UNIVERSIDAD NACIONAL DEL CALLAO

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

ESCUELA PROFESIONAL DE INGENIERÍA ELECTRÓNICA

TESIS

"MEJORA DE LA CAPACIDAD DE COMUNICACIONES DE UN SISTEMA DE TELEFONÍA MEDIANTE LA IMPLEMENTACIÓN DE UNA PBX BASADA EN ASTERISK Y UNA LÍNEA PRIMARIA E1"

PARA OBTENER EL TITULO PROFESIONAL DE INGENIERO ELECTRÓNICO

PRESENTADO POR LOS BACHILLERES:

- ✓ NORIEGA MEZA DANIEL
- ✓ CARRASCO MALUQUISH MARCIAL A.
- ✓ BARRIENTOS SULCA ARTURO P.

Callao, 2019

PERU

DEDICATORIA

Dedicamos nuestra tesis a nuestros padres por todo el apoyo incondicional que siempre nos brindaron de manera y ser parte fundamental de nuestras vidas.

AGRADECIMIENTO

Agradecemos a Dios por darnos fortaleza, nuestros padres y hermanos por el apoyo incondicional que nos brindan y a nuestros docentes en la época de estudiante, por facilitar y brindarnos sus conocimientos para hoy ser un buen profesional.

RESUMEN

En el presente informe de suficiencia se describe el diseño e implementación de la mejora de la capacidad de comunicaciones de un sistema de telefonía mediante la implementación de una central telefónica IP basada en *Asterisk* y una línea primaria E1.

El caso de estudio es un c*all center* encargado de las cobranzas que usa dos locales, donde actualmente el local principal, en donde se concentra la mayoría del tráfico voz IP usa alrededor de 70 anexos y el otro local, donde solo posee una oficina de atención al cliente con no más de 5 anexos.

Los problemas que se experimentaban en ambos locales eran: la baja confiabilidad y disponibilidad, la baja calidad del audio de telefonía, deficiencias en la seguridad de la red, tormentas de broadcast, pérdida de paquetes, etc.

La solución para obtener una red escalable, que posea una buena integración y con calidad de servicio para la voz, esta se desarrolla en varias etapas abarcando tareas como la implementación de redes VLAN, calidad de servicio para los distintos servicios, la implementación de *firewalls-proxy*, realizar una red virtual privada (VPN IPsec) entre ambas sedes; la implementación de centrales telefónicas *Asterisk*, la interconexión de ambas centrales telefónicas IP *Asterisk* y la implementación de una línea primaria E1 como respaldo.

El propósito de la solución descrita en el presente informe es obtener un *call center* capaz de cumplir con los estándares establecidos por la empresa dueña del mismo, como la alta disponibilidad, seguridad e interconexión.

ABSTRACT

In this sufficiency report, the design and implementation of the improvement of the communications capacity of a telephony system is described through the implementation of an IP telephone exchange based on Asterisk and a primary E1 line.

The case study is a call center in charge of collections that uses two premises, where currently the main premises, where most of the voice traffic IP concentrates, uses about 70 annexes and the other premises, where it only has one office of attention. to the client with no more than 5 annexes.

The problems that were experienced in both locations were: low reliability and availability, low telephony audio quality, deficiencies in network security, broadcast storms, packet loss, etc.

The solution to obtain a scalable network, which has a good integration and quality of service for the voice, is developed in several stages covering tasks such as the implementation of VLANs, quality of service for the different services, the implementation of firewalls- proxy, perform a private virtual network (IPsec VPN) between both sites; the implementation of Asterisk telephone exchanges, the interconnection of both Asterisk IP telephone exchanges and the implementation of a primary E1 line as backup.

The purpose of the solution described in this report is to obtain a call center capable of complying with the standards established by the company that owns it, such as high availability, security and interconnection.

ÍNDICE

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

GLOSARIO DE TERMINOS

ACL	Listas de Control de Acceso
ARP	Address Resolution Protocol
Core	Núcleo
CRC	Comprobación cíclica de redundancia
CS – ACELP	Codec-excited linear prediction speech coding
DHCP	Dynamic Host Configuration Protocol
EIGRP	Enhanced Interior Gateway Routing Protocol
FCS	Secuencia de verificación de trama
HSRP	Hot Standby Router Protocol
Hub	Concentradores
IP	Internet Protocol
IEEE	Instituto de Ingenieros Eléctricos y Electrónicos
IOS	Internetwok Operating System
LLC	Logical Link Control
LAN	Red de área local
MAC	Medium Access Control
NAC	Control de admisión a la red
NIC	Tarjeta de interfaz de red
OTDR	Reflectómetro óptico de dominio de tiempo
PCM	Pulse Codification Modulation
PDU	Unidad de Data de Protocolo
PSTN	Public Switched Telephony Network
PoE	Power over Ethernet
QoS	Calidad de servicio.
RTP	Real Time Transport Protocol

RTCP	Real Time Control Protocol
SC	Conector Lucent
SIP	Session Initiation Protocol
SSH	Secure Shell
SSL	Secure Sockets Layer
ST	Punta Recta
STP	Spanning Tree Protocol
TCP	Transmission Control Protocol
TIA	Asociación de las Industrias de las Telecomunicaciones
Time-stamping	Marcas de tiempo
URI	(Uniform Resource Identifier)
User Agent	UA
User agent client	UAC
User Agent Server	UAS
UTP	Cableado de par trenzado no blindado
VLAN	Redes virtuales
VoIP	Voice Over IP
WAN	Wide Area Network
XMPP	Extensible Messaging and Presence Protocol

INTRODUCCIÓN

El trabajo desarrollado en este informe surge por la necesidad de mejorar la capacidad de comunicaciones de un sistema de telefonía mediante la implementación de una central telefónica IP basada en *Asterisk* y una línea primaria E1. Eliminado así una serie de deficiencias que experimentaba la empresa del caso de estudio, por ejemplo:

- La empresa poseía una IP PBX propietaria llamada DENWA, la cual ya había experimentado problemas de hardware (mal funcionamiento del ventilador del equipo, lo cual provoco que se quemara la *mainboard* del equipo).

- El tiempo para establecer una sesión SIP (*Session Initiation Protocol*) con terminales remotos no era el adecuado.

- No se poseía una línea de respaldo ante fallos de los proveedores SIP, ya sea debido a la falla misma de los proveedores SIP, o a la caída de los servicios de internet

- Las llamadas y videoconferencias IP se escuchaban muy distorsionadas, a veces incluso entrecortadas.

- La central IP DENWA al ser propietaria necesitaba pagar para acceder a módulos necesarios, como, por ejemplo, un módulo para la integración de bases de datos.

- La red LAN del *call center* no estaba segmentaba en VLANs, lo que ocasionaba tormentas de *broadcast* innecesarias y problemas de seguridad. Tampoco se poseía QoS para priorizar los paquetes de voz sobre los de datos.

- No poseía un firewall/proxy para crear políticas de seguridad en la red.

La solución plantea realizar un diseño, considerando la escalabilidad, integración y calidad de servicio para la voz, en tres etapas:

- Primera Etapa: Implementación de las VLAN, QoS, *firewall. -proxy* y la VPN IPSec en la sede Camaná.

- Segunda Etapa: Implementación de centrales telefónicas Asterisk en ambas sedes.

- Tercera Etapa: Interconexión de ambas centrales telefónicas IP ASTERISK e implementación de una línea primaria E1 como backup.

El informe de suficiencia profesional está organizado en cuatro capítulos:

- Capítulo I “Planteamiento del problema de ingeniería”. - en donde se explica el problema

de ingeniería y se precisan los objetivos del informe. También se hace una evaluación de la problemática y se establecen los alcances del proyecto desarrollado.

- Capítulo II “Marco teórico conceptual”. - En este capítulo se exponen las bases teóricas conceptuales más importantes para la comprensión del sistema descrito en el presente informe. Estos se enfocan los siguientes ítems: servicios de VoIP, protocolos de comunicación utilizados, el *Asterisk* como solución de comunicaciones de telefonía IP.

- Capitulo III “Hipótesis y variables “. -En este capítulo se definen la hipótesis y variables.

- Capítulo IV “Metodología para la solución del problema”. - se enfoca en la solución de escalabilidad, integración y con calidad de servicio para la voz: Implementación de las VLAN, QoS, firewall. Proxy y la VPN IPSec en la sede Camaná; implementación de centrales telefónicas Asterisk; interconexión de ambas centrales telefónicas IP Asterisk.

CAPÍTULO I

PLANTEAMIENTO DE INGENIERÍA DEL PROBLEMA

En este capítulo se explica el problema de ingeniería y se precisan los objetivos del informe. También se hace una evaluación de la problemática y se establecen los alcances del proyecto desarrollado.

1.1 Descripción del problema

Los problemas son la inadecuada gestión y limitaciones del equipamiento acorde a los nuevos requerimientos de comunicaciones de la organización.

1.2 Objetivos del trabajo

El objetivo consiste en mejorar la capacidad de comunicaciones e interconexión de un sistema de telefonía mediante la implementación de centrales telefónica IP basadas en Asterisk y de una línea primaria E1.

1.3 Evaluación del problema

En este Informe de suficiencia se expone el desarrollo de la solución para un caso de estudio particular, un c*all center* de un estudio jurídico encargado de las cobranzas.

En un *call center* es de suma importancia la disponibilidad y estabilidad de su central telefónica y a su vez otro factor es el económico o sea minimizar los costos.

El *call center* posee dos locales uno el principal donde se concentra la mayoría del tráfico IP con alrededor de 70 anexos y otro donde solo posee una oficina de atención al cliente con no más de 5 anexos IP.

El *call center* deseaba aumentar la cantidad de anexos al alrededor del segundo local mencionado, con lo cual el tráfico iba ser mucho mayor, por ende, ya no era razonable tener terminales remotos debido al gran consumo de ancho de banda de la sede remota. Además de los retardos propios de la red IP del proveedor, es necesario recalcar que la red IP es tipo *Best Effort*.

La situación antes de la implementación de la solución tenía varios inconvenientes:

- La empresa poseía una IP PBX propietaria llamada DENWA, la cual ya tuvo problemas de hardware (mal funcionamiento del ventilador del equipo, lo cual provoco que se quemara la *mainboard* del equipo).

- El tiempo para establecer una sesión SIP (*Session Initiation Protocol*) con terminales remotos no era el adecuado.

- El local principal no poseía una línea de *backup* ante fallos de los proveedores SIP, ya sea debido a la falla misma de los proveedores SIP, o a la caída de los servicios de internet.

- Las llamadas y videoconferencias IP se escuchaban muy robotizadas hasta a veces entrecortadas.

- La central IP DENWA al ser propietaria necesitaba pagar para acceder a módulos necesarios, como, por ejemplo, un módulo para la integración de bases de datos.

- La red LAN del *call center* no estaba segmentaba en VLANs causando tormentas de *broadcast* innecesarias y problemas de seguridad, ni poseía QoS para priorizar paquetes de voz sobre los de datos.

- Ambos locales no poseían un *firewall-proxy* para crear políticas de seguridad en la red.

1.4 Alcance del trabajo

La solución planteada al *call center* se centró en varios puntos:

- La solución para superar los problemas de disponibilidad, problemas con pagos de licencias y compatibilidad se recomendó usar una central telefónica IP basada en Asterisk.

- La solución para la línea *backup* en casos de fallos, se propuso la instalación de una línea digital RDSI, con un E1.

- La solución para superar el problema de las llamadas (*delay*, perdida de paquetes, jitter), como el *call center* poseía *switches* con capacidad para soportar el protocolo 802.1Q y calidad de servicio (QoS), se segmentó la red de la empresa en la VLANs necesarias y a su vez configurar QoS para priorizar los paquetes de voz o videoconferencia a paquetes de datos en los todos los dispositivos de red (*switches, routers, firewalls*, etc.).

En la primera oficina se implementó un firewall y un proxy (por separado, para no saturar mucho tráfico ambos equipos), ambos basados en tecnologías *Open Source*:

- Firewall: Sistema basado en software Open Source Vyatta.

- Servidor proxy-cache: Un software Squid con CentOS como SO.

En la segunda oficina también se implementó un firewall basado en *Open Source* Vyatta, este firewall también ayudará a establecer la VPN entre ambas sedes.

Los anexos remotos en la segunda oficina aumentaron considerablemente, por lo que se optó por integrar otra central telefónica IP basada en Asterisk. Para la interconexión entre ambas sedes se implementó, primero una VPN a nivel de capa 3 del tipo IPSec, entre ambas sedes y luego una troncal IAX2 entre ambas centrales IP Asterisk.

La solución de escalabilidad, integración y con calidad de servicio para la voz se dio en tres etapas:

- Primera Etapa: Implementación de las VLAN, QoS, firewall. Proxy y la VPN IPSec en la sede Camaná.

- Segunda Etapa: Implementación de centrales telefónicas Asterisk.

- Tercera Etapa: Interconexión de ambas centrales telefónicas IP ASTERISK, implementación de la línea primaria E1 como backup.

En el capítulo 4 de explica el desarrollo de la solución mencionada, etapa por etapa.

CAPÍTULO II

CONCEPTOS BASICOS DE PROTOCOLOS USADOS EN TELEFONIA IP

En este capítulo se exponen las bases teóricas conceptuales más importantes para la comprensión del sistema descrito en el presente informe. Estos se enfocan los siguientes ítems: Servicios de VoIP, Protocolos de comunicación utilizados, el Asterisk como solución de comunicaciones de telefonía IP.

2.1 Servicios de VoIP

A continuación, se desarrolla en esta sección, los aspectos generales (PSTN, PBX, RDSI), la telefonía IP y los *Gateways* VoIP.

2.1.1 PBX analógicas e IP

Las antiguas PBX analógicas de la PSTN (*Public Switched Telephony Network*) están siendo actualmente reemplazadas por las centrales telefónicas basadas en el protocolo IP mejor conocidas como centrales telefónicas IP.

Las PBX antiguas se basaban en la tecnología de conmutación de circuitos a diferencia de las de telefonía IP que se basa en la conmutación por paquetes. Las PBX basadas en IP se suelen llamar IP PBX.

La gran ventaja de basar la telefonía en el protocolo IP es que la gran mayoría de tecnología de las actuales, convergen a IP, con lo cual se puede fácilmente desplegar la red de telefonía.

2.1.2 Telefonía IP y VoIP

Los términos telefonía IP y VoIP suelen confundirse, pero indican conceptos distintos. *VoIP (Voice Over IP)* es una tecnología (dispositivos, protocolos), ella se refiere a todos los recursos que necesita la voz analógica para poder ser transportado por una red basada en IP, es decir digitalizar la voz, *media gateways* para la interconexión con otras tecnologías, equipos de conmutación (*switch, routers*), etc. La telefonía IP es el servicio que usa la tecnología *VoIP* para el transporte de la voz.

2.2 Protocolos de comunicación utilizados

En la telefonía IP como en la tradicional, una llamada siempre se divide en dos fases:

- Fase de señalización (establecimiento de la llamada).

- Fase de transporte de voz.

En el caso de la telefonía IP, para la fase de señalización se usa varios protocolos como: H.323, SIP and IAX (secciones 2.2.1 y 2.2.2).

La fase del transporte de la voz usa principalmente el protocolo RTP (*Real Time Transport Protocol*) y su par el protocolo RTCP (*Real Time Control Protocol*) (secciones

2.2.3 y 2.2.4), este último es usado como un protocolo de control de RTP, los cuales son explicados a continuación:

2.2.1 SIP (*Session Initiation Protocol*)

SIP es un protocolo de señalización usado en redes VoIP, por lo tanto, SIP se encarga de:

- Establecer una sesión.

- Administrar una sesión.

- Dar por terminado una SIP.

El protocolo SIP solo de encarga señalizar mas no de transportar paquetes voz en las redes IP, pero es la primera etapa imprescindible para lograr una comunicación, sin esta no sería posible.

Las sesiones SIP establecen: conferencias en Internet, llamadas telefónicas por internet o juegos en línea, etc. Existen dos versiones: SIP, SIP v1 que apareció en 1999 y SIP v2 que apareció en 2002.

El protocolo SIP también tiene la característica de estar basado en HTTP en la forma de funcionar, o sea en texto claro lo cual lo hace más entendible para su estudio y resolución de problemas y SMTP en la forma de estructura del direccionamiento de los terminales.

