# Universidad de las Ciencias Informáticas


# “Procedimiento para el análisis de los protocolos SIP y H.323, usados por los servicios de Voz sobre IP (VoIP).”

**Trabajo de Diploma para optar por el título de**
**Ingeniero en Ciencias Informáticas**

**Autor(es):** “Raúl Concepción González”
“José Antonio Bustio Encinoso”
**Tutor(es):** Ing. Adrián Yera Pérez
**Co-tutor:** MsC. Ing. Omar Yera Pérez
**Asesor:** MsC. Pedro Carlos Pérez Martinto

**Junio 2009**
**Ciudad de La Habana**
**“Año del 50 Aniversario del Triunfo de la Revolución”**

*En el verdadero éxito, la suerte no tiene nada que ver; la suerte es para los improvisados y aprovechados; y el éxito es el resultado obligado de la constancia, de la responsabilidad, del esfuerzo, de la organización y del equilibrio entre la razón y el corazón.*

**DATOS DE CONTACTO**

Tutor: Ing. Adrian Yera Pérez**.**
Graduado en 1990 en el ISPJAE como Ingeniero Eléctrico en la especialidad Máquinas Computadoras. Ha trabajado en temas relacionados con las redes IP y el desarrollo de aplicaciones y sistemas durante más de 15 años.

Co-tutor: MsC .Ing. Omar Yera Pérez.
Graduado en 1984 como Ingeniero en Máquinas Computadoras. Máster en Ciencias. Ha trabajado en temas relacionados con las redes IP y el desarrollo de aplicaciones y sistemas durante más de 20 años.

Asesor: MsC. Pedro Carlos Pérez Martinto.
Graduado en 1990 en el ISPEJV como Defectólogo. Ha estado vinculado a los proyectos de Tecnología Educativa del MINED. Trabaja líneas de investigaciones en la informática aplicada a procesos de rehabilitación en menores discapacitados visuales. Actualmente asesor de investigación de la facultad 5 y profesor principal de Metodología de la Investigación Científica en la UCI. Posee 19 años de experiencias en labores de formación profesional a diferentes niveles educacionales.

**AGRADECIMIENTOS**

*A nuestros padres y familiares que nos dieron el apoyo moral más grande que existe; este fue fundamental para todo el desarrollo de la tesis.*

*A nuestro tutor Adrián Yera Pérez y a su hermano Omar Yera Pérez guías y fundamento inteligente de esta investigación. Por enseñarnos, transmitirnos sus conocimientos acerca de nuestro trabajo investigativo y por su paciencia que fue fundamental para la realización del mismo.*

*A nuestro asesor Pedro Carlos Pérez Martinto por su apoyo y sus consejos que sirvieron de experiencia para la culminación del trabajo.*

*A los especialistas que nos atendieron y apoyaron en algunas de las decisiones importantes.*

*A los amigos que de una forma u otra nos dieron su aporte, amor y cariño y que en momentos de angustias fueron más que amigos, hermanos y padres.*

*Agradecimiento especial a nuestras novias por ser tan pacientes, darnos su apoyo fundamental, el amor que cada día nos hacía sentir bien y nos ponía la autoestima por encima de todo.*

*A nuestra otra madre la Revolución por darnos la oportunidad de estudiar en una Universidad de Excelencia y convertirnos en profesionales que la defenderán siempre.*

*A todos Muchas Gracias.*

## DEDICATORIA

*Al Comandante en Jefe, líder indiscutible de la obra que hoy todos llevamos adelante y padre de la idea de crear la Universidad de Excelencia en la que me he formado como profesional revolucionario.*

*A mis padres por estar siempre a mi lado, por la educación que me han dado, guiándome por el camino correcto, apoyándome en todo y comprendiendo cada decisión mía. La verdad es que sin sus consejos siempre acertados, hoy no fuera lo que soy.*

*Otro agradecimiento especial para mi hermano, por todo el cariño que me profesa, ayudándome también a salir adelante.*

