VNIVERSIDAD NACIONAL
AVTNOMA DE
MEXICO

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

"ESTUDIO DEL COMPORTAMIENTO DINÁMICO ENTRE PROTOCOLOS DE SEÑALIZACIÓN (SIP Y H323) PARA EL SOPORTE DE VOIP"

TESIS

QUE PARA OPTAR POR EL GRADO DE:

MAESTRA EN INGENIERIA (COMPUTACION)

PRESENTA:

SANDRA NAYELI DELGADO VIEYRA

TUTOR.

DR. VÍCTOR RENGEL LICEA

FACULTAD DE INGENIERÍA-UNAM

MÉXICO, D. F. JULIO 2015

Jurado asignado

Presidente: **Dr. Gerardo Vega Hernández**

Vocal: **Dr. Víctor Rangel Licea**

Secretario: **Dr. Javier Gómez Castellanos**

Suplente: **Dr. Jorge Luis Ortega Arjona**

Suplente: **Dra. María Elena Lágarra Ramírez**

Lugar donde se realizó la tesis: México D.F.

Tutor de tesis

Dr. Víctor Rangel Licea

Dedicatoria

A mis padres: Rita Vieyra Bolaños y Víctor Manuel Delgado López por su cariño, dedicación y apoyo incondicional durante todo este tiempo, por educarme y formar a la persona que soy ahora.

A mis hermanas: Nancy y Diana por el apoyo y el cariño que me han brindado siempre.

A Ixchel Melissa Rivera Durán por estar conmigo y apoyarme en mi superación.

A mis familiares y amigos que siempre han estado presentes en todos los momentos de mi vida.

Agradecimientos

A la Universidad Nacional Autónoma de México por brindarme la oportunidad de prepararme tanto de forma profesional como humana para realizar mis estudios de posgrado.

Al Dr. Víctor Rangel Licea por sus enseñanzas, dedicación, compresión y paciencia durante el desarrollo de este trabajo de tesis.

Gracias al Dr. Miroslav Voznak por brindarme herramientas y conocimientos para desarrollar pruebas.

A la Universidad Técnica de Ostrava en Republica Checa, por brindarme un espacio dentro de sus aulas.

Al CONACYT, por la beca otorgada durante mis estudios de maestría.

A la CEP por el apoyo económico brindado, para la realización de una estancia académica.

A la DGAPA por el apoyo recibido en el Proyecto PAPIIT:IN114713

Resumen

Los constantes cambios en las tecnologías han propiciado a mejorar, crear y diseñar nuevos protocolos o bien a dejar atrás algunos de ellos. Tal es el caso del protocolo H.323 que fue el primero en aparecer para dar soporte a la señalización de voz sobre IP. Con el paso del tiempo se ha mejorado y se encuentra en competencia con otros protocolos que hoy en día se encuentran disponibles tales como MEGACO, IAX y SIP. Éste último nació como alternativa de H.323. Actualmente ambos protocolos son los más utilizados para la implementación de voz sobre redes de datos.

Por lo que en este trabajo de tesis se realiza un estudio del comportamiento dinámico entre los protocolos H.323 y SIP saturando la red. Para ello se implementa una red de datos WAN en delta, dónde transitara solo tráfico de voz.

Se explica primeramente los protocolos de señalización a estudiar, en cuanto a características, arquitectura y protocolos que trabajan en conjunto para poder realizar llamadas.

La implementación de la red se basa en equipos de interconectividad como *routers* y *switches* para montar la red de datos, y para implementar lo referente a VoIP se hace en base a *software* abierto (*Open Source)*. En la red de datos se configuran dos usuarios y un servidor. El servidor es una máquina con Asterisk instalado y en los clientes se instalaron las aplicaciones SIPp y Callgen 323, que serán los encargados de generar tráfico SIP y H.323 respectivamente hasta saturar la red a un 90%, para evitar desbordamientos.

Los resultados obtenidos reflejan que la red soporta una mayor cantidad de llamadas con SIP que H.323, el tráfico de señalización generado por H.323 es hasta 3 veces mayor que el generado por SIP.

Ambos protocolos presentan ligeros retardos y *Jitter* considerables, los resultados teóricos obtenidos se apegan la literatura de VoIP.

Índice principal

Índice de figuras

Capítulo 2

Capítulo 3

Capítulo 4

Capítulo 5

Capítulo 6

Apéndice 1.

Índice de tablas

Capítulo 1. Introducción

1.1 Antecedentes

Los antecedentes primordiales de la VoIP[1] (Voice Over Internet Protocol) indican que desde la aparición de algunas tecnologías, la transmisión de voz no pudo haberse desarrollado, las tecnologías a las cuales nos referimos son el teléfono e Internet. Ocho años después de la aparición del teléfono en 1870 aparece la primera central telefónica establecida en New Haven y los teléfonos poco a poco pasaron de ser analógicos a digitales. Casi 100 años después, en 1968 el Internet es desarrollado por ARPANET (Advanced Research Projects Agency Network) y en 1957 fue fundado por el Departamento de Defensa de los Estados Unidos (DOD, United State Department of Defense).

Para ese entonces ya se tenía el equipo y el medio por el cual viajarían los datos pero a un falta el cómo y por dónde viajarían a su destino esos datos. Desde los inicios del Internet muchos investigadores, ingenieros y científicos se han dedicado a definir varios protocolos para las redes de datos. Muchos de éstos han tenido tanto éxito, que hoy en día siguen siendo el sustento de nuestras redes como es el caso del Protocolo de Control de Transmisión y el Protocolo de Internet (TCP/IP), inventado por Vinton Cerf en el DOD en 1972 [1]. Los cuales han sido los más importantes en los últimos 40 años, gracias a ellos se han encontrado nuevas formas de cómo transferir datos y cómo establecer comunicaciones. Y que propiciaron al surgimiento de las redes convergentes que hacen referencia a la integración de los servicios de voz, vídeo y datos sobre una sola red basada en el protocolo de Internet (IP) como protocolo de capa de red.

En la actualidad podemos transferir voz por medio de nuestra red IP sin necesidad de tener una red telefónica tradicional, o bien una combinación de la red telefónica tradicional y una red IP. Pero obviamente la voz sobre IP no ha existido desde que apareció el Internet, esta tecnología inicia un poco antes de 1989 con la invención de un dispositivo llamado Audio *Transceiver*, que permite digitalizar la voz y adaptarse a la red de Internet.

[1] La lista de términos se encuentra en el Glosario partir de la página 124

Entonces para 1989 se fundó la empresa VocalTec, meses después se lanza el primer teléfono de Internet que en realidad era un *softphone* llamado "InternetPhone *Software*". Fue diseñado para usarse en una computadora normal siempre y cuando tuviera tarjeta de sonido, micrófono, parlantes y modem, pero solo funcionaban si ambas PC tenían el mismo *software* y el mismo *hardware* lo que fue un fracaso total. Sin embargo, sirvió de impulso para continuar con investigaciones, porque se dieron cuenta que era posible enviar voz por Internet. Para 1996 VocalTEc entra a la bolsa con 33 millones de dólares y en ese mismo año aparece su primera competencia Net2Phone quien ofreció comunicaciones telefónicas desde PC's a teléfonos tradicionales.

En ese mismo año la empresa de nombre Telcom Finland, operadora de telefonía tradicional, establece comunicaciones entre PC's utilizando el *software* de VolcaTel. También Microsoft entra al mercado de la telefonía IP, al crear el *software* Netmeeting capaz de realizar llamadas entre PC's, y para el año 2001 este software fue llamado Messenger. En 1997 Deutsche Telekom lanza T-NetCall el cual permite hacer llamadas entre teléfonos por medio de Internet. En ese entonces ya muchas empresas dedicadas a las comunicaciones comenzaban a desarrollar equipos que pudiesen manejar voz, tal fue el caso de CISCO. Que en 1998 comenzó a desarrollar *routers* y los primeros *gateways*, que permitían hacer llamadas de una PC a un teléfono y viceversa [2]. Para el 2003 sale a la luz un *software* que revolucionó las comunicaciones de voz, con la llegada de Skype, se establecieron llamadas gratuitas entre ordenadores y con un coste local entre ordenador y teléfonos fijos tradicionales. Desde entonces Skype ha estado en contantes mejoras.

1.2. Definición del problema

La aparición de la telefonía IP ha creado grandes caminos alternos para las empresas, organizaciones y hogares, ofrece grandes ventajas y funciones que la telefonía convencional no es capaz de proporcionar por la infraestructura con la que cuenta. Pero a pesar de ello aún existen puntos que imposibilitan el uso e implementación de una red IP, tales como los altos costos de los equipos necesarios para una red de este tipo.

Entonces si se implementara una red WAN con tráfico de Voz transitando sobre ella, haciendo uso de *software* libre que sustituya una central telefónica de VoIP y generadores de tráfico de señalización para H.323 y SIP. Con el fin de medir el rendimiento que tiene la red con cada uno de estos protocolos, ofreciendo una alternativa de mejora a todos aquellos interesados en mudarse a redes telefónicas IP con una alta demanda de llamadas.

El principal problema son los costos que implica adquirir o rentar conmutadores telefónicos, PBX o centrales telefónicas que soporten la demanda de llamadas telefónicas de una empresa o negocio. Además de la renta de la línea o líneas telefónicas de algún proveedor de servicio, aumentando el costo. Si aunado a ello se tiene una renta de Internet el costo incrementa. Es por ello que se realiza este trabajo utilizando software libre para la implementación de una central telefónica y los generadores de tráfico. Además de elegir el protocolo con el cual trabajaran los quipos para brindar servicio de telefonía.

1.3. Objetivos

Lo que se propone en el presente proyecto es realizar un estudio del comportamiento dinámico entre protocolos de señalización, para el soporte de VoIP. Montando una red WAN y a bajo costo con la utilización de *software* libre como Asterisk que remplazará a una central telefónica física, generando tráfico de señalización para establecer llamadas. Los objetivos a cumplir de forma progresiva se describen a continuación.

Montar una red WAN por la cual transite Voz sobre IP.

Realizar las configuraciones y adaptaciones necesarias para poner en operación la red WAN con equipo de interconexión, donde se transitará voz.

Configurar una central telefónica con *software* libre como Asterisk que permita la señalización de tráfico de voz generado con *software* libre para voz sobre IP.

Instalar y configurar las herramientas necesarias generadoras de tráfico SIP y H.232, en los equipos con el rol de clientes.

Instalar y configurar las herramientas para realizar mediciones, cuando la red se ponga en marcha.

Analizar, comparar e identificar el tipo de tráfico de señalización que se genera con el *software* libre, simulando llamadas de voz.

Obtener conclusiones en cuanto al comportamiento dinámico que presentan los protocolos de señalización en la red WAN, utilizando una central telefónica como Asterisk y cuáles son los resultados obtenidos sometiendo la red bajo estrés.

1.4. Metodología

Se realizará un análisis de los estudios previos sobre los requerimientos y configuraciones que necesita una red WAN para poder soportar tráfico de voz. En base a los estudios se realizarán dichas configuraciones para poner en funcionamiento la red con los equipos CISCO con los cuales ya se cuenta en el laboratorio.

Posteriormente se estudiara el funcionamiento de algunos protocolos se señalización que trabajan con VoIP, así como su respectiva evolución y sus principales ventajas y desventajas, con el fin de determinar que *software* libre es el más adecuado para generar tráfico de señalización para voz. Una vez encontrado el *software* que cumpla con dichas características se pondrá en marcha sobre la red WAN.

Para la parte de la central telefónica se estudiara el funcionamiento y alcances del *software* libre Asterisk, para lograr su implementación junto con la red WAN y entonces tener lista la red para transmitir voz. Una vez que se tenga la red funcionando se analizará el comportamiento de cada protocolo de señalización, saturando la red y observando su comportamiento, así como de la central. Por último se obtendrán los datos que nos ayuden a concluir que protocolo es más eficiente.

1.5. Contribución y relevancia

Los resultados obtenidos y documentados en esta tesis, podrán ser utilizados como base o referencia para entender, mejorar e implementar el servicio de voz sobre IP en una red WAN utilizando *software* libre. Permitiendo elegir que protocolo de señalización es el más adecuado en base a las necesidades requeridas. Además de proveer las configuraciones y técnicas de configuración de *software* libre para el remplazo de centrales telefónicas y el comportamiento que éste puede tener con los protocolos de señalización.

También servirá como base para la realización de pruebas antes de poner en marcha una red bajo condiciones similares y tomar en cuenta parámetros importantes que afectan la red.

1.6. Estructura de la tesis

Este trabajo de investigación consta de 6 capítulos.

En el segundo capítulo se explica de forma general en qué consiste la telefonía convencional y cómo surge la telefonía IP.

En el tercer capítulo se explican los protocolos de señalización SIP y H.323, así como su arquitectura y sus principales componentes. También se describen otros protocolos que trabajan en conjunto para establecer las llamadas.

En el cuarto capítulo se describe el *software* y *hardware* a utilizar, la red de datos está basada en equipo de interconexión de redes y el *software* libre.

El quinto capítulo se explica la implementación y configuración de todas las herramientas como SIPP, Asterisk, Zoiper, Wireshark, H.323plus. etc.

Por último en el capítulo 6 se presentan los resultados y conclusiones.

Capítulo 2. Conceptos básicos de telefonía tradicional y VoIP

2.1 Introducción

En este capítulo se pretende dar a conocer los conceptos necesarios para entender el funcionamiento de VoIP, para ello se necesita comprender en primera instancia cómo es el proceso de establecer una llamada por medio de la telefonía convencional. También conocer cómo es el desempeño que tiene cada elemento perteneciente a la red de telefonía. Posteriormente será más sencillo entender cómo funciona VoIP y sus componentes principales.

2.2 Señal analógica

La voz humana es captada por un micrófono de un teléfono y se convierte en una señal eléctrica que siempre varía en forma continua, debido a los cambios de sonido, al tono de voz y la pronunciación de cada palabra, los cuales son factores que provocan variaciones en la onda de sonido. El micrófono y el circuito analógico entonces convierten la voz (onda de sonido analógica) en una onda eléctrica, que se transmite por medio de un cable de cobre atravesando la PSTN hasta el otro extremo de la red telefónica. Aquí la onda de voz es convertida de nuevo en onda analógica y se envían al receptor.

Una señal analógica es una presentación de funciones que pueden tomar un número infinito de valores en cualquier intervalo de tiempo.

Cualquier señal analógica está conformada por parámetros tales como periodo, amplitud, longitud de onda y frecuencia [3], como se muestra en la **Figura 2. 1**.

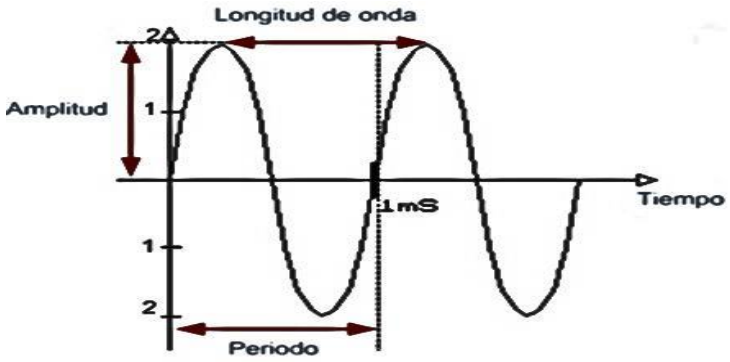


Figura 2. 1 Señal analógica y sus características.

2.2.1 Proceso de digitalización

Las grandes distancias que recorren las señales digitales producen que la señal se degrade y a pesar de utilizar amplificadores, estos distorsionan demasiado la señal es por ello que se hace uso de las señales digitales. Las cuales permiten transportarla de una manera eficiente por medio de un mismo par de cables utilizando la técnica de multiplicación.

Como bien ya se dijo la señal que viaja del bucle local a la central es analógica y al llegar a esta es convertida a digital, y éste es otro concepto importante a tocar ¿que implica el proceso de digitalización de una señal? El proceso de digitalización costa de una serie de pasos: Muestreo, Cuantificación, Codificación, Compresión [4].

Muestreo: Consiste en tomar muestras de una señal de voz analógica en un intervalo de tiempo de n veces por segundo, con el fin de convertir la señal analógica continua a una señal discreta en el tiempo los intervalos de tiempo deben cumplir con el Teorema de Nyquist que dice:

“La mínima frecuencia a la que puede ser muestreada una señal y luego reconstruida sin perder información, es el doble de la frecuencia máxima de dicha señal”.

En la **Figura 2. 2**, se observa del lado izquierdo una señal analógica que al ser muestreada genera una señal muestreada, como se ve del lado derecho de la figura.

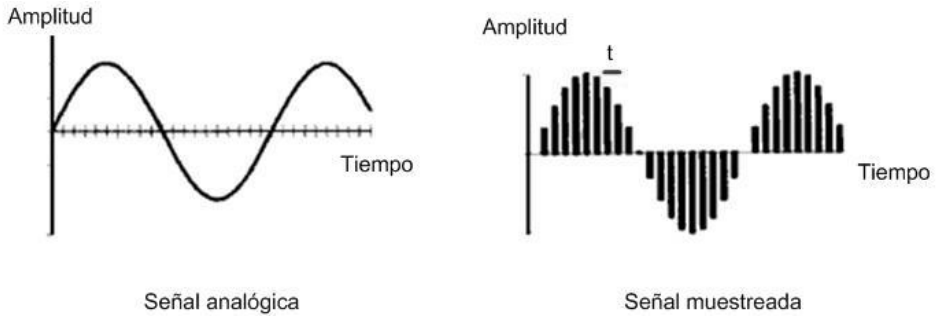


Figura 2. 2 Señal analógica muestreada.

El ser humano escucha sonidos de hasta 18 o 20kHz, pero la voz humana tiene una frecuencia a 3.4kHz que es perfectamente entendible, los sistemas de comunicación se han adaptado minimizando recursos necesarios. Entonces retomando el Teorema de Nyquist, si la voz tiene una frecuencia de 3.4kHz y para poder ser muestreada se necesita como mínimo 6.8kHz. Si bien se debe tomar en cuenta que en la vida real, no se puede muestrear a una frecuencia de 6.8KHz, ya que se harían cortes abruptos a la señal. Por lo que si utilizamos un CODEC G.711 que realiza un muestreo de 8kHz sigue respetando el teorema, entonces con esto se toma una muestra de voz cada 125 microsegundos, siendo legible para la reproducción [5].

Cuantificación: Es el segundo paso de la digitalización, consta de realizar la conversión de las muestras que se tomaron en el proceso de muestreo y asignarles un valor discreto a cada muestra, de acuerdo a la amplitud de la señal. Cuando se realiza la conversión de valores infinitos como es la señal analógica a valores discretos se crea una pequeña distorsión conocida como ruido de cuantificación que es un factor natural de este proceso. Mientras más valores discretos utilicemos disminuirá la distorsión en el proceso, como consecuencia se tendrá una mayor cantidad de información en bits por cada muestra. En la **Figura 2. 3** se muestra la señal analógica y la señal cuantificada.

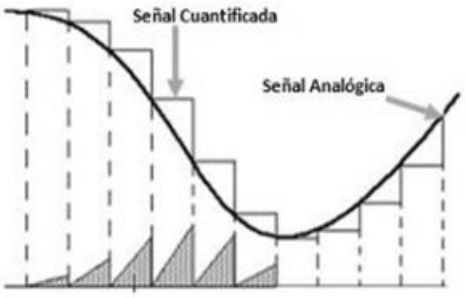


Figura 2. 3 Cuantificación de una señal analógica.

Codificación: Permite optimizar el canal de comunicación, enviando la mayor cantidad de información por un canal de voz, cuidando en no perder calidad. La codificación consta de: a cada

valor cuantificado que representa la señal analógica se le asigna un código binario como se ve en la **Figura 2. 4**.

Para transportar la voz necesitamos de un ancho de banda de 64 kbps, cada muestra es convertida a un código de 8 bits, entonces:
Si tenemos 8,000 x 8 bits =64,000 bits
Taza de muestreo x *N* bits = ancho de banda necesario

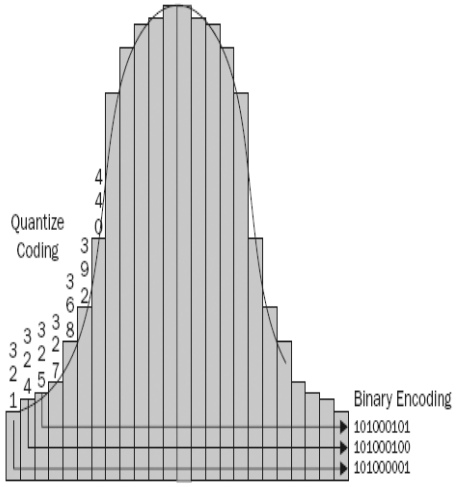


Figura 2. 4 Codificación binaria [4].

El proceso de codificar necesita utilizar un Códec, que es el algoritmo que se encarga de codificar y decodificar la señal, más adelante se detallará este concepto, en éste caso hablamos de un Códec de G.711 que codifica a 64kHz. Existen varias técnicas de codificación pero las más comunes en conversión binaria PCM son, a-law que es usada en casi todo el mundo y μ-law, utilizada en Canadá, Japón y Estados Unidos. Es importante saber que entre ellas no son compatibles y debe hacerse un proceso de tras codificación para su compatibilidad.

Ya una vez que tenemos la señal codificada esta lista para ser enviada por el medio físico, pero tenemos la posibilidad de comprimir antes de enviarlo, se aprovecha la compresión para mandar la mayor cantidad de datos y ser eficientes en el uso del cable.

Compresión: En este paso se trata de a provechar al máximo el canal, evitando tener redundancia en los datos ya codificados, al comprimirlos se reduce el tamaño de los datos que están por enviarse. De tal forma que en un mismo canal pueden establecerse mayor cantidad de conversaciones, cuando la señal llega al destino esta se descomprime en un proceso inverso con una señal muy parecida con la que se comprimió para no alterar la señal.

2.3. La telefonía convencional

La telefonía convencional también suele llamarse Red de Telefonía Básica (RTB) o Red de Telefonía Conmutada (RTC), conformada por un grupo de teléfonos interconectados exactamente de la misma forma como lo están hoy en día. De la central telefónica o bien del abonado a el domicilio donde se encuentra el teléfono por medio de cables de cobre.

En los inicios de la telefonía solo consistía de un teléfono en cada extremo, quién descolgará primero el teléfono iniciaba la conversación, a lo cual se le llamo "circuito ring down ", pero se requería tener un enlace físico para cada circuito. Sin embargo, no era viable en cuanto a infraestructura, seguridad y costos. Como solución se optó por conectar el enlace físico de cada cliente hacia un *switch* con el fin de conmutar la llamada desde allí. Para que un cliente realizara una llamada se requería de mucho tiempo, la persona deseaba realizar una llamada tenía que descolgar el teléfono e indicarle a un operador con quien se quería comunicar. Entonces el operador realizaría una serie de pasos entre diferentes operadores hasta lograr la conexión con el cliente deseado. El operador usaba un tablero con pequeños orificios que en sí eran las conexiones a diferentes lugares o regiones (ver **Figura 2. 5**), se creaba un circuito físico que solo podía ser utilizado entre los dos extremos que establecieron la llamada. Dicho circuito podía ser utilizado de nuevo hasta que la llamada finalizará, este tipo de circuitos son llamados circuitos conmutados.

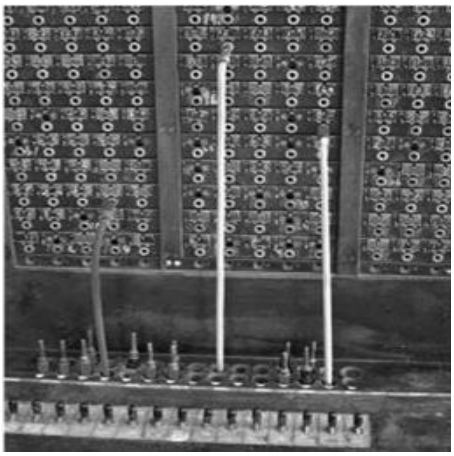


Figura 2. 5 Tablero de central telefónica antigua [6].

La red de telefonía crecía tan rápido que los operadores no se daban abasto, por lo que se remplazó al operador por un sistema mecánico encargado de establecer la comunicación. Éste producía señales eléctricas de acuerdo al número que se quería marcar y de igual manera se creaba un circuito conmutado. Años después la electrónica se apoderó de los mecanismos de las centrales telefónicas.

2.3.1 PSTN

La red telefónica pública conmutada (PSTN) es una red telefónica compuesta por un conjunto de dispositivos físicos que vinculados suministran un servicio de comunicación de voz a cortas y largas distancias de forma digital. Para lo cual es necesario que el sistema telefónico cuente con los medios y recursos adecuados entre los teléfonos para realizar el proceso de conexión y desconexión de la mejor manera. Estos procesos se logran gracias a funciones indispensables como la conmutación, señalización y transmisión.

La conmutación: Dependiendo de la conexión de los enlaces entre los conmutadores estos deben establecer la trayectoria adecuada por medio de la identificación y conexión de los abonados.

La señalización: Es el control sobre la red telefónica y la administración de las conexiones por medio del suministro e interpretación de señales de control y supervisión necesaria para realizar la conmutación correcta.

La transmisión: Se refiere a la forma en que se transmite la información de acuerdo a su contenido ya sea datos, voz o ambos, además de enviar las señales de control, ambas señales enviadas por medio del canal o enlace físico.

2.3.2 Arquitectura de PSTN

La conexión de una llamada telefónica puede ser desde la trasmisión de voz entre dos teléfonos por medio de una sola central telefónica hasta múltiples centrales interconectadas, de las cuales se pueden crear múltiples trayectorias, ya sea en frecuencia o sistemas de onda portadora.

La arquitectura básica usada por PSTN es un bucle local o bien *local loop* mejor conocido como línea de abonado o línea telefónica. Se trata de un circuito de acceso dedicado de 5Km, que va desde el teléfono de un domicilio hasta la primera central telefónica (Central Office Switch) por medio de un enlace físico con un par de cables de cobre. Este medio sigue siendo analógico hasta la fecha, una vez que llega a la central telefónica es convertido a digital. La conexión que enlaza las centrales telefónicas se llama enlace troncal, que son circuitos conmutados.

La gran demanda en el uso de la red telefónica a obligado a los proveedores de servicio a clasificar a los usuarios de acuerdo a zonas geográficas, dichas áreas se asocias con una o varias centrales de conmutación conectadas entre sí. Con el propósito de crear múltiples opciones de conexión que forme una jerarquía en la red. Donde cada central de conmutación tiene un límite de conexiones tanto de usuarios como de centrales. Esta red jerárquica define que cada central de un determinado nivel solo depende de otra central de un nivel superior, idealmente. La distribución geográfica de la red consta de 3 tipos, las urbanas, interurbanas e internacionales.

Las redes urbanas están formadas por los circuitos de abonados y los encales centrales locales, trasmiten en banda base a baja frecuencia.

Las redes interurbanas están formadas por la interconexión de diferentes ciudades. Los enlaces tienen mejores características de *QoS* y pérdidas. Ya sea instalando bobinas de carga cada 1830 metros para reducir la atenuación o bien conectando las ciudades con cable coaxial, fibra óptica, etc., que cuentan con una mayor capacidad de transmisión y calidad.

Por último las redes internacionales interconectan países por medio de enlaces de alta capacidad (miles de circuitos full-dúplex), ya sea por tierra, submarinos o satélite, con doble enlace entre ellos por cuestiones de seguridad.

2.3.2.1 Niveles de centrales de conmutación

Las centrales de conmutación son las encargadas de abrir el circuito para establecer la llamada entre dos usuarios de forma automática. El circuito puede ir desde una central local hasta una central internacional dependiendo donde se encuentren ubicados los usuarios que pretenden comunicarse. Existen niveles de centrales por donde es necesario que lleguen las solicitudes de llamada para encaminar las llamadas. Las centrales están organizadas de forma jerarquita como se muestra en la **Figura 2. 6**.

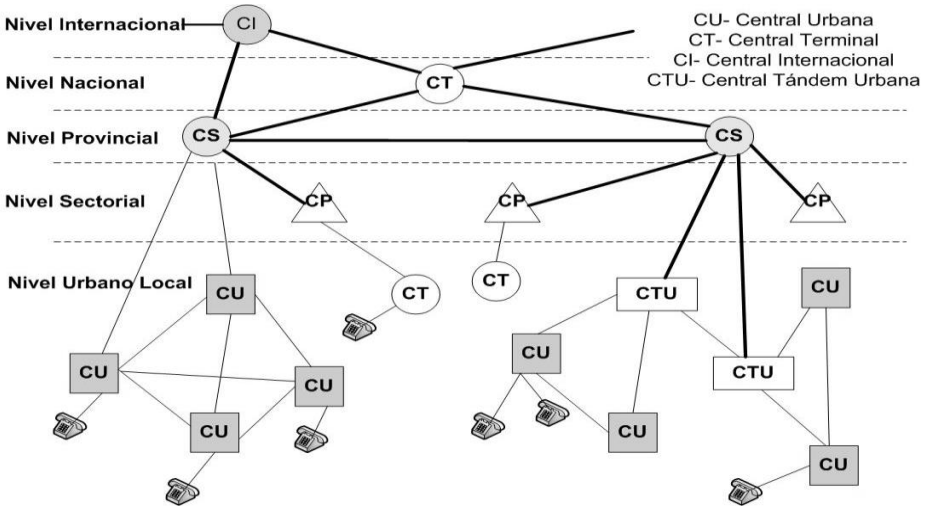


Figura 2. 6 Estructura Jerárquica de una Red Telefónica Convencional.

2.3.2.2 Enlaces de voz analógica

Las interfaces más comunes analógicas que conectan el bucle local con la Oficina central se muestran a continuación.

- **FXS (Foreing Exchange Station Interface).** Interfaz de abonado externo es conectada directamente a una terminal analógica de un teléfono o fax por medio de un puerto RJ-11 (ver **Figura 2. 7**). Proporciona voltaje y señalización a la terminal analógica.

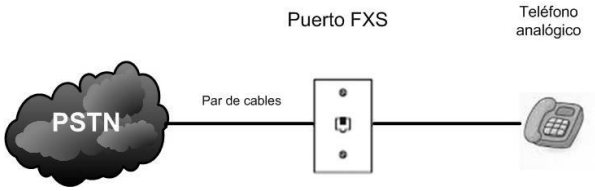


Figura 2. 7 Interfaz FXS.

- **FXO (Foreing Exchance Office Interface**. Interfaz de central externa, se conecta directamente a la PSTN, no genera señalización, normalmente es utilizado para conectar redes IP a las líneas analógicas PSTN o a extensiones analógicas de un PBX.

Para comunicar varias terminales analógicas con la PSTN, se conecta un PBX a la PSTN por medio de una interfaz FXO. Mientras que en el otro extremo del PBX se conectan los teléfonos analógicos, por medio de una interfaz FXS a través de una o más líneas telefónicas (ver **Figura 2. 8**) con el fin de reducir costos, utilizando solo una línea telefónica para todos.

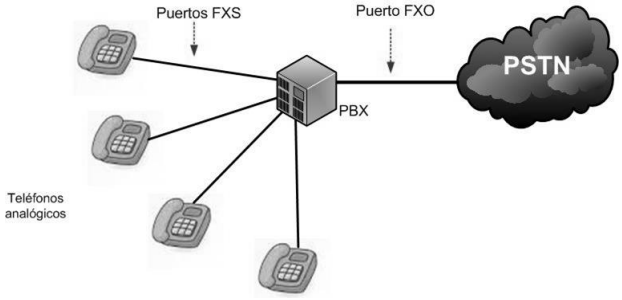


Figura 2. 8 Interfaz FXO.

2.3.2.3 Enlaces troncales digitales

Los enlaces troncales digitales son interfaces lógicas o físicas que pueden tener múltiples interfaces lógicas que se conectan solo a un destino. Son sumamente útiles cuando un negocio necesita más de 10 líneas externas, ya que el tener 10 líneas analógicas no es factible.

Los enlaces digitales más comunes que brindan las redes PSTN son T1, E1 e ISDN, las cuales se explicarían a continuación.

Enlace T1

Este enlace transmite paquetes de voz utilizando TDM (Time Division Multiple Access) y hace uso de la señalización CAS (Señalización por canal asociado). Utilizados en USA, estos enlaces son conocidos como DS-0.

EL enlace T1 tiene una taza de transmisión de 1.54 Mbps, está formado por 24 canales o *time slot*s de 64kbps, en cada canal se puede realizar una llamada de voz y tener el mismo tiempo para cada llamada, cada canal pueden transmitir 8 bits de tráfico de voz. El enlace utiliza un bit de cada canal para la señalización y envía información a través del enlace, a lo cual se le llama señalización por robo de bit o *Robbed Bit*. El bit robado es el que completa el tamaño del *frame* de 193 bits en el enlace, también se dice que la señalización se da dentro de banda porque toma un bit dentro de los canales para la señalización.