El protocolo SIP, según el modelo TCP/IP, está ubicado en la capa de aplicación. Una sesión es una comunicación *two-way* o sea de 2 vías.

a. Tipos de sesiones SIP

Los tipos de sesiones SIP son las siguientes:

a.1 Sesiones SIP establecidos directamente entre dos dispositivos

Nosotros decimos que dos dispositivos están directamente conectados siempre y cuando sean localizables uno al otro mediante una IP estática, o sea, se tratará de una comunicación *device-to-device*. Esto es posible cuando dos terminales SIP pueden alcanzar al otro sin ningún servidor intermediario. En la Figura 2.1 se observa el establecimiento del tipo de sesión mencionado

a.2 Sesiones SIP establecidos mediante un servidor SIP

Por el contrario, también se puede establecer una comunicación de del tipo *user-to-user*, esto es muy usual ya que muy frecuente se usan distintas direcciones IPs, ya sea debido al tener un servicio desde el ISP que asigne direcciones IP dinámicas o al usar distintos dispositivos en lugares remotos. En este último caso se usaría un servidor SIP intermedio que ayudaría a identificar al host remoto, un ejemplo análogo son los servidores DNS los cuales resuelven dirección URL o los servidores SMTP cuando se envía un correo

solo usando una dirección de *e-mail.* En la Figura 2.2 se observa el establecimiento del tipo de sesión mencionado

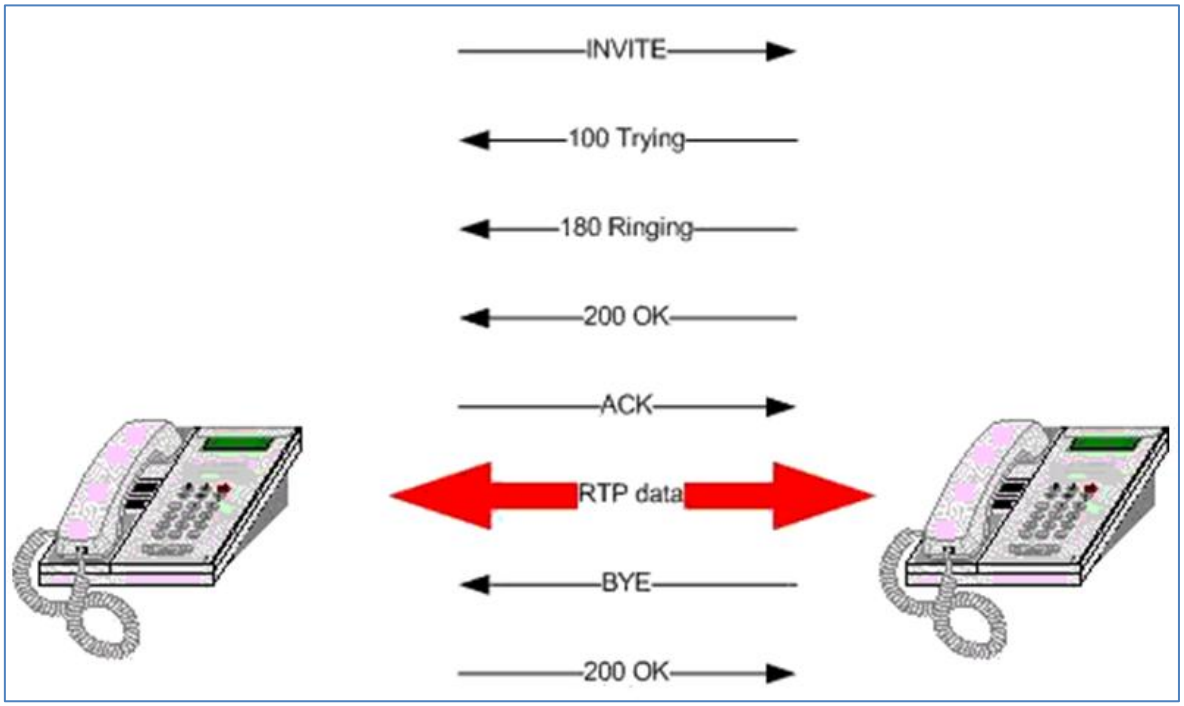

Figura 2.1 Comunicación *device-to-device (*Fuente: Elab. propia)

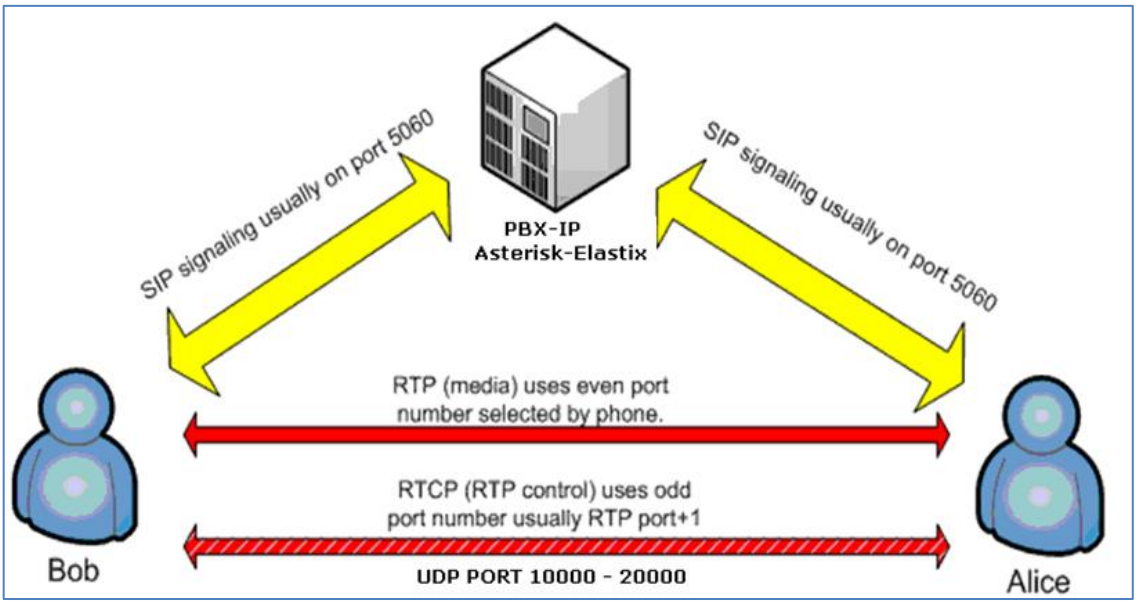

Figura 2.2 Comunicación *Proxy SIP-to-device (*Fuente: Elab. propia)

b. Características de SIP

El protocolo SIP tiene cuatro características que lo hacen muy usable en VoIP:

- Localización de usuario: El protocolo SIP es capaz de poder ubicar dinámicamente cada terminal SIP, esto es posible por el mensaje SIP REGISTER que envía cada terminal al PROXY SIP el cual almacena la información de la ubicación remota.

- Capacidad de negociación: SIP es muy versátil a la hora de negociar parámetros para establecer una sesión, como los puertos (por lo general de tipo UDP) y los *CODEC* (los cuales son los métodos de compresión de datos digitales) a usar, así mismo de las direcciones IP de cada terminal SIP.

- Aviso de disponibilidad: El protocolo de señalización SIP es capaz de informar a otro terminal si esta con su canal ocupado, o está libre para establecer una sesión con otro terminal SIP.

- Administración de la sesión: el protocolo SIP también permite modificar, transferir y finalizar una sesión SIP establecida, a su vez informa del estado actual de la sesión (estado ocupado, libre, intentando establecer o liberando un canal SIP).

c. Esquema de direcciones SIP

SIP usa una dirección conocida como URI (*Uniform Resource Identifier*) que usa un esquema muy parecido a un URL para representar a un usuario:

SIP URI = sip:x@y:Puerto

Donde

- x=Nombre de usuario

- y=equipo (dominio o IP)

Puerto: por defecto usa como puerto destino el 5060 del tipo UDP

Un ejemplo de URI:

sip:support@arribaperu.peru.com

d. Dispositivos en una comunicación SIP

User Agent (UA):

Los UA se clasifican en dos *User agent client* (UAC), los cuales son aquellos que reciben peticiones SIP del otro elemento llamado *User Agent Server* (UAS), el cual es el encargado de aceptar las peticiones SIP hechas por el UAC. Un ejemplo de esto sería un teléfono IP (UAC) y un servidor SIP (UAS).

Proxy/Server SIP:

Usualmente todo dispositivo SIP necesita registrarse en un server SIP el cual guarda la información de la localización (la IP del dispositivo y nombre de usuario), a su vez se necesita un proxy SIP el cual es el encargado de conmutar las llamadas SIP mediante la información dada por el server SIP, generalmente estas dos funciones están incluidas en un solo equipo el cual registra, resuelve y conmuta las llamadas SIP, pero para una mejor escalabilidad y seguridad se puede implementar ambos servicios por separado: un server SIP, que se encargue únicamente del registro y un proxy SIP encargado únicamente de la conmutación de llamadas (enrutamiento).

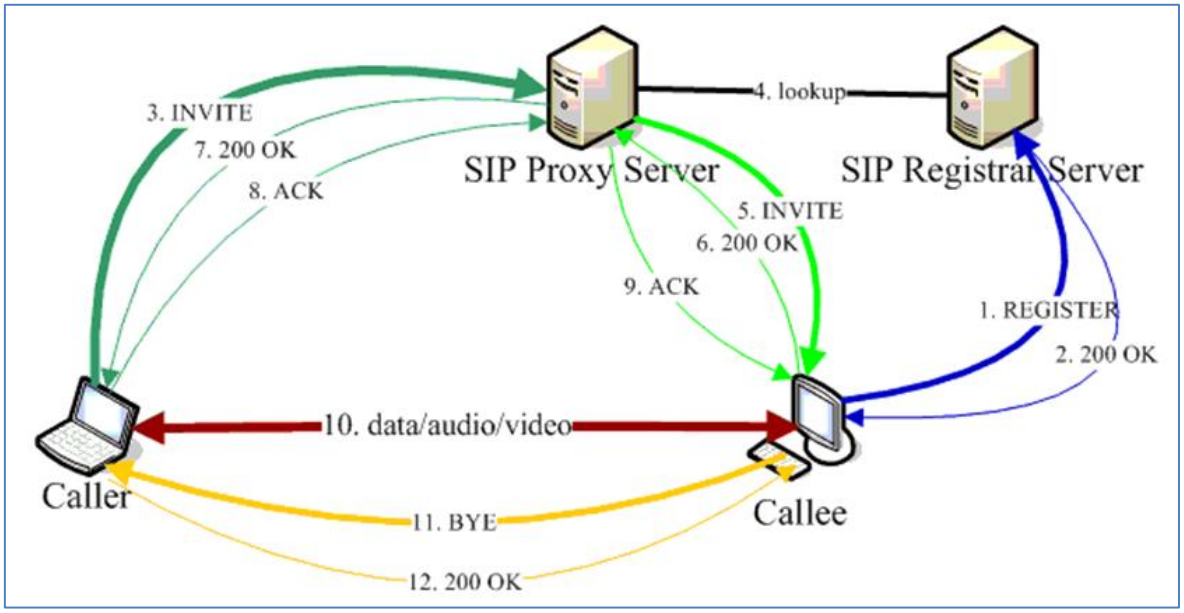

Figura 2.3 Enrutamiento de llamadas (Elab. propia)

En la figura 2,3 se observa que el flujo del registro se da únicamente con el 'SIP registrar server ' y cuando el *'caller'* quiere intenta comunicarse con el '*callee'* este va directamente al SIP proxy server, el cual hace una búsqueda del registro del *callee* en el 'SIP *register server* ', el cual le responde con la ubicación del terminal '*callee'* y donde luego se puede establecer la comunicación ya directamente.

e. Mensajes SIP/SDP

Los mensajes SIP son intercambiados entre UACs y los UASs , los cuales se dividen en dos grandes grupos: peticiones SIP y respuestas SIP.

e.1 Peticiones SIP

Las peticiones SIP son mensajes hechas por los UAC, algunos de los cuales son:

- INVITE: Es el mensaje de petición hecho cuando un terminal desea comunicarse con otro.

- ACK: Es la petición hecha por el que envío el INVITE para hacer saber al otro terminal que recibió su mensaje 200 OK (este es un mensaje del tipo respuesta).

- BYE: Es el mensaje enviado para dar por terminado una sesión SIP.

- CANCEL: Es el mensaje SIP para dar por terminada una sesión SIP que se intentando establecer.

- OPTIONS: Son peticiones SIP usadas por los UA a los UAC para requerir información, por lo general se envían de forma continua en una sesión SIP ya establecida.

- REGISTER: Es una de las primeras peticiones que realiza un UAC para poder registrarse contra un server SIP, esto le permitirá al terminal SIP ser localizado.

e.2 Respuestas SIP

Todo mensaje del tipo petición en el protocolo SIP, va a ir acompañada por un mensaje respuesta SIP, estos mensajes son identificados por números siguiendo un patrón. La tabla 2.1 muestra el total de códigos de respuesta.

Tabla 2.1 Códigos de respuesta (RFC 2616)

Description	Code	Examples	
Informational or provisional response	1XX	100 Trying 180 Ringing 181 Call Is Being Forwarded	182 Queued
Success	2XX	200 OK 202 Accepted	
Redirect	3XX	300 Multiple Choices 301 Moved Permanently 302 Moved Temporarily	303 See Other 305 Use Proxy 380 Alternative Service
Client Errors	4XX	400 Bad Request 401 Unauthorized 402 Payment Required 403 Forbidden 404 Not Found 405 Method Not Allowed 406 Not Acceptable 407 Proxy Authentication Required 408 Request Timeout 409 Conflict 410 Gone	411 Length Required 413 Request Entity Too Large 414 Request-URI Too Large 415 Unsupported Media Type 420 Bad Extension 480 Temporarily not available 481 Call Leg/Transaction Does Not Exist 482 Loop Detected 483 Too Many Hops 484 Address Incomplete 485 Ambiguous 486 Busy Here
Server Errors	5XX	500 Internal Server Error 501 Not Implemented 502 Bad Gateway	503 Service Unavailable 504 Gateway Time-out 505 SIP Version not supported
Global Errors	6XX	600 Busy Everywhere 603 Decline	604 Does not exist anywhere 606 Not Acceptable

1XX: Son respuestas SIP, que indican un estado transitorio de la comunicación.

Ejemplos:

- 100: *Trying*, es el mensaje mediante el cual se está intentando comunicarse un terminal con otro,

- 180: *Ringing*, Este mensaje se ve reflejado en el timbrado del teléfono.

2XX: Son el grupo de respuestas relacionadas a las respuestas exitosas de una petición SIP. Ejemplo:

- 200: Por ejemplo, al establecerse un registro exitoso el server SIP envía un 200 OK

3XX: Son el grupo de respuestas hechas para informar que las peticiones han sido reenviadas a otro dispositivo SIP.

4XX: Son respuestas SIP hechas por el UAC debido a sus errores (el cliente SIP)

5XX: Son respuestas SIP hechas por el server SIP debido a errores.

6XX: Son respuestas SIP relacionados a errores en general.

2.2.2 IAX (*Inter-Asterisk eXchange*)

IAX es otro protocolo usado para la señalización en telefonía IP o *streaming media* (video conferencia, videos en tiempo real, etc.), originalmente fue diseñado por la comunidad Asterisk para la señalización entre PBX Asterisk, lo comúnmente llamado troncales IAX se suele usar en producción la versión 2 del protocolo, que vendría a ser el protocolo IAX2.

IAX2 usa como protocolo de transporte UDP con puerto 4569, por este único puerto se realiza la señalización y el transporte del streaming.

IAX2 como protocolo de señalización tiene grandes ventajas como:

- Es un protocolo de fácil implementación.

- Funciona con una gran cantidad de CODEC.

- Es un protocolo de señalización *in-band* (usa el mismo canal tanto para señalización como para *streaming*).

- Es muy transparente a los *firewalls.*

- Es un protocolo binario, a diferencia de SIP que está basado en texto.

- Usa menos ancho de banda del canal, al usar cabeceras más pequeñas y transportar múltiples tramas en una solo cabecera.

- No necesita de un protocolo adicional como en SIP para transportar el streaming

a. Funcionamiento IAX2

El protocolo IAX2 usa pocos mensajes para establecer, administrar y dar de baja a una sesión, en el siguiente esquema se puede visualizar la secuencia de mensajes en el protocolo IAX2, desde el momento del establecimiento hasta la finalización de la sesión.

Como se puede visualizar en la figura 2.4, IAX2 está compuesta de tres etapas:

- Etapa Establecimiento.

- Etapa del transporte del audio.

- Etapa de la finalización de la sesión.

En la primera etapa un terminal intenta comunicarse con otro enviando un mensaje *NEW*, a su vez el otro terminal responde con un mensaje *ACCEPT*, por último, el primero envía un mensaje *ACK*, haciendo indicar que ha recibido el mensaje *ACCEPT*, luego de esto el terminal remoto envía un mensaje *RINGING*, indicando que el terminal está timbrando, al contestar este último envía un mensaje *ANSWER*.

En la segunda etapa, ya establecida la sesión, se da únicamente el transporte del *streaming*.

En la tercera etapa es la desconexión para lo cual el terminal local envía un mensaje *HANGUP*, lo que a su vez el terminal remoto responde con un mensaje *ACK* dando por terminado la sesión.