*A mi abuelo Ramón, por toda la paciencia, dedicación y preocupación que tuvo para conmigo en los primeros años de mi vida, siendo más que un abuelo.*

*A mis abuelas Rosa y Julia, por la preocupación que siempre han mostrado por mi y por el amor que me brindan.*

*A toda mi familia en general, por estar siempre pendiente de mí.*

*A mi novia, por su compañía y sacrificio en todo este tiempo de desarrollo de la Tesis.*

*Al tutor, Ing. Adrian Yera Pérez y a su hermano MsC. Ing. Omar Yera Pérez, por el tiempo dedicado a atendernos y por sus críticas, las que permitieron que la Tesis tenga la calidad requerida.*

*A mi compañero de tesis Tony, que es casi un hermano para mí.*

*Al Equipo de Tesis, por todos los momentos buenos y malos, tristes y felices que hemos pasada juntos.*

*A todos mis amigos por la ayuda que de una forma u otra me brindaron.*

*En general a todos aquellos que estuvieron pendientes del desarrollo de mi Tesis, que en algún momento preguntaron: ¿Cómo va la Tesis?*

***Raúl Concepción González***

**DEDICATORIA**

*Quisiera dedicar mi tesis a cuatro personas fundamentales que son los pilares que me han encaminado en la vida: la primera mi madre Marlen, por ser una sola en el mundo y una de las personas por la cual vivo; a mi padre Zenén, por enseñarme casi todo lo que sé de la vida, por ser padre y amigo a la vez y por dedicarle tiempo a mis cosas personales; y a una personita especial, a mi abuelo, que con la cantidad de años que tiene todavía no tiene canas en el pelo y lucha cada día para que me haga un hombre correcto. Dedicarle mi tesis a los demás familiares que se preocuparon por cómo me iba. Si tuviera que elegir mi familia no cambiaria la que tengo. Quisiera dedicarle mi tesis a una cuarta persona que aunque no está con nosotros es la que más quiero en el mundo Berta Acosta, ella es la mujer que todos los días vela por mi para que todo me salga bien. Dedico mi tesis a mis tutores porque son las personas que nos atendieron día a día y con sus críticas nos hicieron hombres fuertes capases de lograr cualquier meta trazada .A los amigos que cada día me preguntaban cómo estaba la tesis; son muchos solo mencionare a mi compañero de tesis Raúl Concepción González y a mi mejor amigo Jorge Félix Duque, son dos hermanos importantes para mí. Dedico además mi trabajo al piquete del modulo del MININT, al equipo de trabajo, a nenita, a Sander y a todos los dirigentes que nos ayudaron de una forma u otra. Dejo de última a Yudelkis Caridad Orta por ser una personita especial en mi vida, fue la persona que cada día me levanto el ánimo y me dio amor de pareja un amor que no tendrá remplazo nunca. Se me quedan muchas persona por agradecer, pero sí de agradecimientos se trata necesitamos una tesis para agradecer a todas las personas que hicieron posible que este trabajo se realizara. De todo corazón a mis amigos y enemigos les dedico personalmente mi tesis*

***Jose A.Bustio Encinoso.***

**RESUMEN**

El vertiginoso desarrollo que está teniendo lugar a nivel mundial en la informática y las comunicaciones y en especial la necesidad cada vez creciente de comunicarse por redes *IP* mediante voz (*VoIP*), fundamentan la existencia del presente trabajo.

En la actualidad no existe en nuestro ámbito una herramienta que tenga recogida toda la información posible sobre este tema, de ahí la necesidad de realizar esta investigación, la cual se centra en la descripción de un procedimiento para el procesamiento de la familia de protocolos *H.323* y *SIP* utilizados en la comunicación de voz sobre las redes *IP* (*VoIP*), utilizando el modelo *TCP/IP*.