Entonces si la señal de voz se muestrea a 8000 ciclos por segundo. Necesitamos enviar 8000 *frames* con un tamaño de 193 bits, por cada segundo, de acuerdo a 8000 x 193 = 1544000 bps que es equivalente a una taza de transferencia de 1.544 Mbps. Se debe incluir el bit de señalización, entonces queda de la siguiente manera, 8000 x 192 = 1536000 bps o bien 1.53 Mbps de taza de transferencia real para un enlace T1.

Todos los canales pueden ser reutilizados simultáneamente. Sin embargo, la información de señalización reduce el ancho de banda total. Desafortunadamente por cada canal solo está disponible 56 kbps de 64, entonces se debe buscar una forma de comprimir la voz para poder hacer eficiente la transmisión de los 56 kbps de voz.

Existen dos formatos de codificación.

Súper trama (SF): Una súper trama está formada por 12 tramas de 193 bits, de la cuales la 6 y 12 aportan un bit cada uno para la señalización, a los cuales se les denomina bits A y B, la trama se muestra en la **Figura 2. 9**.

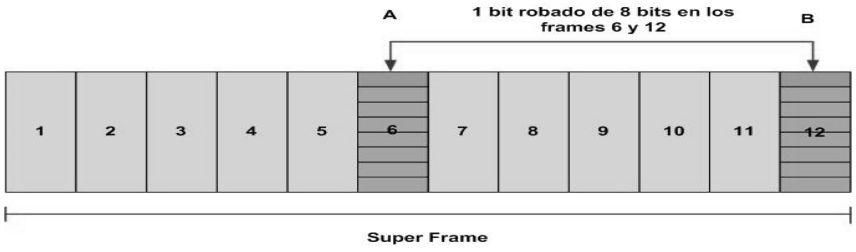


Figura 2. 9 Bit robado de SF.

- **Súper Trama Extendida (ESF), Extended Super Frame:** Formada por 24 tramas, como los grupos son más grandes libera más bits para la señalización, entonces ahora tenemos 4 bits en vez de dos.

Las tramas número 6, 12, 18 y 24 son de señalización conocidos como A, B, C y D respectivamente (ver **Figura 2. 10**), los cuales pueden detectar errores de manera más inteligente y con la capacidad de procesar la verificación de redundancia cíclica (CRC). Actualmente es el más utilizado por los proveedores de servicio.

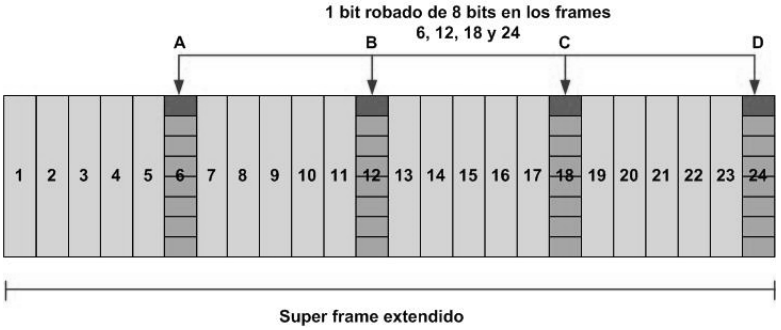


Figura 2. 10 Bit robado de ESF.

Enlace E1

Este tipo de enlaces se utilizan en Europa, Asia, Sudamérica y América Central, el enlace utiliza TDM (Time-Division Multiplexing) para transportar los paquetes.

Cada enlace está compuesto por 32 canales o *time slots* de 64 kbps, teniendo una taza de transferencia de 2.048 Mbps, utiliza 2 canales para control y señalización, de los cuales el canal 1 es para sincronización y el canal 17 para señalización de banda como se muestra en la **Figura 2. 11**. A esta forma de utilizar canales específicos de señalización se le llama fuera de banda.

Pero sigue existiendo robo de bit de señalización, pero aquí los canales son definidos para la señalización.

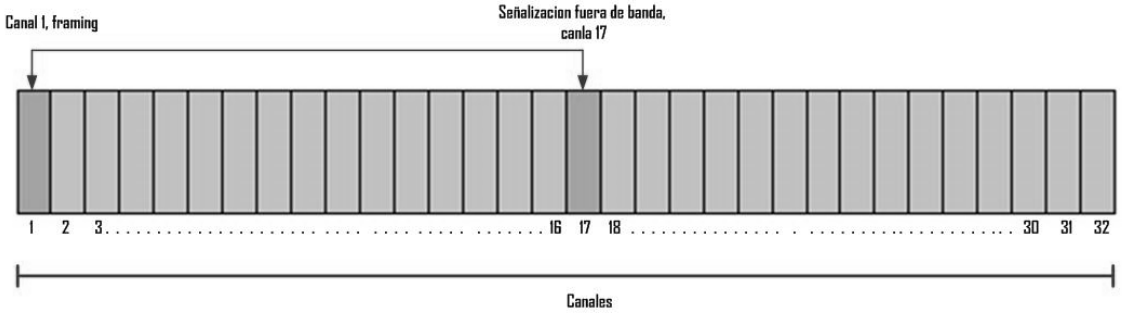


Figura 2. 11 Canales E1.

ISDN

La red se servicios integrados ISDN (Integrate Services for Digital Network), es una red digital de extremo a extremo la cual ofrece servicios con una alta calidad de servicio en comparación con la red de telefónica.

Éste utiliza un canal para la señalización y otro para transmitir voz.

Los canales de voz: son conocidos como *bearer* o canales B, los cuales transmiten a una velocidad de 64 Kbps.

Los canales de señalización o control: se conocen como canales D, transmiten a 16Kbps, éstos pueden ser utilizados para transmitir voz en caso de que no se esté transmitiendo nada referente a señalización o control.

Podemos transmitir datos a grandes velocidades si combinamos lo canales B, en la **Tabla 2.1** de abajo se muestra una pequeña descripción.

Canal	Velocidad	Funciones
B	64 kbps	Transfiere datos y voz
D	16/64 kbps	Señalización y control. Trasmite datos a bajas velocidad.
H_0	384 kbps (6 canales B)	Transmisión de datos a alta velocidad o audio/video de alta calidad.
H_{10}	1472 Kbps (23 canales B)	Transmisión de datos a alta velocidad o audio/video de alta calidad.
H_{11}	1536 kbps (24 canales B)	Transmisión de datos a alta velocidad o audio/video de alta calidad.
H_{12}	1920 Kbps (30 canales B)	Transmisión de datos a alta velocidad o audio/video de alta calidad.

Tabla 2.1 Descripción de canales utilizados en ISDN.

Existen dos interfaces para acceder a ISDN por medio de BRI o PRI.

BRI (Basic Rate Interface)

La interfaz está formada por dos canales B de 64 Kbps y un canal D de 16Kbps. Utilizado en Europa para proveer servicios a oficinas y casas.

PRI (Primary Rate Interface)

Es comúnmente utilizado en la industria debido a su alta demanda de usuarios, principalmente en Norteamérica, Japón y Canadá. Está formado por 23 canales tipo B de 64Kbps y uno tipo D de 64Kbps, lo que equivale a un enlace T1.

2.3.3 Señalización de voz en circuitos digitales

2.3.3.1 Sistema de señalización N. 7 (CCS7)

Estándar de señalización por canal común 7 (CCS, Common Channel Signalling 7) que brinda señalización a la red PSTN, normalizado internacionalmente por la ITU-T que describe una serie de recomendaciones Q. 700-Q.799 Specifications of Signalling System No. 7.

Respecto a canal común se refiere al uso de un canal para enviar la señalización de varios canales de voz. El objetivo de la señalización es crear un lenguaje técnico para intercambiar información de control, con el fin de conectar dos líneas telefónicas ubicadas en cualquier punto dentro de la red telefónica. Se utiliza para proveer la señalización a la PSTN, establecer llamadas, enrutar, operar y tarifar los servicios de dicha red. Actualmente es de gran utilidad para poder enlazar la PSTN con la Red VoIP. Este sistema de señalización funciona de forma óptima con canales digitales de 64 kbps, aunque también puede trabajar con velocidades más bajas y en canales analógicos. Trabaja en enlaces punto a punto terrestre o satelitales

La señalización por canal común es una forma de enviar toda la información de señalización de varios circuitos y la información de gestión de la red como mensajes etiquetados por medio de un solo canal. Cuenta con algunos medios para proteger los mensajes en caso de fallos o perturbaciones en la red, como detección y corrección de errores en cada enlace de señalización. También utiliza la redundancia y la desviación automática del tráfico de señalización por caminos alternos.

2.3.4 Servicios PSTN

La PSTN brinda amplia variedad de servicios, que funcionan sin problemas en la misma red, pero aún existen problemas de compatibilidad con VoIP, se están desarrollando tecnologías para adaptar los servicios de PSTN a telefonía IP. Algunos de los servicios que ofrece PSTN son:

- Plain Old Telephone Service (POST)
- VPN’s, Call-center services, CENTREX
- Servicios SP o Proveedores de servicio.

2.3.4.1 POST

Es conocido como Servicio de Telefonía Estándar y los servicios que ofrece son:

1. **Servicios especializados de llamada**. Estos servicios por lo regular lo ofrecen los proveedores de servicio en paquetes, desde las oficinas centrales se controlan y habilitan directamente en los *switches*. Los servicios más comunes se describen a continuación.

    - Desvió de llamadas
    - Tres a la vez
    - Llamada en espera
    - Número añadido

2. **Marcación rápida:** Asignación de un código a los números frecuentes.

3. **Voice mail.** El Servicio de voz está activo a pesar de que la línea telefónica esté ocupada, no necesita equipo extra para que el servicio funcione. Los dos principales servicios son la mensajería de voz y el FAX.

4. **Servicios de Señalización Especializada de Área Local (CLASS).** El usuario de la línea telefónica puede controlar las llamadas entrantes y salientes de su teléfono. Los CLASS más comunes se presentan son:

    - Remarcado automático: Permite al usuario regresar una llamada perdida.
    - Protección de identidad: Un usuario puede ocultar su número telefónico a las personas a las cuales el llame.
    - Regreso de llamada automática: Cuando un usuario intenta hacer una llamada y dicho número al cual quiere marcar está ocupado, entonces en cuanto la línea se desocupe enviará un tono diferente al solicitante de la llamada y realizará la llamada automáticamente.
    - Customer-originated trace: Si un usuario recibe una llamada de extorción, él puede enviar un código a las autoridades para notificar el hecho.
    - Call-screening. El usuario puede seleccionar las llamadas que acepta, rechaza o deja en modo de espera de acuerdo a un registro previo de llamadas recibidas.

2.3.4.2 Redes Virtuales Privadas de Voz (VPN de voz)

Conocidos como enlaces dedicados virtuales de voz. Son redes que interconectan dos redes físicas de la PSTN pero de forma virtual y no físicamente. Si un usuario requiera de un enlace dedicado tendrá que pagar un alto precio, a cambio tendrá una alta calidad de servicio y nunca tendrá problemas de tráfico. Las VPN`s se identifican en la red por medio de un ID que viaja en la PSTN por el SS7, con este ID el tráfico es enrutado hacia el destino e identifica las llamadas públicas de las privadas internas de la red.

1. Centrex Services

Central Office Exchange Service, son pequeñas centrales virtuales sobre centrales digitales públicas del proveedor de servicios. Centrex brinda servicios de voz y datos en la pequeña red a un bajo costo y sin equipo extra en las instalaciones del usuario.

2. Call center

Éste es uno de los servicios más conocidos por las personas, muchas veces hablamos a alguna compañía y está ocupada la línea. La llamada entra en un tono de espera hasta que algún agente se encuentre disponible para atendernos. Un Call *Center es* un servicio de telefonía que recibe una gran cantidad de llamadas, las cuales tiene que distribuir y enrutar eficientemente hacia el personal adecuado para ser atendidas.

2.3.4.3 Servicios de Proveedor de servicios (SP)

El servicio consta de dar soporte a los usuarios de la PSTN como los siguientes.

1. **Base de datos.** Con este servicio los proveedores de servicio pueden mantener, acceder y traducir determinada información útil para brindar servicios y tener acceso a números especiales como 900 y 800.
2. **Servicios de operador**
    - **Servicios de directorio telefónico**
    - **Servicio de facturación**
    - **Asistencia y tasación**

2.4 ¿Qué es Telefonía IP?

La telefonía IP es la tecnología compuesta de *hardware*, *software*, protocolos y estándares que permiten transportar voz (en forma de paquetes) sobre la red de Internet. Mientras que *VoIP*, *Voice over IP* y Voz sobre protocolo de Internet, son sinónimos del servicio como tal que transita sobre la red que originalmente fue creada para transmitir datos y que ahora es adaptada para transportar la voz.

Es necesario aclarar que el Protocolo de Internet es un protocolo de conexión, quiere decir que es un conjunto de normas que ambas extremos deben utilizar para comunicarse. Haciendo una analogía podemos decir que es un lenguaje que ambas partes (emisor y receptor) deben dominar para que logren entenderse.

Con la llegada de esta tecnología se han facilitado muchos procesos y servicios que sin ella serian complejos al realizarlos con la red PSTN.

Como por ejemplo

- Se pueden realizar más llamadas sobre una misma línea telefonía sin tener que conectar nuevas líneas en caso de que crezca el personal en la empresa.
- Algunas funcionalidades que la telefonía convencional cobra como cargos extra, en la telefonía IP no tiene costo alguno, ya que utiliza la misma red de internet.
- Permite tener sistemas y equipos conectador e integrados para tener acceso desde cualquier punto y en tiempo real.

2.4.1 Ventajas

Una de las principales ventajas en la telefonía IP es la utilización de la infraestructura física, la implementación de un sistema PBX necesita una importante infraestructura de cableado para crear los abonados entre terminales y centrales. Con VoIP solo necesitamos tener acceso a una red TCP/IP, la cual hoy en día cualquier empresas u oficina cuenta con ella.

Ahorro de costos, al enviar tráfico de voz sobre redes IP. Ya que el servicio de Internet es una renta mensual con determinado ancho de banda independientemente de los tipos de tráfico que se envié.

Los estándares abiertos y la interoperabilidad con la que funciona VoIP permiten tanto a negocios como a proveedores de servicios utilizar equipos de diferentes fabricantes sin preocuparse de si estos funcionarán correctamente a pesar de ser de diferentes marcas.

También se tiene acceso a redes corporativas desde pequeñas sucursales por medio de redes entregadas de voz.

2.4.2 Desventajas

Sin embargo también existen algunas desventajas sobre la telefonía IP, como por ejemplo:

El gran costo de los teléfonos IP en comparación con los teléfonos tradicionales.

La total dependencia de la red de datos para las comunicaciones en una empresa, ya sea en una red local que se produzca un fallo en un enlace o en un equipo o el *software* no funcione correctamente, lo cual provoca que la red de voz quede inhabilita. O bien si en la red pública se produce un fallo en la conexión a Internet se pierde toda posibilidad de realizar llamadas [7].

Las redes de datos depende totalmente del uso de energía eléctrica y esto puede ser uno de los principales problemas en las empresas cuando no cuentan con generadores de energía. Si no se cuenta con energía eléctrica los equipos son incapaces de funcionar como hace la telefonía convencional.

2.5 Red IP

La red IP conecta dispositivos IP y convencionales que en conjunto funcionan para realizar llamadas entre dispositivos por medio de Internet a bajos costos o de forma gratuita, en la **Figura 2. 12** se muestra la conexión entre terminales IP y terminales tradicionales por medio de la red IP.

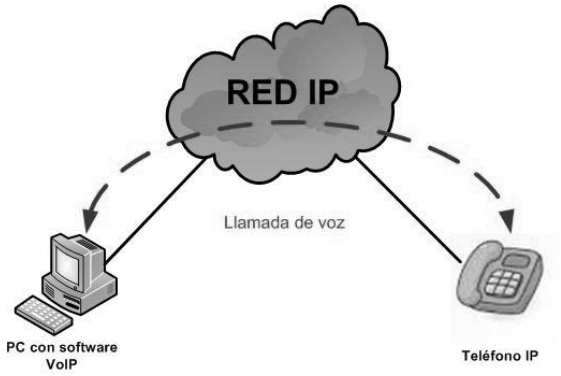


Figura 2. 12 Dispositivos IP conectados.

Los proveedores de Servicios de Telefonía por Internet (ITSP) son los encargados de interconectar la red telefónica pública con Internet mediante *gateways*. Permite a los corporativos realizar llamadas dentro de la red del proveedor a bajos costos, o bien establecer llamadas fuera de la red IP o sea a PSTN.

Una red IP funciona de la siguiente manera, primero la señal analógica que se emite por el micrófono del teléfono es digitalizada en señales PCM (Pulse Code Modulation) utilizando un codificador/decodificador de voz (CODEC). Las muestras obtenidas se comprimen mediante un algoritmo de compresión y se fraccionan en paquetes que pueden enviarse a través de una red privada WAN. Entonces se envían los paquetes por un extremo WAN y en el otro extremo se realiza el proceso inverso, primero se descomprime y después se decodifica la señal, como se ve en la **Figura 2. 13**.

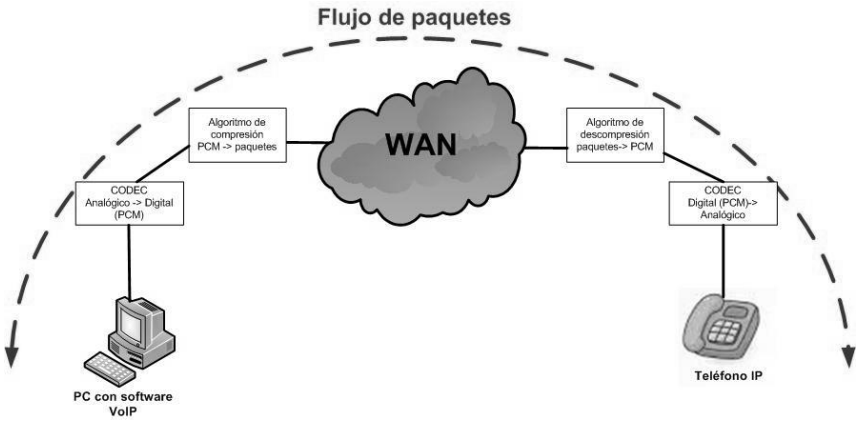


Figura 2. 13 Flujo de paquetes de voz en una red IP.

Existen características muy peculiares de una red VoIP, independientemente si se cuenta con el *hardware* especializado o no para dicho servicio, como son:

VoIP permite el control de tráfico de la red, como resultado reduce las posibilidades de que la red se caiga.

- Proporciona el enlace a la red telefónica tradicional.
- Es independiente del tipo de red física y brinda la conexión a grandes redes de IP
- Es independiente del *hardware*
- Puede implementarse en *hardware* y *software*.

Como ya se mencionó, en vez de utilizar un teléfono IP, se utiliza una PC con un *software*, que remplaza las funciones de un teléfono IP. Además de facilitar la conexión a una red IP y reduce los costó es gran medida. También con la utilización del *software* sobre una PC obtenemos otras funciones extras tales como [8]:

- Una agenda compartida y personal
- Organizador de llamadas
- Remarcación automática
- Reconocimiento de voz.

2.5.1 Arquitectura de la red VoIP

En la telefonía IP el cambio fundamental radica en la red de enlaces, que se sostiene en una red basada en el protocolo IP como la red de Internet. Mientras que en la parte del abonado se conserva el enlace físico de un par de cobre.

En la **Figura 2. 14** se muestras una arquitectura general de una red VoIP, donde el *gateway* convierte las señales provenientes de la telefonía tradicional (POST, T1/E1, ISDN u otras) a VoIP. Y el *Gatekeeper* provee el manejo y funciones administrativas para el enrutamiento de llamadas sobre la red. Finalmente la red IP provee la conectividad entre las terminales, cabe mencionar que la red IP puede ser privada, intranet o Internet.

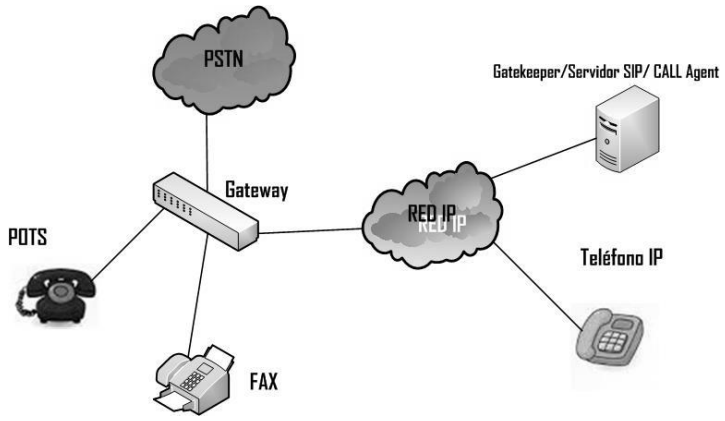


Figura 2. 14 Arquitectura general de una red VoIP.

2.5.2 Elementos de una red VoIP

En base a la recomendación hecha por H.323 existen tres elementos básicos para que la red IP funcione tales como; las terminales, *gateway* y el *gatekeeper*, además de necesitar otros elementos que la red de Internet ya tiene incorporados pero no por ello son menos importantes [2].

2.5.2.1 Terminal

También son conocidos como clientes, existen dos tipos de clientes el primero es un *software* corriendo en una PC, controlado por una interfaz gráfica. El segundo es un cliente virtual que radica en el *Gateway*.

Las funciones principales del cliente son establece y terminar la llamada de voz. Además de codificar, empaquetar y transmitir la información del micrófono del usuario que está hablando y en el otro extremo el cliente recibe, decodifica y reproduce la información de voz en los altavoces.

2.5.2.2 Gateway VoIP

Gateway de VoIP o pasarela, es un dispositivo que sirve para conectar redes que utilizan diferentes protocolos de comunicación con el fin de pasar de un lado al otro la información.

La función de los *gateways* es proveer las interfaces con la telefonía convencional apropiada, o sea un puente de comunicación entre los usuarios. También juegan un papel importante en cuanto a seguridad de acceso, confiabilidad, control de *QoS* y su mejoramiento. Por medio del *Gateway* es posible realizar tres tipos de llamadas de voz sobre IP, llamadas de PC a PC, de PC a teléfono y de teléfono a teléfono, siendo el *gateway* el encargado de enlazar la red VoIP con la red tradicional.

Existen dos tipos de gateways

Media *Gateway* (MGW). Solo se usan para la conversión de datos, desempeña un papel como traductor entre diversas redes de telecomunicaciones.

Signalling gateways. Este *Gateway* es responsable del tráfico de señalización de mensajes, como la facturación, ubicación, mensajes cortos, dirección de conversión etc.

2.5.2.3 Gatekeeper

Un gatekeeper es llamado de diferentes formas de acuerdo al protocolo de señalización que se esté utilizando. Se llama *gatekeeper* bajo el protocolo H.323, servidor SIP bajo el protocolo, (tales protocolos se definirán en el capítulo siguiente).

Las terminales pueden conectarse directamente sin tener de por medio un *gatekeeper*, pero sin él todo es limitado y difícil para el usuario. Cada terminal o usuario antes de realizar una llamada consulta al *gatekeeper* para verificar si la llamada puede realizarse. Entonces el *gatekeeper* accede a los servidores y verifica los datos, para permitir o denegar la conexión. Si el permiso es concedido ahora el *gatekeeper* se encarga de enviar el identificador de usuario destino y la dirección IP equivalente. Cuando el *gatekeeper* establece la llamada deja de intervenir entre las terminales.

2.5.3 Otros elementos de la red VoIP

a) Routers

Los *router* son dispositivos muy conocidos en la redes de datos, son dispositivos de *hardware* que operan en la capa 3 (capa de red) del modelo OSI. Este *hardware* permite interconectar redes de computadoras y se encarga de determinar la ruta y enrutar los paquetes entre redes conectadas a él.

b) ATA

Adaptador de teléfono analógico es un dispositivo electrónico que permite a uno o más teléfonos analógico utilizar VoIP, o sea que crea una conexión física mediante el teléfono e Internet entre los cables de un teléfono tradicional o fax y un ordenador o una pasarela Ethernet. Permite seguir utilizando los teléfonos normales sin necesidad de reemplazarlos y realizar gastos extras.

c) Conmutadores

Es un dispositivo que encamina y dirige de forma ordenada líneas telefónicas.

Existen tres tipos de conmutadores

Conmutadores multilínea: Pueden conectarse hasta 65 teléfonos para brindar atención de llamadas multilínea.
PBX: Soportan desde 30 hasta 10 mil líneas, también mensajes voz y datos. Además de tener conectividad a redes LAN, telefonía inalámbrica y servicios de información.
Centrales telefónicas: Son demasiado complejas y son usadas principalmente en comunicaciones multinacionales y grandes empresas que manejan tráfico de voz mayor a 10 mil líneas telefónicas.

d) Tarjetas de red

Los PBX tienen diversas tarjetas que componen sus sistemas, dependiendo la cantidad y el tipo de tarjetas de red que esta tenga. Ya sean tarjetas digitales y analógicas.

e) Códecs y protocolos

Tanto los códec como los protocolos que utiliza VoIP son sumamente importantes y requieren una amplia explicación la cual será explicada en el Capítulo 3.

2.6 Estado del Arte

Desde mediados de los 90´s la unión de dos servicios, el Internet y la telefonía convencional formaron lo que hoy llamamos VoIP, y con ello ha propiciado a un desarrollo de protocolos, *software*, *hardware,* normas, recomendación, investigación etc. De iniciar como un experimento para poder realizar una llamada entre dos computadoras las cuales debían de tener las mismas características físicas. Ahora no importa si es un teléfono fijo, un celular, una computadora, un *softphone* un teléfono IP, actualmente se cuentan con múltiples tecnologías que hacen posible la interoperabilidad entre ellos. Hoy en día con ya casi 20 años de los inicios de VoIP, es implementada en cualquier dispositivo inteligente, y mejor aún en cuanto a costo y tiempo la implementación en centros telefónicos, empresas, escuelas y por qué no también en hogares.

Cabe señalar que eso no sería posible sin la implementación de protocolos de señalización, puesto que estos son los encargados de generar toda la señalización para que las llamadas puedan establecerse. Actualmente las telecomunicaciones han tenido un crecimiento bastante grande. Con el uso de Internet, aplicaciones, teléfonos inteligentes, tabletas, protocolos y entandares, se tiene un mundo de tecnología que permite la comunicación con personas que se encuentran a grandes distancias de una forma rápida, fácil y sin la necesidad de trasladarse a esos puntos. Gracias a la llegada de la telefonía IP, aparte de ofrecer grandes ventajas y factibilidad, también nos ofrece una comunicación en tiempo real a un bajo precio e incluso en algunos casos de forma gratuita.

La constante mejora de los protocolos de señalización VoIP ha motivado a realizar diversos estudios de dichos protocolos, ya sea analizando el mecanismo de señalización, analizando al calidad de servicio, mejorar los servicios etc., [9], [10], [11].

Entre los protocolos más populares se encuentran H.323 y SIP. Este último cuenta con una ligereza en sus mecanismos que ha sido foco de múltiples investigaciones en los últimos años. Por otro lado H.323 es un estándar más maduro, añade servicios complementarios y funciones junto son PSTN, crea interoperabilidad entre aplicaciones. Mientras que SIP está diseñado bajo una sintaxis y semántica en cuanto a características y descripción de una sesión.

Hoy en día existe una fuerte competencia entre los protocolos de señalización, entre cual puede trabajar y ser más eficiente sobre Internet[12], se realiza una comparación entre el protocolo SIP y IAX (InterAsterisk Exchange). Éste último aún no estandarizado y está causando ruido en tecnologías de VoIP, pero tiene desventajas con respecto a SIP y H.232. Desafortunadamente aún no se encuentra estandarizado y estructuralmente es muy parecido a SIP, tendiendo como ventaja el uso de ancho de banda requerido es menor a otros protocolos. Se utiliza el MOS (Mean Opinion Score) para medir de forma subjetiva la calidad de la voz evaluada con (ACR, Absolute Category Rating).

Los protocolos H.323 y SIP han sido puntos de comparación de diferentes índoles tal es el caso [13], donde se realiza una comparación orientada a la arquitecturas de servicios , haciendo una evaluación profunda en los mecanismos de aplicación de servicios. Se basa en un estado de normalización, servicios soportados, interoperabilidad de los servicios, arquitectura de servicios suplementarios, entro otros.

En [14] se realiza una comparación entre SIP y H.323 en cuanto a *QoS*, escalabilidad, flexibilidad, interoperabilidad y seguridad, considerando escenarios similares para ambos protocolos. Como resultado aseguran que SIP y H.323 han mejorado gracias a que han ido aprendiendo uno del otro. En cuanto a funcionalidad son parecidos pero H.323 están mejor definidos que los de SIP por lo cual hace a H.323 mejor en cuando interoperabilidad y mayor compatibilidad con H.323, mientras que SIP tiene una mayor flexibilidad lo que lo hace adaptable.

Las tecnologías de conmutación y enrutamiento de paquetes se están mudando y evolucionado a redes IP, ATM y Frame Relay y tecnologías de enrutamiento óptico. Creando una oportunidad de migrar el circuito tradicional conmutado de telefonía a una red de tecnología de voz sobre Internet, y a partir de ello poder integrar las redes inalámbricas con tráfico de voz.

Aunque todavía presentan grandes desafíos, como se explica en [15] donde se realizó una comparación de rendimiento de H.323 y SIP bajo redes inalámbricas, toman en cuenta factores externos a la red.

En la publicación [16] se realiza una investigación sobre pruebas de rendimiento en una infraestructura SIP, comparando el rendimiento que entre un servidor SIP y B2BUA. Utilizando dos escenarios respectivamente y simulando usuarios así como de encontrar una metodología para evaluar SIP, los resultados obtenidos fueron rendimientos muy idénticos.

A pesar de que SIP y H.323 siguen en contantes batallas por ganar el mercado, siguen operando y mejorando, lo que propicia a seguir estudiando e investigando las contantes mejoras y el rendimiento que aportan para su implementación.

2.7 Conclusión

Se estudiaron los antecedentes de la telefonía convencional, la arquitectura y el protocolo de señalización utilizada por la PSTN. Así como también los mecanismos y pasos para establecer una llamada telefónica tradicional. Aunado a ello, se estudiaron y analizaron los elementos generales que conforman una red de telefonía IP, lo que conlleva a estudiar la arquitectura y protocolos utilizados por VoIP. Además se analizaron las ventajas y desventajas de utilizar VoIP.

Capítulo 3. Estándares y protocolos utilizados por VoIP

3.1 Introducción

En este capítulo se explica en que consiste la voz sobre el protocolo de Internet, así como los protocolos utilizados en base al modelo OSI y los protocolos de señalización que trabajan en la capa de transporte. Se mencionan a detalle los protocolos de señalización SIP y H.323, así como sus arquitecturas y sus principales componentes.

Se estudian las estructuras y los diferentes tipos de mensajes, respuestas y estados que utiliza cada protocolo, además de explicar cuál es el proceso que sigue cada uno de ellos para iniciar, mantener y finalizar una llamada telefónica.

Se explica también otros protocolos que trabajan junto con SIP y H.323 para trabajar en conjunto y establecer llamadas. También se definen los principales elementos que se necesitan para poder usar esta tecnología.

3.2. Características VoIP

En torno al gran auge que ha tenido la nueva generación de telefonía, se han desplegado elementos importantes que la han impulsado y que sin estos tal vez la telefonía IP se encontraría estancada. Estos elementos son la VoIP, los estándares abiertos y el software libre.

La voz sobre IP, ya vimos que es la transmisión de voz en base al protocolo de Internet, lo cual no quiere decir que se deba de hacer uso concreto de algún mecanismo para enviar la voz por la Internet. Hoy en día existen diversas tecnologías que lo hacen posible, es necesario conocer que la VoIP se encuentra en dos rubros, por un lado encontramos las tecnologías cerradas propietarias y los sistemas abiertos o libres. Por ejemplo Skype o Cisco Skinny (SCCP) que son tecnologías propietarias. Skinny es un protocolo usado en las terminales, fue desarrollado por Selsius Corporation y actualmente está bajo el mando de Cisco Systems, Inc.

Por otra parte con tecnologías abiertas se hace referencia a los protocolos como SIP, *H.323*, IAX2 entre otros. Por ahora nos interesa saber que gracias a estándares abiertos y el código libre se puede construir una red.

Además de poder conectar la red telefónica tradicional, creando nuevos conocimientos y mejoras a lo que ya existe. También cada empresa, persona o comunidad puede decidir que tecnologías usar, adecuadas a sus necesidades y su desarrollo futuro.