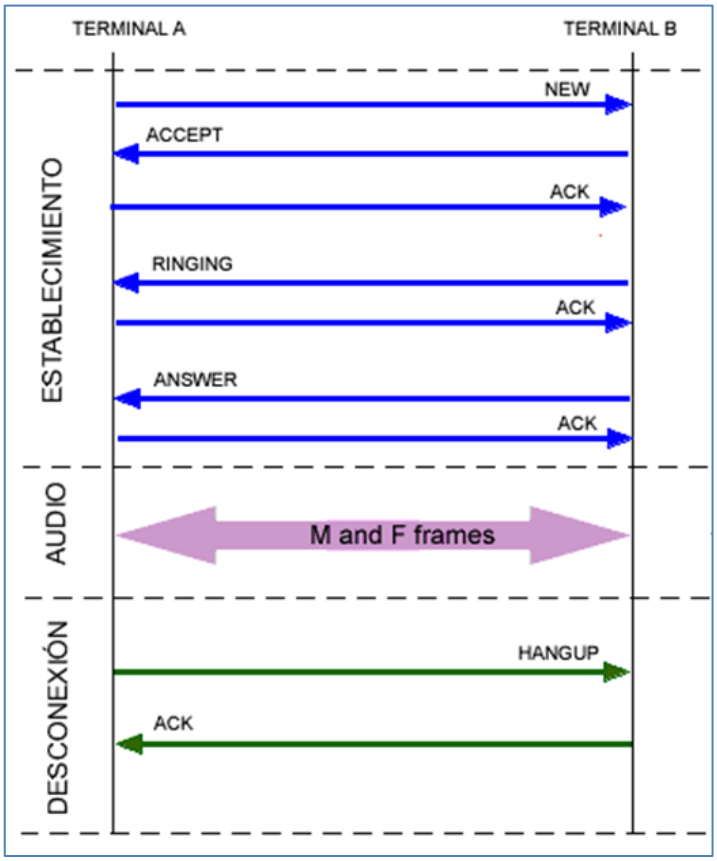

Figura 2.4 Tres etapas de IAX (Elab. propia)

2.2.3 RTP (*Real time Transport Protocol*) and RTCP *(Real time Transport Protocol)*

El protocolo **RTP** se dedica exclusivamente al transporte y problemas relacionados con los servicios multimedia de tiempo real como: la música, la telefonía IP, videoconferencia, televisión and radio IP, etc.

El protocolo RTP funciona de la mano de otro protocolo llamado RTCP (RTP *Control Protocol*), el cual tiene la función de controlar la calidad y seguridad en los servicios proporcionados

Los protocolos RTP como RTCP, según el modelo TCP/IP, están ubicados en la capa de aplicación, ambos usan como protocolo de transporte a UDP, esto último mencionado es muy necesario para evitar retrasos en la llegada de información en tiempo real.

A continuación, se describe brevemente el funcionamiento de ambos protocolos.

El protocolo RTP es el encargado de multiplexar/demultiplexar flujo de información *unicast o multicast* para así poder enviar un solo flujo de información, para lograr esto RTP usa dos componentes principales, reflejados en la cabecera de RTP:

- Número de secuencia

- *Time-stamping* (Marcas de tiempo)

Con el número de secuencia dada por el origen, luego el receptor es capaz de ordenar la secuencia de información, y con el time-stamping igualmente dada en el origen es capaz el receptor de saber en qué tiempo empezó el flujo de información. Con ambos componentes mencionados es posible construir un buffer de la aplicación en tiempo real.

Además, RTP ya que usa UDP, es el encargado de solucionar problemas relacionados a perdida de paquetes mediante la interpolación de datos.

La cabecera RTP es formada por tres palabras cada una de 32 bits, pero esta también puede poseer campos adicionales según se convenga, como se muestra a continuación.

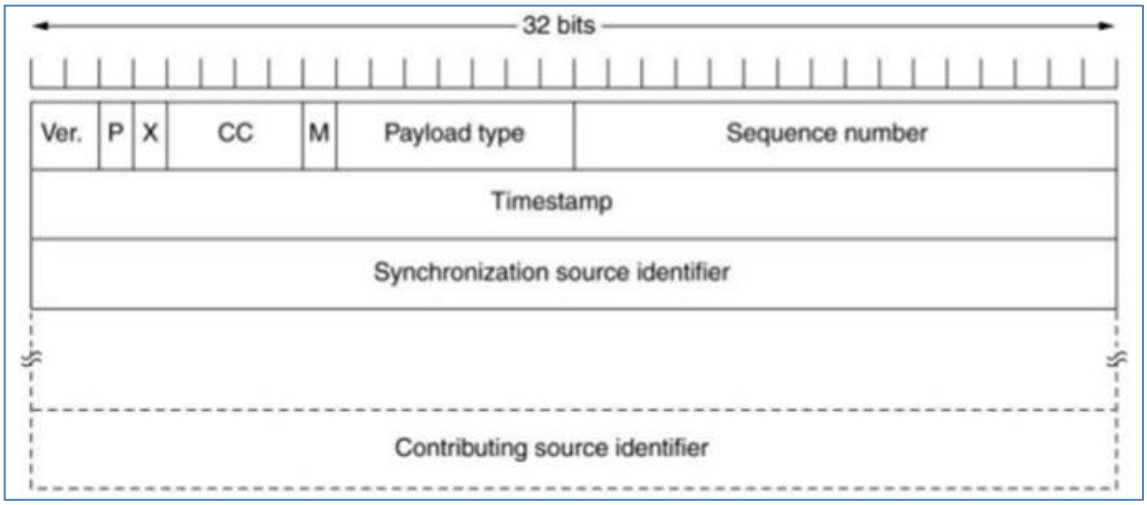

Figura 2.5 Cabecera RTP (Fuente: RFC 1889)

- Ver: Indica la versión usada de RTP (2 bits).

- P: Es 1 si la cabecera se ha llenado con un número de bits múltiplo de 4, en caso contrario P toma el valor de 0. (1 bit)

- X: Es 1 si existe campos adicionales, 0 en caso contrario (1 bit).

- *Payload type*: Indica que algoritmo de codificación se está usando.

- *Sequence Number:* Es el número de secuencia de cada paquete RTP, este número se va incrementando según haya más paquetes RTP (16 bits).

- *Timestamp*: Indica el tiempo en se creó el primer paquete del flujo RTP. (32 bits).

- *Synchronization source identifier* : Es el campo que identifica que flujo RTP pertenece un serie de paquetes RTP.

El protocolo RTCP, es usado junto a RTP para dotarle de seguridad ya que un atacante podría hacerse pasar por un agente RTP, o cambiar campos de la cabecera como el tipo de codificación, con lo cual sería vulnerables a decodificar la información, también es usado como un protocolo de control antes fallas en RTP.

Al igual que el protocolo RTP, RTCP usa UDP como protocolo de transporte además usa puertos adyacentes a los que usa RTP.

El funcionamiento de ambos protocolos se puede observar la figura 2.6.

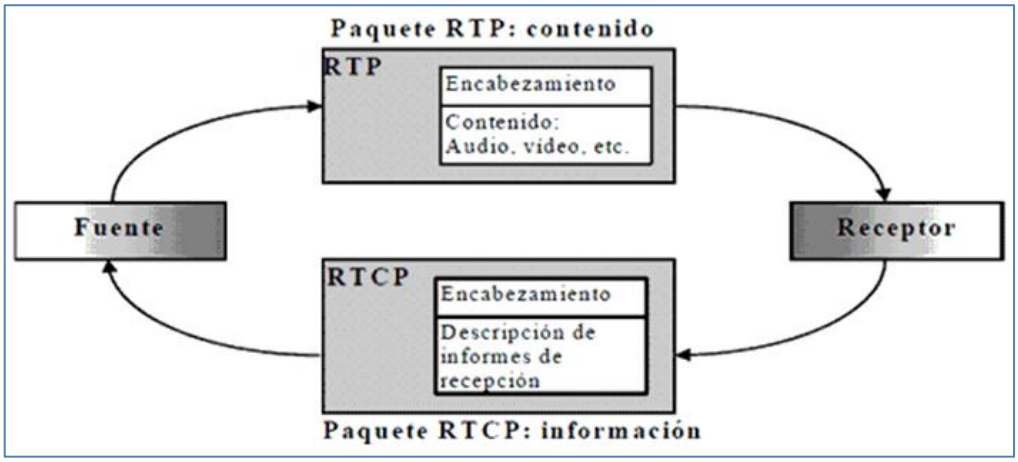

Figura 2.6 Funcionamiento de protocolos RTP y RTCP (Fuente RFC 3605)

El protocolo RTP envía la información desde la fuente al receptor usando UDP, a su vez el receptor envía información de control (por ejemplo, tiempos de envío de flujos RTP, calidad, etc.) RTCP y así de esta manera se sincronizarán dos tipos de flujos (voz e imagen) como por ejemplo en una videoconferencia o un *streaming* de video.

2.3 Arquitectura VoIP

La arquitectura de VoIP está dividida en dos: señalización y flujo de datos, como se puede observar en la figura 2.7 siguiendo el esquema del modelo TCP/IP.

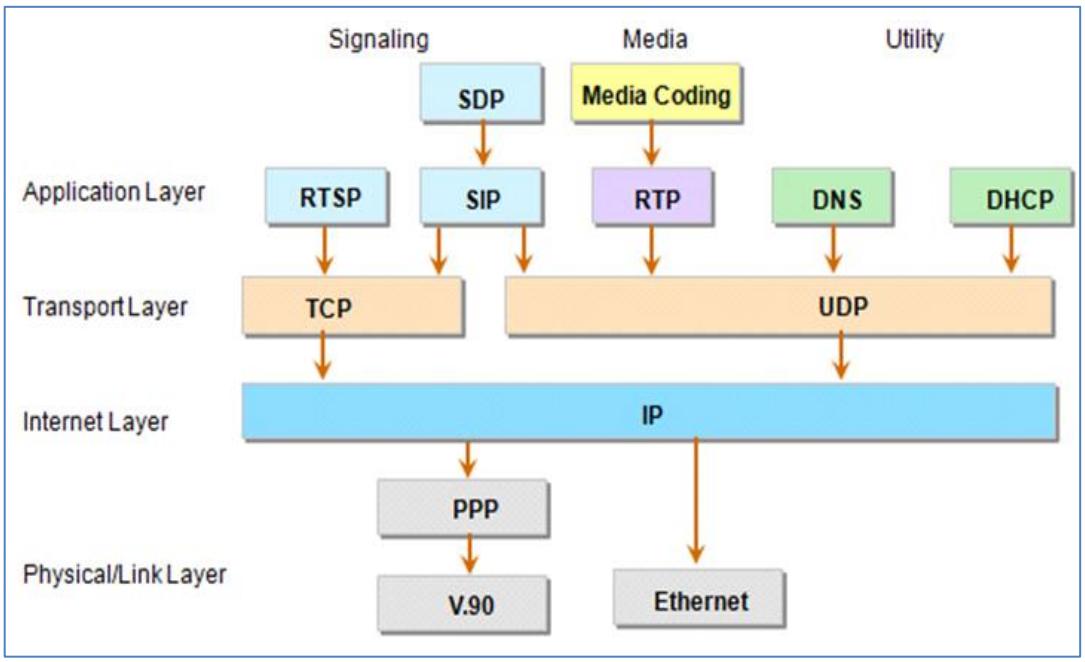

Figura 2.7 Arquitectura de VoIP (Elab. propia)

En la figura 2.7 también se observa que el protocolo SDP, que es un protocolo basado en texto, es usado conjuntamente con SIP en algunos de sus mensajes, como por ejemplo en el *INVITE*, donde se proporciona la información de los *CODEC* a negociar con el otro terminal. También se puede ver como todos los protocolos están basados en el protocolo IP en la capa de internet el cual es el medio de transporte actual más usado en la red.

2.4 CODEC

La telefonía IP necesita un interfaz que codifique y decodifique una señal análoga, en este caso la voz humana que es recibida por un transceptor a una señal digital, la cual luego será transportada por la red IP, este elemento es el CODEC, que en si es un algoritmo de compresión, existen muchos tipos de CODEC dependiendo del tipo del algoritmo usado, los requerimientos computacionales usados, o la calidad del audio o la cantidad del ancho de banda usado.

Algunos de estos CODEC se listan a continuación:

G.711:

G.711 es el CODEC que usa PCM (*Pulse Codification Modulation*), por lo cual no usa compresión, esto lo hace a su vez un CODEC con bajo consumo para el procesamiento debido a su poca complejidad, además poseer audio de buena calidad. El CODEC G.711, toma 8 000 muestras por segundo, usando: 8 bits para los diferentes niveles de las amplitudes de la señal, se tiene entonces que usamos 64 000 bits por segundo o 64kbps,

el cual sería el consumo usando el CODEC G. 711.

G.729:

G.729 es un CODEC que comprime los datos tomando muestras a una tasa de 10 ms, con lo cual genera un bit rate de 8 Kbps usando el algoritmo *codec-excited linear prediction speech coding* (CS – ACELP).

2.5 Media Gateway

El Media Gateway es la interfaz del mundo VoIP con otras tecnologías como la red conmutada tradicional PSTN, la redes RDSI o las redes móviles 2G, 3G o sea la conversión del formato desde una tecnología a otra.

Los equipos media *gateway* presentan las interfaces necesarias para interactuar con las distintas tecnologías a usar, cada una de estas pertenecen a tarjetas que manejan por ejemplo Ethernet, RDSI, PSTN (entradas FXS, FXO).

Los Media Gateway son usados para interconexión con las redes ya mencionadas, sus usos principales son:

- Implementar tecnología SIP *trunking* para PBX antiguas.

- Proteger y extender las redes VoIP a redes TDM.

- Evitar gastos al actualizar la PBX.

CAPITULO III

HIPOTESIS Y VARIABLES

En este capítulo se definen la hipótesis y variables.

3.1 HIPÓTESIS

Implementación de una central telefónica IP basada en Asterisk y una línea primaria E1, eliminando así una serie de deficiencias que experimentaba la empresa del caso de estudio.

3.2 VARIABLES

3.2.1 Variables Independientes

Diseñar una red escalable, que posea una buena integración y con calidad de servicio para la voz y una línea de respaldo.

3.2.2 Variables Dependientes

- alta disponibilidad.
- redundancia o Backup.
- seguridad e interconexión.

CAPÍTULO IV

METODOLOGÍA PARA LA SOLUCIÓN DEL PROBLEMA

Este capítulo se enfoca en la solución de escalabilidad, integración y con calidad de servicio para la voz, la cual se efectuó en tres etapas:

- Primera Etapa: Implementación de las VLAN, QoS, firewall. Proxy y la VPN IPSec en la sede Camaná.

- Segunda Etapa: Implementación de centrales telefónicas Asterisk.

- Tercera Etapa: Interconexión de ambas centrales telefónicas IP Asterisk, implementación de la línea primaria E1 como *backup*.

4.1 Primera Etapa

Consiste en la implementación de las VLAN, QoS, firewall. Se desarrolla en seis partes que se explican a continuación:

- Planteamiento del problema y planificación de la solución.

- Segmentación de la red Camaná e implementación de VLAN.

- Implementación de servidor Proxy SQUID.

- Implementación de servidor firewall VYATTA en sede Camaná.

- Topología de la red de Miraflores e implementación de servidor firewall – proxy VYATTA Miraflores.

- Implementación y configuración de la VPN IPsec.

4.1.1 Planteamiento del problema y planificación de la solución

Antes de empezar las labores se encontró varios problemas:

- La red IP de la sede Camaná carecía de una implementación acorde a la envergadura de la red (solo usaba una VLAN).

- La red IP solo usaba un segmento de red: 192.168.1.0/ 24 la cual era ya insuficiente para el direccionamiento para los hosts con el crecimiento proyectado.

- La seguridad de la red tanto de Camaná como de Miraflores ambos carecían de un *firewall,* lo cual lo hacía vulnerable a ataques informáticos.

- Constante *delay*, latencia en los servicios de voz.

- No existía un servidor *proxy* para la administración de los accesos a los contenidos *web* hacia internet, lo cual causaba congestión en la red de Camaná y mal uso del acceso a las páginas webs.

- El “Estudio Jurídico Romero “posee en Lima dos sedes: una ubicada en la Av. Camaná y otra ubicada en el distrito de Miraflores, la interconexión de ambas redes antes de la implementación era nula, o sea no había una VPN implementada entre ambos sitios para que las redes se comporten como una LAN extendida una de la otra.

La solución planteada para los temas mencionados fue:

- Planificación de *subnetting* y *gateways* adecuados en la red de Camaná, para esto se tuvo en cuenta la cantidad de host actual y la proyección a crecimiento de host en cada sub-red.

