la señal de muestreo. Además existe una agrupación superior en la que un conjunto de 16 tramas constituye una multitrama. Las tramas se numeran del 0 al 15 dentro de la multitrama, y el tiempo de repetición de multitrama es de $16 \times 125\text{mseg.} = 2\text{ms.}$

2.7 Tarjeta Digital TE205P

Las tarjetas de Digium serie TE200 son tarjetas T1/E1 capaces de transportar voz y datos. Apoyan los protocolos estándar de la industria, incluyendo señalización Robbed Bit, E&M, Primary Rate ISDN (PRI), y varios modos de datos (PPP, HDLC, Cisco HDLC y Frame Relay). La Serie TE200 es capaz de funcionar en E1, T1, o modos de J1. También son capaces de canales DACSing de un span a otro. La serie TE200 es ideal para conectar teléfonos a un banco de canal, la conexión a su conmutador T1/E1, o conectarse a una PBX.

Los modos de datos que provee la tarjeta son:

- Cisco HDLC
- HDLC
- PPP

- Multilink PPP
- Frame Relay

Mientras que los modos de voz:

❖ PRI CPE and PRI NET

- NI1
- NI2
- EuroISDN
- 4ESS (AT&T)
- 5ESS (Lucent)
- DMS100

❖ E&M

- Wink
- Feature Group B
- Feature Group D

❖ FXO and FXS

- Ground Start
- Loop Start
- Loop Start with Disconnect Detect

Las tarjetas series TE200 pueden ser usadas para conectar tu maquina Asterisk al mundo PSTN, al banco de canales, o a otro PBX. Esto es complementado via a T1/E1 interface. Las tarjetas permiten al software Asterisk conectar a tu red, creando un ambiente telefónico profesional.

2.7.1 Identificación de la Tarjeta TE205P

Las tarjetas con serie TE200 consiste de dos puertos RJ45 y dos estados de LEDs como se muestra en la **Figura 2.6**. Los puertos son usados para conectar cables T1, E1, ó J1.

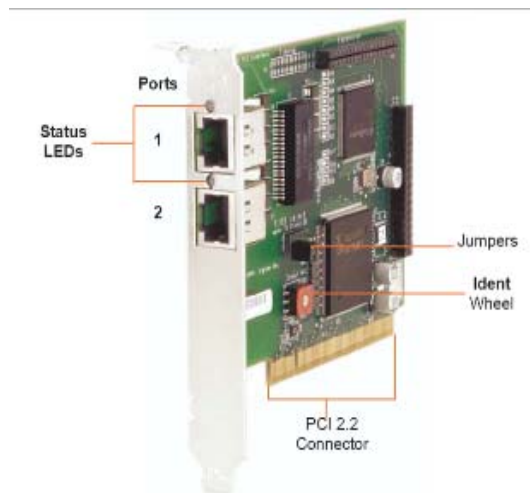


Figura 2.6 Tarjeta TE205P

2.7.2 Selección T1/E1

La tarjeta incluye una columna de jumpers a seleccionar modo T1 or E1 para los spans. Un ejemplo de los jumpers de la tarjeta TE 205 es mostrada en la **Figura 2.7**. El modo T1/E1, en muchos casos, es establecido antes de la instalación de la tarjeta a la Pc.



Figura 2.7 Jumpers E1/T1

Para seleccionar el modo E1, se debe colocar los jumpers en forma cerrada y para seleccionar el modo T1 se debe colocar los jumpers abiertos.

CAPÍTULO III

IMPLEMENTACIÓN DEL PROYECTO

3.1 Asterisk

3.1.1 Requerimientos para iniciar el proyecto en Asterisk

3.1.1.1 Hardware

Las características al momento de seleccionar un servidor pueden ser consideradas fáciles o complicadas, esto depende del presupuesto con el que cuenta el 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 ha configurado 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
- Dos teléfonos VoIP.

3.1.1.2 Software

Para poder aprovechar todos los beneficios que nos brinda Asterisk debemos bajar, extraer, compilar e instalar los paquetes nombrados a continuación:

- Libpri
- Dahdilinux

- Dahditools
- Asterisk
- AsteriskAddons
- Chan_ss7

Todos estos paquetes se los guarda en la ruta: **/usr/src**.

Luego de esto se comienza a extraer cada uno de ellos mediante la ejecución del comando: **tar -xvzf**.

Una vez hecho esto se procede a compilar cada archivo que está en cada carpeta:

Libpri

```
[root@wrks129-213fieclibpri]$ make
```

```
[root@wrks129-213fieclibpri]$ make install
```

Dahdi-linux

```
[root@wrks129-213fiecdahdi-linux-2.2.0.1]# make
```

```
[root@wrks129-213fiecdahdi-linux-2.2.0.1]# make install
```

Dahdi-tools

```
[root@wrks129-213fiecdahdi-tools-2.2.0]# make
```

```
[root@wrks129-213fiecdahdi-tools-2.2.0]# ./configure
```

```
[root@wrks129-213fiecdahdi-tools-2.2.0]# make
```

```
[root@wrks129-213fiecdahdi-tools-2.2.0]# make install
```

```
[root@wrks129-213fiecdahdi-tools-2.2.0]# make config
```

Asterisk

```
[root@ wrks129-213fiec asterisk-1.6.0.10]# ./configure
[root@ wrks129-213fiec asterisk-1.6.0.10]# make
[root@ wrks129-213fiec asterisk-1.6.0.10]# make config
[root@ wrks129-213fiec asterisk-1.6.0.10]# make samples
```

Asterisk-addons

```
[root@ wrks129-213fiec asterisk-addons-1.6.0.2]# ./configure
[root@ wrks129-213fiec asterisk-1.6.0.10]# make
[root@ wrks129-213fiec asterisk-1.6.0.10]# make install
[root@ wrks129-213fiec asterisk-1.6.0.10]# make samples
```

Chan_ss7

```
[root@ wrks129-213fiec chan_ss7-1.3]# make
[root@ wrks129-213fiec chan_ss7-1.3]# make install
```

Cabe recalcar que al ser instalado chan_ss7, el módulo del mismo (chan_ss7.so) se va a ubicar en la ruta: **/usr/local/ss7/lib/modules** teniendo que realizarse el cambio manualmente a la ruta en la que se encuentran los otros módulos de asterisk es decir: **/usr/lib/asterisk/modules**.