Los estándares existentes para VoIP permiten que tanto sistemas como equipos puedan trabajar sin problemas de interoperabilidad. Gracias a ello podemos tener una amplia variedad de opciones para nuestros sistemas. También tenemos los sistemas que emulan el funcionamiento de dichos estándares como los ahora llamados *softphone*, que funcionan sobre una computadora o dispositivo con conexión a Internet. Entonces con este gran grupo de elementos se puede poner en marcha un sistema de VoIP. Es fundamental contar con acceso a los programas y equipos que permiten el uso de voz, además de una red abierta y pública para conectarse a Internet, que permita realizar modificaciones en cada elemento para adaptarla a nuestras necesidades.

3.2.1. Pila de protocolos

Lograr establecer llamadas de VoIP requiere de equipo y *software* especializado que operen con este concepto. Para lograrlo tienen que trabajar bajo ciertos protocolos, cuyo mecanismo de conexión abarca un grupo de transiciones de señalización entre terminales.

VoIP trabaja dentro del modelo OSI, algunas capas de este modelo VoIP utiliza protocolos específicos. De la capa 1 a la 3 son protocolos conocidos que trabajan sobre redes de datos, la capa de transporte y sesión son fundamentales para VoIP, en estos encontramos dos grupos de protocolos para VoIP (ver **Tabla 3.1**), estos grupos son:

- Protocolos de señalización VoIP
  -SIP
  -H.323
- Protocolos de transporte de voz.
  -RTP
  -RTCP

Aplicación	Asterisk, Elastix, etc.
Presentación	Códecs (G.729/G.711)
Sesión	*SIP*, H.323, MEGACO
Transporte	UDP/RTP/RTCP
Red	IP
Enlace	Ethernet
Física	Enlaces físicos

Tabla 3.1 Modelo OSI para VoIP.

3.3 Protocolos de señalización VoIP

3.3.1. SIP (Session Initiation Protocol)

En el año de 1996 la IETF creó un grupo de trabajo llamado MMUSIC (Multiparty Multimedia Session Control, Control de sesión multimedia), con el fin de desarrollar un protocolo que fuese capaz de establecer comunicación multimedia entre usuarios. Se propusieron dos protocolos, el primero fue llamado *SIP* (Session Initiation Protocol, Protocolo de inicio de sesión) y el segundo SCIP (Simple Conference Invitation Protocol, Protocolo simple de invitación a conferencia). Después de varias modificaciones se quedó como protocolo definitivo el *SIP*, quien propuso la utilización del protocolo HTTP y coautor de RTP y RTSP. La versión final de *SIP* fue publicada en Marzo de 1999 en la especificación RFC2542, después se crearon las especificaciones RFC3262 y RFC3266.

SIP fue pensado y hecho para ser utilizado sobre Internet utilizando el protocolo IP. Es un protocolo de señalización de extremo a extremo, lo cual quiere decir que toda la lógica y el estado de conexión son almacenados en los dispositivos finales a excepción de los mensajes de ruteo *SIP*.

A cambio de ello se tiene que pagar un costo muy alto al agregar todas las cabeceras de capa 2, 3 y 4 del modelo OSI en los mensajes. El ancho de banda necesario para generar una llamada incrementa con todas las cabeceras agregadas por las capas inferiores del modelo OSI.

SIP como protocoló de señalización está diseñado como protocoló de aplicación para establecer y gestionar las sesiones de los múltiples usuarios[17].

SIP trabaja en conjunto con otros protocolos, lo que permite gestionar sesiones con independencia de los otros protocolos de transporte que estén por debajo y sin dependencia del tipo de sesión que se establezca. En la cual están incluidos los protocolos: RTP para el transporte de datos en tiempo real, RTCP para la retroalimentación y brindar QoS, PSTN para controlar las puertas de acceso a la red telefónica publica conmutada y SDP para describir sesiones multimedia.

Cuando una sesión es establecida, los participantes de ésta intercambian directamente el tráfico, ya sea audio o vídeo a través del protocolo RTP (Real –Time Transport Ptotocol). Sin embargo, SIP no es capaz de reservar recursos, por lo cual no ofrece calidad de servicio. Solo es capaz de enviar mensajes de señalización cortos con el objetivo de establecer, mantener, modificar y liberar sesiones multimedia. También es capaz de invitar participantes a sesiones ya existentes como las conferencias *multicast*, o bien añadir o quitar contenidos en una sesión ya existente. SIP ha crecido con el fin de soportar servicios tales como mensajería instantánea, transferencia de llamadas, conferencias, y otros servicios complementarios de telefonía, etc.

Para que se establezca y finalice una sesión multimedia, SIP toma en cuenta las siguientes cinco etapas [18].

- Localización de los usuarios: Se refiere al sistema final utilizado para la comunicación.
- Disponibilidad de usuarios: Indica si el usuario quiere o puede ser partícipe de la comunicación.
- Negociación e intercambio de capacidades entre las terminales: Indica el medio y los parámetros del medio a utilizar.
- Establecimiento de la llamada: Se inicia y establece la sesión, así como los parámetros en cada extremo de comunicación (suena el teléfono).
- Gestión y mantenimiento de la llamada: Transferencia y finalización de sesiones, modificando los parámetros de la sesión y llamando a los servicios solicitados.

La seguridad dentro de SIP toma un papel muy importante por los servicios que esté pude ofrecer. Por lo tanto SIP tiene un conjunto de mecanismos de seguridad, tales como prevenir la denegación de un servicio, autenticar a los usuarios y servidores proxys, proteger la integridad y los servicios de privacidad y cifrado. Los clientes SIP usan el puerto 5060 en TCP y UDP para conectarse con los servidores *SIP*, en caso de utilizar un protocolo seguro como *SIP* entonces se utiliza el puerto 5061 y que soporta IPv4 e IPv6.

a) Elementos de una red SIP

El diseño de SIP se basa en el modelo cliente-servidor, el cliente/usuario envía solicitudes (*requests*) al servidor, el cual responde (*response*) ya sea para aceptar, rechazar o re direccionar las solicitudes. El protocolo SIP permite a los usuarios establecer sesiones entre dos o más participantes para el intercambio de información multimedia (datos, voz y vídeo).

La arquitectura SIP también puede hacer uso de otro tipo de servidores para formar redes más complejas. Los dos equipos básicos para *SIP* son las terminales o también conocidos como Agentes de usuario- User Agent (UA) y los servidores de red ver **Figura 3. 1**. Los (UA) son los equipos finales utilizados por los usuarios y se compone de dos partes, el User Agent Client (UAC) y el User Agent Server (UAS). Mientras que los servidores son equipos intermedios que aportan la funcionalidad a las redes *SIP* [17].

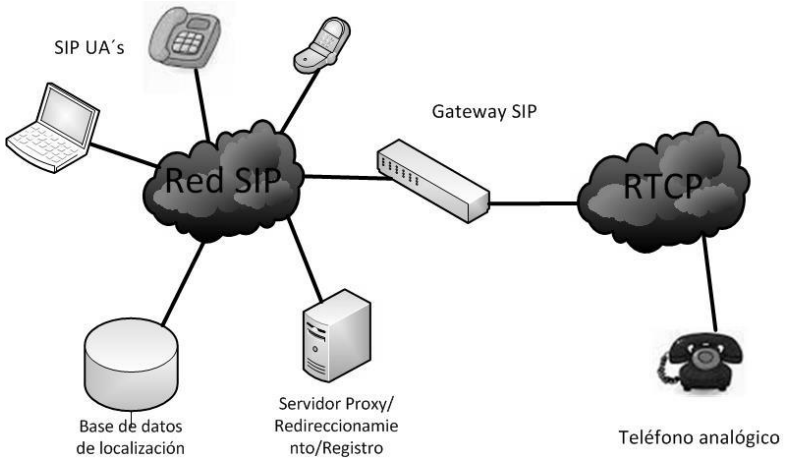


Figura 3. 1 Elementos de una red SIP.

Agentes de usuario (UA)

Los agentes de usuario son las terminales o clientes manejados por una persona o un *software*, que emiten o reciben los mensajes del protocolo *SIP* (videoteléfono, teléfono, *softphone*, etc.).

A continuación se describen los dos tipos de UA.

- **Agente de usuario cliente o User Agent Client (UAC):** Es una aplicación o equipo que le permite a la terminal iniciar una llamada enviado solicitudes *SIP* hacia la red IP, ésta solo envía solicitudes *SIP*.

Los agentes de usuario pueden utilizarse sin que esté de por medio un servidor de red, cuentan con la funcionalidad completa de una terminal.

- **Agente de usuario servidor o User Agent Server (UAS):** Es la aplicación o equipo que permite a la terminal recibir y responder a una llamada iniciada por el UAC, las solicitudes que recibe son de tipo *SIP* y las posibles repuestas a éstas son de aceptación, rechazo o re direccionamiento de dicha solicitud.

En un mismo *software* o dispositivo se tienen tanto el agente de usuario cliente como el servidor (ver **Figura 3. 2**), el rol que desempeñe cada extremo dependerá de quien haya enviado la solicitud de invitación SIP. El que envía dicha solicitud pasa a ser Agente de usuario cliente y quien recibe la solicitud será Agente de usuario servidor. De igual forma cuando se envía el mensaje de BYE.

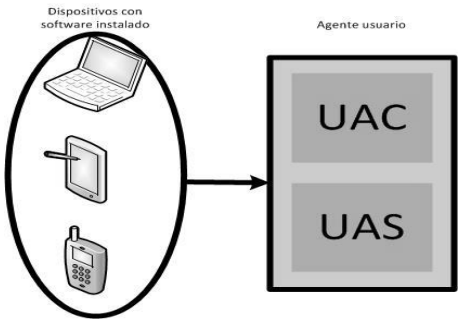


Figura 3. 2 Agente Usuario.

Servidores de red

Existen tres tipos de servidores de red.

- **Servidor proxy:** Estos servidores son los encargados de recibir las solicitudes del cliente, analizarlas y decidir a qué servidor reenviará las solicitudes, en dado caso puede modificar las solicitudes antes de enviarlas. Existen dos tipos de servidores proxy: Statefull Proxy y Stateless Proxy.

- **Proxy con estado (Statefull Proxy):** Mantiene el estado de las transacciones durante el procesamiento de las peticiones para la localización en paralelo de la llamada y así obtener una mejor respuesta para el usuario que solicito el establecimiento de la llamada
- **Proxy sin estado** (**Stateless Proxy):** Solamente reenvían los mensajes sin mantener el estado de las transacciones durante el procedimiento de las peticiones. Estos son muy simples pero funcionan con mayor rapidez que los proxy con estado.

- **Servidor de registro:** Cuando un usuario hace una petición *REGISTER* al servidor, esté último debe saber en qué punto está conectado el usuario y para ello se realiza un mecanismo de registro. Entonces cuando el usuario envía una petición *REGISTER* al servidor, el servidor se encarga registrar y asociar la dirección lógica (SIP-URL) a una IP (es caduca y no renovable)[19]. A esa asociación se le llama *binding* y es almacenada en una base de datos de localización.
- **Servidor de redirección:** El funcionamiento de este servidor consta de generar respuestas de re-direccionamiento a las peticiones que recibe, encamina las peticiones al siguiente servidor.

b) SIP URI

Las direcciones que utiliza *SIP* son del tipo *URI* que significa Uniform Resource Identifier, los *URI* contienen la suficiente información para iniciar y mantener una sesión de comunicación con el recurso. Su estructura es similar a la de una dirección de correo usuario@host, donde el usuario hace referencia al nombre o identificador o número telefónico. Por otro lado el host se refiere al dominio al cual pertenece el usuario o dirección de red.

Por ejemplo:

usuario@dominio, donde dominio es un nombre de dominio completo.
usuario@equipo, donde equipo es el nombre de la máquina.
usuario@dirección_ip, donde dirección_ip es la dirección IP del dispositivo.
número_teléfono@gateway, donde el *gateway* permite acceder al número de teléfono a través de la red telefónica pública

La forma general de un *SIP* URI es:

***SIP*:usuario:contraseña@direccion_del _equipo; parámetros_URI**

Dónde:

Usuario: Se refiere a un nombre, número o palabra que identifique al usuario, el cual nunca debe estar vacío, seguido de una contraseña, el signo @.
Contraseña: La contraseña está asociada al usuario, no es recomendable hacer uso de este campo debido al que el paso de información de autenticación en texto plano expone riegos de seguridad.
Dirección del equipo: Hace referencia a un nombre de dominio o bien a una dirección IP (IPv4 o IPv6) asociado a un dispositivo de red.
Puerto: Número de puerto donde la solicitud debe ser enviada.

c) Mensajes SIP

La comunicación que se establece por medio del protocolo SIP se basa en una serie de mensajes, los cuales pueden trasmitirse de forma independiente sobre la red. Se le llama mensaje a los datos enviados entre elementos SIP como parte del protocolo, tanto una solicitud y una respuesta son considerados mensajes. La comunicación en una red que utiliza SIP está dada por dos tipos de mensajes: Solicitudes-*Request* (métodos) y las repuestas-*Response* (códigos de estado), los cuales utilizan el formado genérico establecido en el RFC2822.

Estos mensajes pueden ser enviados ya sea sobre TCP o UDP, los cuales veremos más adelante. SIP es un protocolo basado en texto y por lo tanto utiliza el conjunto de caracteres ISO 10646 en codificación UTF-8 (RFC 2279). Los Mensajes de texto son muy similares como lo hacen los mensajes HTTP, donde los mensajes están formados por textos fáciles de entender, de fácil desarrollo y compatibles con aplicaciones HTTP. Las cabeceras especifican los detalles de la comunicación, como definir la parte que llama, la ruta y el tiempo de mensaje de la llamada.

El formato de un mensaje SIP se muestra en la **¡Error! No se encuentra el origen de la referencia.**, que incluye el método, el URI del destinatario y la versión SIP.

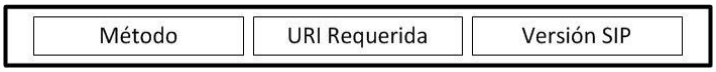


Figura 3. 3 Formato de mensaje SIP.

d) Métodos SIP

El protocolo *SIP*, maneja seis métodos básicos para que los clientes realicen sus peticiones y seis métodos extendidos que se muestran en la **Tabla 3.2**), dichos métodos son definidos en RFC 254. Las peticiones son identificadas por una línea inicial del mensaje llamada *Request-Line*, que contiene el nombre del método, el identificador del receptor y la versión de *SIP*.

Métodos	***Función***
INVITE:	Inicia una sesión entre *UAS*, un UA envía el método *SIP INVITE* y recibe una respuesta a dicha invitación por ejemplo (200 *OK*) donde el confirma con un *ACK*.
ACK	Confirma el establecimiento de una sesión (llamada)
OPTION	Solicita a una gente de usuario o servidor envíe sus capacidades como por ejemplo códecs soportados y tipo de mensajes.
BYE	Indica la finalización de una sesión (llamada), este mensaje puede ser emitido tanto como el que inicia o recibe la llamada.
CANCEL	Cancela una solicitud (llamada) enviada que aún no se ha confirmado
REGISTER	Registra a un agente de usuario entre la correspondencia de su dirección SIP y su dirección de contacto. Dejando un registro sobre la ubicación de usuario actual, tal como su dirección IP y el puerto por el cual se registró el mensaje.
Extensión de los métodos *SIP*	
INFO	Monitorea la llamada (RFC2976)
SUBSCRIBER	Mensajería instantánea
COMET	Precondiciones
PRACK	Reconocimiento provisional de las respuestas
NOTIFY	Mensajería instantánea
MESSAGE	Mensajería instantánea

Tabla 3.2 Funciones de los métodos utilizados por SIP.

e) Respuestas SIP (Código de estado)

Cuando el destinatario de la llamada recibe e interpreta los requerimientos *SIP*, esté contesta dependiendo cual sea el caso con las diferentes clases de repuestas *SIP*. El mensaje es similar al de la solicitud *SIP* en la línea inicial, llamada *Status-line*, que contiene la versión de *SIP*, el código de la repuesta (status-code) y una pequeña descripción (Reason-Phrase). El código de respuesta está formado por 3 dígitos, donde el primer digito identifica el tipo de repuesta. Se describen los códigos en la **Tabla 3.3**.

Código	**Función**
1xx	(Información): Proporciona información sobre si los requerimientos ya fueron recibidos o si está en proceso de tratamiento.
2xx	(Éxito): La recepción, entendimiento y aceptación de los requerimientos fueron exitosos.
3xx	(Re enrutamiento). Utilizada para redirigir una llamada, son enviados por los servidores proxy de una solicitud y no puede procesarla, envía una respuesta para re direccionar la llamada y entonces el usuario que está llamando pone una nueva ubicación para llamar al destino.
4xx	(Error requerimiento cliente): El o los requerimientos no pueden ser aceptados, los cuales tienen que ser modificados antes de ser reenviados al servidor.
5xx	(Error en el servidor): El servidor presenta problemas durante el procesamiento de un requerimiento que aparentaba ser válido.
6xx	(Fracaso global): El requerimiento no puede ser procesado por ningún servidor.

Tabla 3.3 Códigos de estado de llamadas SIP.

f) Cabeceras SIP

Los siguientes campos de cabecera deben de estar contenidos en los mensajes *SIP*, los cuales proporcionan la información necesaria a los agentes o equipos *SIP*, en la **Tabla 3.4** se describe la función de las cabeceras utilizadas por dicho protocolo.

Cabeceras	**Función.**
Via:	Indica el protocolo de transporte utilizado para la transmisión de la llamada e identifica la direcciona donde se debe enviarse el mensaje.
From:	Indica la dirección lógica o IP del solicitante para establecer una llamada.
To:	Indica la dirección lógica o IP del destinatario al cual se le envía la solicitud.
Call-Id:	Es un identificador único para cada llamada y contiene la dirección del host, sirve para agrupar mensajes que pertenecen a una misma llamada. Las solicitudes y repuestas en el proceso del establecimiento de la llamada debe de ser el mismo Call-ID.
Cseq:	Es un número aleatorio que inicia la secuencia de cada petición.
Contact:	Contiene una o varias direcciones que de ser necesario son utilizadas para contactar al usuario.
User Agent	Es el cliente agente que realiza la comunicación
Max-retime	Limita el número de saltos permitidos que puede hacer una solicitud hasta llegar a su destino, este número se disminuye cada vez que se realiza un salto.

Tabla 3.4 Cabeceras de los mensajes SIP.

g) Invitaciones SIP

Las invitaciones SIP son iniciadas por algún agente servidor dirigidos hacia un agente cliente.

Existen tres casos donde se tiene invitación de entre AU´s:

- Invitación SIP donde solo participan los AU´s
- Invitaciones y repuestas SIP utilizando un servidor de redirección
- Invitaciones y repuestas SIP utilizando un servidor de desvió

En el **Apéndice 1** se encuentra la explicación de los procesos antes mencionados.

3.3.1.1 SDP (Session Description Protocol)

El protocolo SDP (Protocolo de Descripción de Sesión), se describe en el RFC 4566. Es el encargado de describir los parámetros o características para iniciar una comunicación multimedia, cubre aspectos como anuncio de sesión, invitación a sesión y negociación de parámetros.

O sea realiza una negociación entre los usuarios que van a participar en una sesión, y establecer parámetros en común, a este conjunto de parámetros se le llama perfil de sesión

Un mensaje SDP se compone de varias líneas llamadas campos, los nombres de los campos son abreviados en una letra, los siguientes campos son obligatorios en un mensaje SDP.

3.3.1.2 Establecimiento de una llamada con señalización SIP

Para establecer una llamada *SIP* se necesitan de cuatro fases, descritas a continuación.

1. **Registro de usuarios:** Los dos primeros mensajes en color rojo pertenecen al proceso de registro de cada usuario. Es necesario que los usuarios se registren para que otros usuarios los puedan encontrar. El registro se inicia cuando las terminales envían una solicitud "*REGISTER*" en donde los campos *from* y *to* corresponden al usuario registrado, entonces el servidor *proxy* realiza una consulta para autenticar y enviar un mensaje de *OK* en caso de restarlo.
2. **Establecimiento de sesión:** EL usuario envía un mensaje *INVITE* al servidor *proxy* para solicitar se establezca una sesión. Posteriormente proxy envía un mensaje *TRYING 100* para detener las retransmisiones *INVITE* del usuario A al proxy. Además el proxy también envía un mensaje *INVITE* al usuario B, donde éste contesta con un *RINGING 180* al *proxy*, el cual retrasmite este mismo al usuario A y en este momento el teléfono comienza a sonar, y el mensaje *OK* 200 indica que el usuario B ha aceptado la llamada. Y contesta con un ACK.
3. **Transmisión de voz:** Los usuarios conversan y se transmite la voz por medio de RTP, los parámetros como direcciones, códecs, etc. son negociados por SDP.
4. **Finalización de sesión:** La finalización de la llamada termina cuando alguno de los dos usuarios (en este caso usuario A) envían un mensaje *BYE* al *proxy*, y éste lo reenvía al otro usuario (usuario B) contestando con un *ACK* que confirma de enterado al usuario.

La señalización de la explicación anterior se puede ver en la **Figura 3. 4**.

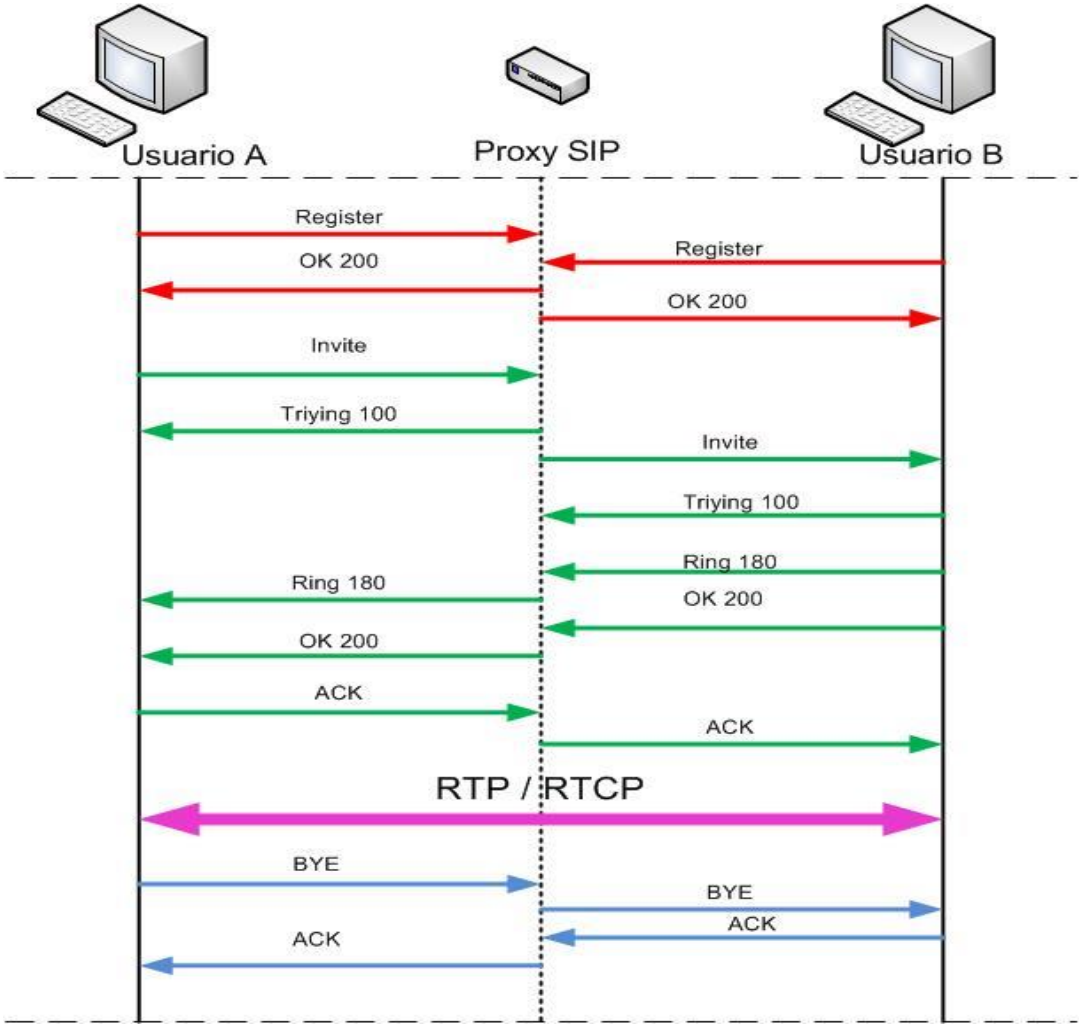


Figura 3. 4 Establecimiento de una llamada con señalización SIP.

3.3.2 H.323

La ITU-T fue el primer comité de estandarización quien creó el estándar de transferencia de tráfico multimedia sobre Internet. El acrónimo H.323 hace referencia a dicho estándar llamado "Sistemas de terminales de telefonía visual sobre redes de área local sin garantías de calidad de servicio" [21]. Estandarizado en Noviembre de 1996. Su gran aportación fue la del desarrolló de un grupo de protocolos de señalización. Dichos protocolos permitían controlar el establecimiento, mantenimiento y finalización de las conexiones multimedia ya sea para la transición de vídeo, voz o datos, haciendo uso de la red de datos, ya que los protocolos RTP y RTCP, para la transición de dichos medios ya se había sido desarrolladas por IETF.

Posteriormente surgieron nuevas versiones del estándar H.323 hasta llegar a la versión 7 que se puede ver [22] de nombre "Sistemas de comunicaciones multimedia basados en paquetes" aprobado en Diciembre del 2009. Para Marzo del 2013 se tiene la versión 7.1 [23], en la cual se anexa en mensaje de señalización denominado *FACILITY* para la transferencia de llamadas. El protocolo H.323 describe los componentes, protocolos, señalización, códecs, etc., que garanticen la comunicación y la interoperabilidad entre dispositivos, también define como se crea, se mantiene y se cierra una sesión entre dos terminales.

En un principio H.323 se creó para que funcionara sobre redes IP y redes de conmutación de paquetes, soportando conferencias de voz y vídeo multipunto, a pesar de que pocos usuarios hacen uso de esta capacidad de multipunto.

Los principales objetivos son:

- Incorporar algunas de las ventajas que ofrecen las redes de conmutación de paquetes para transportar datos en tiempo real.
- Resolver problemas respecto al envió de datos en tiempo real en redes de conmutación de paquetes.
- Hacer uso de entandares ya existentes tales como H.320, RRP y Q.931.
- Permitirle a las empresas fabricantes de equipos diseñar sus propias especificaciones al protocolo, y que puedan definir otras normas complementarias aportando nuevas características y capacidades.

El estándar cuenta con múltiples características, las cuales, como ya se mencionó no es necesario cumplir con todas, pero sí que sea compatible con sus equipos vecinos. Algunos equipos puede que solo cuenten con características contenidas en la primera versión del estándar y algunos otros con las más resientes versiones, algunas de las características son:

- Independencia de la red.
- Interoperabilidad entre equipos de diferentes fabricantes.
- Independencia de la plataforma y de la aplicación (*hardware* y *software*)
- Soporte para multi-conferencia
- Gestión del ancho de banda.
- Soporte para transmisión multicas
- Soporte para el establecimiento de conferencias entre distintas redes multimedia. H.323
- Seguridad (autenticación, integridad y privacidad)
- Establecer una llamada rápida (Fast Call)
- Avisos de requerimientos de calidad de servicio
- Procesos de Control basado en HTTP
- Resolución de dirección (en la versión 5)
- Robustez (ante errores sensillos de comunican)
- Monitoreo en la *QoS* (versión 5)
- Gestión de movilidad, se especifica en H.501, H.520 y H.530.
- Servicios extras. Otros servicios orientados a conferencias son
  - Transferencia de llamada

- Desvió de llamada
- Llamada On Hold (llamada en espera)
- Llamada en espera
- Conferencia sin consulta. Una llamada es contestada por alguien y posteriormente lo conecta con el número deseado.
- Identificar en número de la terminal llamante
- Control de planes de marcado
- Establecimiento de prioridades

3.3.2.1 Arquitectura H.323

La arquitectura general utilizada por H.323 se muestra en **Figura 3. 5**, en la parte central se encuentra la puerta de enlace (*gateway* H.323).

Los componentes principales de H.323 son:

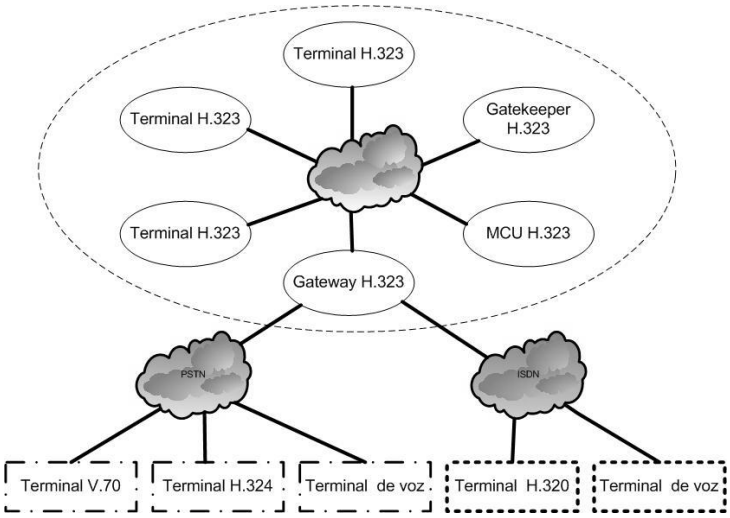


Figura 3. 5 Arquitectura de H.323.

Terminal H.323

Son extremos de la red que proporcionan una comunicación bidireccional en tiempo real, ya sea con otra terminal H.323, *gateway* o MCU, ya sea datos vídeo o datos.

Una terminal de este tipo puedo proporcionar voz, voz y datos, voz y vídeo, o voz, datos y vídeo.

Si se desea transmitir voz, las terminales como mínimo deben soportar la decodificación usada por las líneas telefónicas tradicionales, la codificación y decodificación usada por G.711 (PCM, 8Khz 64kbps). En cuanto a vídeo no es obligatorio pero en caso de serlo debe soportar como mínimo el códec H.261.

Una terminal H.323 está compuesta de varios elementos, tales como códecs de vídeo y voz, equipo telemático, aplicaciones de datos, la interfaces del equipo de usuario, las funciones de control del sistema y la interfaz hacia la red de paquetes.

- **Equipos de adquisición de información:** Conjunto de dispositivos como cámaras, monitores, micrófonos altavoces, aplicaciones e interfaces de usuarios.
- **Códecs de audio y vídeo:** Cada terminal debe tener un códec de audio para codificar y decodificar las señales de voz y poder transmitir y recibir información. En cuanto a códecs de vídeo es opcional en terminales de este tipo.
- **Canales de datos:** Es opcional contar con uno o más canales de datos ya sean unidireccionales o bidireccionales.
- **Unidad de control del sistema:** La unidad es la encargada de proporcionar la señalización para el adecuado funcionamiento de la terminal, dicha unidad está conformada por tres bloques.
  - Función de control H.245.
  - Función de señalización de llamada H.225.
  - Función de señalización RAS.
- **Interfaz de red de paquetes:** Se debe de cumplir con las recomendaciones de H.225, por lo que los servicios de extremo a extremo debe de ser fiable utilizando TCP, es obligatorio para el canal de control, canales de datos y canal de señalización de llamada.

Unidad de control multipunto (MCU)

Es un punto final encargado de conectar dos o más terminales para realizar multiconferencias entre puntos finales bajo el funcionamiento de H.323, el soporte que brinda hace referencia a que realiza la negociación entre terminales, que permite determinar las capacidades que tienen en común dichas terminales y así poder procesar el audio y vídeo.

MCU está formado por dos unidades lógicas: un controlador multipunto (MC) y uno o más procesadores multipunto (MP).

- **Controlar multipunto (MC: Multipoint Controller):** Su funcionamiento es gestionar las conexiones y realizar la negociación entre las terminales, con el fin de determinar las capacidades comunes entre todos los dispositivos que van a participar.
- **Procesador Multipunto (MP: Multipoint Contriller):** Su función es mezclar, conmutar y procesar todos los canales de audio, vídeo y datos, enviándolos a los usuarios de la multiconferencia.

Gateway

Su función se describe en apartado 2.5.2.2 Gateway VoIP.

Gatekeeper

Un *gatekeeper* tiene la función de gestionar una zona de control, dicha zona de control está formada por un conjunto de equipos registrados como lo son terminales, *gateways* y MCU. La comunicación entre los equipos y el *gatekeeper* se realiza por medio del protocolo RAS (Registro, Admisión, Situación)

El uso de un *gatekeeper* es necesario cuando deseamos tener compatibilidad entre una red H.323 y la telefonía tradicional. No es necesario el uso de un *gatekeeper* entre terminales H.323.

Este dispositivo puede ofrecer señalización indirecta en llamadas entre terminales H.323, registrar las llamadas que se relazan y mantener una lista de llamadas en espera.

El uso de estos dispositivos proporcionan las siguientes funciones a todos los elementos dentro de su zona de control.