- La implementación de redes VLAN (802.1Q) en los *switch* de la red de Camaná, con lo cual se logró:

o Dominios de *broadcast* aislados, una por red VLAN, disminuyendo la saturación de tramas *broadcast* en la LAN.

o Una eficiente administración de las diferentes redes VLAN creadas, ya que se puede crear reglas para los accesos de ciertos *host* a ciertos servicios de la red, bloquear tráfico entre segmentos de red, además de emplear configuraciones de QoS y de políticas de protección de puerto por picos de tráfico.

Nota: Todos los *switchs* de la red de Camaná (SG300) soportan el protocolo 802.1Q

- Implementación de la VPN mediante la tecnología *IPsec*, la cual proporciona una VPN segura a nivel de capa 3, protegiendo todo segmento o datagrama de capa 4 además de poseer:

o Integridad, mediante los HASH disponibles para IPsec.

o Confidencialidad, mediante el cifrado de todo el paquete de datos.

o Autenticación, mediante el intercambio de claves con IKE.

La VPN se realizó entre los firewalls VYATTA, los cuales se encuentran detrás de los *routers* instalados por el ISP. Dado que se contaba con un pool de IP públicas, se planteó planificar el *subneting* respectivo entre cada par: firewall VYATTA–*router* ISP, para evitar problemas de NAT y lograr la VPN de manera eficiente. También se planteó implementar,

como se mencionó, dos *firewalls* VYATTA, uno en cada sede, para efectuar políticas de seguridad y la realización de la VPN. Las políticas de seguridad partieron desde el punto de vista restrictivo. Por último, se planteó implementar un *proxy* SQUID en modo no transparente, sobre una distribución LINUX CENTOS v.6, para la administración de los accesos de los servicios *web* y a su vez como un *proxy* cache, para el acceso inmediato.

4.1.2 Segmentación de la red Camaná e implementación de VLANs

Teniendo en cuenta la cantidad de *host* actuales por sub-red y con un proyectado de crecimiento de 25% a 30%, se obtuvo la tabla 4.1, tomando como *gateway* la última dirección valida de cada segmento de red.

Tabla 4.1 Segmentación de la red y asignación del número de VLAN (Elab. propia)

Nombre de la sub-red	Segmento de red	Gateway	Número de VLAN
Servidores	172.16.0.32/27	172.16.0.62	100
Cámaras	172.16.0.64/27	172.16.0.94	120
Teléfonos IP	172.16.0.96/27	172.16.0.126	130
Supervisores	172.16.0.0/28	172.16.0.14	140
Gerencia	172.16.0.16/28	172.16.0.30	150
Sistemas	172.16.0.128/27	172.16.0.158	160
Administración	172.16.0.160/27	172.16.0.190	170
Recaudación	172.16.0.192/27	172.16.0.222	180
Cobranza	172.16.1.0/24	172.16.1.254	190

Además, se usó el segmento 172.16.2.0/24 y la VLAN 1 (por defecto) para el tráfico de la administración de los *switches* Cisco. Las direcciones IP de los *switchs* son mostrados en la tabla 4.2.

Tabla 4.2 Direcciones IP de los switchs (Elab. propia)

Nombre del *switch*	Dirección IP	Tipo	Marca / Modelo
Switch 1	172.16.2.1	Acceso	Cisco / SG300
Switch 2	172.16.2.2	Acceso	Cisco / SG300
Switch 3	172.16.2.3	Acceso	Cisco / SG300
Switch 4	172.16.2.4	Acceso	Cisco / SG300
Switch 5	172.16.2.5	Acceso	Cisco / SG300
Switch 6	172.16.2.6	Acceso	Cisco / SG300
Switch 7	172.16.2.7	Distribución	Cisco / SG300
Switch 8	172.16.2.8	Acceso	Cisco / SG300
Switch 9	172.16.2.9	Acceso	Cisco / SG300

Nota:

- El *switch* 7 se usó como *switch* de distribución para todos los *switches* de acceso, en una topología estrella.

- El *switch* 7 esta funcionado como un *switch* de capa 3 para el enrutamiento del tráfico entre la WAN y las sub-redes LAN.

- El resto de *switchs* están funcionando solo a nivel de capa 2.

En las figuras 4.1 y 4.2, se muestran las configuraciones hechas en el *switch* 7 multicapa que por donde se ha realizado los enlaces troncales con todos los *switchs* de acceso, y se ha configurado las interfaces virtuales y la transformación de un puerto (ge_15) de capa 2 a capa 3 para que se comunique con la interfaz del firewall VYATTA:

```
switch7#sh vlan

Vlan       Name                    Ports                Type     Authorization
---- ---------------- -------------------------- ------------ -------------
 1           1         gi1-4,gi6-12,gi16,gi21,       Default      Required
                       gi23-24,gi26-28,Po1-8
100     servidores     gi4-5,gi10,gi13-14,gi17-19     static      Required
120       camaras           gi20,gi22,gi25-26        static      Required
130    telefonos ip          gi2,gi4,gi7-9           static      Required
140    supervisores            gi2,gi7-8             static      Required
150      gerencia               gi4,gi9              static      Required
160      sistemas               gi4,gi8              static      Required
170   administracion            gi4,gi9              static      Required
180     recaudacion               gi8                static      Required
190      cobranzas            gi1-4,gi7-10           static      Required
200         sql                                      static      Required
210      wireless                 gi2                 static      Required

switch7#
```

Figura 4.1 Configuración de las VLAN (Elab. propia)

```
switch7#
switch7#sh ip int

    IP Address         I/F       Type     Directed   Precedence   Status
                                          Broadcast
------------------- --------- ----------- ---------- ---------- -----------
172.16.0.14/28      vlan 140  Static      disable    No         Valid
172.16.0.30/28      vlan 150  Static      disable    No         Valid
172.16.0.62/27      vlan 100  Static      disable    No         Valid
172.16.0.94/27      vlan 120  Static      disable    No         Valid
172.16.0.126/27     vlan 130  Static      disable    No         Valid
172.16.0.158/27     vlan 160  Static      disable    No         Valid
172.16.0.190/27     vlan 170  Static      disable    No         Valid
172.16.0.222/27     vlan 180  Static      disable    No         Valid
172.16.1.254/24     vlan 190  Static      disable    No         Valid
172.16.2.7/24       vlan 1    Static      disable    No         Valid
172.16.3.2/30       gi15      Static      disable    No         Valid
172.16.4.1/24       vlan 210  Static      disable    No         Valid

switch7#
```

Figura 4.2 Configuración de *gateways* y VLAN de administración (Elab. propia)

4.1.3 Implementación de servidor Proxy SQUID

El proxy fue implementado bajo el software SQUID, el cual se ejecuta sobre una distribución de LINUX llamada CENTOS, versión 6. La configuración de las ACL y las reglas del *proxy* están configuradas en el archivo: /etc/squid/squid.conf

Las listas para páginas *web* filtradas o permitidas están ubicadas en el directorio: /etc/squid/listas.

En primera instancia se tenía planeado usar un proxy transparente pero debido a la envergadura (arriba de los 250 host) y al no contar con un servidor, se convino de usar un computador para el *proxy* y otro para el *firewall*. Se implementó un proxy no transparente, ubicado en la red de servidores (VLAN 100), a donde es enviada toda petición *web,* ya sea del tipo GET o POST.

El proxy tiene los datos de red que se muestran en la tabla 4.3.

Tabla 4.3 Parámetros de red del proxy Camaná (Elab. propia)

Dirección IP	172.16.0.49
Mascara	255.255.255.224
Gateway	172.16.0.62
Puerto Listen (TCP)	3128
Tamaño Cache	10MB
Puerto SSH	22

Nota: La configuración del proxy se realizó de acuerdo con los requerimientos del cliente

4.1.4 Implementación de servidor *firewall* VYATTA en sede Camaná

El *firewall* fue implementado sobre el *software* VYATTA, el cual se ejecuta sobre la distribución LINUX – DEBIAN.

VYATTA es un *firewall* a nivel de *software*, que está orientado a ser usado en soluciones de *networking*, este se encuentra ubicado lógica y físicamente entre el *switch* 7 multicapa y el *router* CISCO proporcionado por el ISP. Como se verá en la topología de la figura 4.3, el *firewall* consta de dos interfaces: eth0 (WAN) y eth1 (LAN) con direcciones Pis:

eth0: 172.16.3.1 / 30

eth1: 200.4.200.194 / 30

I) A nivel de enrutamiento se configuró 2 rutas:

- Ruta hacia la red LAN: 172.16.0.0/16, con siguiente salto la dirección IP de la interfaz ge_15: 172.16.3.2 del switch 7 multicapa:

```
protocols {
  static {
      route 172.16.0.0/16 {
          next-hop 172.16.3.2 {
          }
      }
```

-La ruta default:

```
  gateway-address 200.4.200.193
```

II) También se aplicaron configuraciones estándar de protección contra ataque maliciosos:

```
Firewall {
    all-ping enable
    broadcast-ping disable
    conntrack-expect-table-size 4096
    conntrack-hash-size 4096
    conntrack-table-size 32768
    conntrack-tcp-loose enable
    ipv6-receive-redirects disable
    ipv6-src-route disable
    ip-src-route enable
    log-martians enable
```

III) Como se vio previamente, se implementó un servidor proxy no transparente, para la cual es necesario configurar reglas en el *firewall* con tal que solo el *proxy* sea capaz de hacer consultas HTTP y HTTPS:

```
name proxy {
        default-action drop
```

```
        rule 10 {
            action accept
            source {
                address 172.16.0.49
            }
        }
    rule 20 {
            action drop
            destination {
                address 0.0.0.0/0
                port 443
            }
            protocol tcp
            source {
                address 172.16.0.0/23
            }
        }
          rule 30 {
          action drop
            destination {
                address 0.0.0.0/0
                port 80
}
            protocol tcp
            source {
                address 172.16.0.0/23
```

```
        }
    }
```

IV) Además se abrió un puerto 2235 para el servicio de acceso remoto HTTP hacia el servidor: 172.16.0.48

```
nat {
    rule 21 {
        description avi
        destination {
            address 200.4.200.194
            port 2245
        }
        inbound-interface eth0
        inside-address {
            address 172.16.0.48
            port 80
        }
        protocol tcp
        type destination
    }
```

A continuación, se va a mostrar la topología de lo hablado explicado hasta ahora.

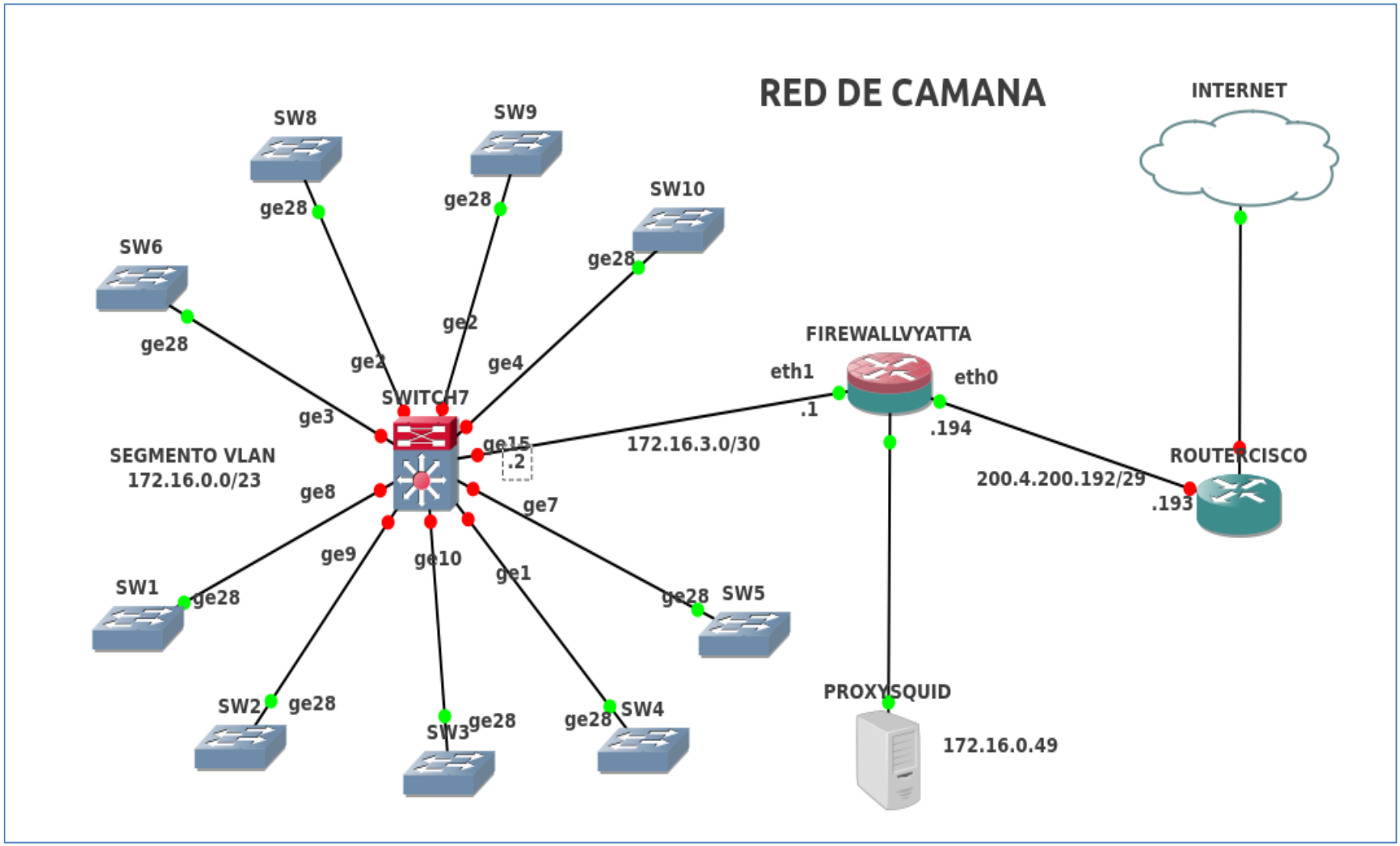

Figura 4.3 Topología de la red Camaná (Elab. propia) Nota:

- En la figura 4.3 se muestra la topología. Era necesario realizar un *subnetting* desde el *pool* de publicas disponibles para asignarse a la interfaz de FW-VYATTA y el router CISCO del ISP. Además, se creó una ruta estática desde en el router CISCO para llegar a la red 172.16.0.0/23.

```
ESTUDIO_JURIDICO_ROMERO_CD104999#sho ip route 172.16.0.0
Routing entry for 172.16.0.0/16, 4 known subnets
  Attached (2 connections)
  Variably subnetted with 4 masks
S       172.16.0.0/16 [1/0] via 200.4.200.194
C       172.16.3.0/30 is directly connected, GigabitEthernet0/1
L       172.16.3.1/32 is directly connected, GigabitEthernet0/1
S       172.16.4.0/24 [1/0] via 172.16.3.2
```

- También en el FW-VYATTA se creó una ruta estática para llegar hasta la red 172.16.0.0/23

```
vyatta@Juridico:~$ show ip route 172.16.0.0/16
Routing entry for 172.16.0.0/16
  Known via "static", distance 1, metric 0, best
  * 172.16.3.2, via eth1
```

4.1.5 Topología de la red de Miraflores e implementación de servidor firewall – proxy VYATTA Miraflores

El *firewall* en Miraflores se implementó por dos causas principales:

- Proteger a la red de Miraflores contra ataques maliciosos informáticos.

- Lograr un VPN hacia la sede de Camaná.

El firewall de Miraflores establece la VPN, NAT, aparte de sus funciones propiamente de firewall. Los datos de red del firewall Miraflores se muestran en la tabla 4.4:

Tabla 4.4 Parámetros de red del Firewall Miraflores (Elab. propia)

Dirección IP WAN (eth0)	190.187.149.146 / 30
Dirección IP LAN (eth1)	192.168.1.200 / 24
Puerto SSH	8412
Next -Hop	190.187.149.145 / 30
Equipo	Vyatta Core 6.3

En la figura 4.4 se muestra la topología respectiva.