El resultado de esta investigación permitió contar con un procedimiento detallado que permite el desarrollo de herramientas y aplicaciones propias e idóneas para el procesamiento en línea de los protocolos *H.323* y *SIP*, que son los protocolos más comúnmente utilizados en las comunicaciones de *VoIP* en redes atendidas por el MININT.

PALABRAS CLAVE:

Tecnologías, Protocolos, Comunicación, *VoIP*, Procedimiento.

# TABLA DE CONTENIDOS

## INTRODUCCIÓN

La constante evolución de los medios informáticos, hace imprescindible dotar a los futuros profesionales y a todo el pueblo de los conocimientos para el manejo y utilización de las herramientas de computación que les permitan a los usuarios tomar decisiones importantes y lograr un mayor aprovechamiento del capital que se invierte actualmente en el país. Por estas razones el estado cubano, la Universidad de las Ciencias Informáticas (UCI) junto al Ministerio del Interior llevan a cabo el proceso de informatización de la sociedad, lo cual eleva además el nivel de vida del pueblo.

Hay que destacar que el uso de las nuevas tecnologías de la informática y las comunicaciones (Tics) en la actualidad se ha incrementado. La creación de redes informáticas, sobre todo el desarrollo de Internet, no es más que un medio de comunicación, de interacción y de organización social. Internet es ya y será aun más el medio de comunicación y de relación esencial sobre el que se basa una nueva forma de sociedad que ya vivimos, lo que se llama la sociedad red.

La evolución del mundo de la comunicación universal por *Internet* ha crecido de manera explosiva y está cambiando su naturaleza. Hoy, a paso acelerado, las operaciones más complejas se hacen fuera de la computadora personal (*PC*), en grandes centros de cómputo a través de prestadores de servicios externos.

El crecimiento e implantación de las redes, el desarrollo de técnicas avanzadas de digitalización de voz, mecanismos de control y priorización de tráfico, protocolos de transmisión en tiempo real, así como el estudio de nuevos estándares que permiten la calidad de servicios en redes *IP*, han creado un entorno donde es posible transmitir telefonía sobre *IP*[1].

Para tener una idea más clara del desarrollo que han tenido las redes telefónicas, tenemos que ver las diferentes transiciones por las que han pasado:

- ✓ 1941 -Se introdujo el sistema de portadoras del tipo L1 (480 canales de voz, 1575 MHz)
- ✓ 1950 -Aparecen los sistemas de microondas

[1] IP: ***(Protocolo de Internet).*** Define la unidad de información enviada entre sistemas, que proporciona un servicio de entrega de paquetes básicos. Tomado de: www.bunam.unam.mx/portal/internet/c07glt01p01.html

- ✓ 1956 -Primer cable submarino transoceánico
- ✓ 1962 -Sistema de portadores digital (T1, *PCM*)
- ✓ 1964 -Conmutación con elementos electromecánicos
- ✓ 1964 -Se introdujo el control por programa almacenado (*SPC*)
- ✓ 1976 -Introducción de sistemas de conmutación digital
- ✓ 1985 -989 -*ISDN*
- ✓ 1991 -*BISDN*
- ✓ 1996 -*H.323*

Si a todo lo anterior se suma el enorme potencial de *Internet*, junto con el potencial de ahorro que este tipo de tecnología acarrea, la conclusión es clara, la tecnología de voz sobre *IP* (*VoIP*), es una fuente de recursos y avances tanto para empresas públicas como privadas.

La telefonía sobre *IP* abrió un espacio importante dentro del universo de las *intranets*, *extranets* e *Internet*. Es la posibilidad de ser comunicados a bajos costos dentro de las empresas o fueras de ellas, la puerta de varios servicios apenas imaginados, la forma de combinar una página *web* con la atención en vivo desde un *call center*. El concepto original es simple, se trata de transformar la voz en paquetes de información manejables por una red *IP.* Gracias a la aparición de algunos protocolos es posible reservar cierto ancho de banda para lograr una buena comunicación.