Al final cargamos asterisk por medio de:

```
[root@ wrks129-213fiec asterisk-1.6.0.10] # asterisk -r
```


Luego de realizar la instalación de las librerías descritas anteriormente, se creara la carpeta Asterisk en el directorio /etc, quedando los archivos de configuración en el directorio:

/etc/asterisk/

3.2 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.
- Añade soporte Neon MWI para el driver wctdm24xxp.
- Controlador native para los cuatro puertos B410P del modulo BRI.

En esta nueva versión se pueden realizar múltiples cambios para hacer el uso de DAHDI lo más fácil posible en usuarios que utilizan dialplans, CDR parsers, aplicaciones AMI y otras.

Cabe destacar que en el cambio que hubo de Zaptel a DAHDI se han modificado los nombres de ciertos archivos, teniendo ahora los siguientes:

chan_zap.so -> chan_dahdi.so

app_zapbarga.so -> app_dahdibarga.so

app_zapras.so -> app_dahdiras.so

app_zapscan.so -> app_dahdiscan.so

codec_zap.so -> codec_dahdi.so

3.3 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 y chan_dahdi.conf, cabe recalcar que el servidor A es aquel que va a recibir las llamadas que se van a realizar desde el servidor B.

3.3.1 CONFIGURACION SIP.CONF

/etc/asterisk/sip.conf

El protocolo SIP se utiliza para configurar los canales 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.

3.3.1.1 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.

3.3.1.2 Configuración de los canales

[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 en la **Figura 3.1** la configuración final del archivo sip.conf, tanto en el servidor A como en el servidor B.

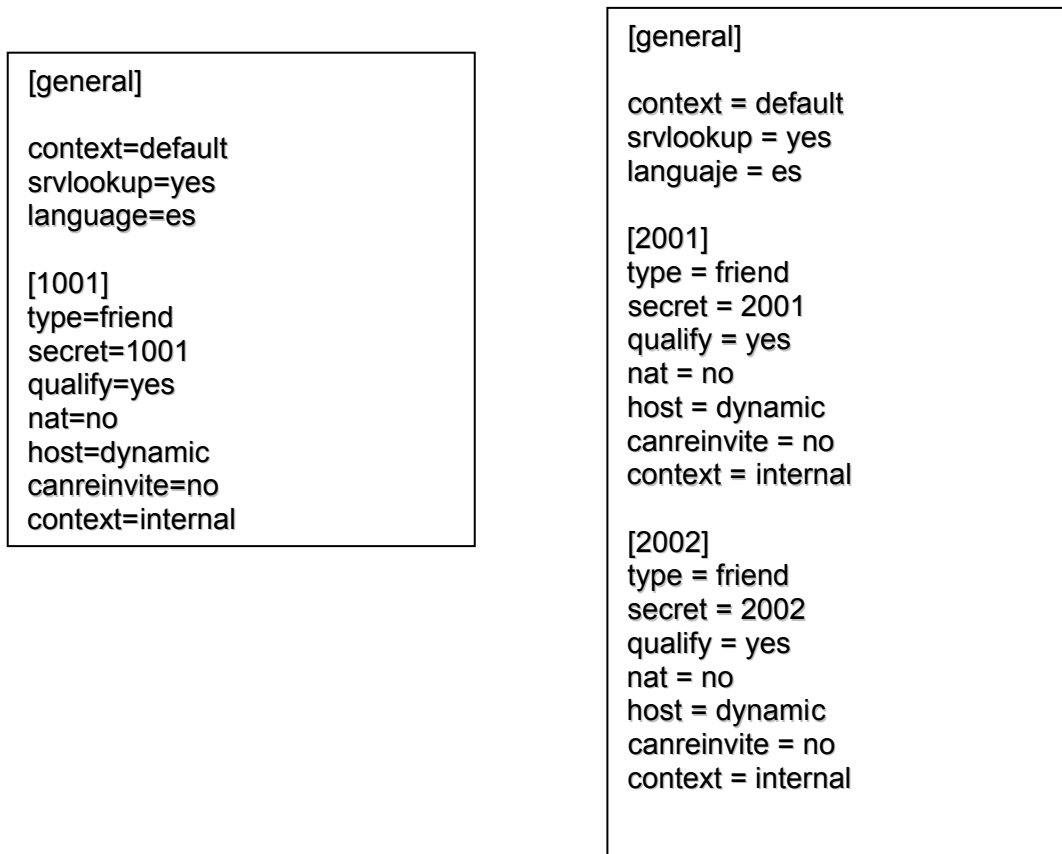


Figura 3.1 Configuración del Archivo sip.conf en Servidor A (a) y B (b)

3.3.2 CONFIGURACION 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.

3.3.2.1 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.

3.3.2.2 Contexto internal

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(90)

Para que espere una mínima 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(SS7/\${EXTEN},10,r)

Se configuran las extensiones de destino al cual queremos llamar conformado con la tecnología SS7, seguido de \${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.

[ss7]

include => internal

De esta manera el servidor A acepta las llamadas que vienen del servidor B y viceversa.

Se usará el contexto internal.