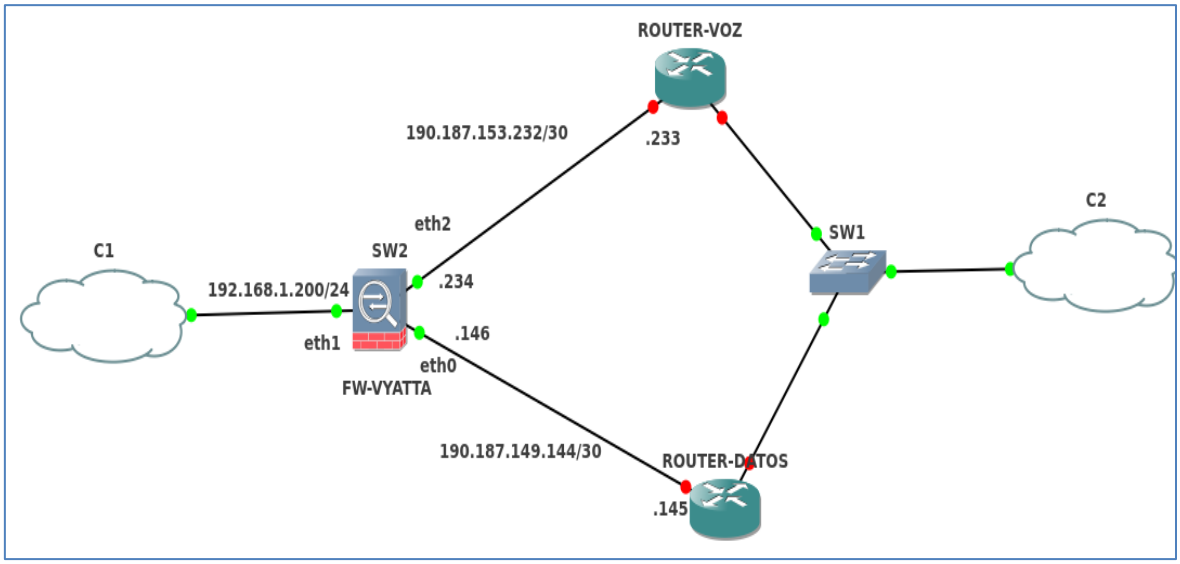

Figura 4.4 Topología Miraflores (Elab. propia)

Notas:

- En el *router* de voz esta deshabilitado todo tráfico HTTP y HTTPS, porque es únicamente usado para voz (proveedores SIP).

- El *router* de datos como el router cisco ISP no contaban con IP publicas planificada, por lo que se subneteo el pool de IP publicas disponibles para asignar IP públicas a cada interfaz:

ethx (Router-Datos) : 190.187.149.143

eth0 (Firewall -VYATTA): 190.187.149.146

- El NAT configurado en el firewall VYATTA es tipo PAT configurado para enmascarar todo el pool privado de la LAN: 192.168.1.0/24

```
rule 9
        {
            outbound-interface eth0
            source {
                address 192.168.1.0/24
            }
            type masquerade
```

- También se abrió un puerto UDP para la conexión de un proveedor SIP y hacer el SIP trunking.

```
rule 2 {
            destination {
                address 0.0.0.0/0
            }
            outbound-interface eth2
            outside-address {
                address 190.187.153.234
            }
            protocol tcp_udp
            source {
                address 192.168.1.133
            }
            type source
```

```
}
rule 7 {
  destination {
      address 190.187.153.234
  }
  inbound-interface eth2
  inside-address {
      address 192.168.1.133
  }
  protocol tcp_udp
  source {
      address 50.30.32.108
  }
  type destination
```

Anotaciones del proxy VYATTA:

El *proxy* esta implementado bajo SQUIDGUARD que viene en el archivo SQUID del propio VYATTA, el cual a su vez está corriendo sobre LINUX para ser más específicos, una distribución DEBIAN.

La configuración para el filtrado *web* está en el archivo: /etc/squid/squidguard.conf

El *proxy* cuenta con 4 grupos los cuales mantienen un rango de direcciones IP, por otro lado, se ha dispuesto integrar una *blacklist* o base de datos de páginas o palabras no permitidas. Ver figura 4.5, figura 4.6 y tabla 4.5 al respecto.

Los rangos de direcciones están configurados como se ha dispuesto en grupos bajo reglas de filtrado de acuerdo con las políticas que se ha establecido para el bloqueo y acceso a las URL tanto de HTTP como de HTTPS

Commit the change.	vyatta@R1# **commit**

Show the updated webproxy-related configuration.	vyatta@R1# **show service webproxy**
	listen-address 192.168.1.254 {
	}
	url-filtering {
	squidguard {
	block-category ads
	block-category spyware
	block-category gambling
	local-block youtube.com
	local-block facebook.com
	local-block-keyword .cn
	local-ok www.company-ads.com
	log local-block
	redirect-url
	“http://192.168.1.254/cgi-bin/squidGuard-simple.cgi?targetclass=%t&url=%u”
	}

Figura 4.5 Reglas globales del proxy (Fuente: Elab. propia)

```
rule 120 {
    local-ok sunat.gob.pe
    local-ok essalud.gob.pe
    local-ok infocorp.com.pe
    local-ok equifax.com
    local-ok netmng.com
    local-ok reniec.gob.pe
    local-ok sunarp.gob.pe
    local-ok paginasblancas.com.pe
    local-ok paginasamarillas.com.pe
    local-ok avg.com
    source-group SUPERVISORES
}
rule 140 {
    default-action block
    local-ok sunat.gob.pe
    local-ok essalud.gob.pe
    local-ok infocorp.com.pe
    local-ok equifax.com
    local-ok netmng.com
    local-ok reniec.gob.pe
    local-ok sunarp.gob.pe
    local-ok paginasblancas.com.pe
    local-ok paginasamarillas.com.pe
    local-ok avg.com
    local-ok sbs.com.pe
    local-ok Afpnet.com.pe
    local-ok .google.com
    local-ok google.com
    source-group TELEFONIAIP
}
source-group CALLCENTER {
    address 192.168.1.201
}
source-group SUPERVISORES {
    address 192.168.1.202
}
source-group TELEFONIAIP {
    address 192.168.1.203
    address 192.168.1.240
```

Figura 4.6 Reglas para grupos de usuarios (Fuente: Elab. propia)

Tabla 4.5 Parámetros del proxy VYATTA (Elab. propia)

Dirección IP	192.168.1.200
Mascara	255.255.255.0
Gateway	172.16.0.62
Puerto Listen (TCP)	3128
Tamaño Cache	10MB
Puerto SSH	22

4.1.6 Implementación y configuración de la VPN IPsec

La VPN se implementó con la tecnología IPsec (VPN a nivel de capa 3). Para ello se empleó los 2 firewalls VYATTA implementados previamente, en donde cada interfaz WAN contaba con IP públicas.

La configuración de la IPsec se puede resumir en las tablas 3.6 y 3.7:

Tabla 4.6 Parámetros Ipsec FW-Camaná (Elab. propia)

IP WAN	200.4.200.194
Cifrado	aes256
HASH	sha-1
IKE	si
Interfaz IPSec	eth0
Sub-red LAN	172.16.0.0/16

Tabla 4.7 Parámetros ipsec FW-Miraflores (Elab. propia)

IP WAN	190.187.149.144
Cifrado	aes256
HASH	sha-1
IKE	si
Interfaz IPSec	eth0
Sub-red LAN	192.168.1.0/24

Luego de configurar estos parámetros se ve que el túnel IPsec se ha establecido:

- Red Camaná :

```
vyatta@Juridico:~$ show vpn ipsec  sa
Peer ID / IP                            Local ID / IP
------------                            -------------
190.187.149.146                         200.4.200.194
    Tunnel  State  Bytes Out/In   Encrypt  Hash  NAT-T  A-Time  L-Time  Proto
    ------  -----  -------------  -------  ----  -----  ------  ------  -----
    2       up     2.3M/113.6M    aes256   sha1  no     2064    3600    all
```

- Red Miraflores:

```
root@fw-romerodec:~# show vpn ipsec  sa
Peer ID / IP                            Local ID / IP
------------                            -------------
200.4.200.194                           190.187.149.146
    Tunnel  State  Bytes Out/In   Encrypt  Hash  NAT-T  A-Time  L-Time  Proto
```

------ ----- ------------- -------- ---- ----- ------- ------- -----

2 up 120.8M/2.5M aes256 sha1 no 2708 3600 all

La topología lógica de la VPN IPsec se muestra en la figura 4.7:

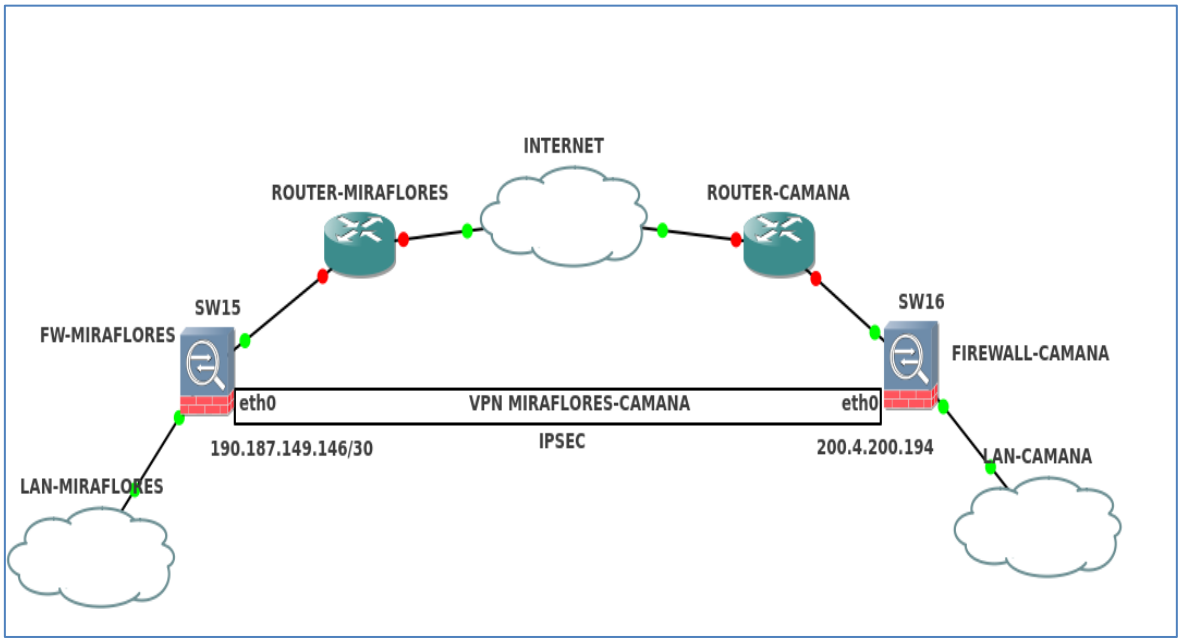

Figura 4.7 Topología VPN ipsec entre las 2 sedes (Elab. propia)

4.1.7 Implementación de políticas de QoS (Quality of Service)

En este punto se va a implementar la calidad de servicio orientada a satisfacer las mayores prioridades del tráfico de voz sobre los datos.

Para garantizar la calidad de servicio mencionado vamos a aplicar una solución muy escalable: DiffServ , el cual analiza los flujos de información en cada dispositivo de conmutación (switch / router) , por lo cual vamos a aprovechar el campo DSCP de cada paquete para identificar a los diferentes flujos de RTP y SIP, para darle prioridad sobre otros tipos de datos. Este es un campo dentro de la cabecera IP que consta de 6 bits.

Como hemos estado viendo los equipos a configurar en cada local (Miraflores y Camaná) son:

-Todos los *switch* CISCO SG300 donde están conectados todos los dispositivos de red y el *switch* central configurado como capa 3 para la comunicación entre todas las LAN.

-El *firewall* VYATTA en ambos locales.

-Los *router* CISCO del proveedor (modelo 1900)

En todos los equipos mencionados se hará una clasificación mediante el campo DSCP que llevará cada paquete, para lo cual en los *softphone* como los teléfonos IP se configurará el valor para etiquetar este campo. Para hallar un valor de diseño seguimos una tabla proporcionada por la empresa CISCO como recomendación:

Application	L3 Classification			L2
	IPP	PHB	DSCP	Cos
Routing	6	CS6	48	6
Voice	5	EF	46	5
Video Conferencing	4	AF41	34	4
Streaming Video	4	CS4	32	4
Mission-Critical Data	3	AF31	26	3
Call Signaling	3	CS3	24	3
Transactional Data	2	AF21	18	2
Network Management	2	CS2	16	2
Bulk Data	1	AF11	10	1
Scavenger	1	CS1	8	1
Best Effort	0	0	0	0

221048

Figura 4.8: *Valores del Campo DSCP*

Con lo cual vemos en la figura que el valor recomendado DSCP para VoIP es 46.

Los dispositivos que etiquetaran este valor son: los *softphones* como los teléfonos IP, en los equipos de conmutación, router *y switch* , no cambiaran dicho campo solo tomaran decisiones a partir de los paquetes ya etiquetados.

Paras las configuraciones en todos los equipos de conmutación siguen los mismos pasos:

Paso 1: Escoger el modo marcado de cada paquete, como ya lo comentamos usamos DSCP.

Paso 2: Habilitar el QoS sobre las interfaces de los *router, switch y firewall,* por donde fluye el tráfico.

Paso 3: Configurar las distintas colas con los algoritmos para administrar las colas, para los *switch* y el *router* del ISP usamos WRR, para los firewalls usamos "*traffic-sharper".*

Paso 4: Mapear los valores del DSCP.

Con la implementación sobre la red ya poseemos por lo menos QoS sobre la red LAN, o sea todo tráfico circulante por la red o entrante a la LAN estará sujeto a las políticas creadas.

4.2 Segunda Etapa

La segunda etapa muestra la implementación de las dos centrales telefónicas IP una ubicada en el local ubicado en la av. Camaná y el otro ubicado en el distrito de Miraflores.

La central IP basada en Asterisk corre mejor sobre un sistema basado en UNIX, como por ejemplo las distribuciones de Linux, las distribuciones más usadas en servidores (ya que poseen bastantes repositorios disponibles y soporte por la comunidad LINUX) son Debian, CentOS, en este caso se usó el segundo, por lo que se instaló el S.O. CentOS 6.0. Por último, se instala la central IP Asterisk. Como se desea interactuar la central con la tecnología digital RDSI, aparte de la instalación del Asterisk propiamente dicha, se necesitan dos paquetes de *software*:

- Protocolos de comunicación de telefonía digital de capa 2 y 3 (Q.921 y Q.931)

- Drivers para la integración de una tarjeta digital RDSI de capacidad de un E1

Para cumplir con todos los requisitos mencionados se va a dividir la instalación en 3 pasos:

Nota: Es necesario en el momento de la instalación deshabilitar la seguridad proveída por el SO, para que no haya problemas de instalación.

Primero se deshabilita el Selinux

#setenforce 0

#getenforce

A continuación, el IPTABLES.

#iptables -F

#service iptables save

Se debe tener presente que al acabar la instalación se deben habilitar ambos servicios de seguridad.

4.2.1 Actualizar el Kernel

El *kernel* en Linux es modular, por lo que al actualizar el kernel muchos módulos siempre se mejoran para:

- Solucionar agujeros de seguridad descubiertos.

- Dar mayor estabilidad al sistema operativo.

- Actualizar drivers que usa el kernel para interactuar con distintos dispositivos.

- Mejorar la velocidad de los procesos realizados por el kernel.

Esto se realiza ingresando a CLI de CentOS el siguiente comando.

 # yum install kernel kernel-devel kernel headers

 # yum update

Luego se procede a reiniciar el servidor para que el CentOS use el nuevo kernel.

Nota: Muchas veces hay problemas de sincronización con los repositorios, por eso es recomendable instalar el servidor ntp para actualizar el tiempo.

#yum install -y ntp && ntpdate pool.ntp.org && chkconfig ntpd on && service ntpd star

4.2.2 Instalación de todas las librerías necesarias que necesita ASTERISK para funcionar de manera correcta

Asterisk para su correcto funcionamiento de necesita de ciertos repositorios para su correcta integración con el S.O. *CentOS*, paquetes, por ejemplo: lenguaje de programación C++, drivers de conexión ODBC para la integración de cualquier base de datos relacionales (Postgres, SQL server, MySQL, etc.) con el Asterisk, la librería de software libxml2 para el análisis de documentos XML, *ncurses* que sirve para crear interfaces basadas en texto, etc. La instalación se realiza con el siguiente comando:

#yum install gcc ncurses-devel make gcc-c++ compat-libtermcap zlib-devel libtool bison-devel bison openssl-devel bzip2-devel wget newt-devel subversion flex gtk2-devel libxml2 libxml2-develunixODBC unixODBC-devel mysql-connector-odbc libtool-ltdl-devel -y

Además, se procederá a instalar paquetes necesarios para dar distintas funcionalidades al Asterisk, como el leguaje PHP y la base de datos MySQL.

#yum install mysql mysql-server mysql-devel httpd php php-gd php-mysql php-pear

4.2.3 Instalación de librerías para los protocolos RDSI y los drivers para las tarjetas digitales y la instalación del central IP ASTERISK propiamente

Los paquetes a instalarse serán códigos fuentes, por lo que se necesitará compilar toda la paquetería para generar el código binario, el cual podrá integrarse con el *kernel* del Linux. Para lograr esto se usan sentencias como:

. /Configure: Mediante la herramienta Autoconf se generan scripts que configuraran los ficheros para la compilación automática.

- *make* : Usado para generar el código binario

- *make install*: para instalar el código binario al sistema operativo.

Además de estos comandos 2 ese usa dos comandos complementarios:

- *make config* : para generar archivos *scripts* de arranque de los programas

- *make samples*: instala files de ejemplos de configuración de Asterisk.