En los últimos años, los protocolos de señalización para el servicio de transmisión de voz, han experimentado una fuerte evolución junto con la tendencia a trasportar dicho tráfico desde las redes de conmutación de circuitos hacia las redes de conmutación de paquetes. Esta tendencia queda reflejada en el desarrollo de estándares en este ámbito y la aparición de productos en el mercado que cubren las necesidades de operadores, grandes empresas y *PYMES*[2]*, lo cual* se verá incrementada durante los próximos 5 años debido a la evolución de las redes móviles basadas en tecnología *UMTS*[3] hacia entornos "*All-IP*".

[2] Acrónimo de Pequeña y Mediana Empresa, Tomado de www.terra.es/personal/antcobo/glosario.htm
[3] Servicio diseñado para permitir transferencias entre dispositivos móviles a velocidades de hasta 2 Mbps Tomado de : www.nachocabanes.com/diccio/ndic.php

En la actualidad las aplicaciones de voz y video están convirtiéndose en herramientas claves para la comunicación entre personas y el Ministerio del Interior actualiza las bases de su desarrollo en esta tecnología.

Existe una gama muy amplia de sistemas de información desarrollados para satisfacer las necesidades de la mayoría de las empresas y organismos. El Ministerio del Interior (MININT) no está ajeno a estos cambios y se integra al proceso de informatización en el cual la creación de la Universidad de las Ciencias Informáticas juega un papel fundamental, especialmente los cadetes insertados del MININT, que son los encargados de desarrollar numerosos proyectos para facilitar el trabajo de dicha institución.

Como consecuencia del aumento acelerado de la utilización de comunicaciones de voz basadas en la tecnología brindada por las redes de conmutación de paquetes y en particular las redes *IP*, surge como:

**Problema a resolver:**

¿Cómo desarrollar un procedimiento para realizar el análisis y procesamiento de los principales protocolos de voz sobre redes *IP* atendidas por el MININT?

**Objeto de estudio:**

Transferencia de datos por las redes *IP* utilizando la familia de protocolos de *VoIP H.323* y *SIP*.

**Objetivo:**

Definir un procedimiento para el procesamiento de los protocolos *H.323* y *SIP*, empleados en las comunicaciones de voz sobre redes *IP* (*VoIP*) mediante la utilización del modelo *TCP/IP*.

**Campo de acción:**

Captura y procesamiento de los datos que circulan por las redes *IP* de los protocolos *H.323* y *SIP* en las redes atendidas por el MININT.

**Idea a defender:**

Un procedimiento para la captura y procesamiento de los protocolos *H.323* y *SIP* utilizando como base el modelo *TCP/IP,* permitirá aplicar herramientas de procesamiento y análisis propias que garantizarán un mejor trabajo del MININT sobre las redes *IP* que atiende.

**Posible resultado:**

Se podrá contar con la descripción detallada de un procedimiento para el procesamiento de los protocolos *H.323* y *SIP*, empleados en las comunicaciones de voz sobre redes *IP* (*VoIP*), en las redes atendidas por el MININT.

Se realizaron las siguientes **tareas de investigación**, para dar cumplimiento al objetivo propuesto:

1. Estudio de las bases teóricas y estado del arte de las tecnologías destinadas a las comunicaciones de voz sobre redes *IP*.
2. Estudio del funcionamiento de la familia de protocolos *SIP* y *H.323*.
3. Realización de pruebas de laboratorio relacionadas con la familia de protocolos *SIP* y *H.323.*

Para dar cumplimiento a las tareas se emplearon los siguientes **métodos científicos:**

**Métodos Teóricos**

- **Histórico – Lógico**: Permitió estudiar la trayectoria histórica de los protocolos *H.323* y *SIP,* su desarrollo, así como sus características.
- **Analítico – Sintético:** Para llegar a conclusiones en la investigación a partir de la información que se procesó de los protocolos *H.323* y *SIP* y precisar características de ambos protocolos.
- **Inducción – Deducción**: Para definir criterios propios a partir de ideas generales encontradas en las distintas fases de la investigación.
- **Modelación**: Con vista a crear, teóricamente, un procedimiento informático que posibilite el procesamiento de los protocolos empleados en las comunicaciones de voz en redes *IP H.323* y *SIP*.