Como se puede apreciar en la Figura 3.2 se encuentra como quedará configurado el archivo extensions.conf para cada uno de los servidores.


```

[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 3.2 Configuración del Archivo extensions.conf Servidor A (a) y B (b)

3.3.3 CONFIGURACION SS7.CONF

En cada equipo, el Asterisk se debe instalar y configurar apropiadamente. Es probablemente 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:

3.3.3.1 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

3.3.3.2 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*.

3.3.3.3 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-wrks129-213fie* o *host-wrks129-214fie*.

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 => I1:1, I2:1

Los enlaces en el host es "I1", conectado a / span conector # 1(para el servidor A) o el host es "I2", 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 3.3**, 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 3.3 Configuración del Archivo ss7.conf para el Servidor A y B

3.3.4 CONFIGURACION 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:

Definicion del formato span:

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; envia tiempo de sincronizacion 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 siempre debería 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: Framing es cas o ccs.

Para el proyecto se utilizó CCS, esto indica que es una Señalización de Canal Común (CCS) en la cual las señales viajan a través de su propio camino. En este sistema se tienen grandes ventajas, no sólo en términos de capacidad. Comparado con el tiempo promedio de llamada, el tiempo total de señalización es muy corto. Esto significa que podemos dejar que un canal de señalización común se encargue de toda la señalización necesaria para un gran número de llamadas.

Podemos resumir las características de la CCS en los siguientes puntos:

- Económico
- Muy rápido
- Confiable
- Gran capacidad
- Flexible y expandible

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

Para E1: código es ami or hdb3 (E1 puede necesitar crc4)

En este caso se usa hdb3, el cual realiza sustentaciones de secuencias de bits que provocan niveles de tensión, constantes por otras que garantizan la anulación de la componente continua y la sincronización del receptor. La longitud de la secuencia queda inalterada, por lo que la velocidad de transmisión de datos es la

misma; además el receptor debe ser capaz de reconocer estas secuencias de datos especiales.

Los objetivos en el diseño de estas técnicas son:

- Evitar la componente en continua.
- Evitar las secuencias largas que correspondan a señales de tensión nula.
- No reducir la velocidad de datos.
- Capacidad para detectar errores.

Bchannels= los canales b por donde se va a ir la información, por tanto los únicos canales que transmiten son 1-31 incluido el canal 16 que es el de señalización.

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

(a)

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

(b)

Figura 3.4 Configuración del Archivo system.conf Servidor A (a) y B (b)

CAPÍTULO IV

CONFIGURACIÓN DE EQUIPOS SDH

4.1 INTRODUCCIÓN

En el laboratorio de Telecomunicaciones de la FIEC (Facultad de Ingeniería en Electricidad y Computación) de la ESPOL (Escuela Superior Politécnica del Litoral) se 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 4.1** esta es la topología anillo con la que trabaja el laboratorio.

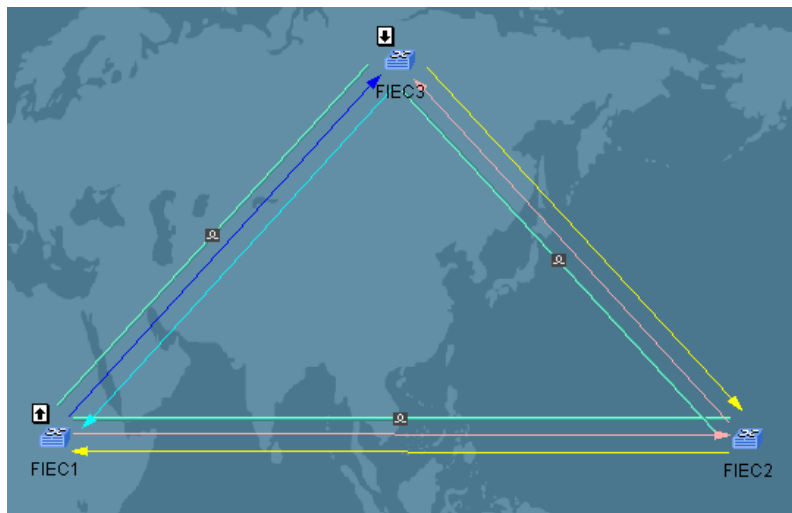


Figura 4.1 Topología anillo del laboratorio de Telecomunicaciones

4.2 CREACIÓN DE SERVICIO (RUTA)

Primero hay que determinar entre que equipos se va a realizar el servicio es decir cuál va a ser la fuente (source) y cuál va a ser el destino (sink), en este caso se va a trabajar con los equipos FIEC1 y FIEC3 como se visualiza en la **Figura 4.2**.