La instalación completa consta de 3 paquetes de software necesarios, todos en código

fuente, para su posterior compilación:

Nota: Para la instalación se usa la carpeta */usr/src,* que es donde regularmente van almacenados los códigos fuente de los programas.

- Libpri: Son las librerías que implementan los protocolos de comunicación RDSI de capa 2 y 3(Q.921 y Q.931), necesarias para poder operar con las tarjetas de comunicaciones que utilizan la tecnología Dahdi con líneas RDSI.

Se descarga el código fuente del Libpri del FTP oficlal de Digium :

 # wget http://downloads.asterisk.org/pub/telephony/libpri/libpri-1.4-current.tar.gz

Para descomprimir:

tar -zxvf libpri-1.4-current.tar.gz

- Dahdi linux: Son los paquetes que se usan como drivers para la integración de tarjetas digitales al servidor donde corre ASTERISK.

- Dahdi tools: Son los paquetes que tienen la función de realizar varios tipos de test a varias tarjetas de telefonía digital.

- Dahdi linux complete: Es la unión de dadhi linux y dadhi tools , por lo general se suele instalar los paquetes juntos.

Igualmente, que Libpri, se descarga el paquete del FTP oficial de DIGIUM:

wget http://downloads.asterisk.org/pub/telephony/dahdi-linux-complete/dahdi-linux-complete-current.tar.gz

Para descomprimir:

tar -zxvf dahdi-linux-complete-current.tar.gz

Por último, el código fuente del ASTERISK, que contiene todos los módulos necesarios para el comportamiento de una central telefónica.

wget http://downloads.asterisk.org/pub/telephony/certified-asterisk/certified-asterisk1.8.15-current.tar.gz

Para descomprimir:

tar -zxvf certified-asterisk-1.8.15-current.tar.gz

Una vez instalado y descomprimido los 3 códigos fuentes necesarios, se procede a la compilación e instalación de estos:

- Para Libpri :

cd libpri-1.4.14

make

make install

Para Dadhi :

cd dahdi-linux-complete-2.6.2+2.6.2

make

make install

make config

Para Asterisk:

cd certified-asterisk-1.8.15-cert2

./configure

make

make install

make config

make samples

Por último, se realiza las pruebas de que los módulos DADHI y ASTERISK están instalados de manera correcta. Para el caso del módulo DAHDI:

- Primero: se inicia el servicio

/etc/init.d/dadhi start

- Segundo: se revisa el módulo con el comando

lsmod | grep dahdi

A lo cual mostrará el siguiente mensaje:

[root@server asterisk]# lsmod | grep dahdi

dahdi_transcode 7928 1 wctc4xxp

dahdi_voicebus 40464 2 wctdm24xxp,wcte12xp

dahdi 196544 12 wctdm24xxp,wcte11xp,wct1xxp,wcte12xp,wct4xxp

crc_ccitt 2096 1 dahdi

Lo cual significara que los drivers y los tools DAHDI han sido instalados de manera correcta En el caso del módulo ASTERISK se usa el *script init* para *Asterisk.*

- Primero: Se inicia el servicio

[root@server asterisk]# /etc/init.d/asterisk start

- Segundo: Se revisa el módulo con el comando: *etc/init.d/asterisk status*, como vemos existen un proceso llamado *Asterisk* corriendo, lo cual significa que el módulo *Asterisk* ha sido instalado correctamente.

[root@server asterisk]# /etc/init.d/asterisk status

asterisk (pid 23712) is running...

4.3 Tercera Etapa

La tercera etapa consiste en la interconexión de ambas centrales telefónicas IP Asterisk, implementación de la línea primaria E1 como *backup*.

La última parte consta sobre la configuración de todos los anexos SIP necesarios para los teléfonos IP, estos en su mayoría son la marca Yeaklink SIP-T20, el cual maneja el protocolo SIP para la señalización y tiene CODEC G.711 y G.729 además de la transferencia de llamadas, y las funcionalidades de cada anexo. Para la interconexión entre centrales Asterisk de ambos locales (Camaná y Miraflores) se emplea el protocolo de señalización IAX2, el cual es muy eficiente ya que solo por un canal pasa tanto la voz como la señalización, además de multiplexar los paquetes de voz además de solo abrir un solo puerto, el puerto TCP 4569.

En el local de Camaná para la salida de llamadas se usan 2 proveedores SIP, aplicando para el mismo una troncal con señalización SIP. Un proveedor será usado para llamadas locales y nacionales fijos y el otro para llamadas celulares, para el local de Miraflores se eligió a los mismos proveedores SIP haciendo los mismos roles mencionados. Se eligió

dos proveedores debido a las tarifas que ofrecía cada proveedor en cada tipo de llamadas. Además, en el local ubicado en Camaná: la implementación del backup de llamadas lo que se hará con un servicio independiente del Internet, un servicio de RDSI para tener a disposición una línea E1. Para esto como parte del hardware para el servidor Asterisk se usa una tarjeta SANGOMA A101, la cual tiene una interfaz E1 con capacidad para 30 llamadas simultáneas (2.048 Mbps) el cual ira conectada a un *modem* SHDL ADTRAN de modelo 4200708 L 1. Actualmente el local de Miraflores no posee una línea *backup* para las llamadas. Por último, se configura el *dial plan* de la central Asterisk, el cual indicará el enrutamiento de las llamadas. Algo para tener en cuenta es el consumo del ancho de banda usado hacia Internet debido a las llamadas realizadas por el proveedor SIP.

El local de Camaná ya tenía arrendado un servicio simétrico de 3 Mbps, esto es suficiente ya que de los 70 anexos que se tienen, en promedio se realizan aproximadamente como máximo 35 llamada en simultaneo aproximadamente y al usar un CODEC G.729, empleando muestras de 10 ms, usando supresión de silencios y usando como tecnología de acceso a Ethernet 802.1Q, se emplea en promedio 37 Kbps. Para el cálculo de este último valor se emplea la herramienta ubicada en http://www.bandcalc.com/es/(Ver la figura 4.9).

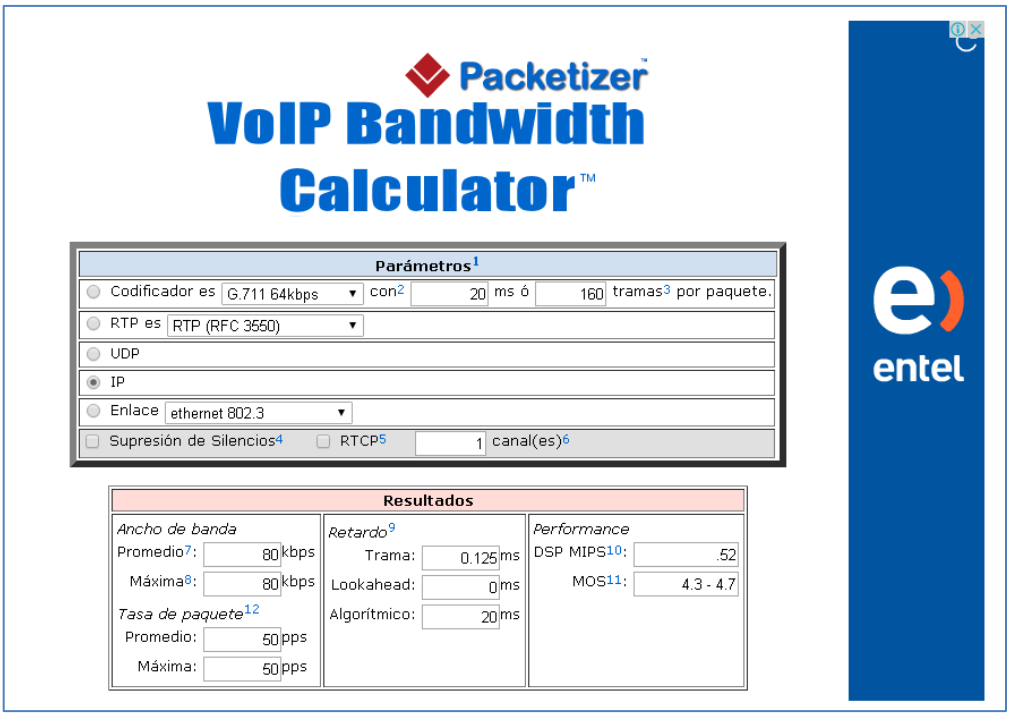

Figura 4.9 Calculador de ancho de banda para VoIP (Fuente: Packetizer)

En la figura 4.10 se muestra el cálculo realizado, empleando los parámetros descritos,

los que son usados usualmente por el CODEC G.729 y debido a la tecnología de acceso usada. Teniendo esto se puede calcular el consumo aproximadamente.

- Consumo total = (Ancho de banda usado por llamada) *(Por número de llamadas)

- Consumo = 37 * 35 = 1.295 Mbps

El cual es un valor menor a 3 Mbps.

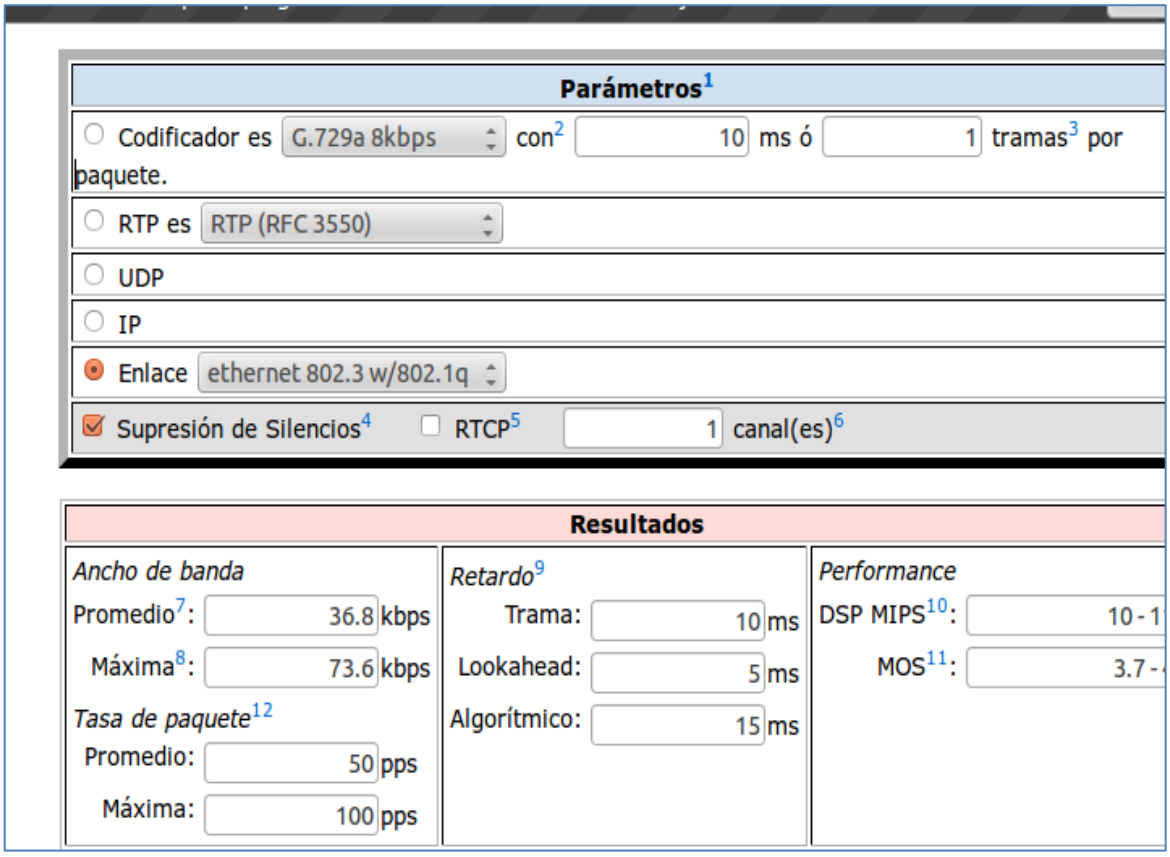

Figura 4.10 Cálculo realizado (Elab. propia)

Para una mejor descripción de la solución se ha dividido está en cinco partes la cuales son:

- Configuración de anexos SIP

- Configuración de la troncales SIP con los proveedores.

- Configuración de troncales IAX para interconexión entre las centrales ASTERISK

- Implementación de la línea primaria E1 como enlace backup para las llamadas

- Configuración del *dial plan* y funcionalidades de la central.

4.3.1 Configuración de anexos SIP

La configuración de cada anexo SIP se realiza en dos etapas:

- Configuración de una cuenta SIP en la central

- Configuración del teléfono SIP, acorde a los parámetros establecidos en la central.

La configuración de la cuenta SIP en el servidor tiene parámetros principales para su correcto funcionamiento los cuales son:

- *transport:* Indica el tipo de protocolo de transporte (UDP o TCP) que va a usar los canales SIP (definidos por el *chan_sip*).

- *qualify*: Si se activa es para enviar mensajes SIP OPTIONS cada 60 segundos para indicar al sistema que se mantiene una sesión activa al ser respondido por el dispositivo SIP en un acierto periodo de tiempo (si solo se escribe “yes” por defecto el tiempo será 2 segundos).

- *type*: Indica si la cuenta SIP puede:

- Solo recibir llamadas, entonces *type=user*
- Solo realiza llamadas, entonces *type=peer*
- Realiza y recibe llamadas, entonces *type=friend*

- host: Indica la dirección IP que tiene la cuenta SIP, en este caso para los teléfonos IP se usará *host = dynamic*, para indicar que el host posee una IP dinámica o estática.

- *context*: Es el nombre del *dial plan* que tiene la cuenta.

- *disallow/allow*: Indica que se desea deshabilitar todos los CODEC a usar, poniendo disallow=*yes* y permitiendo solo los CODEC mencionados en *allow* (G.711, G.729, GSM...)

- *secret*: Es la contraseña que compartirán la cuenta SIP configurada y el dispositivo SIP

- dtmf: Es el tipo de DTMF (*Dual Tone Multi-frecuency)* que se va a usar que pueden ser: rfc2833, *inband o info*, para este caso se usará el rfc2833 para llevar los DTMF, pero fuera de banda (fuera del flujo RTP de la voz).

La configuración de cada anexo seguirá la siguiente plantilla.

[general]

transport=udp

qualify=yes

[anexos_internos](!)

type=friend

host=dynamic

context=sip_user

dtmf=rfc2833

disallow=all

allow=g729

allow=ulaw

context=dialplan_user_sip

[anexo_sip](anexos_internos)

secret=verysecretpassword

Las cuentas en ambos locales se configuraron siguiendo la plantilla anterior describiendo esta:

- Se emplea el protocolo de transporte UDP para la señalización SIP (transport=UDP)

- En los contextos (dial *plan* para los anexos SIP) se va a emplear 3 tipos:

- Agentes del *call center*: *context* = agente
- Administradores de la red: *context* = *manager_network*
- Jefes de call center: *context* = *call_center*

- Para los tonos DTMF van a seguir la RFC2833, que describe como llevar los tonos mediante un *payload* del RTP, pero sin seguir el CODEC usado en el mismo, a esto se le llama *out-band* ya que es un flujo diferente del flujo de información RTP.

- Las contraseñas (*secret*) que se usan para la autenticación de anexos SIP se empleó la

dirección MAC de cada dispositivo.

- Se emplea como CODEC preferido el G.711 (allow=ulaw)

Como se ve en la plantilla de la cuenta SIP, se la ha dividido en 3 partes principales:

- General: Los parámetros configurados acá afectaran a cualquier cuenta SIP.

- Una plantilla especial para anexos internos, los cuales varían uno del otro principalmente por los distintos contextos que toman los cuales darán nombres a las plantillas:

- Agente
- Manager_Network
- Call_center

- Parámetros especiales de cada anexo, el cual varía según el *password* que tendrá cada cuenta, que como se comentó, será la dirección MAC del dispositivo.

Todas las configuraciones se realizan en el archivo de configuración.

/etc/asterisk/sip.conf

4.3.2 Configuración de la troncales SIP con los proveedores

La solución consta de implementar 2 troncales SIP una con cada proveedor distinto, por razones económicas, aunque en ambos casos se pueda usar para todo tipo de llamadas. Al igual que las cuentas SIP las troncales también se configuran en el archivo de configuración */etc/asterisk/sip.conf.* Las configuraciones en ambos locales son las mismas, en el cual se registra (el servidor Asterisk) como clientes SIP a cada uno de los proveedores, estos registros se hacen desde el encabezado general.

El proveedor SIP NetVoice que será usado para a teléfonos fijos locales y nacionales e internacionales, el segundo proveedor SIP es Net2Phone, el cual se usará para realizar llamadas a teléfonos celulares. Los pasos para la configuración son tres:

- Registro de cada proveedor SIP en el encabezado general.

```
[general]

#se registra cada proveedor con la sentencia

#register => username:password@hostname_proveedor
```

register => netvoice_call1007:T0pS3cr3t@sip.netvoice.com

register => net2phone_peru453:xxxyyyy@net2voicesip@netvoice.com

- Registro de la cuenta del primer proveedor.