**Métodos Empíricos**

- **Observación:** Mediante el uso de herramientas que permitieron la observación y el análisis del funcionamiento de los protocolos *H.323* y *SIP* en las redes, la relación e integración con otros protocolos que intervienen en el transporte de los datos, así como el análisis de la información que aporta cada protocolo.

El trabajo de diploma consta de 3 capítulos, conclusiones, recomendaciones, bibliografía y anexos. La información en el documento esta dividida de la siguiente forma:

**Capítulo 1:** Fundamentación Teórica, se muestra el resultado de la investigación bibliográfica sobre el objeto de estudio y lo referente a los antecedentes de protocolos, así como las tendencias que adquieren actualmente. Se detallan las tecnologías, analizando sus características, ventajas y desventajas.

**Capítulo 2:** Describe el procedimiento creado para la captura y procesamiento de los protocolos *H.323* y *SIP*, su funcionamiento, así como los protocolos de la familia *TCP/IP* relacionados con ellos.

**Capítulo 3:** Contiene las pruebas de laboratorio realizadas con las familias de protocolos *SIP, H.323* y protocolos *TCP/IP* relacionados con ellos, así como las herramientas utilizadas para realizar tales pruebas.

# CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

## 1.1 Introducción

En el presente capítulo se desarrolla la fundamentación teórica de la investigación, tiene como objetivo abordar los diferentes elementos que brindan la base teórico conceptual para el desarrollo de este trabajo, los principales conceptos y definiciones, la evolución histórica de los protocolos de voz sobre redes *IP* (*VoIP, por sus siglas en inglés*). Se exponen además los antecedentes y desarrollo de protocolos de redes conmutadas por paquetes. Se describe una serie de aspectos teóricos y características de dichos protocolos además de su integración a sistemas informáticos. Además se realiza un análisis de las tecnologías sobre las cuales se lleva a cabo el presente trabajo.

## 1.2 Tecnología de voz sobre *IP* (*VoIP*)

En la década de los 90, irrumpe en el escenario de las telecomunicaciones mundiales la Telefonía *IP*, como importante protagonista que apunta hacia la convergencia de medios (voz, datos, vídeo) y para la transformación requerida en las redes de telecomunicaciones actuales.
Aunque no se ha llegado a un consenso sobre la definición exacta del término Telefonía *IP*, a los efectos del presente trabajo, se utilizará "como un término genérico para la prestación de servicios vocales, facsímil y servicios conexos, parcial o totalmente por redes basadas en *IP* con conmutación de paquetes. La Telefonía *IP* también puede incluir aplicaciones que integren/incorporen la transmisión de señales vocales y facsímil con otros medios tales como textos e imágenes". Esta es la definición utilizada en el Informe del Secretario General de la Unión Internacional de Telecomunicaciones (*UIT*, *por sus siglas en inglés*) en el Foro Mundial de Políticas de las Telecomunicaciones, efectuado en Marzo del 2001 en Ginebra [1].
En otro documento posterior de la *UIT* [2], en una definición de trabajo señalan que la Telefonía IP se refiere a la transferencia de Voz sobre el Protocolo Internet (*VoIP*), por lo cual en ocasiones los términos Telefonía *IP* y *VoIP* son indistintamente utilizados.

Las redes han evolucionado con respecto a la voz *IP* [3], el ***Anexo*1** en la ***figura1*** muestra como pasa de una red tradicional a una red telefónica, también se muestran los componentes de dichas redes.

### 1.3 Sistemas de *VoIP*

Esta tecnología se conoce también como telefonía por *Internet*. Es un método de digitalización de la voz, encapsulada en paquetes que son enviados a través de una red de conmutación de paquetes *IP* compuesta por una colección de tecnologías o dispositivos. Además de proveer de forma mejorada los servicios de comunicaciones de voz actuales, extiende las capacidades de red hacia nuevas aplicaciones de voz, datos, video y presenta convergencia.