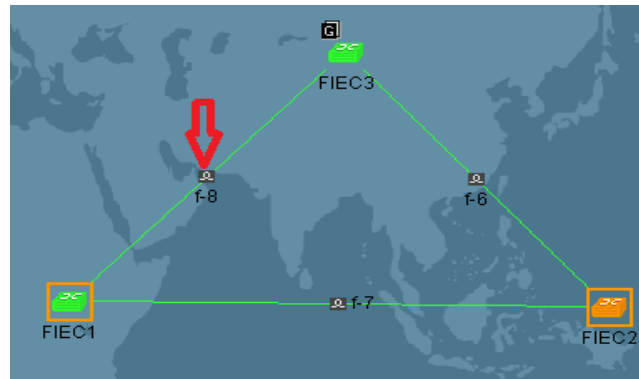


Figura 4.2 Selección del enlace sobre el cual se va a trabajar

Una vez seleccionado el enlace hacemos click derecho y seleccionamos la opción **Query Relevant Trails**. Aquí se muestran los servicios (caminos) que ya han sido creados anteriormente (véase la **Figura 4.3**).

Status	Alarm Status	Name	Source	Source Timeslot	Sink	Sink Timeslot	ID	Cre
Non-Alarmed		FIEC1-FIEC2-VC12-0001	FIEC1-13-P01-1(SDH_TU-1)		FIEC2-13-P01-4(SDH_TU-4)		0	2010-01-

Total: 1 Selected: 1 Relevant View Filter Create ▼ Alarm ▼ Performance ▼ Maintenance ▼ Report ▼ Print Save As

Figura 4.3 Servicios que ya han sido creados

Procedemos a hacer click en el botón **Create** y seleccionamos la opción **Create Trail**.

Tal como se muestra en la **Figura 4.4**, se va a mostrar una pantalla en la cual se puede visualizar cada uno de las tarjetas que tiene instaladas el equipo SDH.

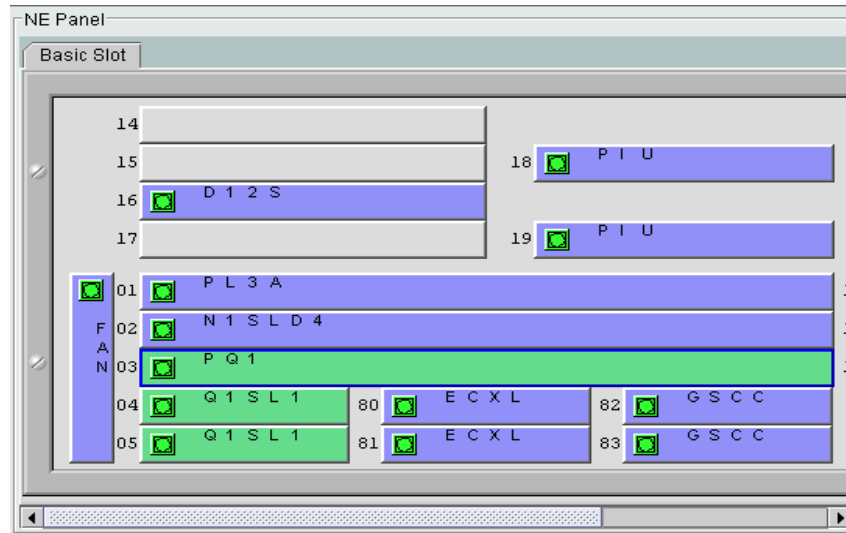


Figura 4.4 Tarjetas que posee el equipo SHD

Seleccionamos 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 4.5**); en este caso, el laboratorio de telecomunicaciones cuenta sólo con cuatro puertos hábiles para trabajar, así que se va a tener que escoger únicamente los puertos habilitados para poder configurar el servicio tanto para la fuente como para el destino.

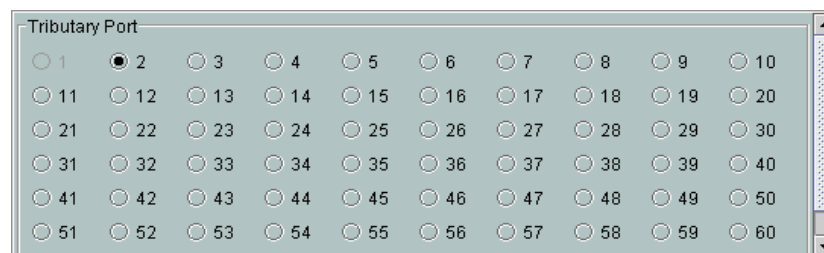


Figura 4.5 Puertos que posee la tarjeta PQ1

Como se visualiza en la **Figura 4.5** el puerto 1 ya ha sido utilizado así que se procede a escoger un puerto disponible (puerto dos) para realizar nuestro camino; hacemos click en **OK** y con esto ya hemos creado la fuente (source); de la misma manera tenemos que realizar la configuración para el destino (sink). Finalmente al tener los dos puertos configurados hacemos click en **Apply** y como se visualiza en la **Figura 4.6** se va a poder apreciar la nueva ruta creada que en este caso sería:

Fuente (Source): FIEC1-13-PQ1-2(SDH_TU-2)

Destino (Sink): FIEC3-13-PQ1-1(SDH_TU-1)

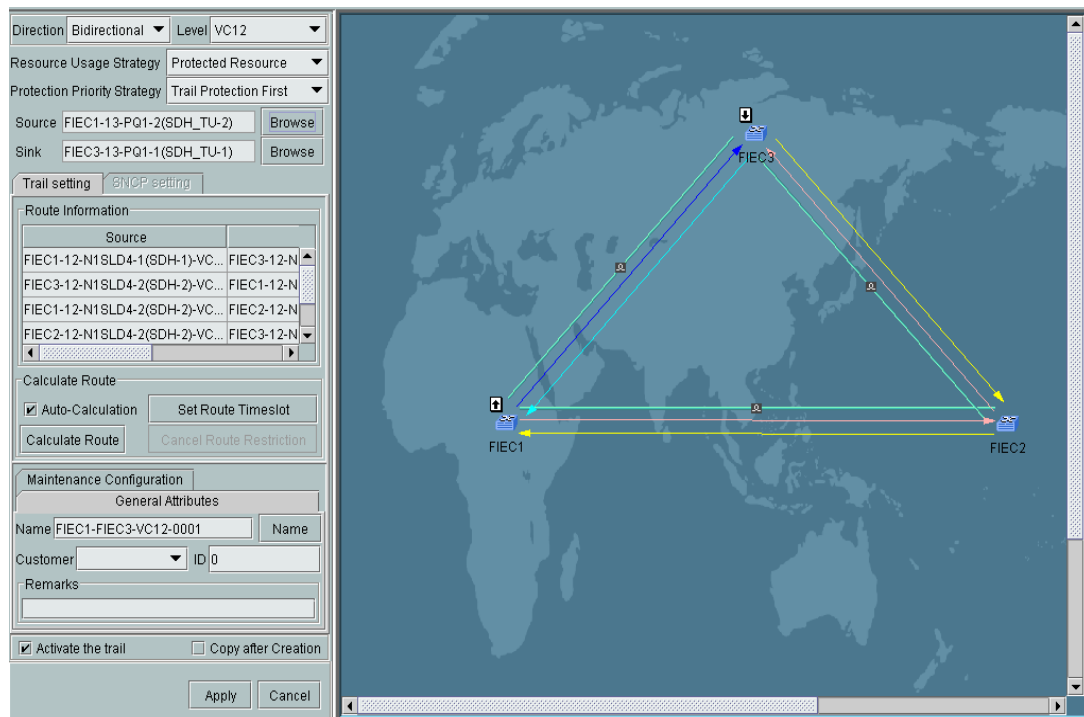


Figura 4.6 Nuevo servicio creado entre FIEC1 Y FIEC3

CAPÍTULO V

FUNCIONAMIENTO Y PRUEBAS DEL PROYECTO

5.1 ETAPA DE PRUEBAS ENTRE SERVIDORES

Una vez instalado los paquetes necesarios para el correcto funcionamiento de Asterisk en ambos servidores, se procedió a comprobar si existe comunicación entre ellos, utilizando un cable E1 crossover, de tal manera que se puedan llamar.

Pruebas con Hardware

Para realizar las pruebas con el hardware, debemos tener un cable E1 crossover, el cual tiene características especiales como solo utilizar cuatro hilos, en la **Figura 5.1** se detallan los pines que se utilizan:


	Pin	Description
	1	Rx
	2	Rx
	3	Not used
	4	Tx
	5	Tx
	6	Not used
	7	Not used
	8	Not used

Figura 5.1 Extremo de cable E1 crossover (Conector RJ-45)

El primer indicador del correcto funcionamiento de la tarjeta es el color verde del led en ambas tarjetas TE205P como se visualiza en la **Figura 5.2**.

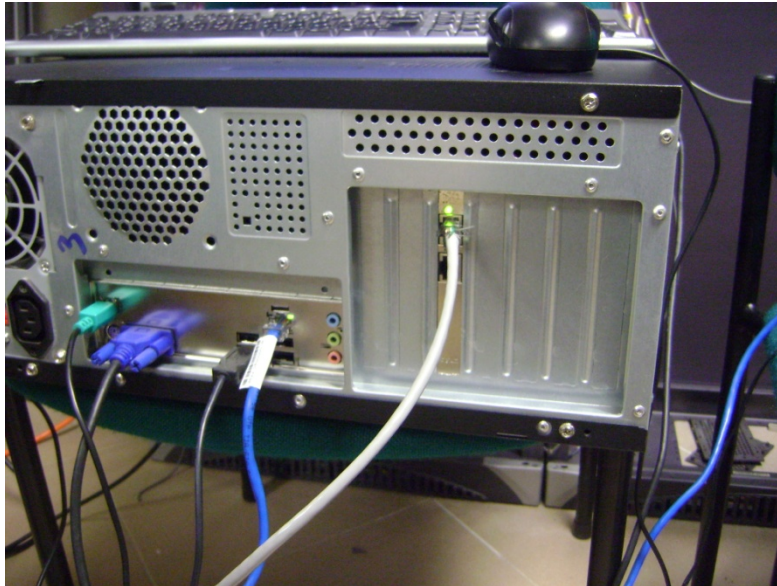


Figura 5.2 Color verde encendido en la tarjeta TE205P

5.2 INICIALIZANDO CON ASTERISK

Los siguientes son comandos que permiten iniciar o detener el servicio de asterisk:

<i>service asterisk start</i>	Iniciar servicio.
<i>service asterisk stop</i>	Detener servicio.
<i>service asterisk status</i>	Obtener estado del servicio.
<i>service asterisk restart</i>	Reiniciar servicio.


Comandos que permiten ingresar y salir de la consola remota de asterisk así como también ejecutar acciones sin necesidad de ingresar a la misma:

Asterisk Arrancar Asterisk.
asterisk -r Ingresar a la consola remota

Así mismo se debe de revisar en la consola del terminal el correcto funcionamiento de la tarjeta con los siguientes comandos:

dahdi_cfg
service dahdi restart

De tal manera que si la tarjeta esta correcta nos muestre una ventana como la de la **Figura 5.3**.