[netvoice]

type=peer

host=sip.netvoice.com

fromdomain=sip.netvoice.com

username=netvoice_call1007

secret=T0pS3cr3t

insecure=port,invite

context=incoming

canreinvite=no

disallow=all

allow=g729

allow=ulaw

- Registro de la cuenta del segundo proveedor.

[net2phone]

type=peer

host=net2voicesip@netvoice.com

fromdomain=net2voicesip@netvoice.com

username= net2phone_peru453

secret=xxxyyyy

insecure=port,invite

context=incoming

canreinvite=no

disallow=all

allow=g729

allow=ulaw

Con estas dos cuentas SIP están establecidas las dos troncales SIP hacia ambos proveedores.

4.3.3 Configuración de troncales IAX2 para interconexión entre las centrales Asterisk

La solución para la interconexión de ambos locales se usa el protocolo de señalización IAX2, el cual fue desarrollado por la comunidad *Digium* para emplearlo entre centrales Asterisk. Ahora, como ya se implementó una VPN entre ambos locales, solo se va a usar direcciones IP privadas para apuntar desde uno a otro servidor y viceversa.

En la tabla 4.8 se puede observar que las direcciones IP que le corresponde a cada PBX Asterisk además de usar el puerto 465 UDP en ambos locales, para la señalización de IAX y el flujo de paquetes de voz.

Tabla 4.8 Direcciones IP que le corresponde a cada PBX (Elab. propia)

LOCAL	IP	PUERTO	PROTOCOLO
Camaná	172.16.0.33	465	UDP
Miraflores	192.168.1.240	465	UDP

La configuración de la troncal IAX, se va a seguir el concepto de *master-slave,* o sea que primero se configura un PBX como master y el otro como *slave,* en ese sentido la PBX *slave* solo puede realizar llamadas hacia el maestro en sentido inverso no se puede. Siguiendo el modelo *master-slave* se va a dividir la configuración en 2 pasos:

Nota 1: Todas las configuraciones para el canal IAX se realiza en el archivo de configuración */etc/asterisk/iax*

Nota 2: En ambas PBX el encabezado 'general' tiene la misma configuración:

[general]

bindport = 4569

bindaddr = 0.0.0.0

disallow=all

allow=ulaw

allow=alaw

Paso1: PBX master=>PBX Miraflores, PBX slave=>PBX Camaná

Configuración PBX en Miraflores

-Se crea una cuenta para la PBX Camaná

[Account_for_Camaná]

type=friend

host=dynamic

qualify=yes

trunk=yes

secret=iax_miraflores_Camaná

-Configuracion del *dial plan* a seguir, este será de nombre *internal.*

context=internal

;Se deniega cualquier IP y luego se permite solo la IP de la PBX de ; ;Camaná

deny=0.0.0.0/0.0.0.0

permit=172.16.0.33/255.255.255.255

Configuración de la PBX en Camaná

-Se configura la cuenta, la cual se autenticará contra la PBX en Miraflores

[Trunk_to_Miraflores]

type=peer

host=192.168.1.240

qualify=yes

trunk=yes

username=Account_for_Camaná

secret=iax_miraflores_Camaná

context=interna

Paso2: PBX master=>PBX Camaná, PBX slave=>PBX Miraflores

Configuración de la PBX en Camaná

Los pasos para la configuración de la central telefónica IP son tres:

-Se crea una cuenta para la PBX Miraflores

[Account_for_MIraflores]

type=friend

host=dynamic

qualify=yes

trunk=yes

secret=iax_miraflores_Camaná

-El contexto en el *dial plan* a seguir será de nombre *internal*

context=internal

-Se deniega cualquier IP y luego se permite solo la IP de la PBX de Miraflores

deny=0.0.0.0/0.0.0.0

permit=192.168.1.240/255.255.255.255

Configuración de la PBX en Miraflores

-Se configura la cuenta la cual se autenticará contra la PBX en Camaná.

[Trunk_to_Camaná]

type=peer

host=172.16.0.33

qualify=yes

trunk=yes

username=Account_for_Miraflores

secret=iax_miraflores_Camaná

context=internal

Las configuraciones realizadas permiten que la comunicación será bidireccional, y que ambas centrales IP *Asterisk* están registradas una a la otra, con lo cual ya se puede: señalizar las llamadas y realizar el transporte de los paquetes de voz.

4.3.4 Implementación de la línea primaria E1 como enlace *backup* para las llamadas

En la sede de Camaná se va a implementar una línea *backup* de las llamadas, así que cuando haya problemas con el servicio de internet o con el proveedor, las llamadas podrán ser ruteadas por la línea *backup*, la cual será una línea primaria E1. Para lo cual se tienen dos elementos:

- Una interfaz física en el servidor Asterisk: El cual es una tarjeta de marca SANGOMA y modelo A101, la cual posee una interfaz E1, para la conexión con la red RDSI, la cual provee de 30 canales, cada uno con una capacidad de 30 canales.

- Un proveedor de una línea primaria E1, al cual se conectará mediante un modem de tecnología SHDL.

La configuración de la tarjeta SANGOMA necesita primero los drivers para esta. El driver lleva el nombre de *wanpipe.* Luego se procede a la instalación del *driver*:

Paso 1: Se descarga el driver del FTP oficial del fabricante.

wget ftp://ftp.sangoma.com/linux/current_wanpipe/wanpipe-current.tgz

Paso 2: Se descomprime el driver descargado y se abre la carpeta.

tar xvfz wanpipe-current.tgz

cd wanpipe-<version>/

Paso 3: Para instalar el *wanpipe,* se aplica el comando*:/Setup install* desde dentro del directorio abierto en el paso anterior.

./Setup install

Paso 4: Se sigue el asistente del driver para la instalación, siguiendo los pasos descritos a continuación:

- Se acepta instalar el *wanpipe* tecleando Y. Ver la Figura 4.11.

- Se escoge el modo de compilación para el driver. En este caso se escoge la opción 2, el

cual es el adecuado para la integración del *wanpipe* y la central IP *Asterisk,* como se observa en la Figura 4.12.

- Se preguntará por la ubicación del *driver* para las tarjetas DAHDI, por defecto mostrará la ruta /usr/src que es donde se instala los paquetes. Así que solo se le da aceptar, como se observa en la Figura 4.13.

- Se da aceptar a las siguientes ventanas del asistente para la instalación del *wanpipe*, a lo cual por último se presenta una ventana indicando que se ha completado satisfactoriamente la instalación. Figura 4.14.

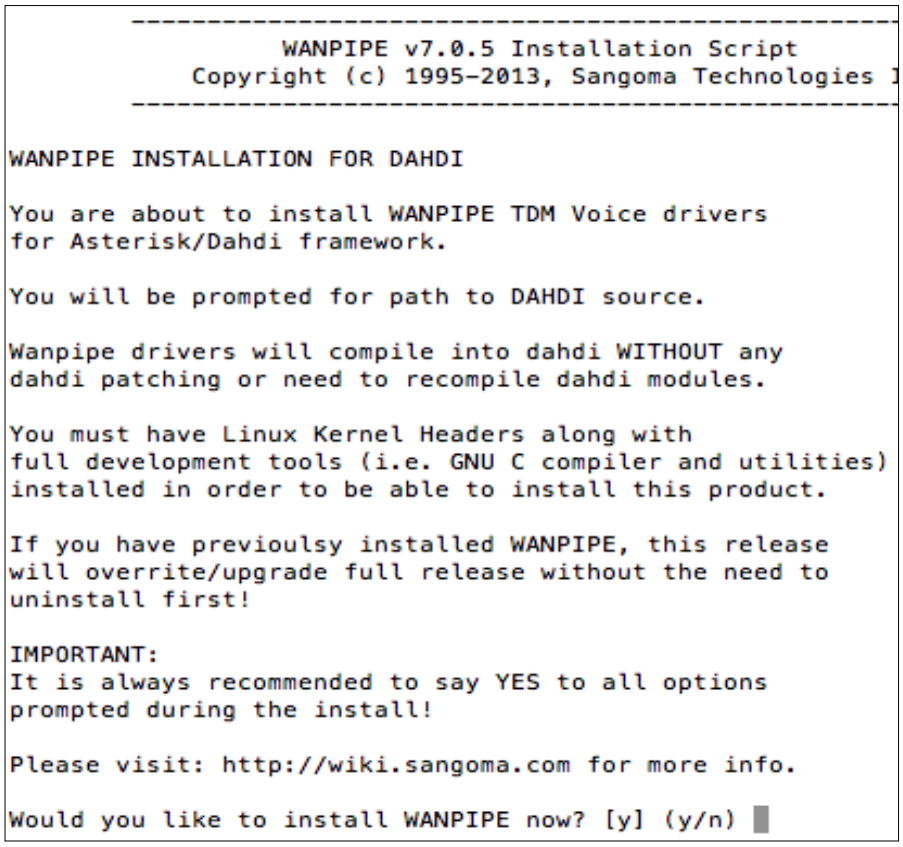

Figura 4.11 Asistente del driver para la instalación (Elab. propia)

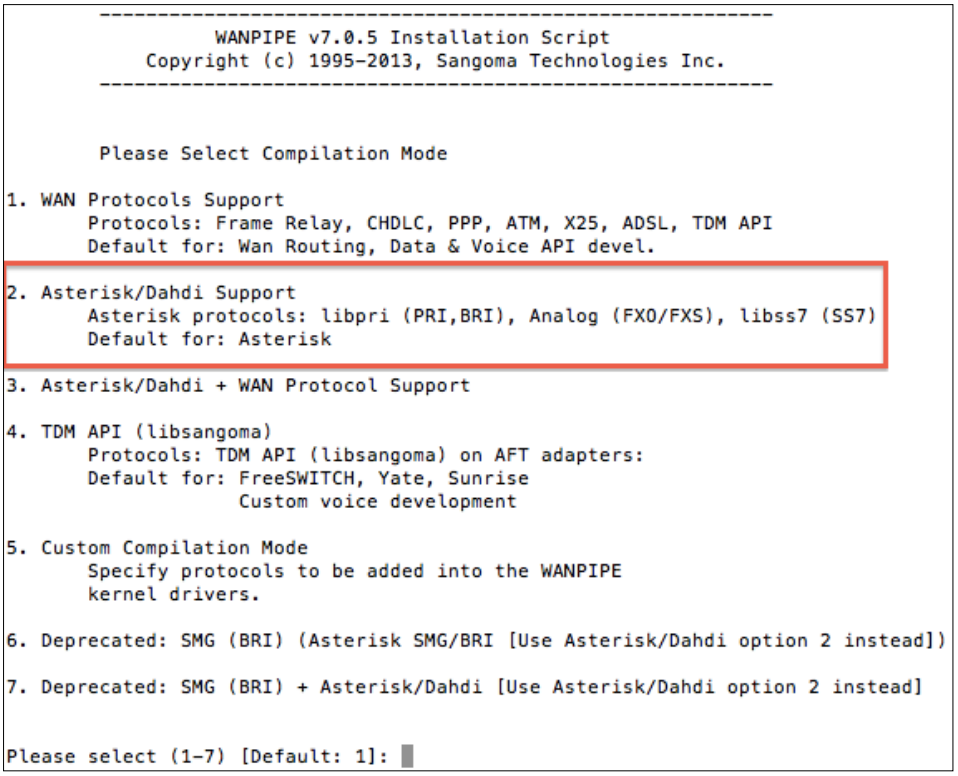

Figura 4.12 Modo de compilación para el driver (Elab. propia)

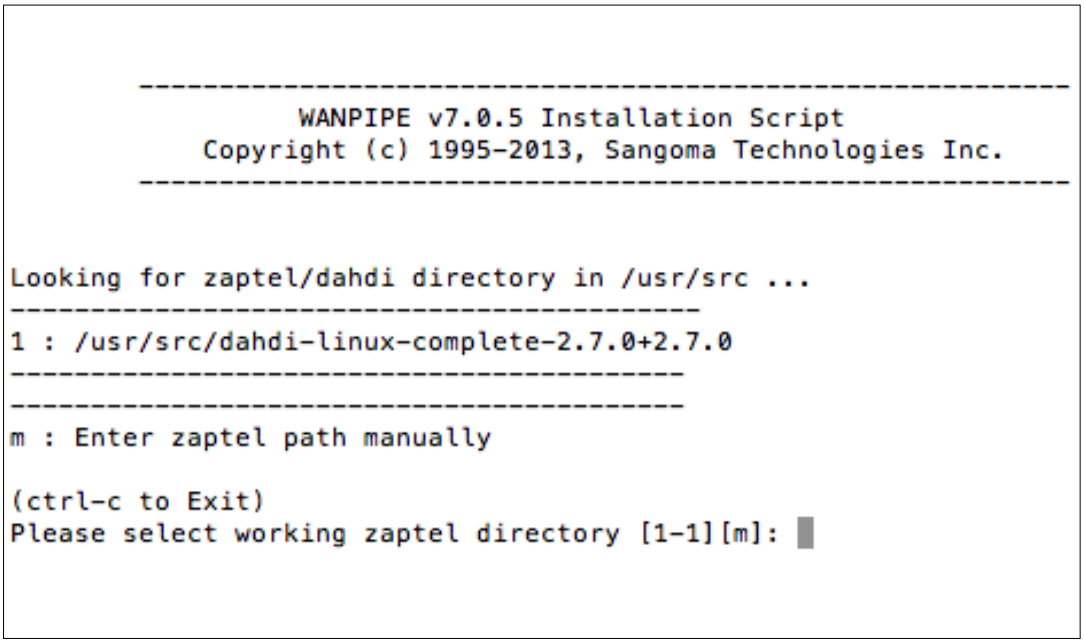

Figura 4.13 Ubicación del driver (Elab. propia)

```
WANPIPE INSTALLATON: COMPLETE

WANPIPE installation is now complete. WANPIPE kernel drivers
and configuration/debug utilities have been compiled and installed.

        1) Proceed to configure the WANPIPE drivers:
                Asterisk/Dahdi   : /usr/sbin/wancfg_dahdi
                Asterisk/Zaptel  : /usr/sbin/wancfg_zaptel
                TDM API          : /usr/sbin/wancfg_tdmapi
                SMG SS7/BRI/PRI  : /usr/sbin/wancfg_smg
                WAN Routing/API  : /usr/sbin/wancfg
        2) Use the /usr/sbin/wanrouter startup script to start and stop
           the router. (eg: wanrouter start)
        3) To uninstall WANPIPE package run ./Setup remove

Please read http://wiki.sangoma.com for further instructions.


Wanpipe / Zaptel Configuration
==============================

wancfg_zaptel configurator can create all wanpipe config files
for ZAPTEL including /etc/zaptel.conf file.
Optionally: the configurator can also create Asterisk zapata.conf
------------------------------------------------------------
Would you like to configure wanpipe devices for DAHDI? (y/n)
```

Figura 4.14 Instalación completa (Elab. propia)

- Para verificar la instalación de la tarjeta E1 SANGOMA se teclea el siguiente comando

wanrouter hwprobe

El cual mostrará detalles de la tarjeta SANGOMA como se ve en la figura 4.15.

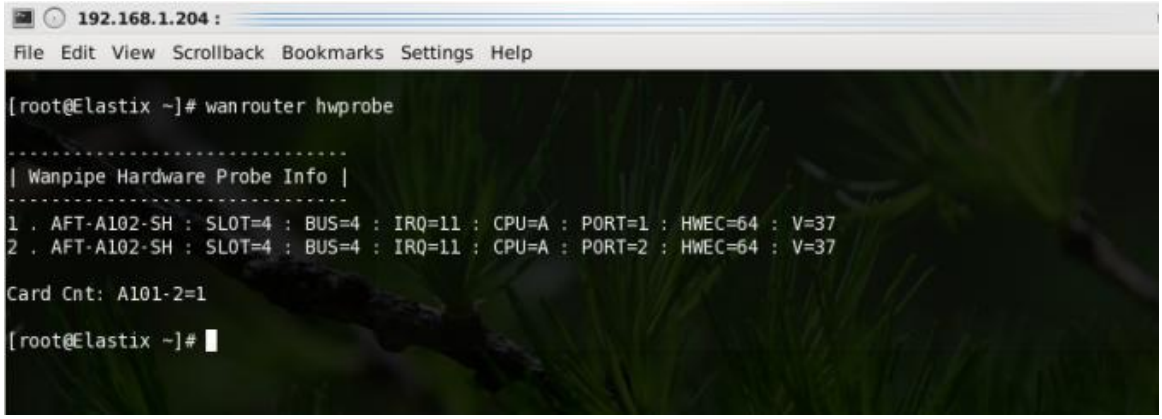

Figura 4.15 Detalles de la tarjeta SANGOMA (Elab. propia)

Al instalarse el driver se puede conectar el equipo SHDL del proveedor a la tarjeta E1 de la PBX.