### 1.4 Ventajas de la Tecnología de Voz sobre *IP* (*VoIP*)

La telefonía *IP* es vista por muchos como una tecnología efectiva, mientras los que han tenido experiencias desfavorables en el pasado, vinculadas con la realización de llamadas telefónicas por Internet, la consideran un obstáculo. Sin embargo, *VoIP* es de sumo interés para la industria de las comunicaciones. A continuación se enumeran algunas de las ventajas que la potencian.

- Integración sobre su intranet de la voz como un servicio más de su red, tal como otros servicios informáticos. La voz sobre *IP* está cambiando el paradigma de acceso a la información, fusionando voz, datos, facsímil y funciones multimedia en una sola infraestructura de acceso convergente.
- Eficiencia en la utilización del ancho de banda. En la Telefonía *IP* se incrementa la eficiencia en la utilización del ancho de banda para la transmisión en tiempo real de voz en un factor de 6 veces o más.
- Con relación a las tarifas los clientes de negocios obtienen beneficios al saltar las redes telefónicas públicas conmutadas y utilizar como *backbone*[4] su intranet. En el caso de

[4] Estructura de transmisión de datos de una red o conjunto de ellas en Internet. Literalmente: "columna vertebral" Tomado de: www.angelfire.com/biz/HUMBERTOLOTNAVARRO/glosario.html

clientes particulares igualmente resultan beneficiados. En estos casos se evaden los costos de las llamadas de larga distancia*,* así como las llamadas internacionales.

- La Telefonía sobre *IP* ha captado la atención de los proveedores de servicios en todo el mundo, ofreciendo una amplia gama de estosque ha reducido al mismo tiempo sus costos de infraestructura.
- Interoperabilidad de diversos proveedores, a partir de un constante y creciente trabajo en el desarrollo de estándares (*H.323, SIP, MGCP y otros*)
- Despliegue progresivo. La Telefonía *IP* puede utilizarse en unión de la *PSTN*[5]*,* líneas arrendadas y conmutadas, *PBXs*[6] y otros equipos de abonados, *LANs*, y conexiones de Internet.

Estas son algunas de las ventajas de la *VoIP.* En diversos foros hay quienes botan a favor y en contra de esta tecnología, pero nuestro trabajo muestra características de la *VoIP* que evidencian su importancia para el desarrollo de las redes, además de ayudarnos en el ahorro del capital monetario que se dispone.

Pero como toda tecnología, *VoIP* no es perfecta, tiene desventajas y algunos factores que pueden fallar, o sea cosas que atentan contra su funcionamiento. El primer factor es la calidad de la red, el segundo la interacción con otras redes, el transporte de *fax* y *modem*[7], costos ocultos y cómo elegir la tecnología que usaremos. Estos son algunos de los factores que pueden influir en el mal uso o implementación de la *VoIP*.

[5] Es la sigla para "Public Switched Telephone Network", que es el servicio de telefonía tradicional provisto por las compañías telefónicas. Tomado de: proyectopica.wordpress.com/glosario/

[6] Comúnmente llamado conmutador, es el sistema de intercambio de líneas telefónicas. Tomado de: www.clevertech.com.mx/glosario4.htm

7 Dispositivo que permite transmitir datos digitales a través de dispositivos de transmisión analógicos, como las líneas telefónicas utilizadas para la transmisión de información a larga distancia. La vía de transmisión puede consistir en un cable largo o en una conexión telefónica. El módem contiene un modulador (para enviar datos) y un demodulador (para recibirlos). Tomado de: http://www.bunam.unam.mx/portal/internet/c07glt01p01.html

### 1.5 ¿Qué es la Telefonía *IP*?

La Telefonía *IP* es un servicio de voz que, a diferencia de la telefonía convencional de la Red Telefónica Pública Conmutada (*PSTN*, por sus siglas en inglés), realiza el transporte sobre redes de conmutación de paquetes.