```

root@wrks19-228f1ec:~# dahdi_cfg
root@wrks19-228f1ec:~# dahdi_cfg
root@wrks19-228f1ec:~# dahdi_cfg
root@wrks19-228f1ec:~# service dahdi restart
Unloading DAHDI hardware modules: ERROR: Module wct4xsp is in use
ERROR: Module dahdi is in use by wct4xsp
error
Loading DAHDI hardware modules:
  wct4xsp:                [ OK ]
Running dahdi_cfg:       [ OK ]
root@wrks19-228f1ec:~#

```

Figura 5.3 Comando service dahdi restart

Previamente, en Asterisk, se tiene que realizar la comprobación de que los canales de la tarjeta TE205P estén habilitados en los dos servidores; eso lo hacemos

mediante la ejecución del comando **ss7 show channels** como se indica en la **Figura 5.4**.

```
wrks19-228fiec*CLI> ss7 show channels
Linkset: siucc*CLI>
CIC  1 Idle
CIC  2 Idle
CIC  3 Idle
CIC  4 Idle
CIC  5 Idle
CIC  6 Idle
CIC  7 Idle
CIC  8 Idle
CIC  9 Idle
CIC 10 Idle
CIC 11 Idle
CIC 12 Idle
CIC 13 Idle
CIC 14 Idle
CIC 15 Idle
CIC 17 Idle
CIC 18 Idle
CIC 19 Idle
CIC 20 Idle
CIC 21 Idle
CIC 22 Idle
CIC 23 Idle
CIC 24 Idle
CIC 25 Idle
CIC 26 Idle
CIC 27 Idle
CIC 28 Idle
CIC 29 Idle
CIC 30 Idle
CIC 31 Idle
wrks19-228fiec*CLI> █
```

Figura 5.4 Canales habilitados por Asterisk

Una vez registrado y comprobado la correcta configuración de los canales, modificamos el archivo sip.conf al cual añadimos un nuevo usuario, también modificamos el archivo extensions.conf, agregando el usuario que se creó en el

archivo sip.conf. Luego se debe cargar en Asterisk los modulos que hemos modificado, para esto utilizamos los siguientes comandos:

module reload chan_sip.so (sip.conf)

module reload pbx_config.so (extensions.conf)

module reload chan_ss7.so (ss7.conf)

Luego de que se haya actualizado el software, en el servidor B ingresamos el comando:

originate SS7/1001 application echo

El comando originate, genera una llamada, en nuestro caso a través del protocolo SS7, especificando la extensión que recibe la llamada, es así como se puede comprobar la comunicación entre los servidores (véase la **Figura 5.5**).