- Realiza la conversión de números telefónicos a direcciones nativas de H.323, para cuando queremos comunicarnos con alguien que hace uso de la telefonía tradicional, la traducción universal es la asignación de números de extensión.
- Gestiona los elementos que pertenecen a la zona H.323 llevando a cabo el registro y admisión de los equipos su zona de control a través del protocolo H.225.0/RAS.
- Controla el ancho de banda disponible en la red H.323, donde el administrador limita el número de conexiones simultáneas con el fin de evitar congestiones en la red y perder *QoS* y rechaza las nuevas peticiones que se encentre por encima del número de conexiones limite.

Además el *gatekeeper* provee servicios de control de administración [24], donde el *gatekeeper* puede rechazar llamadas procedentes de una terminal sin autorización o *gateways* con acceso restringido o bien a cauda de determinadas franjas horarias.

Proxy H.323

Es un servidor que provee a los usuarios accesos a redes seguras confiando en la recomendación hecha por H.323, este proxy trabaja como dos puntos remotos que envían mensajes call- set up e información en tiempo real a su destino que se encuentran del otro lado.

- Control y gestión de ancho de banda: Tiene el control sobre el número de terminales H.323 que tienen acceso simultáneamente a la red o no tengan el suficiente ancho de banda.
- Gestión de la zona: Realiza el registro y admisión de la terminal y *gateways* de su zona, siempre sabe que *gateways* están involucrados en el encaminamiento hacia terminales RCC.

3.3.2.2 Pila de protocolos H.323

H.323 hace uso de otros entandares, normas y protocolos tales como H.323, H225.0, H.245, RAS y RTP/RTCP, a pesar de que H.323 trabaja en conjunto con muchos otros estándares y normas no es obligatorio cumplir con todas y cada uno de ellos, pues solo puede hacer uso de algunos cuantos según las necesidades requeridas. Es un estándar robusto pero a la vez flexible.

El protocolo H.323 utiliza los procedimientos de señalización de los canales lógicos contenidos en la norma H.245, la cual dice que cada uno de los canales se define cuando éste es abierto, dichos procedimientos indican cómo será la presentación entre el emisor y el receptos, como se establece, se mantiene y se finaliza la llamada, como se intercambia la información y como se codifica/decodifica la señal. Cuando se origina una llamada vía Internet los dispositivos finales llegan a un acuerdo en quien llevará el control para enviar mensajes especiales de control.

H.323 trabaja bajo ciertos protocolos que gestionan la preparación, establecimiento, control de estados, mensajería, códec de audio/vídeo, transferencia de datos y fin de llamada. Dichos protocolos funcionan sobre cualquier protocolo que se encuentre por debajo de la capa de transporte, pues se basa también en el modelo OSI. La pila de protocolos de H.323 se ven en la **Figura 3. 6**.

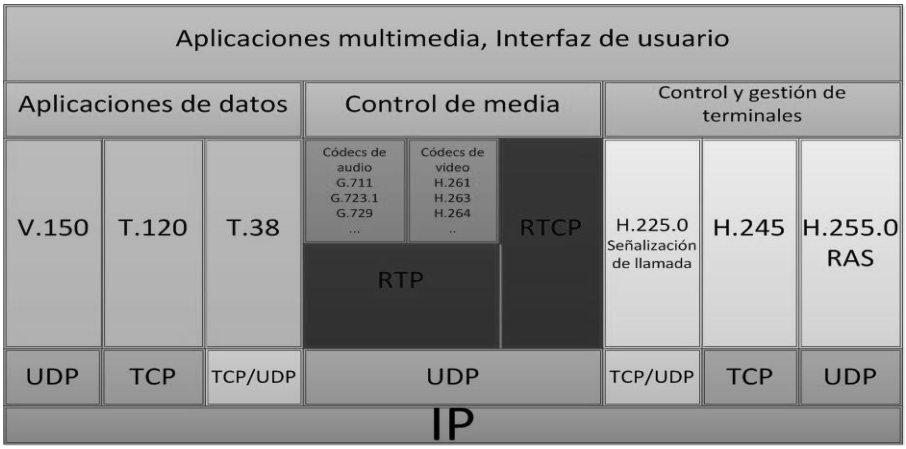


Figura 3. 6 Pila de protocolos de H.323.

En cuanto al transporte del flujo, puede utilizar TCP o UDP, si utiliza una comunicación no fiable como UDP entonces hace uso de RTP/RTCP, de lo contrario deja a cargo a TCP realice su trabajo.

H.323 trabaja sobre IP, por ultimo para el tratamiento que se le da al audio o vídeo se utilizan algoritmos de compresión y códecs, H.323 utiliza como códecs obligatorios básicos G.711 para audio y H.261 para vídeo, mientras que para conferencia T.120, comunicaciones de modem V.150 y para fax T.38.

Para definir la sintaxis de representar los datos de señalización y control, H.323 hace uso de ASN.1 (Abstract Syntax Notation 1) el cual se puede consultar en las especificaciones X.680-683, mientras que para realizar la conversión de esos datos a binario utiliza PER disponible en la X.691, de la ITU-T. El manejo del flujo de datos de audio y vídeo está a cargo de RTP y RTCP.

La comunicación que establece H.323 consta de una combinación de señales de audio, vídeo, datos, así como también de señalización y control. Es obligatorio que cada terminal cuente con señales de capacidad de audio, señales de llamada Q.931, control RAS, y señalización H.245.

La principal función de H.323 es controlar las llamadas, lo que implica la señalización para establecer la llamada, intercambio de capacidades, señalización de comandos e indicaciones, así como mensajes de apertura y descripción del contenido de los canales lógicos: Tales funciones las llevan a cabo los protocolos H.225.0, RAS, H.245 y RTP/RTCP, este último se explicara más adelante.

Algunos de los protocolos relevantes con los que trabaja H.323 se describen en el **Apéndice 2.**

3.3.2.3 Establecimiento de una llamada

Una llamada consta de cuatro fases donde están involucrados los protocolos RAS, H.245 y H.225.0. [26], la señalización de H.323 se muestra en la **Figura 3. 7**.

1. **Establecimiento**

La terminal A envía una solicitud de registro al *Gatekeeper* (identificándose con su alias) utilizando el protocolo RAS a través del mensajes ARQ, el GK aceptará enviándole a la terminal un mensaje de confirmación ACF o bien rechazará la solicitud ARJ.

Después se hace uso del protocolo H.225 (establece y libera una llamada) para enviar un mensaje SETUP para iniciar la llamada, el contenido de este mensaje son las direcciones IP y los puertos tanto de la terminal A (llamante) y B (llamado).

Posteriormente la terminal B contesta con un Call Proceeding para advertir el intento de llamada. Y se registra con el *Gatekeeper* mediante RAS como lo hizo la terminal A, y acepta la llamada la terminal B envía un Alerting que indica la generación de tono de llamada. Y por último Connect indica el comienzo de la conexión.

2. Señalización de control

Se establece un canal H.245 en una nueva conexión TCP. Iniciando un intercambio de mensajes para lograr la negociación de quien será maestro y quien esclavo, además de enviar las capacidades de cada participante. Los códecs de audio y vídeo a utilizar, esta conexión debe mantenerse mientras las terminales intercambien información permitiendo modificar parámetros.

Por último se abre el canal de comunicación tomando en cuenta las direcciones IP y los puertos.

Los mensajes que se envían son:

TerminalCapability (TCS). Mensaje que intercambia las capacidades soportadas por cada terminal.

OpenLogicalChannel (OLC). Mensaje que abre el canal lógico de información que contiene la descripción que permite la recepción y codificación de datos así como el tipo de datos que se trasmiten.

3. Transmisión de información multimedia

En esta fase inicia la comunicación directa entre terminales, transmitiendo el audio o vídeo (lo que se habla entre los usuarios de la terminales) a través de canales de información mediante el protocolos RTP/UDP/IP. Además se establecen canales de control a través de los protocolos RTCP/UDP/IP para retroalimentación.

4. Desconexión

Cuando alguno de los usuarios desea terminar la llamada cuelga y envía un mensaje CloseLogicalChannel y EndSessionComand de H.245 para cerras en canal H.245.

Después se cierra la conexión con un mensaje Release Complete de H.225

Y finalmente se liberan los registros y recursos contenidos en el *Gatekeeper* a través del protocolo RAS.

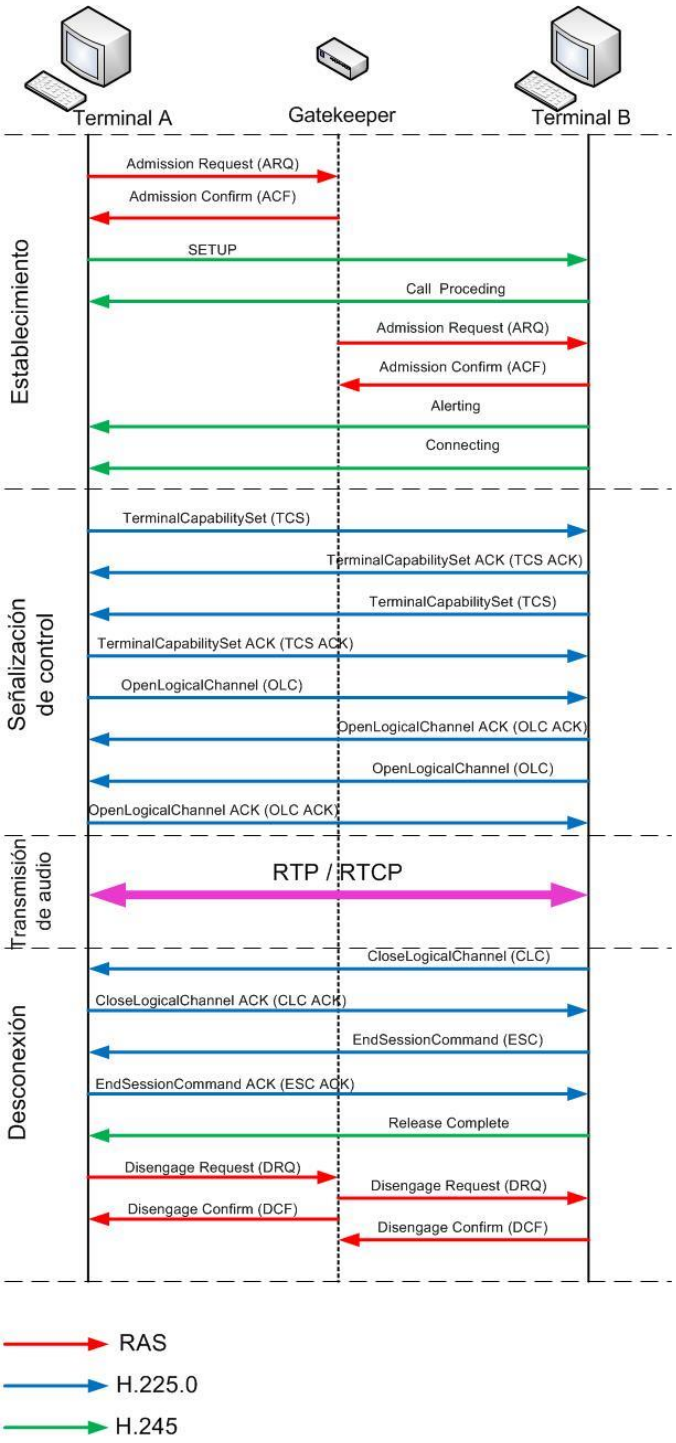


Figura 3. 7 Señalización de una llamada con H.323.

3.4 Protocolos de transporte

Los protocolos de transporte como bien lo dice su nombre trabajan en la capa de transporte del modelo OSI, pues bien se sabe que el protocolo IP se basa en dos protocolos de tráfico, UDP y TCP. Donde TCP brinda una fiabilidad mucho mayor que UDP en cuanto a conexión, pero UDP brinda mayor rapidez en cuando a la entrega de paquetes que para VoIP es de suma importancia al ser sensible al retraso de tiempo. Es por ello que se utiliza un protocolo llamado RTP, entonces la voz queda por encima de RTP y por encima de ambos queda IP, de la siguiente manera RTP/UDP/IP.

En esta capa llegan los datos provenientes de capas superiores, son divididos en partes pequeñas de ser necesario, pasándolos posteriormente a la capa de red asegurándose de que lleguen correctamente del otro lado de la red.

Los protocolos que trabajan en esta capa para implementación de VoIP son:

- RTP (Real-time Transport Protocol) Protocolo de transporte en tiempo real
- cRTP (Compress RTP)
- RTCP (Real-time Transport Control Protocol) Protocolo de control de transporte en tiempo real.

RTP trabaja con cualquier protocolo de señalización para VoIP, ya sea *SIP*, H.323, IAX o bien Skinny, todos estos necesitan un protocolo que permita transportar el audio o vídeo entre usuarios finales. No basta con enviarlos, también se necesita de la ayuda de otro protocolo llamado RTCP quien se encarga de monitorear la red, con el fin de brindar mayor calidad de servicio.

RTP y RTCP fueron originalmente definidos en el RFC 1889 [19] el cual fue remplazado por RFC 3550[28].

RTP y RTCP utilizan puertos diferentes, RTP utiliza puertos pares mientras que RTCP utiliza el siguiente puerto impar. Los puertos por defecto para RTP y RTCP son 5004 y 5005 respectivamente, pero también pueden hacer uso de los puertos comprendidos entre 1025 y 65535. Cuando una sesión RTP se abre al mismo tiempo se abre una sesión RTCP.

3.4.1 RTP (Real-Time Transport Protocol)

EL protocolo RTP (Real-time Transport Protocol), en español Protocolo de Transporte en Tiempo Real. El cual surge con la necesidad de controlar la gran demanda de recursos multimedia.

RTP permite:

- Identificar el tipo de información transportada.
- Indicar el instante en el que se pueden enviar paquetes, utilizando marcadores temporales los cuales también permiten sincronizar los flujos, medir retardos y la fluctuación.

- Incluir números de secuencia a la información que se transporta con el fin de detectar perdida de paquetes y entregar con bien los paquetes a la aplicación destino.

Este protocolo hace la entrega de voz, vídeo o ambos, estos datos son el resultado de la operación que llevan a cabo los códecs o sea las muestras que obtuvo, y posteriormente RTP los empaqueta para ser tratados por UDP.

En el payload de los paquetes se indica el tipo de códec que género las señales de voz o vídeo. Entonces el protocolo RTP identifica el payload y la secuencia de dichas señales generadas. Ahora el paquete RTP es encapsulado en una cabecera UDP, tratándose de datos en tiempo real. UDP también asigna una prioridad o secuencia pues se debe evitar los retardos o los paquetes perdidos.

RTP multiplexa diversos flujos de datos en tiempo real en un solo paquete UDP, capaz de enviar a uno o varios destinos (unicast y multicast). RTP no es capaz de reservar recursos o controlar la calidad de servicio, además de no garantizar la entrega de paquetes al receptor.

3.4.2 cRTP

El protocolo cRTP reduce 40 bytes del grupo de encabezados de IP/UDP /RTP a 2-5 bytes. La información contenida en UDP/IP/RTP en dos extremos que están interactuando, será la misma información durante todo la sesión, por q que cRTP quita esa información y reduce el tamaño de los paquetes enviados, por lo cual conserva ancho de banda.

Dicho protocolo es más eficiente en enlaces WAN con velocidades T1 o menores.

3.4.3 RTCP

Protocolo definido en los mismos RFC que RTP, RTCP es un protocolo de comunicación que brinda información de control sobre el flujo de datos para una aplicación multimedia. Su función es transmitir paquetes de control a los participantes de una sesión multimedia de *streaming*. (Ver u oír un archivo en una página web sin tener que descárgalo previamente). Dicho protocolo se basa en las transmisiones periódicas de paquetes de control y de calidad de servicio para los participantes en la sesión. RTCP trabaja junto con RTP para con el fin de monitorear la transmisión de datos RTP encapsulados, o sea que RTCP proporciona un respaldo sobre la de las transiciones de RTP. Con el fin de proveer información en tiempo real y así poder modificar parámetros, como por ejemplo modificar el códec a uno d menor calidad en caso de que la red se encuentre congestionada.

Varios paquetes RTCP pueden ser enviados en un mismo mensaje UDP.

3.5 CODECS

En toda comunicación multimedia se necesita hacer uso de códecs ya sea de audio o vídeo, así como de algunos formatos en caso de querer almacenar la información de audio o vídeo.

El proceso de convertir una señal analógica a digital está definido por varios estándares donde los procedimientos son complejos, basta con mencionar que la mayoría de ellos están basados en la modulación codificada mediante pulsos (PCM) o variaciones. En el apartado **2.2.1 Proceso de digitalización** se explica este proceso de convertir una señal analógica a digital.

Códec proviene del inglés coder-decoder (Codificador- Decodificador), ya sea *software* o *hardware* o una combinación de ambos. Es el encargado de convertir una señal de audio analógica a un formato de audio digital. Las señales se transmiten por algún medio y posteriormente cuando son recibidas por el usuario final las convierte a una señal analógica, y finalmente poder ser reproducido (audio/vídeo). Para que el emisor codifique la señal analógica y el receptor pueda decodificar la señal enviada por el emisor, es necesario que tanto emisor como receptor soporten el mismo códec.

En términos generales un códec es un algoritmo de compresión y descompresión de datos (vídeo y audio). Éste hace uso y aprovecha la potencia de un CPU ya sea para mejorar o degradar la calidad. Todos los algoritmos realizan una mezcla de tres elementos indispensables para su funcionamiento, estos elementos son [21] :

- Uso de recursos de CPU
- Consumo de ancho de banda
- Calidad de la comunicación

El algoritmo realiza una combinación de ellos en base a los resultados que se pretendan tener, si se hace uso de un algoritmos muy sofisticados, donde sea posible realizar una gran compresión y se tenga una buena calidad de comunicación, entonces estamos sacrificando los recursos del CPU, por lo tanto éste tiene que realizar grandes esfuerzos para lograr la compresión. Por otro lado si hacemos uso de un algoritmo sencillo donde el CPU no trabaje demasiado y se conserve la calidad de comunicación, entonces lo que se sacrifica es el gran uso de ancho de banda.

La mayoría de algoritmos (códecs) provocan perdidas de información con el fin de obtener una archivo más pequeño, se debe tener en cuenta que si los datos tendrán otro tratamiento posterior se debe asegurar que no existan demasiadas perdidas en el proceso de codificación

Lo ideal es lograr un equilibrio entre estos tres elementos, muchos de estos códecs son capaces de mantener ese equilibrio, lamentablemente la mayoría de ellos se encuentran bajo licencias muy costosas, ya dependerá de cada quien invertir en ellos para mantener una alto nivel de comunicación.

Existe también una amplia variedad de códecs, como ya se dijo existen de acuerdo a la calidad de voz, al ancho debanda que necesitan o al consumo de CPU para codificar/decodificar. Aunque también los podemos clasificar en base su uso libre o bajo licencia, o bien de acuerdo a la institución y/o empresa que los estandarizo, a continuación se mencionarán algunos códecs estandarizados por la ITU-T se muestran la **Tabla 3.5**, para mayor información consultar [22].

Información del códec				**Cálculos de ancho de banda**					
Velocidad de bits (kbps)	**Tamaño del codec**	**Intervalo del codec**	**Mean Opinion Score (MOS)**	**Tamaño de la carga útil de voz (bytes)**	**Tamaño de la carga útil de voz (ms)**	**Paquetes por segundo (PPS)**	**Ancho de banda MP (kbps)**	**Ancho de banda c/cRTP MP (kbps)**	**Ancho de banda Ethernet (kbps)**
G.711 (64kbps)	80 bytes	10 ms	4.1	160 bytes	20 ms	50	82.8 kbps	67.6 kbps	87.2 kbps
G.729 (8 kbps)	10 bytes	10 ms	3.92	20 bytes	20 ms	50	26.8 kbps	11.6 kbps	32.3 kbps
G.723.1 (6.3 kbps)	24 bytes	30 ms	3.9	24 bytes	30 ms	33.3	18.9 kbps	8.8 kbps	21.9 kbps
G.723.1 (5.3 kbps)	20 bytes	30 ms	3.8	20 bytes	30 ms	33.3	17.9 kbps	7.7 kbps	20.8 kbps
G.726 (32 kbps)	20 bytes	5 ms	3.85	80 bytes	20 ms	50	42.8 kbps	27.6 kbps	47.3 kbps
G.726 (16 kbps)	15 bytes	5 ms		60 bytes	20 ms	50	42.8 kbps	27.6 kbps	47.2 kbps
G.228 (16 kbps)	10 bytes	5 ms	3.61	60 bytes	30 ms	33.3	28.5 kbps	18.4 kbps	31.5 kbps
G722_64k (64 kbps)	80 bytes	10 ms	4.13	160 bytes	20 ms	50	82.8 kbps	67.6 kbps	87.2 kbps

Tabla 3.5 Características de los códecs.

Otros códecs y desarrollador.

- GSM
- ETSi|
- iLBC –
- LPC10 – Gobierno USA
- Speex
- EVRC- 3GPP2
- SILK- Skype

3.6 Conclusiones

Se presentaron los protocolos de señalización que se utilizaron en este trabajo (SIP y H.323), así como el conjunto de protocolos con los que trabaja cada uno de ellos. También se mostraron los mensajes, métodos, y direcciones con las que SIP y H.323 funcionan para la señalización con el fin de establecer llamadas.

Además se estudiaron los pasos que sigue cada protocolo para iniciar, mantener y finalizar una llamada.

Capítulo 4. Descripción del hardware y software utilizado

4.1 Introducción

En este capítulo se describe los elementos de la red por la cual viajará voz, para lo cual es indispensable montar una red WAN con equipo CISCO, así como realizar sus respectivas configuraciones que se adapten a los requerimientos necesarios.

Por otro lado, también se explica el software que se utiliza en el desarrollo de esta tesis, así como el uso y funciones que tendrá dentro de nuestra red.

4.2 Hardware

EL equipo utilizado para interconectar los elementos de la red es de la marca CISCO que se encuentran disponibles dentro del laboratorio de redes en el edificio Valdez Vallejo.

4.2.1 Routers CISCO 2811

El modelo de los *router*s CISCO 2800, cuenta con dos puertos FastEthernet (10/100) para conexión de la red local y una tarjeta WIC-2T (Tarjeta de interfaz WAN de puerto serial dual) para conexiones WAN ver **Figura 4. 1**. Que cuentan con dos puertos seriales que soportan protocolos de interconexión Frame Relay y PPP, el equipo terminal de datos (DTE) o de un equipo de comunicación de datos (DCE). Además soporta enlaces T1/E1 para interconectar *router*, los enlaces permiten brincar servicios de voz, vídeo y datos de forma segura. Cada puerto utiliza un conector Smart Serial, con un soporte asíncrono que alcanza una velocidad máxima de 8Mbps, y de forma asíncrona de hasta 115.2kps por puerto [29].

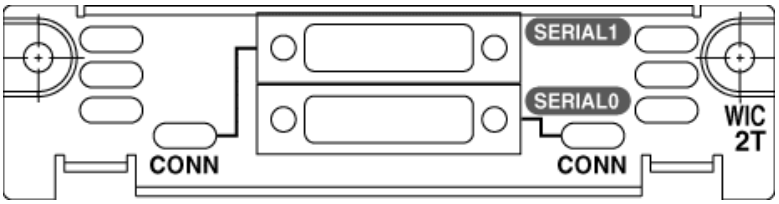


Figura 4. 1 Conector de tarjetas WIC.

Los puertos seriales necesitan cables Smart Seriales. Además cuenta con un slost VIC (Voice Interface Card) que incluye 24 puertos telefónicos (FXS, FXO o E&M), se instalan de forma adicional.

4.2.2 Switches CISCO 2960

Los *switches* que se utilizaron cuentan 24 puertos Ethernet 10/100 Base-TX. Soportan DHCP, listas de control de acceso (ACL´s), VLAN, *QoS* y seguridad en la LAN. Ofrecen un alto rendimientos, gran escalabilidad y administración de la LAN.

4.2.3 Equipos terminales

Se utilizaron dos lap tops como clientes y una PC como servidor Asterisk, las características son las siguientes.

Lap top 1

Toshiba Satellite
SO. Ubuntu 12.04 (precise)
Kernel 3.5.0-54-generic
4 CPUs Intel (R) Core (TM) i5-3210M CPU 2.50GHz.

Lap top 2

ACER Aspire V3
SO. Ubuntu 12.04 (precise)
Kernel 3.5.0-54-generic
Intel(R) Core(TM) i5-3210M CPU @ 2.50GHz

PC

SO. Ubuntu 12.04 (precise)
Kernel 3.5.0-54-generic
2 CPU`s Intel (R) Pentium (R) D CPU 2.66GHz.

4.3 *Software*

El *software* utilizado fue seleccionado con el fin de proporcionar bajo costo al implementar una red, utilizando *software* libre, principalmente bajo la distribución de Linux que nos permite mayor flexibilidad y compatibilidad en el sistema.

- Asterisk
- SIPp
- Open H.323
- Wireshark
- Jperf
- Zoiper
- Ubuntu 12.04

4.3.1 ASTERISK

Asterisk es un proyecto que fue lanzado en 1999 bajo licencia de código abierto escrito en lenguaje C por Mark Spencer. Quien pensó en una alterativa para ahorrar dinero utilizando *software* libre en vez de sistemas operativos propietarios, así que decidió ofrecer servicio a usuarios Linux. Ya que no se contaba aún con un apoyo comercial para éste, así que creo una empresa llamada Servicios de soporte para Linux (LSS), brindaba asistencia técnica telefónica para los profesores de TI (Tecnologías de la Información), los cuales llamaban para obtener ayuda sobre Linux. Así pasaron algunos meses hasta que Mark se dio cuenta que la pequeña oficina ya no era suficiente y los expertos en Linux no se daban abasto. Decidió implementar un sistema telefónico, pero las cotizaciones de algunas empresas superaban el presupuesto que Mark podía tener.

Por lo tanto Mark tomo una decisión importante para las telecomunicaciones, decidió crear su propio sistema telefónico. Código por código su proyecto fue creciendo, creando el núcleo de Asterisk, meses después el código fuente lo público en Internet bajo la licencia GPL de Linux.

Asterisk comenzó a atraer las miradas en varias partes del mundo, los cuales también comenzaron a contribuir en la presentación de nuevas características y funciones al sistema.

Para el 2001 el Servicio de Apoyo a Linux cambio el nombre a Digium, Digium Asterisk sigue desarrollando junto con la colaboración de la comunidad.

Actualmente Asterisk sigue creciendo y desarrollando nuevas funcionalidades, con el fin de satisfacer las necesidades de los usuarios. Cuenta con un soporte más amplio así como herramientas que comunicación, además ofrece recursos que brinda un panorama sobre la transición que vive Asterisk hoy en día. Logotipo de Asterisk **Figura 4. 2**.

Figura 4. 2 Logo de Asterisk.

4.3.1.1 ¿Qué es Asterisk?

Asterisk es un *framework* de código abierto para la creación de aplicaciones de comunicaciones. Una computadora se convierte en un servidor de comunicaciones cuando instalamos Asterisk, porque brinda un sistema IP PBX, puertas de enlace para VoIP, funciones de conferencia además de soluciones personalizadas.

Esté fue diseñado para trabajar sobre Linux pero actualmente existe la portabilidad de trabajar con Mac y Windows sin mucho éxito y sin soporte. Es capaz de conectar y trabajar con diversos protocolos de telefonía.

Asterisk permite construir aplicaciones, soluciones y protocolos en comunicaciones de tiempo real, pues abstrae la complejidad de los protocolos y tecnologías de las comunicaciones. Lo que facilita al usuario concentrarse solo en crear productos y soluciones innovadoras.

Con Asterisk también se pueden construir aplicaciones para sistemas de telefonía empresarial (PBX), distribuidores de llamadas, puerta de enlace VoIP y puentes para establecer vídeoconferencia. Es un sistema hibrido.

Asterisk se compone de una gran variedad de aplicaciones las más comunes son:

- *Gateway* VoIP
- Correo de voz del servidor
- Puente de conferencia
- Call Center
- IVR servidor
- Negocio sistema de teléfono IP PBX

Asterisk es considerado un PBX de código abierto, en sus inicios solo se trataba de un sistema telefónico para pequeñas empresas. Asterisk es más que un simple PBX es capaz de realizar mucho más funciones que las de un PBX, como lo son: *gateway* VoIP, call center, puente de conferencias. Y como servidor de correo de voz entre otras aplicaciones que necesiten comunicación en tiempo real, lo que lo hace una herramienta universal para las comunicaciones.

En pocas palabras Asterisk es una aplicación servidor de comunicaciones, lo que es el servidor web Apache es a las aplicaciones web. Así como Apache se encarga de los detalles de bajo nivel del envío y recepción de los datos, de lo misma forma lo hace Asterisk haciendo uso de diferentes protocolos de comunicaciones.

Para Apache se desarrollan aplicaciones web construidas en base a paginas HTML, hojas de estilo CSS, imágenes, bases de datos, servicios web, imágenes etc. En Asterisk las aplicaciones de comunicación se construyen por medio de guiones *Dialplan* (*scripts*), grabaciones de audio, configuraciones de audio, protocolos etc. Con Apache para que una página web funcione el servidor web debe estar conectado a Internet y en caso de una aplicación de comunicación es necesario que el servidor esté conectado a los servicios de comunicación (PSTN o VoIP).

Para tener acceso a una página web se necesita que previamente el desarrollador registre un nombre de dominio y configurar el o los DNS que correspondan a su página. Mientras que, con las aplicaciones de comunicación, se necesita que las personas que deseen tener acceso al sistema cuenten con un número telefónico o un URI de voz para que las terminales tipo clientes se conecten a él, con el propósito de establecer llamadas y trasmitir voz y vídeo en tiempo real, utilizando cualquier protocolo y códecs disponibles en Asterisk.

Reconoce varios protocolos de VoIP como IAX, MGCP. H.323 y *SIP*, además de que puede interoperar con terminales IP pues es capaz de registrar y actuar como *gateway* entre

4.3.1.2 Arquitectura Asterisk

Asterisk está integrado por varios componentes que al relacionarse crean un sistema complejo (ver **Figura 4. 3**), entender sus componentes es necesario antes de iniciarse a usarlo, pues nos brinda un panorama fácil de su funcionamiento, además de cómo se relaciona con los componentes fuera de él.

Algunos conceptos se deben tener claros antes de sumergirse en el uso de Asterisk, los cuales se describen a continuación.

Núcleo

El principal componente es el núcleo, que interactúan con módulos y proporciona una gran infraestructura, carga los diferentes módulos y componentes a utilizar, lee y ejecuta los archivos de configuración incluyendo el *dialplan*, relaciona componentes fuera y dentro de éste.

Módulos

Los módulos de Asterisk complementan su funcionamiento, la mayoría de los módulos son independientemente configurables y cuentan con sus propios archivos de configuración. Los módulos son archivos con una extensión .so que por lo regular se encuentran en el directorio de módulos (/usr/lib/asterisk/modules). El formato de los archivos es pbx_xxxxx.so

Llamadas y canales

Para Asterisk una llamada hace referencia a uno o más canales contenidos en Asterisk

Canales

Los canales son creados por Asterisk haciendo uso de *drivers*, estos canales pueden estar puenteados a otros canales, también puedes hacer uso de otros recursos como módulos o librerías externas.

Dialplan

El *dialplan* es el plan de marcado, principal método que establece el comportamiento de Asterisk, usa aplicaciones y funciones que afectan los canales, configuración y características de Asterisk.

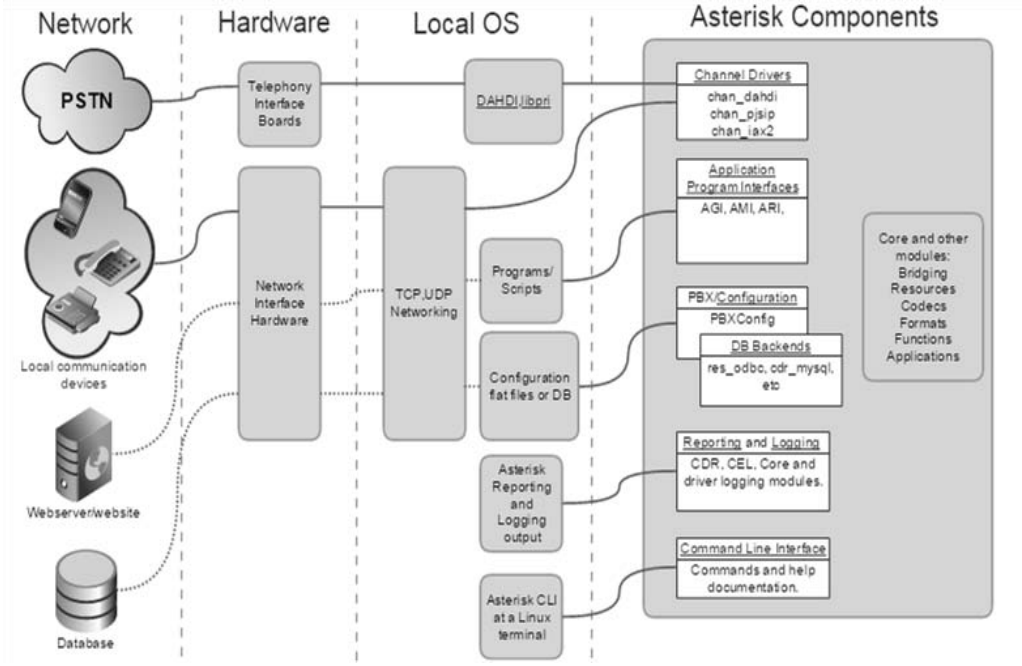


Figura 4. 3 Sistema Asterisk [30]**.**

Módulos de Asterisk

Asterisk cuenta con una arquitectura modular lo que hace que sea flexible y casi ilimitado, el administrador de Asterisk puede elegir que módulos cargar de acuerdo a sus requerimientos, no es necesarios hacer uso de todos los módulos, los cuales se verán más adelante.