En las redes de conmutación de paquetes la información de audio, datos o vídeo se digitaliza y divide en pequeños paquetes de información, que poseen una dirección o encabezado de datos que los identifican. Estos son temporizados con precisión, de tal forma que no se pierda ningún paquete en la transmisión, lleguen a tiempo a su destino y en el orden correcto.

La Telefonía *IP* abarca todas las aplicaciones que, basadas en computadoras personales y otros terminales como teléfono y fax, enlazan a dos usuarios de una misma red o en redes distintas. Una situación particular, es cuando el intercambio de paquetes de voz sobre *IP* se produce por Internet, en cuyo caso se habla de Telefonía por *Internet.* Aquí los paquetes pueden sufrir demoras, ser afectados por el *jitter*[8], perderse o retrasarse, ya que están expuesto al problema de la congestión que provoca la disminución de la calidad en el transporte de señales de voz, en comparación con la telefonía tradicional, que se conduce mediante tecnologías orientadas a conexión en redes de conmutación de circuitos.

Para la materialización de la Telefonía *IP* se utilizan diversos tipos de terminales, que permiten agrupar los servicios brindados por los mismos, ver en el ***Anexo1 figura 2*** que muestra la arquitectura de los sistemas de telefonía *IP* según los grupos que se muestran en la ***Tabla 1.1.***

| Tecnología | Descripción | Barreras actuales |
|---|---|---|
| PC a PC | Primera generación | Calidad baja, llamadas limitadas a dueños de PCs en línea |
| PC a Teléfono/Fax | Segunda generación, que permite a los usuarios de PCs conectarse a teléfonos tradicionales o dispositivos de fax. | Calidad baja pero mejorando. Las llamadas únicamente pueden ser iniciadas por dueños de PCs. |
| Teléfono a teléfono | Tercera generación. Análogo al servicio telefónico tradicional. | Tecnología en constante mejoría. Precio y disponibilidad dependientes del proveedor local |

Tabla 1.1.Tipos de Telefonía IP.

[8] Distorsión de una señal producida por la variación inesperada de una de sus características (por ejemplo la fase)'Tomado de: www.dednet.net/institucion/itba/cursos/000183/demo/@glosario.html

## 1.6 Evaluación de los Sistemas basados en voz sobre *IP*

*VoIP* ha sido declarada por las grandes empresas como *Forbes.com* como “las cuatro letras para el crecimiento.” Resulta ampliamente sabido que la tecnología puede impulsar el crecimiento y los negocios, incluyendo a las *Pymes*.

La *VoIP* introduce eficiencia, eficacia, profesionalismo y una orientación muy clara hacia los clientes de parte de las *Pymes*, provocando la introducción de ventajas antes imposibles de obtener por su costo de difícil alcance para cualquier emprendimiento.

Como se muestra la *VoIP* va incrementando su auge en el mundo de las redes y da solución a problemas de costo de las inversiones, lo que no quiere decir que implementar el uso de redes en *IP* sea cosa fácil.

## 1.7 Actualidad

Actualmente se pueden encontrar tres tipos de redes *IP* [4] [5]:

- *Internet.* El estado actual de la red no permite un uso profesional para el tráfico de voz (se experimentan retardos no despreciables).
- *Red IP pública.* Los operadores ofrecen a las organizaciones la conectividad necesaria para sus redes de área local en lo que al tráfico *IP* se refiere. Se puede considerar como algo similar a *Internet*, pero con una mayor calidad de servicio y con importantes mejoras en seguridad. Hay operadores que ofrecen garantías de bajo retardo y/o ancho de banda, lo que los hace interesantes para el tráfico de voz.
- *Intranet.* La red *IP* implementada por la propia organización suele constar de varias redes *Lan* (*Ethernet conmutada, ATM*[9], etc.) que se interconectan mediante redes *WAN* tipo *Frame-relay*[10]*/ATM*, líneas punto a punto, *RDSI* para el acceso remoto, etc. En este caso