```

wrks19-228fieci*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
wrks19-228fieci*CLI> █

```

Figura 5.5 Servidor A, el cual acepta la llamada proveniente del servidor B

5.3 Etapas de Pruebas con equipos SDH

Una vez establecida la conexión, por medio del cable E1 crossover, procedemos a establecer comunicación con los equipos SDH del laboratorio de telecomunicaciones.

Para conectar la tarjeta TE205P con los equipos SDH utilizamos 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 así, que viendo el sistema de gestión (**Figura 5.6**) en el servidor de los equipos, se ve que están 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 5.6. Gestión de los equipos SDH Huawei

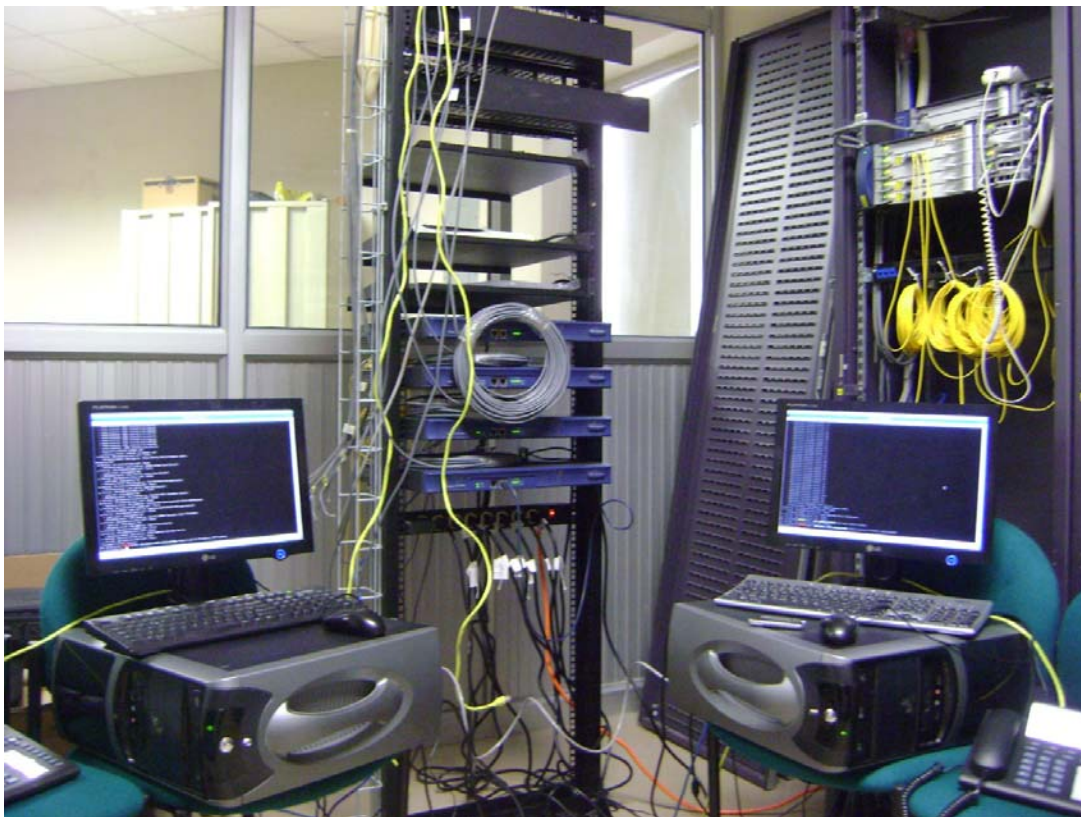


Figura 5.7 Conexión de los servidores con los equipos SDH

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

```

wrks19-228fiiec*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
wrks19-228fiiec*CLI> █

```

Figura 5.8 Recepción de la llamada en el servidor A, proveniente de B.

Comprobado que existe comunicación entre servidores y equipos SDH tan solo con un canal del E1 (como se visualiza en la **Figura 5.8**), procedemos 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 5.9**):

/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 5.9 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.

Como se indica en la **Figura 5.9** se va a tener un contador de extensiones (a) el cual se va a ir reduciendo de uno en uno mediante la utilización del comando **for**, dentro del mismo se va a encontrar el comando **origíname** que es el que va a realizar cada una de las llamadas.

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

Como se visualiza en la **Figura 5.10** este va a ser el resultado de la ejecución del script cuando se digita el numero de extensión antes mencionado.

A terminal window with a black background and white text. The text shows a command being executed: "-- Executing [123@internal:1] System('SIP/2001-00000012', '/etc/asterisk/script') in new stack". The word "System" is highlighted in green, and the path "/etc/asterisk/script" is highlighted in pink.

```
-- Executing [123@internal:1] System('SIP/2001-00000012', '/etc/asterisk/script') in new stack
```

Figura 5.10 Ejecución del script desde el servidor B

En la **Figura 5.11** se puede ver las llamadas que se generan del servidor B hacia el servidor A, habilitándose una a una.