4.3.5 Configuración del *Dial plan* y funcionalidades de las centrales

Llegados a este punto, ya se puede implementar la parte del enrutamiento de todas las llamadas, mediante el *dial plan.* Para el cual se crean varios contextos:

Para el caso de la oficina de Camaná se tienen estos contextos:

- Contexto para llamadas entre anexos. Anexo
- Contexto para llamadas a teléfonos fijos. locales Locales Fijos
- Contexto para llamadas a teléfonos fijos nacionales Nacional Fijo
- Contexto para llamadas a teléfonos celulares Celular
- Contexto para llamadas internacionales Internacional

Desde los cuales se va a crear 3 contextos que los agrupan teniendo en cuenta los privilegios siguiendo la tabla 4.9:

Tabla 4.9 Privilegios (Elab. propia)

Contextos	Local	Nacional	Celular	Internacional
Agente	x	-	x	-
Manager_Network	x	x	x	-
Manager Call	x	x	x	x

Nosotros debemos tener en cuenta que los anexos de Camaná tienen la numeración: 1XX y los anexos de Miraflores tienen la numeración: 2XX

Primero se crean los contextos para cada tipo de llamada:

- **Contexto para llamadas entre anexos**: Como se manejan dos locales cada uno tiene un *dial plan* distintos para realizar las llamadas entre anexos:

Para Camaná:

```
[anexos]

exten => 1XX, n,Answer ()
```

same => n,Dial(SIP/${EXTEN},30,Ttr)

same => n,Hangup()

Para Miraflores:

[anexos]

exten => 2XX,n,Answer()

same => n,Dial(SIP/${EXTEN},30,Ttr)

same => n,Hangup()

Como se posee una troncal IAX2 establecida entre ambas sedes se puede realizar llamadas entre anexos de distintas localidades, teniendo en cuenta las troncales IAX2 previamente establecidas, el *dial plan* seria configurado de la siguiente manera:

-Para Camaná:

[anexos_iax]

exten => 2XX,n,Answer

same => n,Dial(SIP/Trunk_to_Miraflores/${EXTEN},60,tTr)

same => n ,Hangup()

-Para Miraflores:

[anexos_iax]

exten => 1XX,n,Answer

same => n,Dial(SIP/Trunk_to_Camaná/${EXTEN},60,tTr)

same => n ,Hangup()

- **Contexto para llamadas locales y nacionales:** Los dígitos para números locales son de 7 dígitos empezando todos desde el número 2 hasta el 8, en el caso de números nacionales la cantidad de dígitos es 6 también se debe tener en cuenta el escalamiento para sacar la llamada; primero *NetVoice,* segundo *Net2SipPhone* y por último sacar la llamada por el primario. A continuación, describimos las configuraciones realizadas en cada local.

-Para el local de Camaná:

[local]

exten => [2-8]XXXXXX,Answer()

same => n,Dial(SIP/netvoice/511${EXTEN},30,tTr)

same => n,Dial(SIP/net2phone/511${EXTEN},30,tTr)

same => n,Dial(DAHDI/g0/${EXTEN},60,tTr)

same => n,Busy(2)

same => n,Hangup()

[nacional]

exten => XX[2-8]XXXXX,Answer()

same => n,Dial(SIP/netvoice/51${EXTEN},30,tTr)

same => n,Dial(SIP/net2phone/51${EXTEN},30,tTr)

same => n,Dial(DAHDI/g0/${EXTEN},30,tTr)

same => n,Busy(2)

same => n,Hangup()

- Para el local de Miraflores: Como se comentó anteriormente el local de Miraflores no posee una línea primaria como backup de llamadas, por lo que se omite sacar la llamada por una línea digital `DADHI/g0`.

[local]

exten => [2-8]XXXXXX,Answer()

same => n,Dial(SIP/netvoice/51${EXTEN},30,tTr)

same => n,Dial(SIP/net2phone/51${EXTEN},30,tTr)

same => n,Busy(2)

same => n,Hangup()

[nacional]

exten => _0NX[2-8]XXXXX,Answer()

same => n,Dial(SIP/netvoice/51${EXTEN},30,tTr)

same => n,Dial(SIP/net2phone/51${EXTEN},30,tTr)

same => n,Busy(2)

same => n,Hangup()

- **Contexto de llamadas para líneas celulares:** Como se sabe la cantidad de dígitos es 9, además que todo número celular empieza con 9.

Para el local de Camaná:

[celular]

exten => _9XXXXXXXX,Answer()

same => n,Dial(SIP/netvoice/51${EXTEN},30,tTr)

same => n,Dial(SIP/net2phone/51${EXTEN},30,tTr)

same => n,Dial(DAHDI/g0/${EXTEN},60,tTr)

same => n,Busy(2)

same => n,Hangup()

Para Miraflores:

exten => _9XXXXXXXX,Answer()

same => n,Dial(SIP/netvoice/51${EXTEN},30,tTr)

same => n,Dial(SIP/net2phone/51${EXTEN},30,tTr)

same => n,Busy(2)

same => n,Hangup()

- **Contexto para llamadas internacionales:** Para las llamadas internacionales la cantidad de dígitos varia de país en país por los que se usará en el contexto el punto (.) para indicarle al *dial plan* que después de los 2 ceros marcados (toda llamada hacia el extranjero se realiza primero marcando 2 ceros) puede ir cualquier cantidad de dígitos sin restricción de valores. El contexto para cada local será como sigue:

-Para Camaná:

exten => _00.,1,Answer()

same => n,Dial(SIP/netvoice/51${EXTEN},30,tTr)

same => n,Dial(SIP/net2phone/51${EXTEN},30,tTr)

same => n,Dial(DAHDI/g0/${EXTEN},60,tTr)

same => same => n,Hangup()

-Para Miraflores:

exten => _00.,1,Answer()

same => n,Dial(SIP/netvoice/51${EXTEN},30,tTr)

same => n,Dial(SIP/net2phone/51${EXTEN},30,tTr)

same => same => n,Hangup()

Como ya se tiene los contextos creados, solo se los asigna según la tabla descrita con los permisos para cada tipo de usuario, para cada local tendrá el mismo formato:

[Agente]

include => anexos

include => anexos_iax

include => local

include => celular

[Manager_Agente]

include => anexos

include => anexos_iax

include => local

include => nacional

include => celular

[Manager_Call]

include => anexos

include => anexos_iax

include => local

include => nacional

include => celular

include => internacional

Con estos últimos contextos creados, se asigna a cada anexo SIP el parámetro '*context'* poniendo el contexto que le corresponde: Agente, *Manager_Agente* o *Manager_Call.*

4.4 Análisis de costos

El *call center* en ambos locales ya contaban con *switch CISCO SG300* y SG500 para conectar todos los dispositivos de red, también el cableado estructurado necesario en ambos locales, UPS necesarios para los servidores, proveedor de servicios de internet y proveedor de servicios de telefonía IP, por lo cual solo se adquirieron el equipamiento físico necesario como listamos a continuación.

Tabla 4.10 Hardware (Elab. propia)

Equipos	Unidades	Descripción	Costo
Servidor HP ProLiant DL320E Gen8 Intel Xeon E3-1220v2 1P	1	Para el servidor de telefonía IP en Camaná	$1750
SERVIDOR IBM X3200 XEON System x3200 M3 (7328-C2U), procesador Intel XeonX3430 (2.4GHz/L3 8MB/1333MHz), memoria 2GB DDR3 ECC 1333MHz,	4	Para los 2 firewalls implementados en ambos locales, el proxy implementado en el local de Camaná y servidor de telefonía en Miraflores	$1300
Tarjeta Sangoma A-101	1	Tarjeta para la línea RDSI en Camaná	$ 520.00

Garantía: Todos los servidores como la tarjeta tienen garantía de 1 año, por causas de desperfectos en los equipos o dispositivos.

4.5 Estado final de la implementación

Al finalizar la implementación de la solución, se superaron los inconvenientes principales que tenía el *call center*, con lo cual el cliente: *call center* Estudio Jurídico y el ingeniero de sistemas Ruben Chuspe, dieron la conformidad de los impactos positivos en la red del lugar.

Figura 4.16 *Call center* en producción.

Figura 4.17 ***Data*** *center* en producción

CONCLUSIONES

Al finalizar el proyecto se concluye lo siguiente:

1. Se logró superar los problemas de disponibilidad, problemas con pagos de licencias y compatibilidad mediante el uso de una central telefónica IP basada en ASTERISK.
2. Se brindó la solución para la línea backup en casos de fallos, mediante la instalación de una línea digital RDSI, con un E1.

RECOMENDACIONES

Al finalizar el proyecto se recomienda lo siguiente:

1. La implementación de un proxy en el local de Miraflores, para evitar un mal uso de los servicios del Internet y de los servidores locales.
2. La suscripción a Vyatta Subscription Edition (VSE), ya que la solución *open source* para *networking* viene con ciertas limitaciones en su versión gratuita
3. Dar mantenimiento a la central de telefonía, así como los *upgrades* de la misma. Este último punto mencionado esta soportado por la comunidad Asterisk, resolviendo los bugs en el sistema.
4. La implementación de constantes actualizaciones y mejoras en las políticas de seguridad en los firewalls / proxy, ya que todo sistema no es 100 % seguro .
5. La implementación de una línea *backup* de telefonía mediante una línea primaria en el local de Miraflores, ya que como se vio actualmente solo posee dos proveedores SIP los cuales están sujetos a la línea de internet arrendada, al caer esta no se podrán realizar llamadas, lo cual es un punto muy crítico en un *call center*.

ANEXOS A CONFIGURACIONES

Se desarrolla la configuración de teléfono IP Yealink SIP T-20, del *softphone* 3CXPhone, y del *switch* Cisco mediante puerto consola.

A.1 Configuración de teléfono IP Yealink SIP T-20

- Paso 1**:** Se configura una IP estática a teléfono IP Yealink SIP T-20.

- Paso 2: Una vez configurada la IP en el teléfono, se escribe la dirección IP en un navegador web. Las credenciales son:

  - *user: admin*
  - *pass: admin*

- Paso 3: Se va a la pestaña “cuenta” y se llenan los campos necesarios para registrarnos en la central. Tanto en “nombre de registro” como en “nombre de usuario”, se escribe el número o nombre de la cuenta registrada en el servidor de telefonía. En “contraseña”, se introduce la contraseña configurada previamente en la central para esta cuenta. Como la central de telefonía va a realizar ambas funciones de proxy y servidor SIP, en ambos casos se va a escribir la dirección IP de la central. Solo para resaltar, el protocolo de transporte para la señalización SIP va a ser UDP.

- Paso 4: Configurar los *codec* para el teléfono IP: En la misma pestaña de “cuenta” su hace *click* en “*codecs*” para elegir los CODEC a usar por el teléfono IP, se debe tener en cuenta que los CODEC elegidos también deben estar configurados en la cuenta creada en la central telefónica IP.

A.2 Configuración de *softphone* 3CXPhone.

Antes de la configuración del *softphone* se debe descargar el aplicativo desde la página oficial (http://www.3cx.com/voip-ip-pbx/), que es la versión con funciones básicas gratuitas, previo registro de usuario. Esta versión es suficiente para el objetivo. Una vez descargado el aplicativo se procede a configurar los parámetros básicos de configuración para empezar una llamada VoIP.

- Paso 1: Hacer doble-click sobre el icono del software 3CXPhone.

- Paso 2: La primera vez que se abre el programa, automáticamente aparece una ventana para crear una cuenta SIP. En el cual se completan los parámetros:

- -Account name: Es el nombre local que se coloca a la cuenta para distinguir unas de otras creadas en el softphone

- -Caller ID: Es el nombre con que se mostrará ante otro teléfono IP.

- -Extensión/ID: Es el número o nombre con el que fue creado la cuenta en la central IP.

- Password: La contraseña para la autenticación de la cuenta SIP.

- -My location: Es este caso se tienen 2 opciones; "*I am the office* "el cual indica que se está en la LAN de la PBX , y "*I am out the office*" , el cual indica que se está fuera de la LAN de la PBX , para este caso se usará la primera opción , ya que no se crearán anexos remotos , y se escribirá la dirección IP del servidor de telefonía.

- Paso 3: Por último, solo le da "OK", con lo cual ya se tiene la cuenta SIP creada.

A.3 Configuración de switch Cisco mediante puerto consola.

Muchas veces al empezar una configuración de un equipo nuevo de *networking* o tener problemas para el acceso al mismo (olvidarse la dirección de administración del equipo o no tener la contraseña), se debe acceder a él mediante la comunicación tipo serial RS-232, para lo cual se detalla a continuación el procedimiento:

- Paso 1: Descargar el software de acceso PuTTY (u otros) para el acceso al equipo mediante comunicación serial. Esto se puede hacer ingresando a la página web: http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html; en el cual se descarga según el sistema operativo de la PC.

- Paso 2: Hacer doble-click sobre el icono del software PuTTY

- Paso3: Hacer click en "Connection" y luego en "Serial" con lo cual se accede a la siguiente ventana en la cual se llenan los parámetros:

- Serial line: El cual es el nombre de la interfaz serial por donde se conecta al *switch*.

- Speed: La velocidad para la comunicación serial por defecto es 115200 *baud* (en este caso tasa de baudios = tasa de datos, ya que cada símbolo representa solo 1 bit).

- Data bits: El número de bits para representar cada carácter, por defecto es 8 bits.

- Stop bits: Bits enviados al final de cada carácter para diferenciar uno de otro, por defecto se pone 1.

- Parity: Es el método para usar para detectar errores en la comunicación, por defecto no

se elige alguno.

- Flow control: El método de control para la comunicación serial, es este caso no se usa alguno.

- Paso 4: Por último, se da open con lo cual se ingresa a CLI del *switch*. Las credenciales por defecto son:

- admin: cisco
- password: cisco

BIBLIOGRAFÍA

[1] Asterisk: The Definitive Guide, 4th Edition - O'Reilly Media

[2] SIP: Understanding the Session Initiation Protocol Alan B. Jhonston.

[3] CCNP ROUTE 642-902 Official Certification Guide (Exam Certification Guide): Wendell Odom:

[4] Gómez López, Julio, “Voip y Asterisk: redescubriendo la telefonía”,

[5] Sybex's LPIC-1: Linux Professional Institute Certification Study Guid - O'Reilly's

ENLACES

[1] Fundamentals of Asterisk , https://wiki.asterisk.org/wiki/display/AST/Fundamentals

[2] RTP (Real-time Transport Protocol), http://www.voip- info.org/wiki/view/RTP

[3] Voip-info.org, “Asterik”, http://www.voip-info.org/wiki/view/Asterisk

[4] Vwiki, “Vyatta”, http://vwiki.co.uk/Vyatta

[5] Sangoma, “Manual Installation for TDM Cards for Asterisk” http://wiki.sangoma.com/wanpipe-linux-asterisk-dahdi

MATRIZ DE CONSISTENCIA

PROBLEMA	OBJETIVOS	HIPÓTESIS	VARIABLES	TIPO DE INVESTIGACIÓN
Problema General La inadecuada gestión y limitaciones del equipamiento acorde a los nuevos requerimientos de comunicaciones de la organización. **Problemas específicos** 1. El tiempo para establecer una sesión SIP (Session Initiation Protocol) con terminales remotos no era el adecuado. 2. El local principal no poseía una línea de backup ante fallos de los proveedores SIP, ya sea debido a la falla misma de los proveedores SIP, o a la caída de los servicios de internet.	**Objetivo General** Mejorar la capacidad de comunicaciones e interconexión de un sistema de telefonía mediante la implementación de centrales telefónica IP basadas en Asterisk y de una línea primaria E1. **Objetivos específicos** **1.** Obtener un call center capaz de cumplir con los estándares establecidos por la empresa, como la alta disponibilidad, seguridad e interconexión. **2.** Redundancia para Interconexión de ambas centrales telefónicas IP ASTERISK, para lo cual se requiere la implementación de la línea primaria E1 como backup.	**Hipótesis** Implementación de una central telefónica IP basada en Asterisk y una línea primaria E1, eliminando así una serie de deficiencias que experimentaba la empresa del caso de estudio.	**1.Variables Independientes** Diseñar una red escalable, que posea una buena integración y con calidad de servicio para la voz y una línea de respaldo. **2.Variables Dependientes** • alta disponibilidad. • redundancia o Backup. • seguridad e interconexión.	**1. Tipo de Investigación** ○ Documental ○ Experimental vía Explicativo. **2. Nivel de investigación** ○ Descriptivo ○ Explicativo **3. Diseño** Por objetivos. **4. Población** No aplica. **5. Muestra** No aplica. **6. Técnicas de recolección de datos.** ○ Análisis documental. ○ Análisis de datos obtenidos en Cisco, Linux Centos y Asterisk. **7. Plan de Análisis Estadístico de Datos.** ○ Agrupación de datos que permitan sintonizar las variables .