Drivers del canal.

Estos comunican a los dispositivos externos de Asterisk y envían la señalización o protocolo a usar al núcleo. Todas las llamadas provenientes fuera de Asterisk pasan por un conductor del canal antes de llegar al núcleo y las llamadas salientes pasan por un conductor del canal con dirección a un dispositivo externo.

Aplicaciones *dialplan*

Su función es realizar llamadas al sistema, donde una aplicación puede contestar la llamada, reproducir un mensaje de sonido, colgar la llamada, o bien realizar más complejas como correo de voz, o poner en modo de espera.

Funciones diaplan

Se utilizan para recuperar, fijar o modificar ajustes sobre la llamada, por ejemplo determinar el identificador de una llamada saliente.

CODEC

Modulo para la codificación o decodificación de audio y vídeo, los archivos tienen un nombre con el siguiente formato códec_xxxxx.so. Asterisk utiliza los módulos de códecs para enviar y recibir datos y para realizará conversiones entre formatos.

Los módulos de códec con los que cuenta Asterisk por default son:

ADPCM, 32kbit/s
G.711 A-law, 64kbit/s
G.711 μ-law, 64kbit/s
G.722, 64kbit/s
G.726, 32kbit/s
GSM, 13kbit/s
LPC-10, 2.4kbit/s

Si se desea utilizar otros códecs es necesario descargarlos e instalarlos, o bien existen algunos módulos que se pueden modificar.

Archivos de configuración Asterisk

Asterisk cuenta con archivos de configuración específicos, los cuales se configurar dependiendo de las funciones que queremos que realice en nuestra red.

/etc/asterisk/

Contiene los archivos de configuración Asterisk, aquí se almacenan y se leen tales archivos, con extensión .conf, .lua y ael.

Módulos Asterisk: Contiene todos los módulos como aplicaciones, códecs, formatos y canales que utiliza Asterisk.

/usr/lib/asterisk/modules/

Librerías: Contiene una base de datos y subdirectorios.

/var/lib/Asterisk

Directorio de base de datos: Aquí se almacenan los datos para la base de datos interna de Asterisk, llamada astdb.sqlite3

/var/lib/Asterisk

Llaves de encriptación: Al configurar el cifrado basado en clave, Asterisk buscará en el subdirectorio llaves de este lugar para las claves necesarias.

`/var/lib/asterisk`

Directorio AGI (Asterisk Gateway interface): Asterisk busca en este directorio los scripts por default para las aplicaciones AGI

/var/lib/asterisk/agi-bin

Directorio de cola: En este directorio se encuentran los archivos de cola de diversos núcleos y módulos proporcionados por Asterisk.

/var/spool/Asterisk

Directorio de procesos corriendo: Mientras se está ejecutando Asterisk se crean dos archivos en este directorio asterisk.ctl y asterisk.pid, el primero se refiere a la toma del control de Asterisk y el segundo al archivo que contiene el PID (identificador de proceso) de los procesos.

/var/run/Asterisk

Registro de salida: En este directorio se almacenan la configuración para proporcionar una salida de archivo de registro.

/var/log/asterisk

4.3.2 SIPp

SIPp (ver **Figura 4. 4**) es una herramienta generadora de tráfico *SIP* de *software* libre, dicha herramienta cuenta con escenarios básicos SIPStone de usuario agente y cliente *UAC* y *UAS*, capaz de iniciar, establecer, comunicar y terminar múltiples llamadas haciendo uso de los métodos *INVITE* y *BYE*.

Utiliza archivos XML predefinidos o personalizados, que son los escenarios que describen desde muy simples a complejos flujos de llamadas. En donde se define el flujo que se llevara a cabo durante la llamada, dicho flujo se define en base a los métodos, mensajes y códigos utilizados por *SIP*. Cuenta con la visualización dinámica de las estadísticas sobre la ejecución de pruebas (velocidad, retardo de ida y vuelta, y las estadísticas mensaje de llamada), periódicos CSV estadísticas vertederos, TCP y UDP sobre tomas múltiples o multiplexado con la gestión de retransmisión y dinámicamente ajustables tarifas de llamadas.

Otras características avanzadas incluyen soporte de IPv6 , TLS , SCTP SIP autenticación , escenarios condicionales , retransmisiones UDP, robustez error (llamado tiempo de espera, la defensa de protocolo), etc. SIPp también puede enviar el tráfico de voz o vídeo (RTP) a través de RTP eco y RTP/pcap repetición. Aunque están optimizadas para el tráfico, el estrés y las pruebas de rendimiento, SIPp se puede utilizar para ejecutar una sola llamada y salida, corroborando si la llamada se establece o no.

Un escenario SIPp es un archivo XML que contiene los pasos a seguir de una o más llamadas que se desea ejecutar, utilizando los métodos propios de SIP y etiquetas de XML. El escenario se ejecuta de manera secuencial y muestra en pantalla las etapas por las que pasan las llamadas. Adicionalmente se muestra estado en el que se encuentren si se terminaron o no con éxito, SIPp tomo como error cualquier circunstancia que no suceda como se especificó.

SIPp no es capaz de enviar audio como tal, pues al solo manejar dos protocolos *SIP* y RTP no puede capturar y reproducir el audio. La única forma de que SIPp envié audio es reproduciendo una captura de paquetes registrados de un flujo RTP, la captura de paquetes RTP se logra con Wireshark, el procedimiento se verá en el siguiente capítulo.

Figura 4. 4 Logo de SIPp.

SIPp a la hora de ejecutar sus escenarios cuenta con varios parámetros, algunos necesarios para correr el escenario y otros opcionales para obtener información sobre el curso de las llamadas, errores en la ejecución o bien detalles sobre el establecimiento de las mismas.

Para ejecutar un escenario se debe ingresar a la carpeta donde se tienen todos los escenarios *SIP* y los archivos pcap.

Una vez dentro el formato para correr un escenario es el siguiente,

./sipp –sn nombre_archivo.xml –s ip_servidor -otros_parametros

Ejemplo: ejecutar SIPp en 7 llamadas cada 2 segundos (3,5 llamadas por segundo)

./sipp -sn uac -r 2000 127.0.0.1 7 –rp

Parámetros para correr un escenario SIPp

-trace_stat: Guarda todas las estadísticas en un archivo con formato “nombre_escenario_pid.csv entre la información contenida:

- Hora y fecha de inicio
- Tiempo transcurrido
- Tasa de llamadas
- Número de llamadas entrantes y salientes.
- Número de llamadas en curso, finalizadas, iniciadas y fallidas
- Número de mensajes recibidos, contestados y procesados.

-**rtp_echo:** Reenvía el tráfico que le llega por el mismo puerto definido con –mp.
-trace_err: Traza todos los mensajes inesperados en <nombre de archivo de Escenario> _ <pid> _errors.log.
-**error_file:** Establecer el nombre del archivo de registro de errores.
-**error_overwrite:** Sobrescribir el archivo de registro de Errores (Por Defecto cierto).
-sf: Ejecuta un escenario creado por el usuario.
-sn: Ejecuta un escenario de default dentro de *SIP* tales como *UAC*, *UAS*, *regexp, branch, branchs* entre otros.
-d: Define la duración de la llamada o mejor dicho el tiempo que realiza una pausa antes de colgar una llamada por default es o y la unidad es en milisegundos
-s: Establezca el nombre de usuario parte de la solicitud URI. El valor predeterminado es "servicio".
-r: Para especificar el tipo de referencia en el número de llamadas por segundo
-rp: Para especificar la " **r** con periodo "en milisegundos para la tarifa de llamadas (por defecto es 1000 ms / 1 seg). Esto le permite tener n llamadas cada m milisegundos (utilizan

4.3.3 Callgen 323

Es una aplicación del proyecto Open H.323, su función es generar llamadas con diferentes parámetros del protocolo H.323.

Las aplicaciones de Open H.323 se basan en librerías de PWLib y OpenH323 por lo tanto también Callgen 323 se basa en ellas. La forma de interactuar con la aplicación es por medio de comandos de entrada y salida, las llamadas generadas son configurables a las necesidades del usuario.

Callgen está escrita en C++ y es multiplataforma.

Los parámetros que pueden configurarse son los siguientes.

Opciones de ejecución

-1 <Segundo>: Tiempo de espera máximo para realizar llamadas. Especifica el máximo tiempo intentando llamar a la otra persona y el valor por defecto es de 60 segundos.

-2 <Segundos>: Después de que las llamadas se conectan (o tiempo de espera), espera este tiempo y si no se contesta se empezarán a colgar todas las líneas. Sólo significativo para la persona que llama Callgen 323, y el valor predeterminado es de 2 segundos.

-3 <Segundos>: Espere a que todas las llamadas está desconectado o que haya transcurrido el tiempo de espera. Cuando el tiempo de espera transcurrido, el Callgen 323 continuará a la siguiente fase. Solamente es significativo para la persona que llama, y el valor predeterminado es de 60 segundos.

-4 <Segundos>: Espera hasta antes de comenzar la próxima secuencia / lote.

-g <host>: Especifica manualmente el host / dirección del *gatekeeper*.

-i <IP addr>: Específica la interfaz IP a la que se unirá el Callgen 323 STI oyente. Iniciar en modo pasivo. Establezca esta opción para el receptor / destinatario Callgen 323.

-n <número>: Número de llamadas / líneas. Para recibir Callgen 323, Especifica el número máximo de llamadas que puede recibir simultáneamente. Esto específica el número de llamadas simultaneas que Callgen 323 debe hacer.

-N: No requiere registrarse con el *gatekeeper*. Significa que si el registro *gatekeeper* falla, el Callgen 323 todavía continuará

La ejecución de las ITS.

-r <número>: Número de lotes para repetir. Un lote es una secuencia de hacer llamadas y colgar llamadas.

-u <nombre de usuario>: Nombre o Id de usuario

4.3.4 Wireshark

Wireshark es una herramienta multiplataforma utilizada para realizar análisis sobre paquetes de red, cuenta con una interfaz amigable al usuario.

Es capaz de trabajar con una gran variedad de protocolos y permite examinar datos de una red activa o bien de archivos capturados y almacenados previamente. Contiene una amplio lenguaje para filtrar la información que queremos ver.

Wireshark es de *software* libre y se puede ejecutar en la mayoría de los sistemas operativos.

4.3.5 Iperf

Iperf es una herramienta útil para realizar pruebas en redes de datos, tales como medir el rendimiento de una red bajo los protocolos de transporte UDP o TCP.

Iperf fue desarrollado el Distributed Applications Support Team (DAST) en el National Laboratory for Applied Network Research (NLANR) y está escrito en C++, es de código abierto y se ejecuta en plataformas como Linux, Unix y Windows.

La herramienta permite ajustar parámetros que el usuario necesita para sus pruebas o bien optimizar los resultados. Es capaz de funcionar como cliente o servidor, para realizar las pruebas siempre debe de existir un cliente y un servidor.

El comportamiento bajo UDP y TCP.

UDP: Al trabajar bajo UDP permite al usuario especificar el tamaño de los datagramas a enviar, proporcionando resultados de rendimiento y de paquetes perdidos.

TCP: Al trabajar con TCP se mide el rendimiento de la carga útil entre el cliente y el servidor.

Iperf cuanta con una amplia variedad de parámetros para medir el rendimiento de la red, ver más detalles en [31].

4.3.6 Zoiper

Zoiper es un *Softphone*, un *softphone* es un programa de computadora que emula un teléfono y nos permite realizar llamadas desde nuestro computador a otros que contengan un *software* igual o similar o hacia teléfonos de red fija o celulares en cualquier país del mundo. Utiliza los recursos de su computador (procesador, memoria, parlantes y micrófono) para hacerlo.

Zoiper es una marca registrada (ver **Figura 4. 5**) y es uno de los mejores en su tipo, por su fácil manejo, poco peso y excelente calidad técnica.

Existe la versión para PC y también está disponible en Play Store para Smartphone.

Soporta los protocolos *SIP* y IAX para la parte de señalización.

Figura 4. 5 Logotipo de Zoiper[32]**.**

4.4 Conclusiones

Se presentaron los elementos en *software* y *hardware* que en conjunto conforman nuestra red WAN, dichos elementos en *hardware* se tienen *rauters*, *switches*, enlaces WAN, equipos terminales (lap top´s, PC´s). Por la parte del software es de *Open Source*, que van desde una mini central telefónica como lo es Asterisk, *softphone* como Zopiper, generadores de tráfico para SIP y H.323 como lo es SIPp y Callgen 323, y por ultimo herramientas de medición como Wireshark, Iperf y medidor de rendimiento incluido en el sistema operativo Ubuntu.

Capítulo 5. Configuraciones y mediciones

5.1 Introducción

En este capítulo se dan a conocer todas las configuraciones realizadas para montar nuestra red WAN tanto en *software* como en *hardware,* el direccionamiento utilizado para nuestra red. Así como el tipo y ejemplos de los parámetros que se pretende medir con este estudio. Se realiza la configuración de Asterisk, Zoiper, Wireshark, kSar, SIPp y Callgen 323. Se muestran las configuraciones en los archivos sip y extensions de Asterisk, además se hace la grabación del audio de una llamada telefónica para posteriormente añadirlo a nuestros escenarios SIP.

5.2 Topología de la red

La red WAN es implementada en una topología delta, donde cada *router* conecta a una red LAN, en la red LAN3 se encuentra PC con Asterisk. También se encuentra un cliente en cada una de las otras 2 redes LAN, cada cliente cuenta con dos aplicaciones generadoras de tráfico SIPp y Callgen 323. Los enlaces WAN tienen un ancho de banda equivalente a un enlace E2 a 8.448 Mbps. Las configuraciones de direccionamiento se diseñaron tenido en cuenta una holgura para fututo crecimiento. La topología utilizada se observa en la

Figura 5. 1.

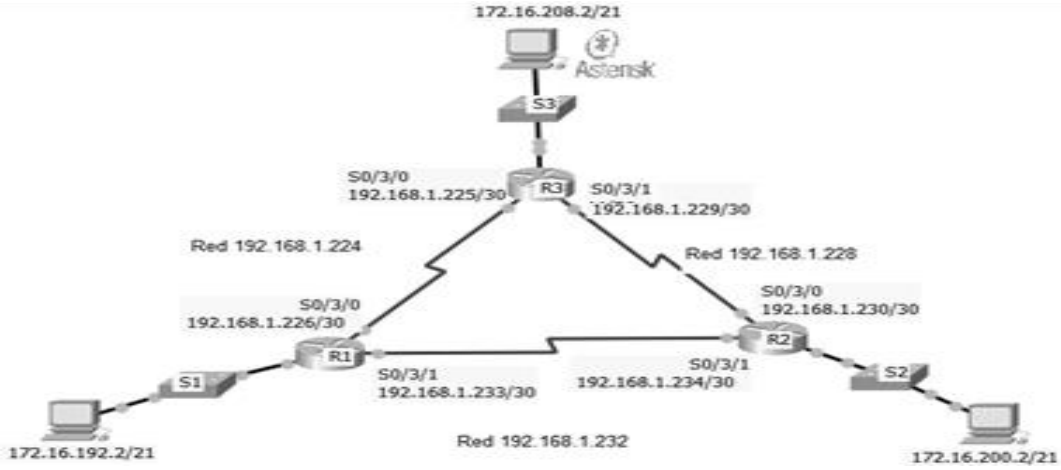


Figura 5. 1 Topología de la red.

5.3 Direccionamiento de la red

Se utilizó la red de clase B 172.16.192.0 para realizar el direccionamiento apropiado de las redes LAN con direccionamiento hasta 1000 usuarios por red. Y para el direccionamiento de las redes WAN se utilizó la red de clase C 192.168.1.0. Para ambas redes se realizaron los procedimientos de subneteo apropiados.

EL direccionamiento para nuestra red se puede observar en la **Tabla 5.1**, donde se R1, R2 y R3, corresponde a los routers respectivamente. También se muestra el direccionamiento de las computadoras utilizadas.

Equipo	Interfaz	IP	Mascara
R1	F0/0	172.16.192.1	255.255.255.248
	S0/3/1	192.168.1.233	255.255.255.252
	S0/3/0	192.168.1.226	255.255.255.252
R2	F0/0	172.16.200.1/21	255.255.255.248
	S0/3/1	192.168.1.234	255.255.255.252
	S0/1/0	192.168.1.230	255.255.255.252
R3	F0/0	172.16.208.1/21	255.255.255.248
	S0/3/0	192.168.1.229	255.255.255.252
	S0/3/1	192.168.1.225	255.255.255.252
PC1	Fast Ethernet	172.16.192.2	255.255.255.248
PC2	Fast Ethernet	172.16.200.2	255.255.255.248
PC3	Fast Ethernet	172.16.208.2	255.255.255.248

Tabla 5.1 Direccionamiento.

5.4 Pruebas de rendimiento sin tráfico

Con la ayuda de la herramienta Iperf se verifica el ancho de banda máximo que puede soportar nuestra red, los enlaces WAN soportan 8Mbps, el ancho de banda máximo que se puede tener sin descartar la utilización real.

En las siguientes imágenes se puede observan un servidor (**Figura 5. 2**) y un cliente (**Figura 5. 3**), en los cuales se intentó trasmitir 9Mbps, arrojando como máximo 8.13Mbps de *System Throughput.*

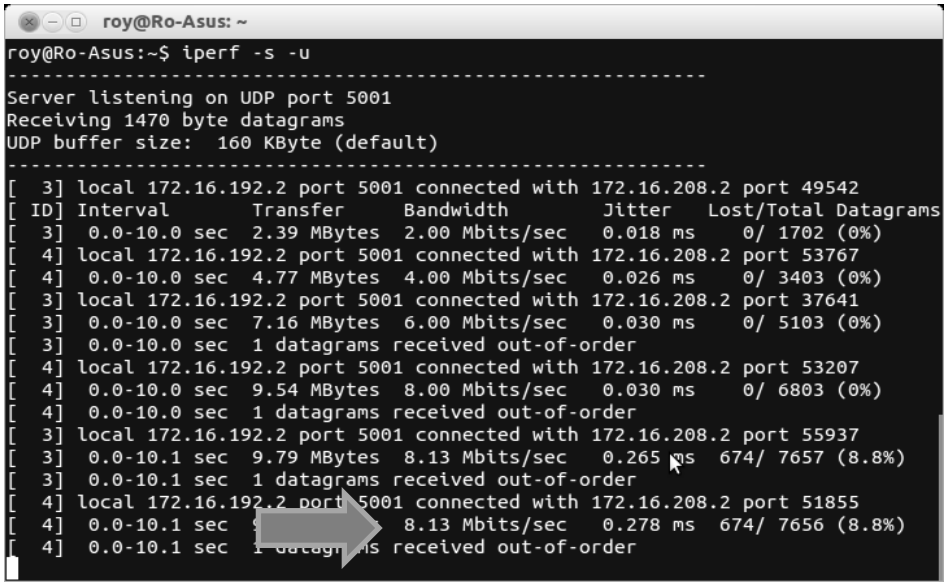


Figura 5. 2 Iperf como servidor.

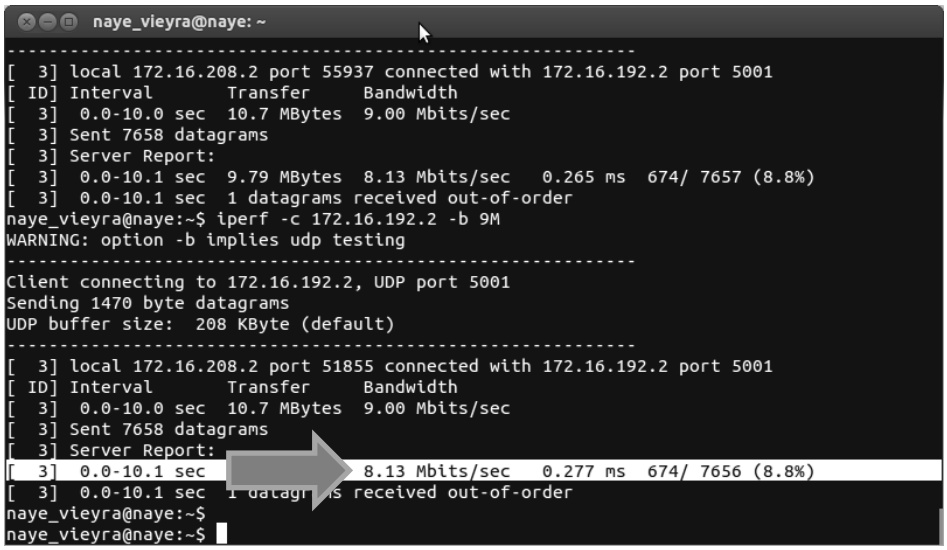


Figura 5. 3 Iperf como cliente.

También se realizó una medición del rendimiento del procesador donde se encuentra instalado Asterisk para tener como referencia las mediciones antes de poner en marcha las llamadas. El procesador del servidor se encuentra a un 20% de su capacidad como se muestra en la **Figura 5. 4**, corriendo las aplicaciones Asterisk y Wireshark.

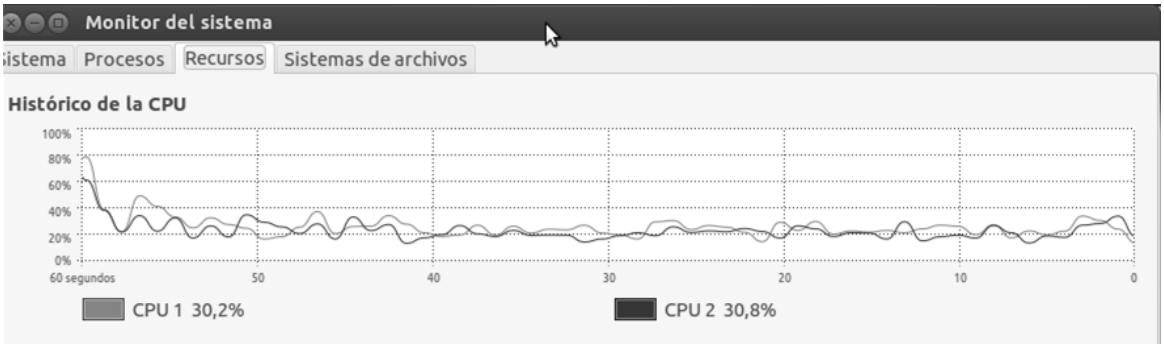


Figura 5. 4 Rendimiento del procesador en Asterisk.

5.5 Configuración de Asterisk

En el capítulo anterior se explicó la arquitectura y funcionamiento de Asterisk. En este apartado se explicará la instalación y configuración de los archivos correspondientes para que Asterisk conteste las llamadas provenientes de los clientes.

La instalación de Asterisk se explica en el **Apéndice 4.** Pasos para la instalación de Asterisk**.**

5.5.1 Configuración de los archivos

Se realiza la configuración de los archivos sip y extensions de Asterisk, para cada usuario se creó una configuración.

Primero se configuraron dos usuarios 80 y 81, tales usuarios funcionaran con una aplicación llamada Zoiper instalado en un Smartphone y en una laptop, para establecer una llamada de prueba entre ellos y grabar la conversación con el fin de añadirle archivo de audio el escenario que utilizaremos para generar llamadas bajo el protocolo SIPp.

Posteriormente se crean las cuentas 1001 y 1002 para cada cliente, donde la terminal en la LAN 1 le corresponde la cuenta 1001 y a la terminal en la LAN 2 le corresponde la cuenta 1002.

Además de crear un *dial plan* para cada una de las cuentas creadas como se muestra en las configuraciones de cada uno de los archivos mostrados a continuación.

Archivo extesnsions.conf

La configuración realizada del plan de marcación para los clientes 1001 y 1002 en el archivo extensions.conf.

```
;########################################
; dialplan
; configuracion del dialplan para los anexos SIP
; servidor asterisk >naye<
;
[general]
;Configuración para usuarios Zoiper
[internal]
exten => 80,1,Answer
exten => 80,1,Dial(SIP/80)
exten => 80,n,Playback(hello-world)
exten => 80,n,Wait(113)
exten => 80,n,Hangup()
;
exten => 81,1,Dial(SIP/81)
exten => 81,n,Answer
exten => 81,n,Playback(hello-world)
exten => 81,n,Hangup
;
; dial plan para usuarios de SIPp
[sipp]
: dial plan para el usuario 1001
exten => 1001,1,Answer
exten => 1001,n,Playback(hello-world)
exten => 1001,n,Wait(110)
exten => 1001,n,Hangup
; dial plan para el usuario 1002
```

```
exten =>1002,1,Answer
exten =>1002,n,Playback(hello-world)
exten =>1002,n,Wait(110)
exten =>1002,n,Hangup
```

Archivo sip.conf

Configuración para los clientes SIP

```
;#########################################
; sip.conf
; configuracion de los clientes sip
; servidor asterisk
;
;#########################################
[general]
;
port=5060
disallow=all
allow=g726
allow=ulaw
allow=alaw
;
;=========================================

; 80 y 81 fueron configuradas para una llamada entre dos terminales
;con Zoiper
[80]
type=friend
host=dynamic
language=es
context=internal
secret=80
username=80
callerid=80
dtmfmode=rfc2833
qualify=yes
;
[81]
type=friend
host=dynamic
language=es
context=internal
secret=81
username=81
callerid=81
dtmfmode=rfc2833
qualify=yes
;

; Configuración para las llamadas con SIPp
[sipp]
type=friend
```

```
context=sipp
host=dynamic
port=6000
user=sipp
canreinvite=no
disallow=all
allow=alaw
allow=ulaw
srvlookup=yes
qualify=in
nat=yes
```

5.6 Extracción de audio con Wireshark para crear un archivo PCAP

Como bien se mencionó en el capítulo anterior, Wireshark nos permite hacer una captura de mensajes de flujo RTP, para el propósito del proyecto, se realiza una captura de audio de una llamada de voz IP utilizando dos terminales con Zoiper instalado. El tiempo de la llamada es de 1.30 minutos y la grabación se guarda en una archivo PCAP.

Las configuraciones pertinentes para Asterisk, que permite que la llamada se establezca se encuentran en el apartado 5.5.1 Configuración de los archivos y la configuración de Zoiper se encuentra en el apartado 6.2 Llamada de prueba con SIP.

Se inicia primero Wireshark para la captura de mensajes pues se corre el riesgo de no capturar mensajes si se inicia después de iniciar la comunicación entre los *softphone*. Con el *softphone* Zoiper ya instalado en las terminales se inicia la marcación para establecer la llamada, después de 1.30 minutos la llamada se cuelga y se debe detener Wireshark. Una vez con todos los paquetes capturados por Wireshark se aplica un filtro para obtener solo los paquetes RTP, este filtro actúa solo sobre la IP fuente y la IP destino y los paquetes RTP.

El formato de filtro es el siguiente:

ip.src == IP_addresse_of_A y ip.dst == IP_addresse_of_B y RTP

ip.src == 192.168.181.34 and ip.dst == 192.168.181.32 and rtp.

Verificamos en el campo *Payload Type* o tipo de carga que se trata de ITU-T G.711 PCMA (8), es decir usa el códec audio G.711, estandarizado por ITU, la frecuencia de muestreo 8KHz, como se muestra en la **Figura 5. 5**.

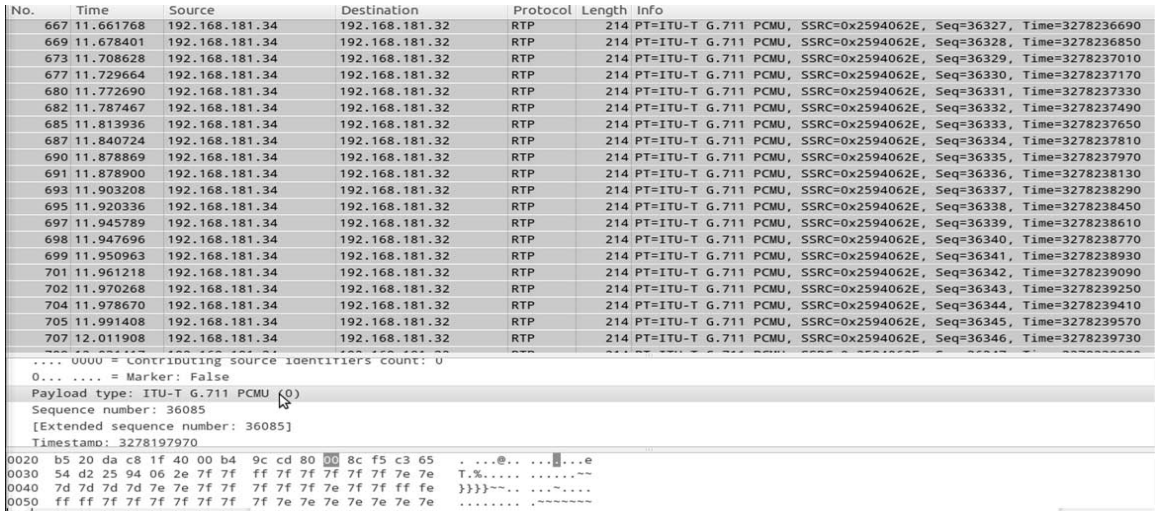


Figura 5. 5 Captura de flujo RTP.

En este apartado solo nos interesa capturar el audio de la conversación por lo cual se hace casó omiso de las direcciones IP que se observan en las imágenes, pues se hace una prueba a través de una conexión inalámbrica. Solo con el fin de capturar el audio y guardarlo como archivo pcap.

En la siguiente **Figura 5. 6** se tiene un análisis gráfico de la conversación establecida entre los dos usuarios. Para visualizar esta ventana no dirigimos a la pestaña *Telephony->*VoIP *Calls*, nos aparece una nueva ventana nos colocamos en el segundo *item* y pulsamos *Graph*.

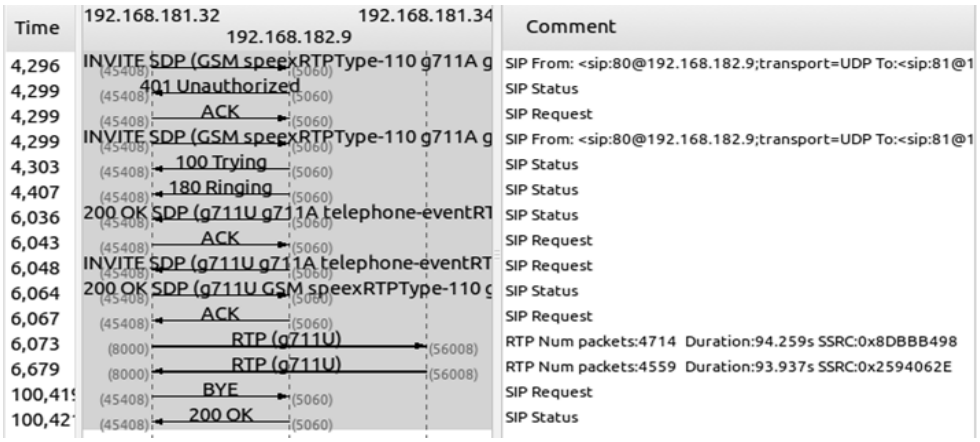


Figura 5. 6 Señalización.

Se observa la grabación, donde el primer espectro es el que inicia la llamada y el segundo quien contesta la llamada. Para ello nos dirigimos a VoIP *Calls* y pulsamos Player en el *item* 1 y pulsamos ahora Decode y obtenemos el audio.

Se observa que sea crean dos canales para la llamada, un canal de ida y otro de regreso en la **Figura 5. 7**.