```

Archivo  Editar  Ver  Terminal  Solapas  Ayuda
-- Recv IAM CIC=30 ANI= DNI=1033 RNI= redirect=no/0 complete=1
-- Executing [1033@ss7:1] Answer("SS7/siuc/30", "") in new stack
-- Executing [1033@ss7:2] Wait("SS7/siuc/30", "90") in new stack
-- Recv IAM CIC=28 ANI= DNI=1032 RNI= redirect=no/0 complete=1
-- Executing [1032@ss7:1] Answer("SS7/siuc/28", "") in new stack
-- Executing [1032@ss7:2] Wait("SS7/siuc/28", "90") in new stack
-- Recv IAM CIC=26 ANI= DNI=1031 RNI= redirect=no/0 complete=1
-- Executing [1031@ss7:1] Answer("SS7/siuc/26", "") in new stack
-- Executing [1031@ss7:2] Wait("SS7/siuc/26", "90") in new stack
-- Recv IAM CIC=24 ANI= DNI=1030 RNI= redirect=no/0 complete=1
-- Executing [1030@ss7:1] Answer("SS7/siuc/24", "") in new stack
-- Executing [1030@ss7:2] Wait("SS7/siuc/24", "90") in new stack
-- Recv IAM CIC=22 ANI= DNI=1029 RNI= redirect=no/0 complete=1
-- Executing [1029@ss7:1] Answer("SS7/siuc/22", "") in new stack
-- Executing [1029@ss7:2] Wait("SS7/siuc/22", "90") in new stack
-- Recv IAM CIC=20 ANI= DNI=1028 RNI= redirect=no/0 complete=1
-- Executing [1028@ss7:1] Answer("SS7/siuc/20", "") in new stack
-- Executing [1028@ss7:2] Wait("SS7/siuc/20", "90") in new stack
-- Recv IAM CIC=18 ANI= DNI=1027 RNI= redirect=no/0 complete=1
-- Executing [1027@ss7:1] Answer("SS7/siuc/18", "") in new stack
-- Executing [1027@ss7:2] Wait("SS7/siuc/18", "90") in new stack
-- Recv IAM CIC=14 ANI= DNI=1026 RNI= redirect=no/0 complete=1
-- Executing [1026@ss7:1] Answer("SS7/siuc/14", "") in new stack
-- Executing [1026@ss7:2] Wait("SS7/siuc/14", "90") in new stack
-- Recv IAM CIC=12 ANI= DNI=1025 RNI= redirect=no/0 complete=1
-- Executing [1025@ss7:1] Answer("SS7/siuc/12", "") in new stack
-- Executing [1025@ss7:2] Wait("SS7/siuc/12", "90") in new stack
-- Recv IAM CIC=10 ANI= DNI=1024 RNI= redirect=no/0 complete=1
-- Executing [1024@ss7:1] Answer("SS7/siuc/10", "") in new stack
-- Executing [1024@ss7:2] Wait("SS7/siuc/10", "90") in new stack
-- Recv IAM CIC=8 ANI= DNI=1023 RNI= redirect=no/0 complete=1
-- Executing [1023@ss7:1] Answer("SS7/siuc/8", "") in new stack
-- Executing [1023@ss7:2] Wait("SS7/siuc/8", "90") in new stack
-- Recv IAM CIC=6 ANI= DNI=1022 RNI= redirect=no/0 complete=1
-- Executing [1022@ss7:1] Answer("SS7/siuc/6", "") in new stack
-- Executing [1022@ss7:2] Wait("SS7/siuc/6", "90") in new stack
-- Recv IAM CIC=4 ANI= DNI=1021 RNI= redirect=no/0 complete=1
-- Executing [1021@ss7:1] Answer("SS7/siuc/4", "") in new stack
-- Executing [1021@ss7:2] Wait("SS7/siuc/4", "90") in new stack
-- Recv IAM CIC=2 ANI= DNI=1020 RNI= redirect=no/0 complete=1
-- Executing [1020@ss7:1] Answer("SS7/siuc/2", "") in new stack
-- Executing [1020@ss7:2] Wait("SS7/siuc/2", "90") in new stack
-- Recv IAM CIC=31 ANI= DNI=1019 RNI= redirect=no/0 complete=1
-- Executing [1019@ss7:1] Answer("SS7/siuc/31", "") in new stack
-- Executing [1019@ss7:2] Wait("SS7/siuc/31", "90") in new stack
-- Recv IAM CIC=29 ANI= DNI=1018 RNI= redirect=no/0 complete=1

```

Figura 5.11 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 proceden a cerrar de forma automáticamente cada llamada que se estableció (véase **Figura 5.12**).