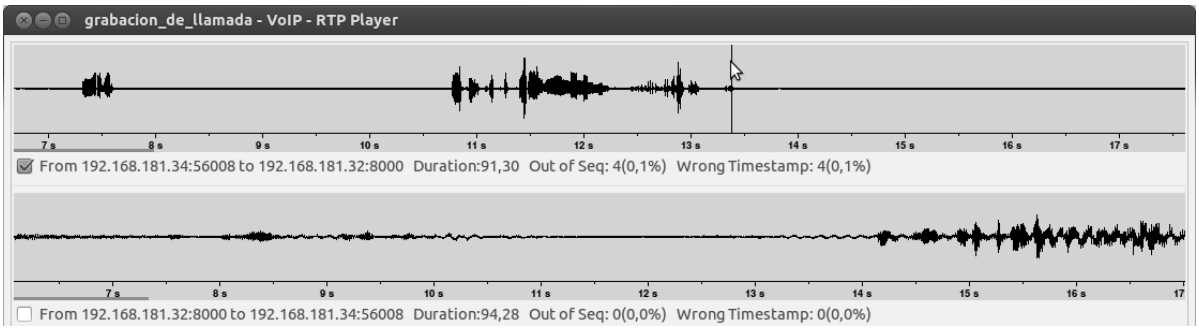


Figura 5. 7 Espectro del flujo RTP.

Guardamos con la opción visualizada para registrar los paquetes filtrados anteriormente, el archivo se guarda con la extensión .pcap en la carpeta de PCAP dentro SIPp, al cual de daremos el nombre de "grabación_de_llamada.pcap".

Para poder incluirlo en nuestro escenario SIP tenemos que cambiar de propietario y de grupo a nuestro archivo .pcap pues debe tener el mismo propietario y grupo que el resto de los archivos .pcap utilizados por SIPp. Para realizar el cambio utilizamos el comando "Chown" desde Ubuntu. Una vez realizado los pasos anteriores podemos reproducir el archivo en un escenario XML.

5.7 Configuraciones SIPp

Se instaló SIPp en su versión 3.3 en las dos laptops que serán los clientes ubicados en la LAN1 y LAN2 respectivamente.

Es necesaria la instalación previa de las siguientes librerías.

- Compilador C++
- Librería curses o ncurses
- Soporte para TSL. Librería OpenSSL 0.9.8
- Soporte para reproducir archivos PCAP: librerías libcap y libnet
- Soporte para SCTP: librería lksctp-tools

Para consultar una fácil instalación de las librerías consultar [33]. Para la consultar los pasos de la instalación de SIPp ver **Apéndice 4.** Pasos para la instalación de SIPp. La descarga e instalación de SIPp se hizo en el directorio /home/nayeli/Descargas/ como lo muestra la **Figura 5. 8**.

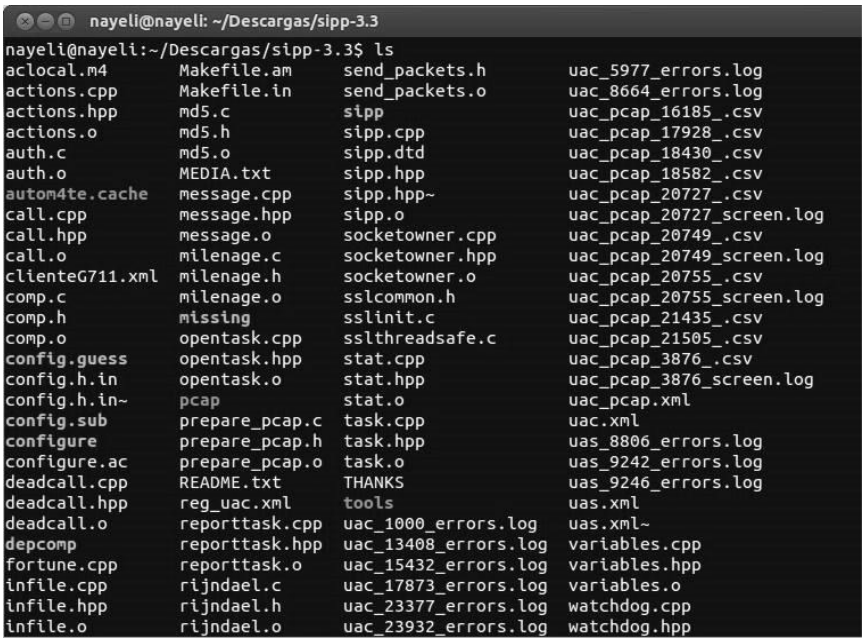


Figura 5. 8 Contenido de carpeta sipp-3.3.

Allí debemos guardar los escenarios creados, además se encuentra la carpeta de pcap donde están los audios predefinidos o bien los creados como es el caso, los escenarios buscan en dicha carpeta los audios para enviarlos en la ejecución del escenario.

Ya instalado SIPp realizaremos modificaciones a un escenario al cual incluiremos el archivo **grabación_de_llamada.pcap** previamente creado.

Nuestro escenario uac_pcap.xml es cargado en ambas laptops que son nuestros clientes,

Se corre el mismo escenario para ambos clientes la diferencia radica en que tiene un número de cliente diferente, las estadísticas obtenidas se observan en el siguiente capítulo.

Corrida del escenario del cliente 1 con cuenta 1001 en la LAN 1 que genera 86 flujos de llamadas se observa en la **Figura 5. 9**.

Figura 5. 9 Ejecución del escenario para la cuenta 1001.

Corrida del escenario del cliente 2 con cuenta 1002 en la LAN 2 que genera 86 flujos de llamadas como se observa en **Figura 5. 10**.

Figura 5. 10 Ejecución del escenario para la cuenta 1002.

El escenario **uac_pcap.xml** utilizado está escrito en lenguaje xml, las palabras reservadas, los comandos, los atributos y la sintaxis para crear estos archivos que trabajan bajo las especificaciones del protocolo SIP, se pueden ver en [34]. Este archivo genera la señalización SIP para iniciar llamadas telefónicas. Utiliza los métodos y mensajes correspondientes al protocolo SIP.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE scenario SYSTEM "sipp.dtd">
<!-- This program is free software; you can redistribute it and/or -->
<!-- modify it under the terms of the GNU General Public License as -->
<!-- published by the Free Software Foundation; either version 2 of the -->
<!-- License, or (at your option) any later version. -->
<!-- -->
<!-- This program is distributed in the hope that it will be useful, -->
<!-- but WITHOUT ANY WARRANTY; without even the implied warranty of -->
<!-- MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the -->
<!-- GNU General Public License for more details. -->
<!-- -->
<!-- -->
<!-- Sipp 'uac' scenario with pcap (rtp) play -->
<!-- -->
<scenario name="UAC with media">
<!-- In client mode (sipp placing calls), the Call-ID MUST be -->
<!-- generated by sipp. To do so, use [call_id] keyword. -->
<send retrans="500">
<![CDATA[
INVITE sip:[service]@[remote_ip]:[remote_port] SIP/2.0
Via: SIP/2.0/[transport] [local_ip]:[local_port];branch=[branch]
From: sipp <sip:sipp@[local_ip]:[local_port]>;tag=[pid]SIPpTag09[call_number]
To: [service] <sip:[service]@[remote_ip]:[remote_port]>
Call-ID: [call_id]
CSeq: 1 INVITE
Contact: sip:sipp@[local_ip]:[local_port]
Max-Forwards: 70
Subject: Performance Test
Content-Type: application/sdp
Content-Length: [len]
v=0
o=user1 53655765 2353687637 IN IP[local_ip_type] [local_ip]
s=-
c=IN IP[local_ip_type] [local_ip]
t=0 0
m=audio [auto_media_port] RTP/AVP 8 101
a=rtpmap:8 PCMA/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-11,16
]]>
```

```
</send>
<recv response="100" optional="true">
</recv>
<recv response="180" optional="true">
</recv>
<!-- By adding rrs="true" (Record Route Sets), the route sets -->
<!-- are saved and used for following messages sent. Useful to test -->
<!-- against stateful SIP proxies/B2BUAs. -->
<recv response="200" rtd="true" crlf="true">
</recv>
<!-- Packet lost can be simulated in any send/recv message by -->
<!-- by adding the 'lost = "10"'. Value can be [1-100] percent. -->
<send>
<![CDATA[
ACK sip:[service]@[remote_ip]:[remote_port] SIP/2.0
Via: SIP/2.0/[transport] [local_ip]:[local_port];branch=[branch]
From: sipp <sip:sipp@[local_ip]:[local_port]>;tag=[pid]SIPpTag09[call_number]
To: [service] <sip:[service]@[remote_ip]:[remote_port]>[peer_tag_param]
Call-ID: [call_id]
CSeq: 1 ACK
Contact: sip:sipp@[local_ip]:[local_port]
Max-Forwards: 70
Subject: Performance Test
Content-Length: 0
]]>
</send>
<!-- Play a pre-recorded PCAP file (RTP stream) -->
<nop>
<action>
<exec play_pcap_audio="pcap/llamada_prueba.pcap"/>
</action>
</nop>
<!-- Pause 8 seconds, which is approximately the duration of the -->
<!-- PCAP file -->
<pause milliseconds="100000"/>

<!--<exec rtp_stream = "5000" />-->
<!-- Play an out of band DTMF '1' -->
<nop>
<action>
<exec play_pcap_audio="pcap/dtmf_2833_1.pcap"/>
</action>
</nop>
<pause milliseconds="10000"/>
<!-- The 'crlf' option inserts a blank line in the statistics report. -->
<send retrans="500">

<![CDATA[
BYE sip:[service]@[remote_ip]:[remote_port] SIP/2.0
Via: SIP/2.0/[transport] [local_ip]:[local_port];branch=[branch]
From: sipp <sip:sipp@[local_ip]:[local_port]>;tag=[pid]SIPpTag09[call_number]
```

```
To: [service] <sip:[service]@[remote_ip]:[remote_port]>[peer_tag_param]
Call-ID: [call_id]
CSeq: 2 BYE
Contact: sip:sipp@[local_ip]:[local_port]
Max-Forwards: 70
Subject: Performance Test
Content-Length: 0
]]>

</send>
<!--<recv response= BYE crlf="true">
</recv>   MIO -->
<recv response="200" crlf="true">
</recv>
<!-- definition of the response time repartition table (unit is ms) -->
<ResponseTimeRepartition value="10, 20, 30, 40, 50, 100, 150, 200"/>
<!-- definition of the call length repartition table (unit is ms) -->
<CallLengthRepartition value="10, 50, 100, 500, 1000, 5000, 10000"/>
</scenario>
```

5.8 Instalación de Callgen 323 323

Esta aplicación genera tráfico de señalización para llamadas telefónicas en base al protocolo H.323, hace uso de librerías como PWLib y Open 323.

Además permite:

- Recepción de un número exacto de las llamadas.

- Ajustar el retardo entre cada lote de llamadas.

- Establecer el número de lotes a repetir.

-La única función es compatible G.711 ulaw y 64k de usuario

Para la instalación de Callgen necesitamos instalar las librerías PWLib y Open323, para ello consultar.

Instalación

Descargamos de la siguiente página web.

http://iweb.dl.sourceforge.net/project/opalvoip/v3.10%20Luyten/Stable%206/opal-3.10.6.tar.bz2

```
$ cd ~
$ tar -xzf Callgen 323323.tar.gz
```

Compilación

```
$ cd Callgen 323323
```

```
$ make
```

En circunstancias normales, es necesario ejecutar dos instancias de Callgen 323, uno para recibir las llamadas (pasivo), y otro para hacer las llamadas (activo). Para ambos casos, por lo general, tiene que especificar el mismo número de llamadas que desea recibir / hacer con la opción -n.

Especifica los nombres de usuario para asignar a la Callgen 323.

Se realizan 86 llamadas en cada cliente, las llamadas fueron generadas por Callgen 323.

Los comandos utilizados fueron los siguientes

Correr en cliente 1

Callgen 323 -n 86 172.16.208.2 -N -l

Correr en cliente 2

Callgen 323 -n 87 172.16.208.2 –u

5.9 Conclusiones

Se realizó el direccionamiento IP apropiado todos los elementos de la red WAN y se configuraron los equipos CISCO, así como también se hicieron pruebas de comunicación y rendimiento de la red sin ningún tipo de tráfico.

Además se instalaron y configuraron todas las hermanitas en software que se utilizó en esta tesis.

Capítulo 6. Resultados

6.1 Introducción

Este capítulo presenta los resultados obtenidos de las llamadas generadas con SIPp y Callgen 323, los datos obtenidos sobre Jitter, paquetes trasmitidos, latencia y el número de llamadas exitosas y fallitas, de cada protocolo.

Además de la conclusiones finales.

6.2 Llamada de prueba con SIP

Se realizó una prueba con un *softphone* para verificar cuantos canales se abren con una llamada en curso. Se utilizó Zoiper como *softphone* el cual se debe instalar y configurar en los equipos terminales, en cada uno de ellos se configura una cuenta SIP que incluyen el nombre de la cuenta, nombre del usuario, clave y la IP de Asterisk como se ve en la **Figura 6. 1**. Dichos parámetros deben ser configurados en el archivo sip.conf de Asterisk.

6.2.1 Configuración de las cuentas

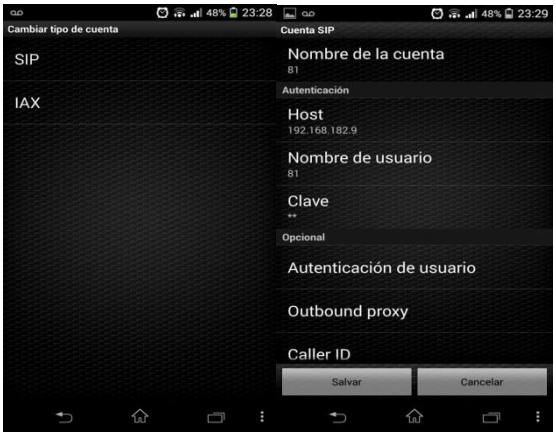


Figura 6. 1 Configuración de Zoiper en un celular.

En la **Figura 6.2** se observa que la llamada se estableció, el número 81 es el usuario fuente y el 80 el destino mostrando su URI.

Figura 6. 2 Establecimiento de la llamada entre softphone.

6.2.2 Archivos de configuración en Asterisk

Las cuentas configuradas en los *softphone* se deben configurar en los archivos sip y extensions de Asterisk para poder reprocesar la llamada.

Las configuraciones de los archivos extensions.conf y sip.conf son los mismos que se utilizaron para realizar una llamada y guardar el audio en un archivo pcap, que se realizó en el apartado 5.6.

extensions.conf

```
[internal]
exten => 80,1,Answer
exten => 80,1,Dial(SIP/80)
exten => 80,n,Playback(hello-world)
exten => 80,n,Wait(113)
exten => 80,n,Hangup()
;
exten => 81,1,Dial(SIP/81,15)   ; m es para cambiar
el tono de llamado
exten => 81,n,Answer
exten => 80,n,Playback(hello-world)
exten => 81,n,Hangup
```

sip.conf

```
[80]
type=friend
host=dynamic
language=es
context=internal
secret=80
username=80
callerid=80
```

```
dtmfmode=rfc2833
qualify=yes
;
[81]
type=friend
host=dynamic
language=es
context=internal
secret=81
username=81
callerid=81
dtmfmode=rfc2833
qualify=yes
;
```

En la **Figura 6. 3** se muestra la consola de Asterisk donde se está procesando la llamada entre un usuario 80 y 81, se ven las aplicaciones del *dial plan* como reproduce un audio, espera 110 segundos. Se abren dos canales para la llamada.

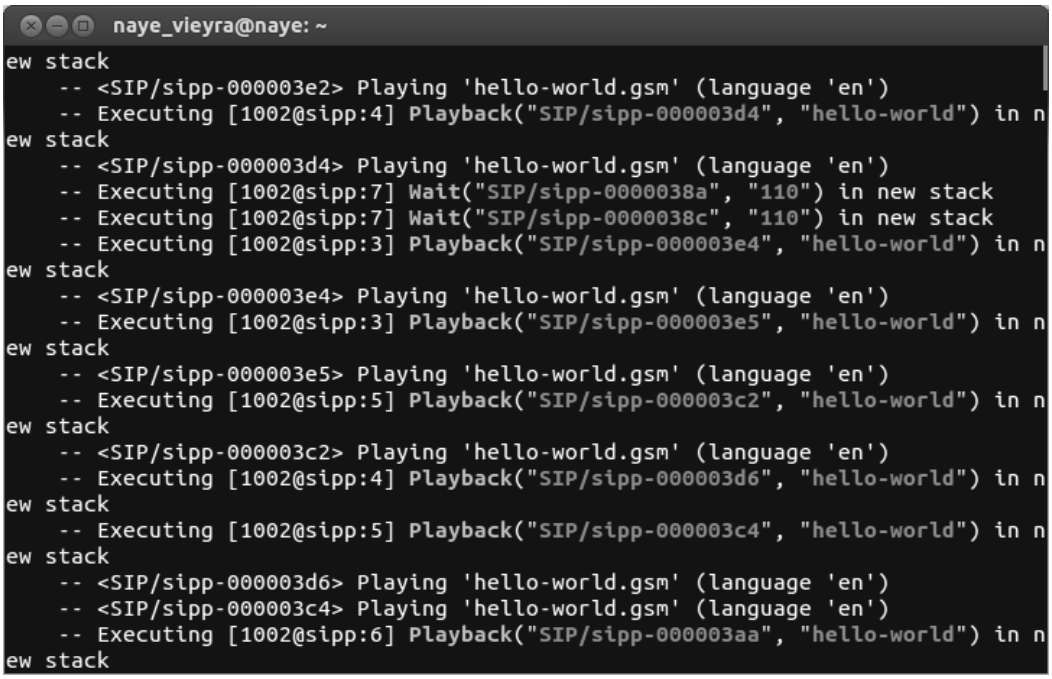


Figura 6. 3 Consola de Asterisk procesando llamadas.

6.2.3 Paquetes transmitidos

Es la siguiente **Figura 6. 4** se muestra los paquetes por segundo de los dos canales abiertos para la llamada. Por cada canal se trasmiten 50 paquetes pos segundo. La línea negra refleja los paquetes de que viajan de *softphone* 1 (usuario 80) al *softphone* 2 (usuario 81), y las líneas rojas son los paquetes del *softphone* 2 al *softphone* 1. El códec utilizado es el G.711 el cual genera 50 paquetes por segundo.

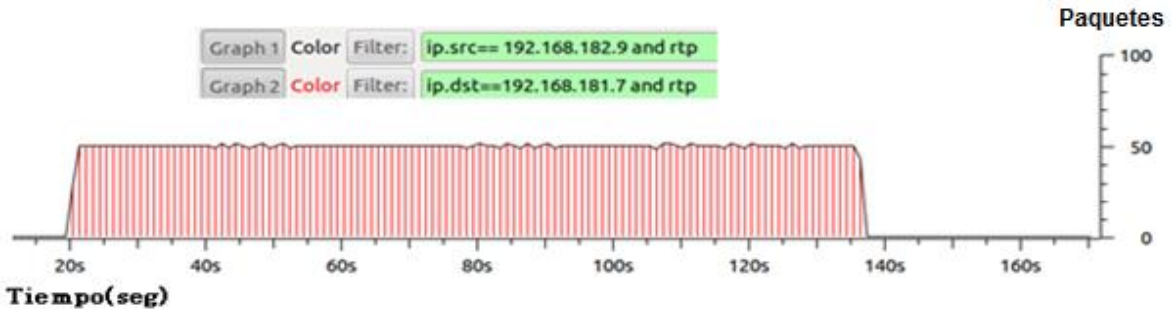
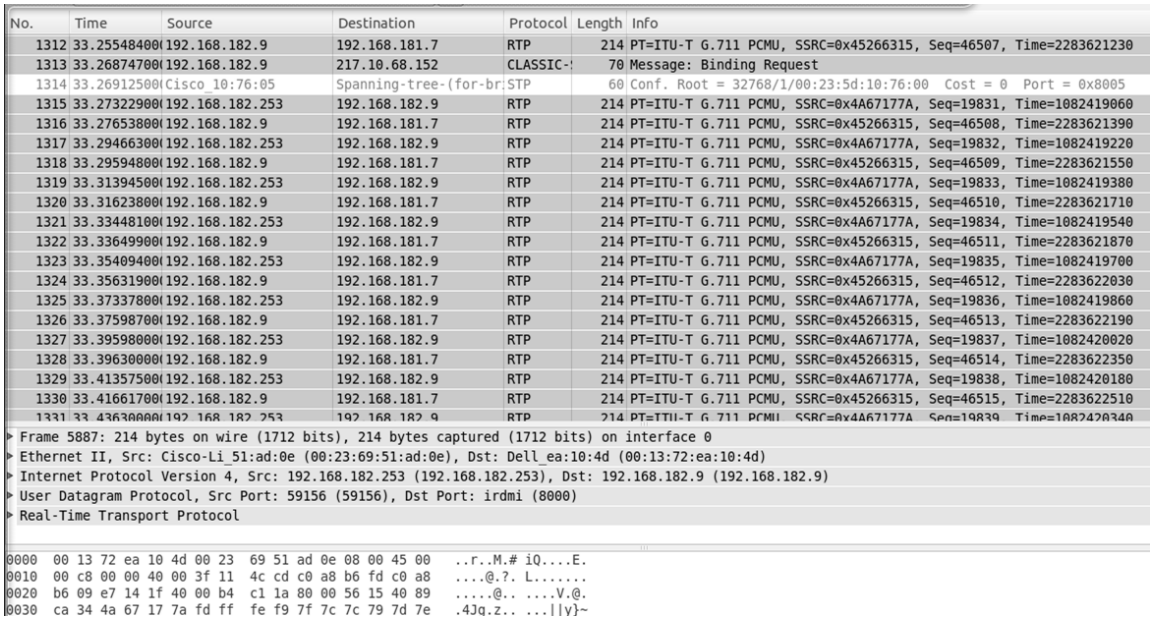


Figura 6. 4 Paquetes transmitidos entre llamada con Zoiper.

En la **Figura 6. 5** se muestran algunos de los paquetes RTP capturados por Wireshark.



No.	Time	Source	Destination	Protocol	Length	Info
1312	33.25548400(	192.168.182.9	192.168.181.7	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x45266315, Seq=46507, Time=2283621230
1313	33.26874700(	192.168.182.9	217.10.68.152	CLASSIC-(	70	Message: Binding Request
1314	33.26912500(	Cisco_10:76:05	Spanning-tree-(for-br	STP	60	Conf. Root = 32768/1/00:23:5d:10:76:00 Cost = 0 Port = 0x8005
1315	33.27322900(	192.168.182.253	192.168.182.9	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x4A67177A, Seq=19831, Time=1082419060
1316	33.27653800(	192.168.182.9	192.168.181.7	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x45266315, Seq=46508, Time=2283621390
1317	33.29466300(	192.168.182.253	192.168.182.9	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x4A67177A, Seq=19832, Time=1082419220
1318	33.29594800(	192.168.182.9	192.168.181.7	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x45266315, Seq=46509, Time=2283621550
1319	33.31394500(	192.168.182.253	192.168.182.9	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x4A67177A, Seq=19833, Time=1082419380
1320	33.31623800(	192.168.182.9	192.168.181.7	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x45266315, Seq=46510, Time=2283621710
1321	33.33448100(	192.168.182.253	192.168.182.9	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x4A67177A, Seq=19834, Time=1082419540
1322	33.33649900(	192.168.182.9	192.168.181.7	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x45266315, Seq=46511, Time=2283621870
1323	33.35409400(	192.168.182.253	192.168.182.9	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x4A67177A, Seq=19835, Time=1082419700
1324	33.35631900(	192.168.182.9	192.168.181.7	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x45266315, Seq=46512, Time=2283622030
1325	33.37337800(	192.168.182.253	192.168.182.9	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x4A67177A, Seq=19836, Time=1082419860
1326	33.37598700(	192.168.182.9	192.168.181.7	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x45266315, Seq=46513, Time=2283622190
1327	33.39598000(	192.168.182.253	192.168.182.9	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x4A67177A, Seq=19837, Time=1082420020
1328	33.39630000(	192.168.182.9	192.168.181.7	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x45266315, Seq=46514, Time=2283622350
1329	33.41357500(	192.168.182.253	192.168.182.9	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x4A67177A, Seq=19838, Time=1082420180
1330	33.41661700(	192.168.182.9	192.168.181.7	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x45266315, Seq=46515, Time=2283622510
1331	33.43630000(	192.168.182.253	192.168.182.9	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x4A67177A, Seq=19839, Time=1082420340

```
▸ Frame 5887: 214 bytes on wire (1712 bits), 214 bytes captured (1712 bits) on interface 0
▸ Ethernet II, Src: Cisco-Li_51:ad:0e (00:23:69:51:ad:0e), Dst: Dell_ea:10:4d (00:13:72:ea:10:4d)
▸ Internet Protocol Version 4, Src: 192.168.182.253 (192.168.182.253), Dst: 192.168.182.9 (192.168.182.9)
▸ User Datagram Protocol, Src Port: 59156 (59156), Dst Port: irdmi (8000)
▸ Real-Time Transport Protocol

0000  00 13 72 ea 10 4d 00 23  69 51 ad 0e 08 00 45 00   ..r..M.# iQ....E.
0010  00 c8 00 00 40 00 3f 11  4c cd c0 a8 b6 fd c0 a8   ....@.?. L.......
0020  b6 09 e7 14 1f 40 00 b4  c1 1a 80 00 56 15 40 89   .....@.. ....V.@.
0030  ca 34 4a 67 17 7a fd ff  fe f9 7f 7c 7c 79 7d 7e   .4Jg.z.. ...||y}~
```

Figura 6. 5 Captura de paquetes de una llamada entre softphone con Wireshark.

6.2.4 Parámetros

El *Jitter* se encuentra por debajo de los 15 ms como se muestra en las líneas negras como se muetra en la **Figura 6. 6**, lo cual no es problema ya que se encuentra muy por debajo de lo permitido.

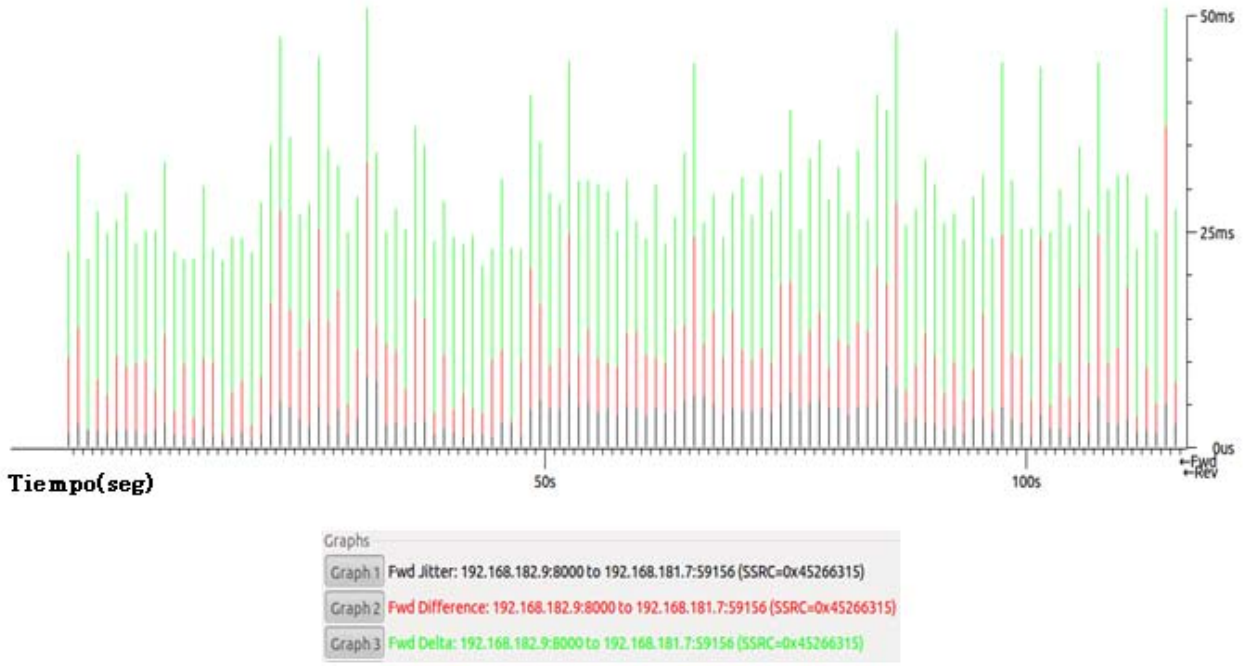


Figura 6. 6 Jitter y delta de una llamada con Zoiper.

En la figura de abajo se corrobora que se transmiten 50 paquetes por segundo esta fue tomada por la herramienta kSar como se muestra en la captura **Figura 6. 7** . En la primera grafica la línea azul representa los paquetes recibidos y la verde los paquetes transmitidos. La segunda grafica muestra los bytes transmitidos y recibidos. Esta medición se realizó en el servidor Asterisk mientras estaba activa la llamada entre el usuario 80 y 81.

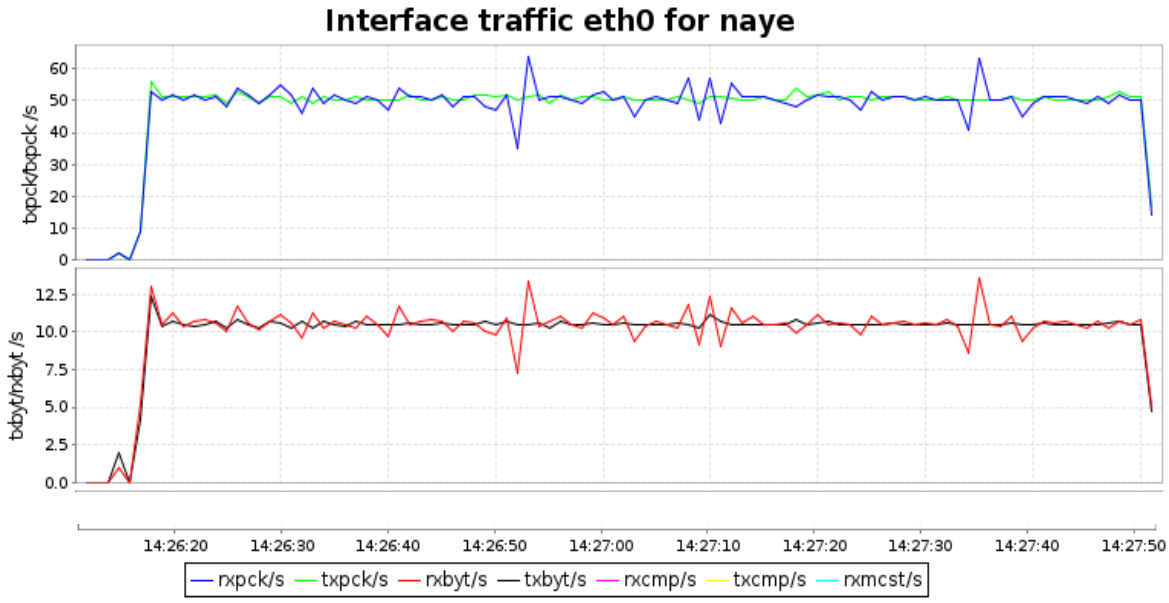


Figura 6. 7 Número de paquetes y bytes transmitidos.

6.3 Cálculo del ancho de banda para una llamada VoIP

Calcular correctamente el ancho de banda requerido para una llamada de VoIP es crucial para estimar el rendimiento de la red y brindar una calidad de servicio adecuada, el conocer el ancho de banda necesario para una llamada permite calcular el número de llamadas simultáneas que se pueden tener en un enlace. Teniendo en cuanta que en un futuro la red puede crecer y el número de llamadas aumentar, así como también circule otro tipo de tráfico aunque en este trabajo no es el caso.

Calcular el tamaño del paquete en bytes para el códec G.711, el cual tiene un tamaño de carga útil de 160 bytes. Existen dos formatos para calcular el tamaño del paquete esto dependerá del protocolo que se maneje a nivel capa MAC, si es Ethernet (**Tabla 6.1**) o PPP (**Tabla 6.2**), se muestra a continuación.

Ethernet encapsulamiento

Ethernet	IP	UDP	RTP	Datos de Voz
14 bytes	20 bytes	8 bytes	12 bytes	160 bytes

Tabla 6.1 Encapsulamiento para Ethernet.

PPP encapsulamiento

PPP	IP	UDP	RTP	Datos de Voz
4 bytes	20 bytes	8 bytes	12 bytes	¿?

Tabla 6.2 Encapsulamiento para PPP.

Calcular el tamaño del paquete de voz que es la sumatoria de los encabezados de la capa 2,3 y 4 más los datos de voz. Ver **Tabla 6.3**

Ethernet	IP	UDP	RTP	Datos de Voz o Payload	**Total tamaño del paquete**
14 bytes	20 bytes	8 bytes	12 bytes	160 bytes	**214 bytes**

Tabla 6.3 Encapsulamiento Ethernet y payload G.711.

Para este trabajo se toma como referencia el cálculo para Ethernet en la capa MAC, nuestro flujo de voz atravesara tanto redes LAN como una WAN, al usar Ethernet se utiliza un mayor tamaño del paquete por mayor número de bytes agregados en esta capa que PPP[35].

Donde el Payload depende del códec a utilizar.

Tamaño del paquete total en bytes = 214 bytes

Obtener el tamaño del paquete en bits basta multiplicar 214 bytes por 8;

Tamaño del paquete total en bits= 214 bytes *8 = 1712bits

El número de paquetes por segundo generados por G.711 está dado:

$$PPS = \frac{velocidad\ de\ bits\ en\ codec}{tamaño\ de\ la\ carga\ \text{útil}\ \ de\ voz}$$

Para el códec utilizado tenemos que:

$$PSS = \frac{64Kbps}{160\ bytes} = \frac{64000bps}{(160bytes)(8)} = 50pps$$

Ancho de banda (BW) por llamada

$$BW \times llamada = (Tamaño\ de\ paquete\ de\ voz)(paquetes\ por\ segundo)$$

$$BW \times llamada = (1712bits)(50pps) = 85600bps = 85.6Kbps$$

6.3.1 Calcular número de llamadas concurrentes en un enlace E2

Calcular el número de llamadas simultáneas soportadas en un enlace nos permite conocer el milite de llamadas soportadas, pues una vez que se sobrepase este limite la red comenzará a tener conflictos ya sea en cuanto a retardos, perdida de paquetes, llamadas cortadas o una perdida en la calidad de servicio.

Los enlaces utilizados son E2 con un ancho de banda de 8.448 Mbps y 128 canales de 64Kbps cada uno. Este enlace utiliza 256 Kbps de relleno.

$$(128\ canales)(64Kbps) = 8192Kbps$$

$$8192Kbps + 256Kbps = 8448Kbps$$

Para realizar el cálculo debemos discriminar los 256Kbps de relleno por lo que solo tomaremos en cuanta 8192Kbps a lo cual se descuenta el 10% como recomendación, no se debe saturar la red al 100%, dentro del 10% se incluyen 2 canales de señalización.

$$\frac{8192Kbps}{7372.8Kbps} \frac{=> 100\%}{=> 90\%}$$

El ancho de banda a tomar en cuanta es 7372.8Kbps y cada llamada requiere 85.6Kbps entonces basta con hacer una simple división.

$$Num\ llamadas = \frac{BW\ enlace}{BW\ por\ llamada}$$

$$\frac{7372.8Kbps}{85.6Kbps} = 86.1\ llamadas$$

Al hablar de llamadas se debe de tener número cerrados por lo que 86 llamadas son las soportadas por el enlace.

6.4 Llamadas SIP con SIPP a Asterisk

6.4.1 Número de flujos abiertos por Asterisk

En realidad son 86 flujos de señalización, lo que se tiene en realidad son 43 llamadas simultaneas, pues se comprobó con la llamada entre *softphones* que se crean dos canales de 85.6Kbps por llamada. Se realizaron 86 llamadas en cada cliente (LAN 1 = 1001 y LAN2 =1002) en base al cálculo del número llamadas simultaneas soportadas por un enlace con ancho de banda de 7372.8 Kbps

Las llamadas en cada cliente con petición a Asterisk, se muestra en las Figuras de señalización obtenida para cada uno de los clientes al generar las llamadas.

El cliente 1 en la red LAN 1 genera 86 flujos los cuales uno no se completa. Ver **Figura 6. 8**, como bien se mencionó por cada dos flujos generados se tiene una llamada, así que se considera que solo se establecieron 42 llamadas exitosas.

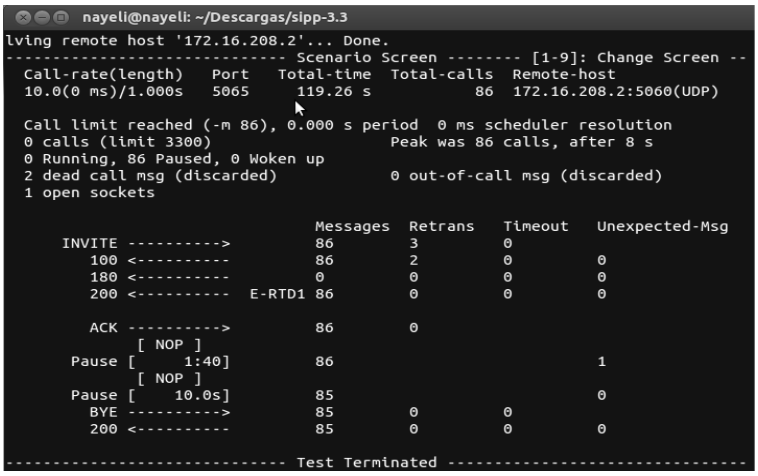


Figura 6. 8 Señalización SIP con SIPP cliente 1.

Se muestran las estadistas de las llamadas exitosas y fallidas, así como el tiempo de análisis en la **Figura 6. 9**.

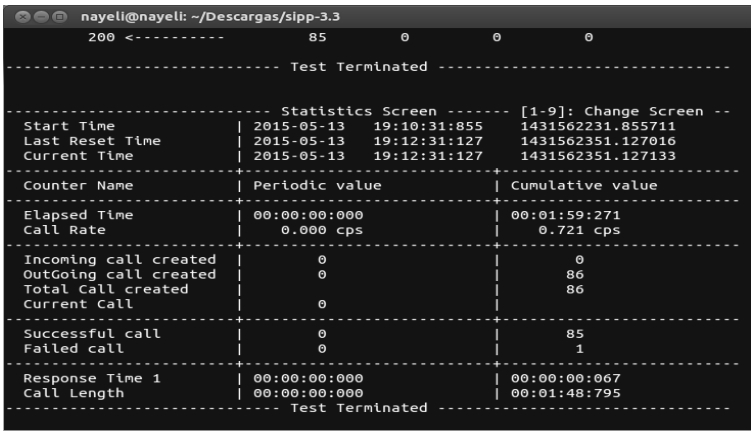


Figura 6. 9 Estadísticas SIPP cliente 1.

La consola de Asterisk ver **Figura 6. 10**, recibe las peticiones para establecer la conexión de llamadas telefónicas realizado por los clientes, se establecen 172 diálogos o flujos, 86 flujos correspondientes a cada cliente, de los cuales son 43 llamadas simultaneas.

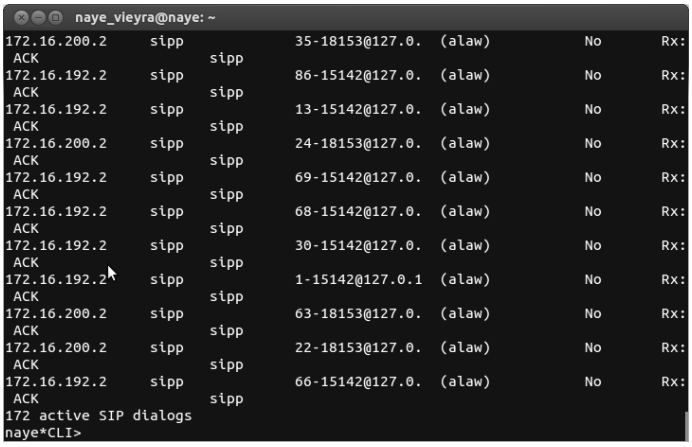


Figura 6. 10 Flujos establecidos en Asterisk para los dos clientes.

6.4.2 Señalización SIP entre clientes y Asterisk

En la **Figura 6. 11** se muestra la señalización *SIP* generada por cada uno de los clientes hacia Asterisk en color rojo y verde, y la suma de la señalización de ambos clientes da como resultado la gráfica en azul, Asterisk responde a esa señalización enviado por los clientes. Los paquetes de señalización que genera SIP al inicio (13s-25s) es para el establecimiento de llamadas y la señalización de los 120s es para finalizar las llamadas y liberar recursos.

se muestra la señalización *SIP* generada por cada uno de los clientes hacia Asterisk en color rojo y verde, y la suma de la señalización de ambos clientes da como resultado la gráfica en azul, Asterisk responde a esa señalización enviado por los clientes. Los paquetes de señalización que genera SIP al inicio (13s-25s) es para el establecimiento de llamadas y la señalización de los 120s es para finalizar las llamadas y liberar recursos.

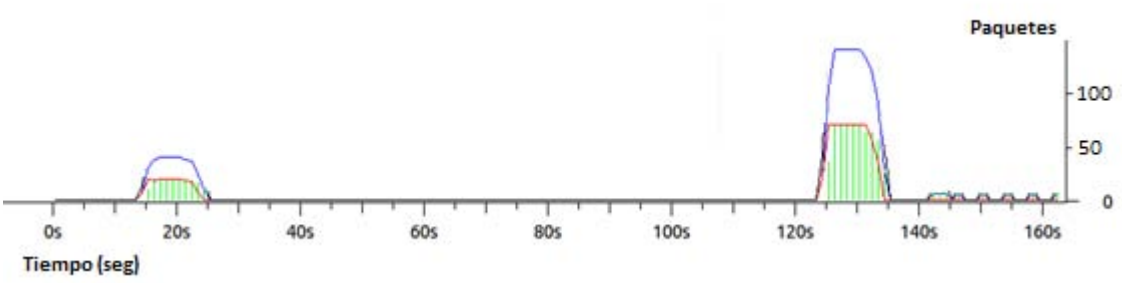


Figura 6. 11 Paquetes de señalización SIP.

6.4.3 Paquetes enviados

En la **Figura 6. 12** se muestra una gráfica en la cual, la línea negra representa los paquetes RTP enviados. La línea verde (cliente 1001) y roja (cliente 1002) representan los paquetes enviados de cada cliente y por último, las líneas en color azul representan los paquetes que llegan el servidor Asterisk que es la suma de los paquetes de los clientes, por lo cual la gráfica azul y negra se encentran igual.

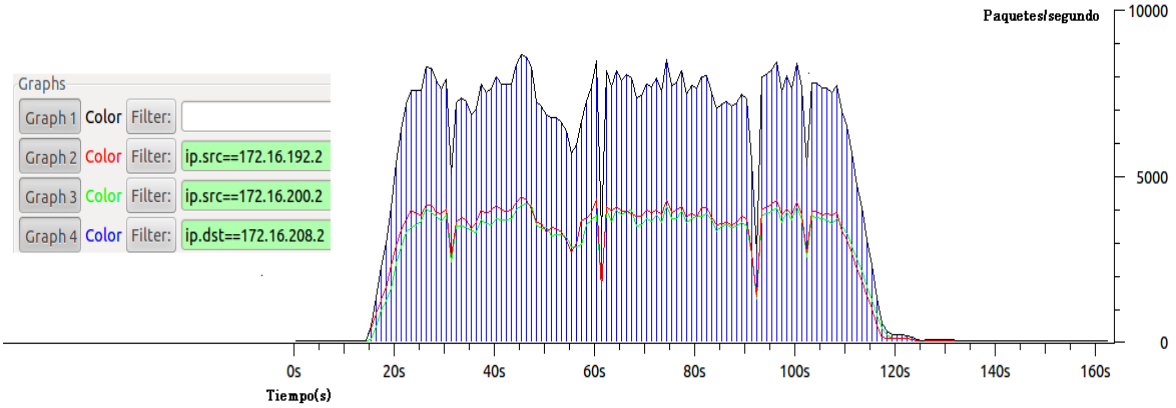


Figura 6. 12.Total de paquetes enviados.

6.4.4 Paquetes RTP

En la **Figura 6. 13** la gráfica negra y roja se observa el tráfico RTP enviado de cada uno clientes (4250 pk y 4300pk) y en verde el tráfico total enviado de 8550 aproximadamente y en cuanto al cálculo teórico se tiene que:

Cada cliente genero 86 llamadas, teniendo en cuanta que para el cliente uno solo 85 llamadas fueron exitosas mientras que para el cliente 2 todas fueron exitosas ; y si cada segundo se transmiten 50 paquetes por segundo, y tomando en cuenta el primer cliente solo género.

85*50=4250 paquetes por segundo

86*50=4300 paquetes por segundo

4250paquetes +4300paquetes = 8550

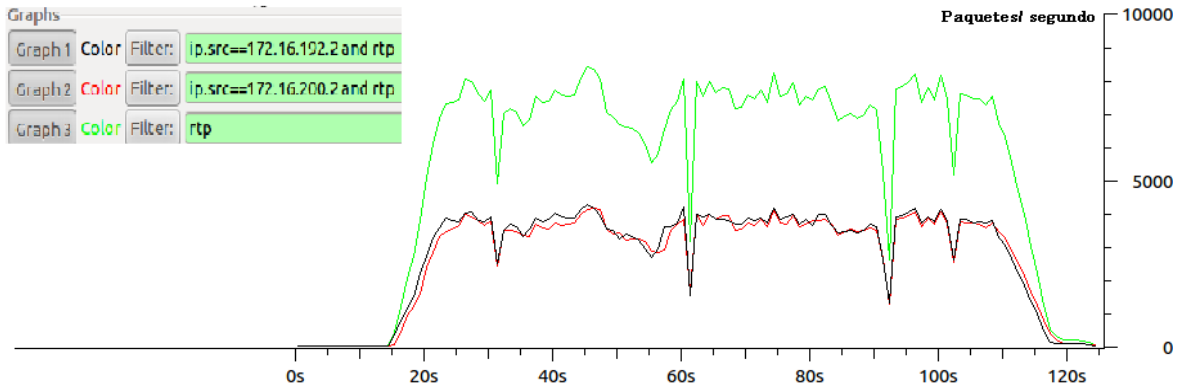


Figura 6. 13 Paquetes RTP transmitidos entre cliente y Asterisk.

6.4.5 Jitter y latencia

Jitter

El *Jitter* promedio de las llamadas realizadas fue de 16.07 ms lo cual es un muy buen parámetro. El mínimo obtenido 0.08ms y el máximo de 99.46ms en la transmisión de un *stream* del cliente hacia Asterisk **Figura 6. 14**.

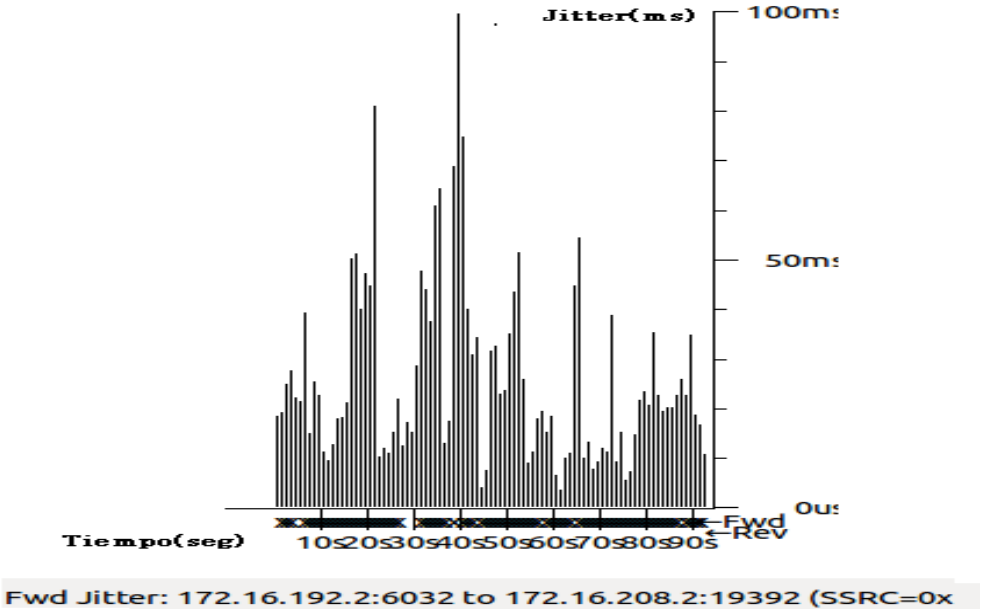


Figura 6. 14 Jitter entre cliente y Asterisk.

Delta

En el delta (latencia) de 1074.14 ms en él envió de paquetes como se muestra en la **Figura 6. 15**.

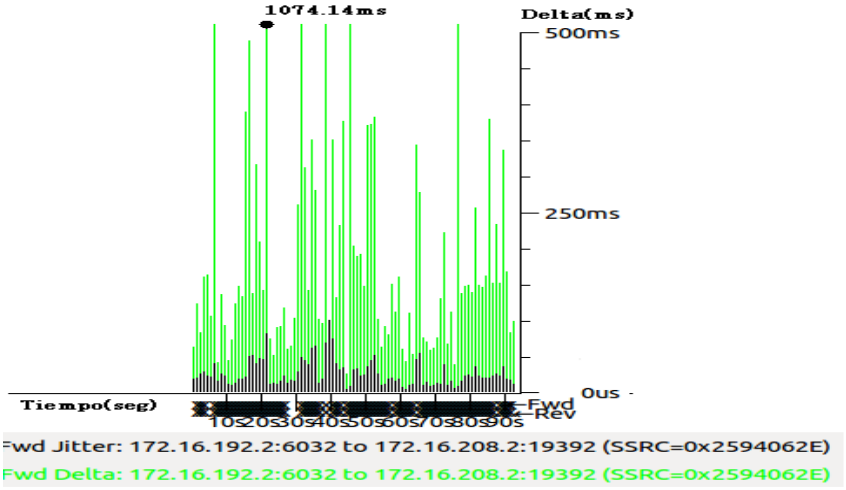


Figura 6. 15 Delta entre cliente y Asterisk.

6.4.6 Resultados para SIP

Las estadísticas obtenidas utilizando Wireshark se muestran en la **Figura 6. 16**.