```

Archivo  Editar  Ver  Terminal  Solapas  Ayuda
-- Executing [1033@ss7:3] Hangup("SS7/siuc/30", "") in new stack
== Spawn extension (ss7, 1033, 3) exited non-zero on 'SS7/siuc/30'
-- SS7 hangup 'SS7/siuc/30' CIC=30 Cause=16 (state=5)
-- Executing [1032@ss7:3] Hangup("SS7/siuc/28", "") in new stack
== Spawn extension (ss7, 1032, 3) exited non-zero on 'SS7/siuc/28'
-- SS7 hangup 'SS7/siuc/28' CIC=28 Cause=16 (state=5)
-- Executing [1031@ss7:3] Hangup("SS7/siuc/26", "") in new stack
== Spawn extension (ss7, 1031, 3) exited non-zero on 'SS7/siuc/26'
-- SS7 hangup 'SS7/siuc/26' CIC=26 Cause=16 (state=5)
-- Executing [1030@ss7:3] Hangup("SS7/siuc/24", "") in new stack
== Spawn extension (ss7, 1030, 3) exited non-zero on 'SS7/siuc/24'
-- SS7 hangup 'SS7/siuc/24' CIC=24 Cause=16 (state=5)
-- Executing [1029@ss7:3] Hangup("SS7/siuc/22", "") in new stack
== Spawn extension (ss7, 1029, 3) exited non-zero on 'SS7/siuc/22'
-- SS7 hangup 'SS7/siuc/22' CIC=22 Cause=16 (state=5)
-- Executing [1028@ss7:3] Hangup("SS7/siuc/20", "") in new stack
== Spawn extension (ss7, 1028, 3) exited non-zero on 'SS7/siuc/20'
-- SS7 hangup 'SS7/siuc/20' CIC=20 Cause=16 (state=5)
-- Executing [1027@ss7:3] Hangup("SS7/siuc/18", "") in new stack
== Spawn extension (ss7, 1027, 3) exited non-zero on 'SS7/siuc/18'
-- SS7 hangup 'SS7/siuc/18' CIC=18 Cause=16 (state=5)
-- Executing [1026@ss7:3] Hangup("SS7/siuc/14", "") in new stack
== Spawn extension (ss7, 1026, 3) exited non-zero on 'SS7/siuc/14'
-- SS7 hangup 'SS7/siuc/14' CIC=14 Cause=16 (state=5)
-- Executing [1025@ss7:3] Hangup("SS7/siuc/12", "") in new stack
== Spawn extension (ss7, 1025, 3) exited non-zero on 'SS7/siuc/12'
-- SS7 hangup 'SS7/siuc/12' CIC=12 Cause=16 (state=5)
-- Executing [1024@ss7:3] Hangup("SS7/siuc/10", "") in new stack
== Spawn extension (ss7, 1024, 3) exited non-zero on 'SS7/siuc/10'
-- SS7 hangup 'SS7/siuc/10' CIC=10 Cause=16 (state=5)
-- Executing [1023@ss7:3] Hangup("SS7/siuc/8", "") in new stack
== Spawn extension (ss7, 1023, 3) exited non-zero on 'SS7/siuc/8'
-- SS7 hangup 'SS7/siuc/8' CIC=8 Cause=16 (state=5)
-- Executing [1022@ss7:3] Hangup("SS7/siuc/6", "") in new stack
== Spawn extension (ss7, 1022, 3) exited non-zero on 'SS7/siuc/6'
-- SS7 hangup 'SS7/siuc/6' CIC=6 Cause=16 (state=5)
-- Executing [1021@ss7:3] Hangup("SS7/siuc/4", "") in new stack
== Spawn extension (ss7, 1021, 3) exited non-zero on 'SS7/siuc/4'
-- SS7 hangup 'SS7/siuc/4' CIC=4 Cause=16 (state=5)
-- Executing [1020@ss7:3] Hangup("SS7/siuc/2", "") in new stack
== Spawn extension (ss7, 1020, 3) exited non-zero on 'SS7/siuc/2'
-- SS7 hangup 'SS7/siuc/2' CIC=2 Cause=16 (state=5)
-- Executing [1019@ss7:3] Hangup("SS7/siuc/31", "") in new stack
== Spawn extension (ss7, 1019, 3) exited non-zero on 'SS7/siuc/31'
-- SS7 hangup 'SS7/siuc/31' CIC=31 Cause=16 (state=5)
-- Executing [1018@ss7:3] Hangup("SS7/siuc/29", "") in new stack

```

Figura 5.12 Cierre de la llamadas pasado los 90 segundos, en el Servidor A

**CONCLUSIONES Y
RECOMENDACIONES**

CONCLUSIONES

1. 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.
2. 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.
3. 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.
4. 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.

RECOMENDACIONES

1. 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.
2. 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.
3. 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.
4. 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.
5. Una recomendación de trabajo a futuro sería la de montar servidores Asterisk en diferentes partes de una ciudad, pero ya no solo con 2 sino colocando 3 y hasta más servidores ubicados en lugares estratégicos para su respectiva conexión utilizando las aplicaciones web (véase **Figura T.1**).
6. Un ejemplo de servicios que se podría utilizar en una empresa gracias a SS7 sería el identificador de llamadas, los números gratuitos 1-800 y características de portabilidad del número telefónico.

7. 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 .

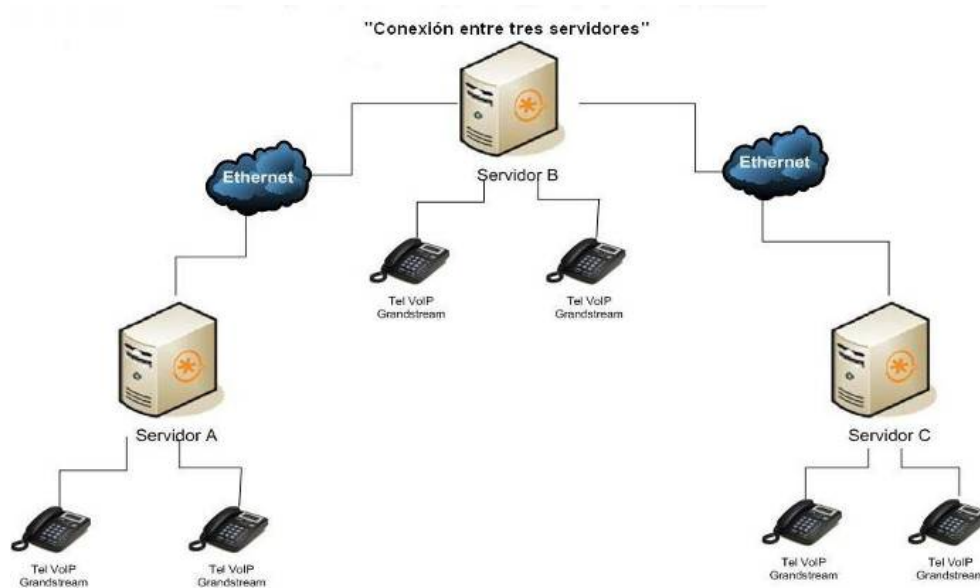


Figura T.1 Conexión de 3 servidores Asterisk usando plataforma Web

GLOSARIO

Add-on: Pequeños programas opcionales que sólo funcionan anexados a otro y que sirven para incrementar o complementar sus funcionalidades.

ATM: Asynchronous Transfer Mode. A modo de transferencia en la cual la información está organizada dentro de células. Es asíncrono en el sentido de que la repetición de las células contienen información de un usuario individual no necesariamente periódico. Se trata de un protocolo dentro de la capa 1 de OSI. Una celda ATM consiste de una cabecera de 5 octeto seguido por 48 octetos de datos.

Cabecera de sesión SOH: Bytes de control añadidos a las tramas STS-1 o STM-1, proporcionando funciones como facilidades de OAM, alineamiento de trama, conmutación de protección, etc.

Dialplan: Un dial plan (plan de marcado) es un software residente en algunos gateways y en casi todas las IP-PBX que permite determinar el tratamiento que debe darse a un número discado.

Gateway: Puerta de enlace, acceso, pasarela. Nodo en una red informática que sirve de punto de acceso a otra red.

MSP: Este procedimiento opera con una sección de tráfico ubicada entre dos nodos adyacentes, en dichos nodos hay dos enlaces separados o dos diferentes fibras: la operativa y la de protección. Ante un evento de fallo del enlace, la señal entrante debe ser conmutada de la fibra activa a la de protección.

Open Source: es el término con el que se conoce al software distribuido y desarrollado libremente. El código abierto tiene un punto de vista más orientado a los beneficios prácticos de compartir el código que a las cuestiones morales y/o filosóficas las cuales destacan en el llamado software libre.

PBX: (Central Telefónica Digital). Sistema telefónico dentro de una organización que maneja las llamadas entre sus usuarios en líneas locales mientras permite que entre todos los usuarios compartan un número determinado de líneas telefónicas externas.

T2000: El T2000 es un sistema de gestión de subred (SNMS). En la arquitectura de la red de telecomunicaciones de gestión, la T2000 se encuentra entre el nivel de NE y de la red, que soporta todas las funciones de nivel NE y parte de las funciones de gestión de nivel de red.

Voip: Voice Over Internet Protocol (Voz Sobre el Protocolo de Internet). Básicamente VoIP es un método por el cual tomando señales de audio analógicas del tipo de las que se escuchan cuando uno habla por teléfono se las transforma en datos digitales que pueden ser transmitidos a través de internet hacia una dirección IP determinada.

BIBLIOGRAFÍA

- [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%BA_n.%C2%BA_7, 2010
- [8] CINIT, SS7 , <http://www.cinit.org.mx/articulo.php?idArticulo=7>, 2010