```
Max delta = 1074.14 ms at packet no. 541465
Max jitter = 99.46 ms. Mean jitter = 16.07 ms.
Max skew = -4134.19 ms.
Total RTP packets = 4565  (expected 4565)  Lost RTP packets = 486 (10.65%)  Sequence errors = 237
Duration 93.94 s (-2768 ms clock drift, corresponding to 7764 Hz (-2.95%))
```

Figura 6. 16 Estadísticas SIP.

Parámetros obtenidos de 172 flujos realizados son SIPp, equivalente a 86 flujos en cada cliente con 43 llamadas cada uno.

6.5 Llamadas con Callgen 323 a Asterisk

La instalación y configuración de Callgen 323 se encuentra en el punto 5.6

6.5.1 Señalización H.323

En la **Figura 6. 17** se muestra el número de paquetes de señalización generados por H.323. La línea azul y las líneas rojas representan los mensajes de señalización generados por cada cliente hacia Asterisk, y la suma de la señalización de ambos clientes da como resultado la gráfica en color verde. Los paquetes entre los 20s son de inicio y establecimiento de las llamadas, mientras que los paquetes de señalización entre los 125s y 135s son para finalizar y liberar los recursos de las llamadas.

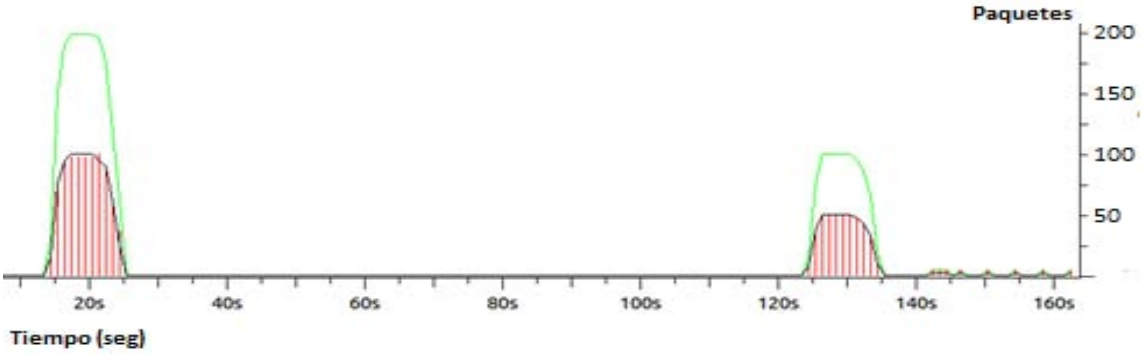


Figura 6. 17. Número de paquetes de señalización H.323.

6.5.2 Paquetes transmitidos

En la **Figura 6. 18** se muestra todo el tráfico enviado tanto de señalización como RTP, en número de paquetes oscila entre los 8000 incluida la señalización como se observa en la primera grafica en color azul, en la segunda se muestra en bytes.

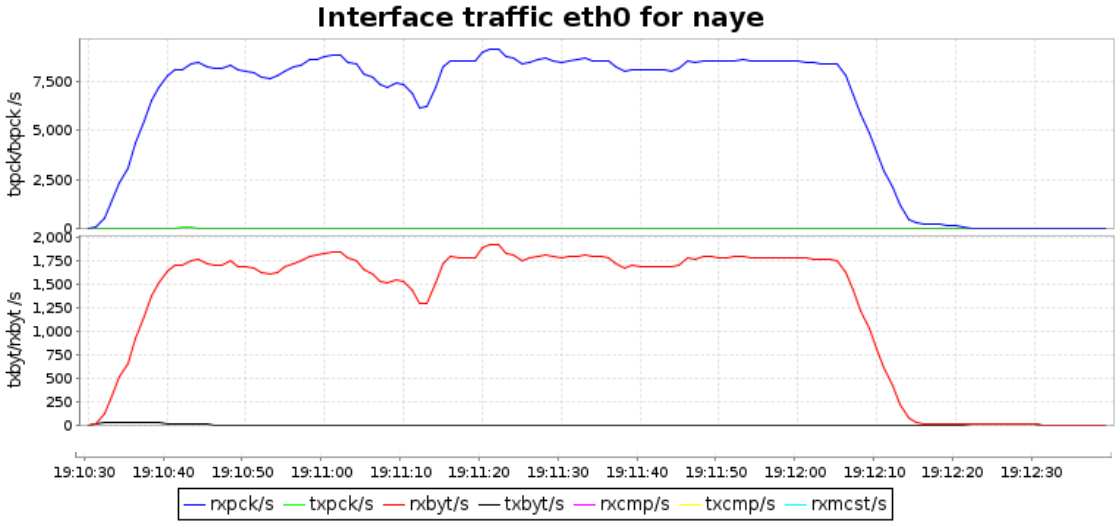


Figura 6. 18 Paquetes transmitidos y el ancho de banda.

En la **Figura 6. 19** se encuentra de nuevo los paquetes que oscilan entre 8000 y 8500 y un ancho de banda 7.5Mbps acordes con lo calculado de forma teórica, pero la señalización ocupa una mayor cantidad de ancho de banda.

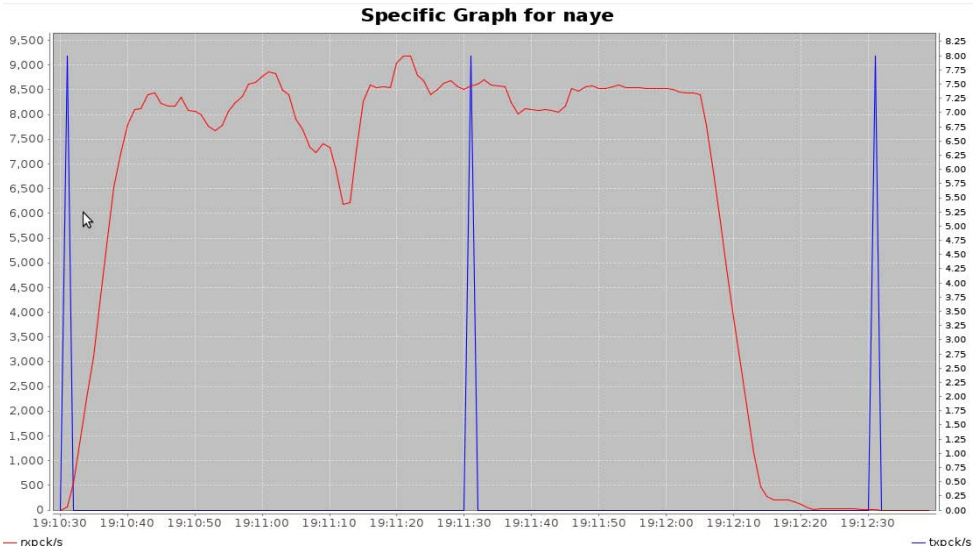


Figura 6. 19 Paquetes y ancho de banda para H.323.

6.5.3 Jitter y latencia

El protocolo H.323 obtuvo un Jitter promedio de 25ms un poco más elevado que SIP, pero considerable. En el delta (latencia) de 2.13s como se observa en la **Figura 6. 20**.

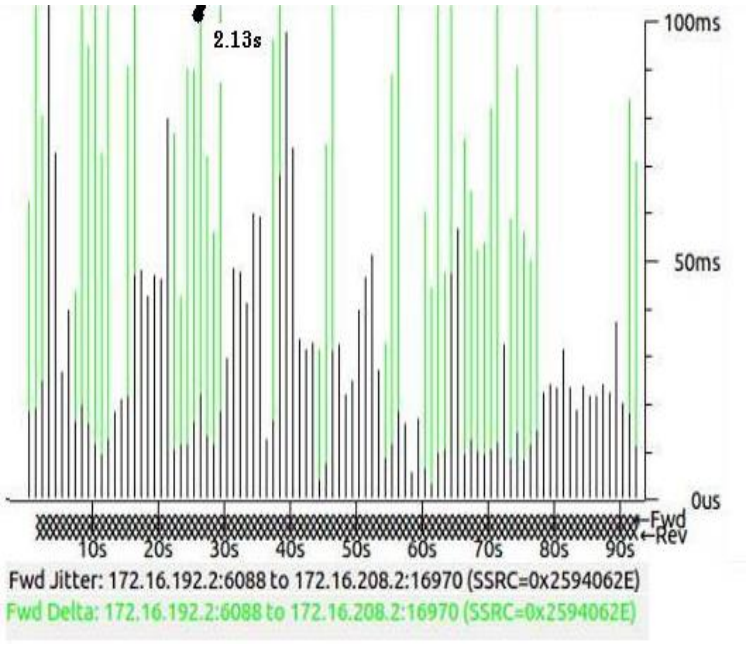


Figura 6. 20 Jitter y latencia de H.323.

6.6 Resultados generales

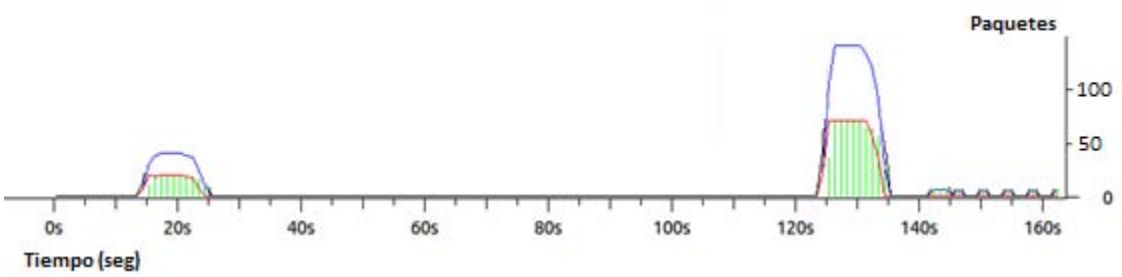
En la **Tabla 6.4** se muestras los resultados obtenidos de los protocolos SIP y H.323.

Protocolo/ características	**SIP**	**H.323**
Implementación	Sencilla	Compleja
Llamada realizadas	86 llamadas (172 flujos)	86 llamadas
Llamadas exitosas	173	169
Llamadas perdidas	1	4
Máximo *Jitter*	99.46ms	97.67ms
Mean *Jitter*	16.07ms	25ms
Máximo Delta	1s	2.13s

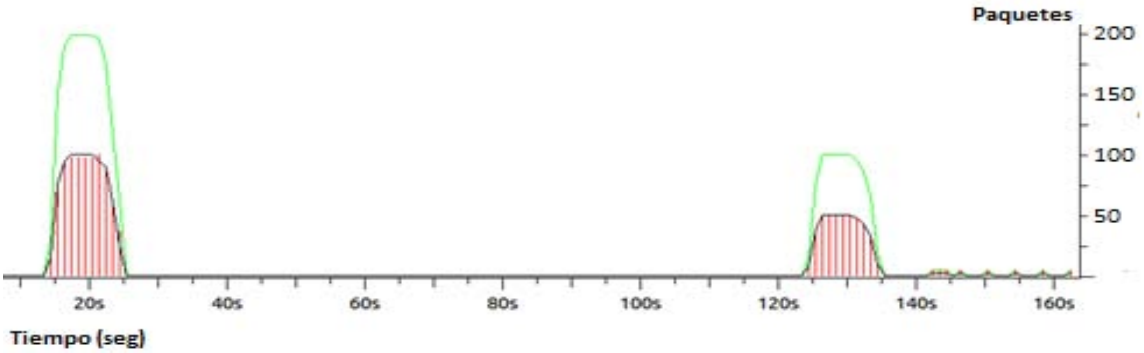
Tabla 6.4 Resultados generales.

En la **Figura 6. 21** se ven los graficas correspondientes a la señalización generada por protocolos SIP y H.323, para el establecimiento de llamadas de los clientes hacia Asterisk.

En la cual se observa que la señalización generada por H.323 para iniciar y establecer las llamadas es tres veces mayor que la que genera SIP.



a) SIP



b) H.323

Figura 6. 21. Señalización SIP vs H.323

Conclusiones

En conclusión en cuanto a complejidad H.323 es un estándar demasiado complicado para su implementación y más si es bajo *Open Source* ya que requiere una gran cantidad de librerías, pero tiene una arquitectura muy completa. Las terminales H.323 como mínimo deben ofrecer el servicio de voz, si se requieren otros servicios es cosa del usuario.

La complejidad de H.323 se enfoca en la gran cantidad de diferentes tipos mensajes codificados en binario con los que cuenta este protocolo, en comparación con los pocos y ligeros mensajes de SIP. Los resultados obtenidos en este trabajo dicen que H.323 genera hasta tres veces más tráfico de señalización que SIP, por lo cual algunas llamadas no fueron establecidas bajo H.323.

La genialidad de SIP son los formatos de mensajes que al basarse en HTTP o XML se muestran comprensibles para los humanos, además de que HTTP ya cuenta con una larga experiencia trabajando sobre Internet. Por el contrario H.323 está basado en mecanismo de SS7 y utiliza codificación binaria lo que lo hace incompresible al humano. Para el procesamiento computacional es totalmente al revés lo cual provoca que SIP tarde más en procesar los mensajes enviados y recibidos y tiene un mayor retardo

También es necesario mencionar que la robustez de H.323 complica la implantación de una red de VoIP, además de dificultar la detección de problemas. Mientras que SIP es de fácil programación e implementación, como la fácil detección de errores. SIP puede moldearse a las necesidades futuras de los usuarios, por el contrario H.323 es rígido, pero con la gran ventaja de que al ser bien implementado es eficiente en redes conectadas a la PSTN, lo cual se complica con SIP

En cuanto al direccionamiento SIP no brinda demasiadas alternativas más que su dirección URI mientras que H.323 es capaz de trabajar con direcciones IP:puerto, alias, número telefónico tradicional o un URL. Lo que le quita claridad y mayor facilidad de compresión a H.323 pues el manejar gran variedad de direccionamiento lo hace confuso.

Uno de los problemas iniciales es tener que batallar con la apertura de los puertos utilizados por los protocolos pues se necesita de la desactivación de los firewalls, puesto que H.323 realiza la negociación de los puertos a utilizar durante la conexión y SIP definir los puertos mientras el establecimiento de la llamada encargado de ello el protocolo SDP.

El protocolo SIP cumple con los 150ms de latencia recomendados por G.114 de la ITU- T, mientras que H.323 rebasa este parámetro, lo cual perjudica el comportamiento de la red.

A grandes rasgos se concluye que la el protocolo SIP es mucho más ligero y amigable, sin generar grandes volúmenes de mensajes, lo que lo hace eficiente en redes completamente IP. Y el protocolo H.323 su gran robustez lo hace pesado y generador gran cantidad de carga para redes completamente IP.

Referencias

[1] J. Hallock, “A brief history of VoIP,” 2004.

[2] “El Impacto de la Telefonía IP en las Organizaciones.UNIVERSIDAD VERACRUZANA.acultad de Contaduría y Administración.” [Online]. Available: http://cdigital.uv.mx/bitstream/123456789/28505/1/Ferraez Rodriguez.pdf. [Accessed: 05-Aug-2014].

[3] J. G. Rodrigo, *Instalaciones de radiocomunicaciones*. Editorial Paraninfo, 2012, p. 224.

[4] “CCNA® Voice Study Guide > Traditional Telephony > Comparing Analog and Digital Circuits : Safari Books Online.” [Online]. Available: http://my.safaribooksonline.com/book/certification/ccna/9780470527665/traditional-telephony/understanding_analog_network_signaling#X2ludGVybmFsX0h0bWxWaWV3P3htbGlkPTk3ODA0NzA1Mjc2NjUlMkZjb21wYXJpbmdfYW5hbG9nX2FuZF9kaWdpdGFsX2NpcmN1aXRzJnF1ZXJ5PQ==. [Accessed: 24-Jul-2014].

[5] F. de I. Dr. Ing. José Joskowicz, Instituto de Ingeniería Eléctrica, Niería, U. de la República, U. Montevideo, and M. 2013, “Microsoft Word - Codificacion de voz y video V3.doc - Codificacion_de_voz_y_video_textos_completos_.pdf,” 2013.

[6] R. Y. OTROS, *Servicios en Red*. Editorial Paraninfo, 2010, p. 254.

[7] M. Vallina and Miguel, *Infraestructuras de redes de datos y sistemas de telefonía*. Editorial Paraninfo, 2013, p. 160.

[8] J. D. C. Pozo, *Sistemas de telefonía*. Editorial Paraninfo, 2007, p. 355.

[9] “Evaluation of Transport Protocols for the Session Initiation Protocol.” [Online]. Available: https://aaltodoc.aalto.fi/bitstream/handle/123456789/5113/publication3.pdf. [Accessed: 20-Dec-2014].

[10] P. Papageorgiou, “Comparison of SIP and H.323 Protocols,” *University of Maryland at College Park*. [Online]. Available: http://www.cs.umd.edu/~pavlos/papers/unpublished/papageorgiou01comparison.pdf. [Accessed: 20-Dec-2014].

[11] H. S. y J. Rosenberg, “A Comparison of SIP and H.323 for Internet Telephony,” p. 4.

[12] T. Abbasi, S. Prasad, N. Seddigh, and I. Lambadaris, “A comparative study of the SIP and IAX VoIP protocols,” *Can. Conf. Electr. Comput. Eng. 2005.*, no. May, pp. 179–183, 2005.

[13] J. Glasmann, W. Kellerer, and H. Muller, “Service architectures in H.323 and SIP: A comparison,” *IEEE Commun. Surv. Tutorials*, vol. 5, no. 2, pp. 32–47, 2003.

[14] H. F. Ismail Dalgic, “Comparison of H.323 and SIP for IP Telephony Signaling,” *Technology Development Center and Cisco Systems a*. [Online]. Available: file:///C:/Users/Administrador/Downloads/Dalg9909_Comparison.pdf. [Accessed: 18-Dec-2014].

[15] S. K. Das, E. Lee, K. Basu, and S. K. Sen, “Performance optimization of VoIP calls over wireless links using h.323 protocol,” *IEEE Trans. Comput.*, vol. 52, no. 6, pp. 742–752, Jun. 2003.

[16] “Methodology for SIP Infrastructure Performance Testing.” [Online]. Available: http://startrinity.com/voip/resources/sip311.pdf. [Accessed: 24-Dec-2014].

[17] VoIPForo, “VoIP Foro - Arquitectura SIP : Session Initiation Protocol - RFC 3261.”

[18] J. M. B. Ordinas, *Protocolos y aplicaciones Internet*. Editorial UOC, 2008, p. 240.

[19] J. Andreu, *Voz IP (Servicios en red)*. Editex, 2011, p. 44.

[20] S. S. Gokhale, “Signaling performance of SIP based VoIP: a measurement-based approach,” *GLOBECOM ’05. IEEE Glob. Telecommun. Conf. 2005.*, p. 5 pp.–765, 2005.

[21] “H.323 : Packet-based multimedia communications systems.” [Online]. Available: http://www.itu.int/rec/T-REC-H.323-200912-I. [Accessed: 27-Aug-2014].

[22] “H.323 : Use of Facility message to enable call transfer.” [Online]. Available: http://www.itu.int/rec/T-REC-H.323-201303-I!Amd1. [Accessed: 27-Aug-2014].

[23] VoIPForo, “VoIP Foro - H323 Componentes H.323 : Gatekeeper, gateway, proxy.”

[24] “ProtocolsSeñalizacionv2.PDF - protocolssenalizacion.pdf.” [Online]. Available: http://www.it.uc3m.es/~jmoreno/articulos/protocolssenalizacion.pdf. [Accessed: 27-Aug-2014].

[25] S. C. R. F. and V. H. Schulzrinne, Jacobson., “RTP: A Transport Protocol for Real-Time Applications, RFC1889.” [Online]. Available: https://www.ietf.org/rfc/rfc1889.txt. [Accessed: 17-Nov-2014].

[26] S. C. R. F. and V. H. Schulzrinne, Jacobson., “RTP: A Transport Protocol for Real-Time Applications, RFC3550.” [Online]. Available: https://www.ietf.org/rfc/rfc3550.txt. [Accessed: 17-Nov-2014].

[27] “Codecs y Formatos en Telefonia IP | ElastixTech - Aprende Telefonia IP Asterisk - Elastix.” [Online]. Available: http://elastixtech.com/codecs-y-formatos-en-telefonia-ip/. [Accessed: 18-Nov-2014].

[28] ITU-T, “ITU-T G Series: Transmission systems and media, digital systems and networks.” [Online]. Available: http://www.itu.int/net/itu-t/sigdb/speaudio/Gseries.htm#G.720.1. [Accessed: 24-Nov-2014].

[29] CISCO, “Understanding 2-Port Serial WAN Interface Card (WIC-2T).” [Online]. Available: http://www.cisco.com/c/en/us/support/docs/routers/3600-series-multiservice-platforms/7261-wic-2t.html. [Accessed: 20-Aug-2014].

[30] “Asterisk Architecture, The Big Picture - Asterisk Project - Asterisk Project Wiki.” [Online]. Available: https://wiki.asterisk.org/wiki/display/AST/Asterisk+Architecture%2C+The+Big+Picture. [Accessed: 26-Apr-2015].

[31] “Iperf - The TCP/UDP Bandwidth Measurement Tool.” [Online]. Available: https://iperf.fr/. [Accessed: 27-Apr-2015].

[32] “Zoiper - Free VoIP SIP softphone dialer with voice, video and instant messaging :: Zoiper.” [Online]. Available: http://www.zoiper.com/en. [Accessed: 18-May-2015].

[33] “sipp / Mailing Lists.” [Online]. Available: http://sourceforge.net/p/sipp/mailman/message/10740282/. [Accessed: 27-Apr-2015].

[34] “SIPp. Create your own XML scenarios.” [Online]. Available: http://sipp.sourceforge.net/doc/reference.html#Create+your+own+XML+scenarios. [Accessed: 10-Jun-2015].

[35] “Voz sobre IP - Consumo de ancho de banda por llamada.” [Online]. Available: http://www.cisco.com/cisco/web/support/LA/7/73/73295_bwidth_consume.html. [Accessed: 26-Nov-2014].

[36] M. C. E. Boquera, *Servicios avanzados de telecomunicación*. Ediciones Díaz de Santos, 2003, p. 816.

Glosario

UTF-8	8-bit *Unicode Transformation Format*
(ITSP	Internet Telephony Service Provider/ Proveedores de Servicios de Telefonía por Internet
ACR,	Absolute Category Rating
ARPANET	Advanced Research Projects Agency Network, Red de la Agencia de Proyectos de Investigación Avanzada
ATM	Asynchronous Transfer Mode, Modo de transferencia asíncrono.
AUC	User Agent client, Agente de usuario cliente
AUS	User Agent Server, Agente de usuario servidor
Banda base	Banda base se refiere a un conjunto de señales que no sufren ningún proceso de modulación a la salida de la fuente que las origina, es decir son señales que son transmitidas en su frecuencia original.
CAS	La señalización y el tráfico de voz viajan a través de la misma ruta por medio de la red.
CCS	Canales de señalización común
CRC	Código de detección de errores usado en redes digitales para detectar cambios accidentales en los datos.
DCE	Data Communications Equipment, Equipo para comunicaciones de datos
DOD	United State Department of Defense. Departamento del brazo ejecutivo del gobierno federal de Estados Unidos encargado de coordinar y supervisar todas las agencias y funciones del gobierno relacionadas con seguridad y las fuerzas armadas.

DTE	Data Terminal Equipment/Equipo Terminal de Datos
full-duplex	Se refiere a que los enlaces son capaces de enviar y recibir datos al mismo tiempo.
FXO	Foreing Exchance Office Interface/ Interface de central externa
FXS	Foreing Exchange Station Interface/ Interfaz de abonado externo
HTTP	HyperText Transfer Protocol, Protocolo de transferencia de hipertexto
IAX2	Inter-Asterisk eXchange protocol
ITU-T	Intégrate Services for Digital Network / Red digital de servicios integrados
LAN	Local Area Network / Red de Area Local
PBX	Private Branch Exchange
PCM	Pulse Code Modulation / Modulación por impulsos codificados
PDU	protocolo data unit / Unidades de datos de protocolo
PPP	Point-to-Point Protocol / Protocolo Punto-a-Punto,
PPS	Paquetes por segundo
PSTN	Public Switched Telephone Network / Red telefónica pública conmutada
QoS	Quality of Service / Calidad de servicio
RJ11	Se refiere a que los enlaces son capaces de enviar y recibir datos al mismo tiempo.
RJ45	Conector utilizado en tarjetas de red Ethernet, que trasmite información a través de cables de par trenzado.
RTB	Telefonía Básica
RTC	Red de Telefonía Conmutada
RTP	Real-time Transport Protocol /Protocolo de Transporte de Tiempo real
SCCP	Skinny Call Control Protocol
SDP	Session Description Protocol / Protocolo de descripción de sesión

SP	Service provider / Provedor de servicio
SS7	Signalling System No. 7 / Sistema de señalización No. 7
SSP	Red inteligente para los conmutadores de clase 4 y 5, los SSP cuentan con una interfaz abierta a la entrada para señalización de conmutación, control y rechazo.
streaming	Término que hace referencia al hecho de escuchar música o ver vídeos sin necesidad de descargarlos, sino que se hace por fragmentos enviados secuencialmente a través de la red (como lo es Internet).
TCP	Transmission Control Protocol/ Protocolo de control de transmisión.
TCP/IP	Transmission Control Protocol/ Internet Protocol
TDM	Time-Division Multiplexing / Multiplexación por División de Tiempo: Tipo de multiplexación más utilizado en la actualidad, especialmente en los sistemas de transmisión digitales. En ella, el ancho de banda total del medio de transmisión es asignado a cada canal durante una fracción del tiempo total.
TDM	Time Division Multiple Access / Acceso múltiple por división de tiempo: Es una técnica que permite la transmisión de señales digitales compartiendo un canal de transmisión entre varios usuario todo el ancho de banda en intervalos de tiempo.
UDP	User Datagram Protocol/ Protocolo de datagrama de usuario
XML	eXtensible Markup Language /Lenguaje de marcas extensible.

Apéndice 1

Ejemplo 1. Invitación SIP donde solo participan los AU´s

Un escenario donde solo participan los AU´s en ambos extremos es el escenario más simple que podemos encontrar. EL procedimiento es el siguiente.

Cuando un usuario_1 quiere establecer comunicación con otro usuario_2. Entonces el agente de usuario cliente perteneciente al usuario_1 envía una invitación al agente de usuario servidor del usuario_2, entre ellos se intercambian solicitudes y respuestas a lo cual se le llama invitación. La solicitud que envía el usuario_1 es enviada a la dirección especificada en el *URI SIP* por el usuario_1. Si la dirección es un nombre de dominio entonces primero se consulta el DNS, el cual realizará la conversión correspondiente, este proceso se puede realizar siempre y cuando la dirección del usuario_2 sea fija, pues no es necesario hacer uso de algún servicio de localización del cual se hablará más adelante.

La solicitud para establecer comunicación entre el usuario_1 y el usuario_2, se inicia cuando el usuario_1 (*UAC*) envía una solicitud de invitación (*INVITE*) al usuario_2 (*UAS*), con el fin de reiniciar una sesión ya establecida anteriormente o bien inicia una nueva sesión entre ellos. Si el usuario_2 está en condiciones para aceptar la solicitud responde con una respuesta afirmativa hacia el usuario_1, esta repuesta es (200 *OK*). Y para terminar y confirmar que la solicitud fue exitosa el usuario_1 envía al usuario_2 un último mensaje de confirmación llamado *ACK*, como se muestra en la Figura 3.4.

En el mismo instante que el usuario_2 responde con un *OK*, este mismo puede enviar información sobre cómo se encurta el proceso de la llamada, por ejemplo indicar que su timbre de llamada sonó.

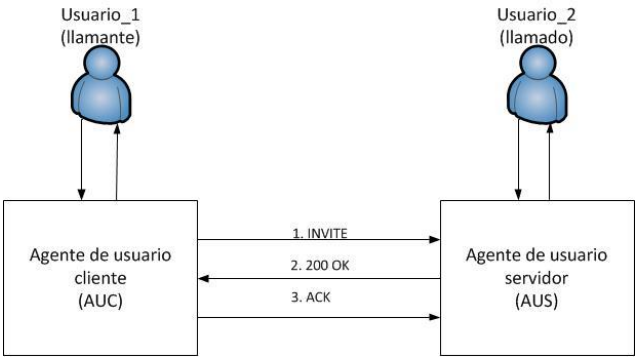


Figura de Apéndice 1. 1 Procedimiento de invitación SIP entre UA´s.

Ejemplo 2. Invitaciones y repuestas SIP utilizando un servidor de redirección

Un segundo escenario se presenta cuando utilizamos de por medio un servidor para la localización de un usuario conocido solo su dirección o el nombre de dominio.

En caso el UAC del usuario_1 envía una solicitud *INVITE* a el UAS del usuario_2, donde el usuario_1 indica a su UAC el URI *SIP* del usuario al que quiere llamar en este caso usaurio_2, entonces lo que realmente sucede es que al UAC (usuario_1) contacta a un servidor de redirección asignando de forma preestablecida y le envía la solicitud de *INVITE* con los datos del usuario_2.

El servidor representante acepta la solicitud y ahora envía la dirección del usuario _2 que fue enviada por el usuario_1 con el fin de realizar consulta a un servidor de localización y así obtener la ubicación precisa del destinatario (usuario_2). Cuando ya se conoce la dirección de usuario_2 el servidor de redirección envía una solicitud de *INVITE* al agente usuarios servidor del usuario_2, si esta parte llamada acepta la solicitud su UAS regresa una respuesta con un *OK* al servidor de redirección, el cual reenvía una respuesta *OK* hacia UAC. Cuando esta repuesta es recibida por el AUC inmediatamente contesta con un *ACK* con destino al Usuario_2, ya sea utilizando de nuevo el servidor representante para localiza al usuario final o bien que él envió se realice directamente con el usuario_2, [20] .

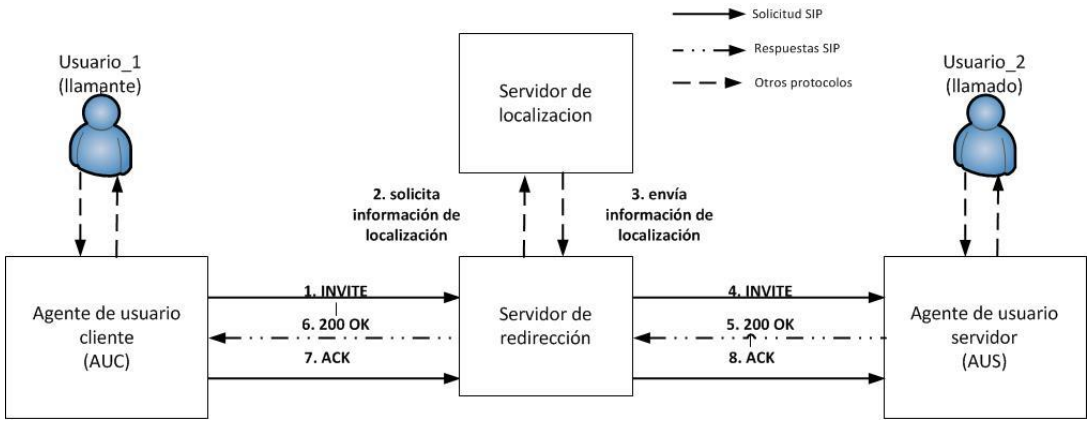


Figura de Apéndice 1.2 Procedimiento de invitación SIP utilizando un servidor de redirección.

Ejemplo 3. Invitaciones y repuestas SIP utilizando un servidor de desvió

En este tipo de escenarios cuando al solicitud *INVITE* llega al servidor de desvió esté localiza al usuario_2 y envía la dirección de dicho usuario al usuario_1, posteriormente el usaurio_1 envía de nuevo una solicitud *INVITE* directamente al usuario_2.

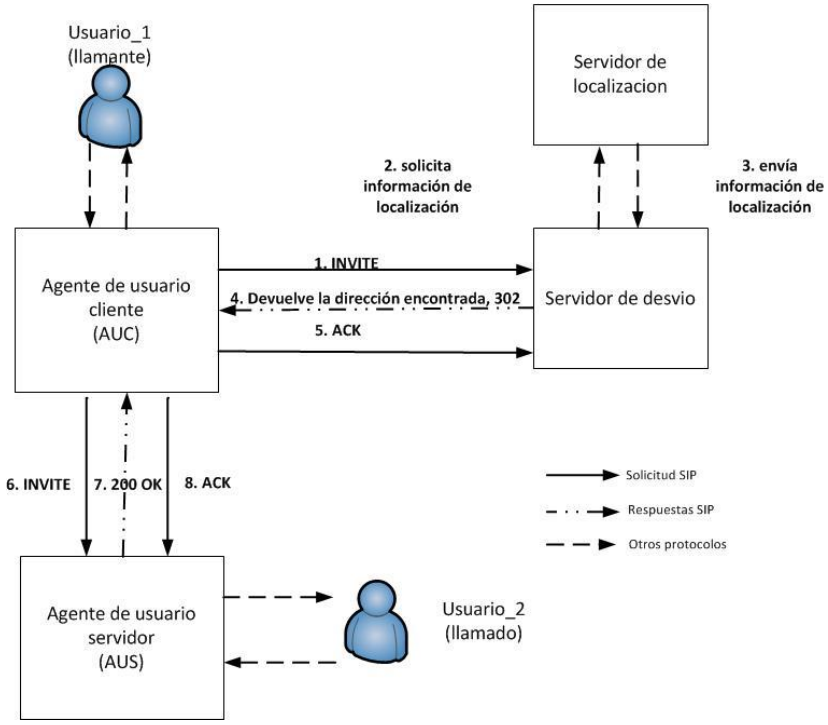


Figura de Apéndice 1.3 Procedimiento de invitación SIP utilizando un servidor de desvió.

Apéndice 2

1. Notación Abstracta ASN.1 para H.323

Lenguaje sintético abstracto (Abstract Syntax Notation 1), la definición de la ITU es que ANS.1 es una notación formal utilizado para describir los datos trasmitidos por los protocolos de telecomunicaciones, independientemente del lenguaje y la representación física de los datos, ya sea compleja y o simple la aplicación. Es capaz de enviar información de cualquier tipo, ya sea vídeo, datos, audio, etc.

Las estructuras de datos y constructores definidos por ASN.1 que es junto con las reglas de codificación como BER y PER, definen la forma en que se pueden traducir a bytes en condiciones necesarias para poder transmitirlas directamente por un medio de transporte. ASN.1 es utilizado para representar los mensajes RAS que utiliza H.323

2. H.225.0 (Señalización de control de llamada)

EL protocolo H.225.0 tiene dos partes:

Por un lado la recomendación Q.931 de la ITU-T, las especificaciones de la capa 3 ISDN, esta recomendación sirve para establecer y determinar las conexiones entre los puntos finales de H.323, esta señalización es llamada Call signalling o Q.931 signalling.

La segunda parte consiste en la señalización RAS que trabaja con los puntos finales y los *gatekeeper*s, pues permite al *gatekeeper* administra los puntos finales de su zona. La señalización RAS es usada para registrar a un punto final en el *gatekeeper* y que éste acepte o deniegue el acceso a la red. EL protocolo RAS se describe con mayor detalle más adelante.

3. Q.931 (Digital Subscriber Signalling)

El canal de señalización de llamada se basa en Q.931, el cual sirve para establecer la primera conexión entre dos terminales. Las llamadas son enviadas sobre TCP por el puerto 1720, en este puerto se iniciar los mensajes de control de llamada Q.932 entre dos terminales para la conexión, mantenimiento y desconexión.

Los mensajes más comunes son:

- Setup: Mensaje enviado con el fin de realizar una llamada H.323, por lo cual sirve para establecer conexión con una entidad H.323, el contenido del mensaje tiene las IP´s y puerto del llamante y del llamado.
- Alerting: Indica la fase de generación de tono.

- Connect: Indica el inicio de la conexión.

- Release Complete. Mensaje enviado por la terminal para la desconexión.
- Facility: Mensaje de la norma Q.932 que sirve como petición o reconocimiento de un servicio adicional.

4. RAS (Registration, Admission and Status)

EL protocolo de Registro, Admisión y Estado, es el encargado establecer un canal para las comunicaciones entre las terminales y el *gatekeeper*. El registra y admite a los participantes en el establecimiento de la comunicación, así como de controlar el ancho de banda, estado y desconexión de los participantes dentro de la conexión. También guarda la información del estado de cada terminal que pertenece a su zona.

El protocolo RAS es encargado de definir las comunicaciones entre cada terminal y su respectivo *gatekeeper*, en cada zona.

Los procedimientos de RAS son:

- Descubrimiento del *Gatekeeper*.
- Registro de la terminal en el *Gatekeeper*
- Solicitud de localización
- Admisión de una terminal en el *Gatekeeper*
- Liberación de procesos después de terminar una llamada.
- Solicitud de intercambio de requerimientos entre la terminal y el *gatekeeper*
- Solicitud de estado a la terminal
- Disponibilidad de recursos del *Gatekeeper*.
-

5. H.245 Control protocol for multimedia communication

Protocolo de control de llamada para comunicaciones multimedia que utilizan H.323, la señalización de H.245 debe realizarse al mismo tiempo que la señalización de H.225.0 y de preferencia antes que el mensaje “Connect” ya que podría perderse algunos datos trasmitidos.

En si H.245 es un conjunto de mensajes ASN.1 usados para el establecimiento y control de una llamada.

- Intercambio de capacidades de las terminales: Las terminales definen los códecs (audio y/o vídeo) con los que disponen asiéndole saber a la otra terminal de comunicación. Teniendo en cuanta los siguientes aspectos:

- El formato multimedia a utilizar (G.711, G.723, H.261 o T.120).
- Número máximo de muestras de audio por paquete.
- Si se admitirá supresión de silencio.

- Determina quién es el maestro y el esclavo en la comunicación.

- Abre y cierra canales lógicos: Se encarga de crear dos canales lógicos por cada comunicación que se desee establecer, pues los canales de audio y vídeo son punto a punto y unidireccionales.

- Controlar el flujo en caso de que ocurriera algún problema.

- Control y composición de la señal de canal lógico.

Un canal lógico se entiende como un camino o bien una conexión por donde se transmitirá información entre dos terminales. Así que todos los mensajes H.245 se transmiten por un canal especial, llamado canal de control H.245. Para abrir este canal necesita una conexión TCP separada. Cuando se utiliza UDP es obligatorio que sea tunelizado dentro de los mensajes H.225.0 para facilitar su paso a través de los cortafuegos.

H.245 dispone de cuatro tipos de mensajes.

- Request
- Response
- Command
- Indication

Los primero mensajes que viajan por el canal H.245 son de tipo Terminal Capability Set (TCS), en el cual se describen los códecs y capacidades multimedia que puede soportar el canal y que capacidades puede soportar al mismo tiempo. Cada una de las capacidades tiene asociado un número que se describen en las tablas de capacidades especificadas en este protocolo. Y tiene la capacidad de describir nuevas capacidades no descritas en las especificaciones.

Este protocolo es muy complejo y resulta difícil explicarlo todo, pero cabe mencionar que cuenta con 53 mensajes más otros 15 mensajes de repuestas.

En cuanto a la señalización de canal lógico, los canales se pueden abrir cuando se intercambian mensajes de tipo “OpenLogical Channel“(OLC), el cual contiene las capacidades que informa la terminal. Cada terminal envía un OCL, para una comunicación asimétrica en códecs, a cada OCL se le asigna un *SessionID* o sea un identificado de sesión, como la más alta prioridad de 1 se le asigna al audio, 2 al vídeo y 3 a datos. Para cada *SessionID* se abre una sesión RTP/RTCP.

EL canal H.245 se cierra cuando la terminal cierre todos los canales lógicos y esperar los *ACK*s respectivos, y así podrá enviar el mensaje *endSession* y esperar su *ACK*, para cerrar el canal.

Apéndice 3

Configuraciones CISCO

Router 1

```
Building configuration...
Current configuration : 1419 bytes
!
version 12.4
no service timestamps debug uptime
no service timestamps log uptime
no service password-encryption
!
hostname R1
!
boot-start-marker
boot-end-marker
!
enable secret 5 $1$xRE/$Gu9V7PYrlAfaT9y.MNyiz0
!
no aaa new-model
!
ip cef
no ip dhcp use vrf connected
ip dhcp excluded-address 172.16.192.1
!
ip dhcp pool LAN1
   network 172.16.192.0 255.255.248.0
   default-router 172.16.192.1
   option 150 ip 172.16.192.1
!
no ip domain lookup
!
multilink bundle-name authenticated
!
voice-card 0
 no dspfarm
!
username R2 password 0 cisco            dress
archive
 log config
  hidekeys
```

```
!
interface FastEthernet0/0
 ip address 172.16.192.1 255.255.248.0
 duplex auto
 speed auto
!
interface FastEthernet0/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface Serial0/3/0
 bandwidth 2048
 ip address 192.168.1.226 255.255.255.252
!
interface Serial0/3/1
 ip address 192.168.1.233 255.255.255.252
!
router eigrp 1
 passive-interface FastEthernet0/0
 network 172.16.0.0
 network 192.168.1.224 0.0.0.3
 network 192.168.1.232 0.0.0.3
 no auto-summary
!
!
ip http server
no ip http secure-server
!
!
control-plane
!
line con 0
 password cisco
 login
line aux 0
line vty 0 4
 password cisco
 login
!
scheduler allocate 20000 1000
!
webvpn cef
!
end
```

Router 2

```
Building configuration...
Current configuration : 1667 bytes
!
version 12.4
no service timestamps debug uptime
no service timestamps log uptime
no service password-encryption
!
hostname R2
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
enable secret 5 $1$VCqg$xcUWnqh4U9y1VyUTobstK.
!
no aaa new-model
!
dot11 syslog
ip source-route
!
!
ip cef
ip dhcp excluded-address 172.16.200.1
!
ip dhcp pool LAN2
   network 172.16.200.0 255.255.248.0
   default-router 172.16.200.1
   option 150 ip 172.16.200.1
!
!
no ip domain lookup
no ipv6 cef
!
multilink bundle-name authenticated
voice-card 0
!
archive
 log config
  hidekeys
!
interface FastEthernet0/0
 ip address 172.16.200.1 255.255.248.0
 duplex auto
```

```
 speed auto
!
interface FastEthernet0/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface FastEthernet0/2/0
 shutdown
!
interface FastEthernet0/2/1
 shutdown
!
interface FastEthernet0/2/2
 shutdown
!
interface FastEthernet0/2/3
 shutdown
!
interface Serial0/3/0
 ip address 192.168.1.230 255.255.255.252
 no fair-queue
 clock rate 8000000
!
interface Serial0/3/1
 ip address 192.168.1.234 255.255.255.252
 clock rate 8000000
!
interface Vlan1
 no ip address
 shutdown
!
router eigrp 1
 passive-interface FastEthernet0/0
 network 172.16.0.0
 network 192.168.1.228 0.0.0.3
 network 192.168.1.232 0.0.0.3
 no auto-summary
!
ip forward-protocol nd
ip http server
no ip http secure-server
!
control-plane
!
ccm-manager fax protocol cisco
```

```
!
mgcp fax t38 ecm
!
line con 0
 password cisco
 login
line aux 0
line vty 0 4
 password cisco
 login
line vty 5
 login
!
scheduler allocate 20000 1000
end
```

Router 3

```
Building configuration...

Current configuration : 1793 bytes
!
version 12.4
no service timestamps debug uptime
no service timestamps log uptime
no service password-encryption
!
hostname R3
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
enable secret 5 $1$PmB1$hdis1gwnK8zIMpYnEWtbS/
!
no aaa new-model
!
dot11 syslog
ip source-route
!
!
ip cef
no ip dhcp use vrf connected
ip dhcp excluded-address 172.16.192.1
ip dhcp excluded-address 172.16.208.1
```

```
!
ip dhcp pool LAN1
  network 172.16.192.0 255.255.248.0
  default-router 172.16.192.1
  option 150 ip 172.16.192.1
!
ip dhcp pool LAN3
  network 172.16.208.0 255.255.248.0
  default-router 172.16.208.1
  option 150 ip 172.16.208.1
!
!
no ip domain lookup
no ipv6 cef
!
multilink bundle-name authenticated

!
voice-card 0
!
username R1 password 0 cisco
username R2 password 0 cisco
archive
 log config
  hidekeys
!
interface FastEthernet0/0
 ip address 172.16.208.1 255.255.248.0
 duplex auto
 speed auto
!
interface FastEthernet0/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface Serial0/3/0
 bandwidth 2048
 ip address 192.168.1.229 255.255.255.252
!
interface Serial0/3/1
 bandwidth 2048
 ip address 192.168.1.225 255.255.255.252
 clock rate 8000000
!
router eigrp 1
```

```
 passive-interface FastEthernet0/0
 network 172.16.0.0
 network 192.168.1.224 0.0.0.3
 network 192.168.1.228 0.0.0.3
 no auto-summary
!
ip forward-protocol nd
ip http server
no ip http secure-server
!
control-plane
!
ccm-manager fax protocol cisco
!
mgcp fax t38 ecm
!
line con 0
 password cisco
 login
line aux 0
line vty 0 4
 password cisco
 login
!
scheduler allocate 20000 1000
end
```

Apéndice 4

Pasos para la instalación de Asterisk

1. Instalar las librerías y dependencias utilizadas por Asterisk.

- Abrir una terminal en Ubuntu y escribir los siguientes comandos

```
apt-get update && apt-get upgrade
apt-get install build-essential wget libssl-dev libncurses5-dev libnewt-dev libxml2-dev linux-headers-$(uname -r) libsqlite3-dev uuid-dev
```

2. Descargar los archivos correspondientes para instalar Asterisk en el siguiente directorio.

- Entramos al directorio /usr/src/ y descargamos los archivos.

```
cd /usr/src/
wget http://downloads.asterisk.org/pub/telephony/libpri/libpri-1.4.14.tar.gz
wget http://downloads.asterisk.org/pub/telephony/asterisk/releases/asterisk-11.5.0-rc1.tar.gz
wget http://downloads.asterisk.org/pub/telephony/asterisk/releases/asterisk-sounds-1.2.1.tar.gz
wget http://downloads.asterisk.org/pub/telephony/asterisk/releases/asterisk-addons-1.6.2.4.tar.gz
```

3. Descomprimir cada uno de los archivos dentro del mismo directorio que se descargaron.

```
tar -xzf libpri-1.4.14.tar.gz
tar -xzf asterisk-11.5.0-rc1.tar.gz
tar -xzf asterisk-sounds-1.2.1.tar.gz
tar -xzf asterisk-addons-1.6.2.4.tar.gz
```

4. Crear las configuraciones de libpri e instarlo.

- Entramos al directorio libpri-1.4.14 con el siguiente comando.

```
cd libpri-1.4.14
```

- Dentro del directorio tecleamos los siguientes comandos

```
make
make install
cd ..
```

5. Entrar al directorio Asterisk y crear e instalar los archivos.

- Entramos al directorio asterisk-11.5.0-rc.

```
cd
```

- Tecleamos los siguientes comandos

```
./configure
make menuconfig
make
make install
make samples
make config
cd ..
```

6. Instalar los sonidos que provee Asterisk

```
cd asterisk-sounds-1.2.1
make install
cd ..
```

7. En este archivo se encuentra el modulo H.323 para poder realizar llamadas con este protocolo.

```
cd asterisk-addons-1.6.2.4
./configure
make
make install
cd ..
service asterisk start
```

8. Instalar AsteriskGUI.

```
wget http://downloads.asterisk.org/pub/telephony/asterisk-gui/asterisk-gui-2.1.0-rc1.tar.gz

tar -xzf asterisk-gui-2.1.0-rc1.tar.gz
cd asterisk-gui-2.1.0-rc1
./configure
make
make install
```

Pasos para la instalación de SIPp

Descargamos SIPp y procedemos a descomprimir y realizar la instalación, en este caso el archivo se descargó en la carpeta Descargas y desde allí mismo se realizó la instalación con los siguientes comandos.

```
#wget -m -nd http://downloads.sourceforge.net/pro...3.3.src.tar.gz
# tar -xvzf sipp-3.3.tar.gz
# cd sipp
# ./configure --with-sctp --with-pcap –with-openssl
# make
#cd sipp-3.3
```