[9] ATM (Asynchronous Transfer Mode): Una tecnología de redes de alta velocidad que transmite múltiples tipos de información (voz, vídeo, datos) mediante la creación de "paquetes de datos”. Tomado de: http://www.alien-tech.com.ar/glosario.htm

[10] Frame relay: Sistema de transmisión basado en la conmutación de paquetes. Tomado de: http://www.uv.es/selva/Glossary/def.htm

la organización tiene bajo su control prácticamente todos los parámetros de la red, por lo que resulta ideal para su uso en el transporte de voz.

Actualmente se puede partir de una serie de elementos disponibles en el mercado que, según diferentes diseños, permiten construir las aplicaciones *VoIP*. Estos elementos son: teléfonos *IP*, adaptadores para *PC*, *hubs*[11]telefónicos, *gateways* (pasarelas *RTC/IP*, *RDSI/IP*), *gatekeeper*, unidades de audio conferencia múltiple (*MCU* Voz, por sus siglas en inglés), servicios de directorio, etc. El ***Anexo 1 figura 3*** muestra elementos de una red *VoIP*.

### 1.8 Proveedores de Tecnologías *VoIP*

Numerosos son los proveedores de tecnologías de *VoIP*. Algunos proceden del mercado de datos existente, como es el caso de *Cisco,* otros son tradicionales proveedores de tecnologías para la *RTPC* y muchos han desarrollado sus propias plataformas de redes públicas de datos, como *Alcatel*. Otros proveedores importantes son *Lucent*, *Nortel*, *Siemens* y *Ericsson*.

Dichos proveedores son responsables de las actuales transformaciones de las redes de circuitos hacia las redes de paquetes. Importantes empresas dan soluciones en este sentido como *VocalTec*, promotor de la telefonía *IP* en sus inicios junto a *Cisco*. También se ha producido una fusión muy importante entre antiguos proveedores de tecnologías de redes de datos y de equipamiento de *RTPC*, como es el caso de *Cisco* e *Italtel*, de cuya unión ha surgido la solución de red de paquetes multiservicios, que se está utilizando en Italia.

Otros proveedores de tecnologías debían ser estudiados, debido a sus soluciones atractivas como: *China Telecom Corporation*, *SIEMENS* con la plataforma *SURPASS* [2] y el Proveedor *CIRPACK* [7].

[11] **Hubs (concentradores):** dispositivo que centraliza la conexión de los cables procedentes de la estaciones de trabajo. Existen dos tipos de concentradores: pasivos y activos. Los concentradores pasivos son simplemente cajas que disponen de unos puertos a los que se conectan las estaciones de trabajo dentro de una configuración en forma de estrella. Únicamente se trata de un cuadro de uniones. Tomado de: http://genesis.uag.mx/edmedia/material/comuelectro/glosario.cfm

## 1.9 Ámbito Nacional

En el mundo de la telefonía se han dado importantes pasos con la introducción de nuevas tecnologías y servicios, lo cual ha permitido brindar soluciones interesantes a las crecientes necesidades de los clientes. Así surgieron sistemas telefónicos que posibilitan la comunicación dentro de una empresa. Dichos sistemas son conocidos como *PBXs* (*Private Branch Exchange*, Pizarra Telefónica Privada, por sus siglas en inglés) o *PABXs* (*Private Automatic Branch Exchange*, Pizarra Telefónica Automática Privada, por sus siglas en inglés) y se han ido desarrollando junto con los sistemas telefónicos de mayor escala, aunque por sus características pueden prestar servicios adicionales. Un proveedor con gran presencia en las redes de telecomunicaciones de Cuba es *Alcatel.*

### 1.9.1 CIMEX y BFI (Banco Financiero Internacional, por sus siglas en inglés)

La Agencia Reguladora del MIC (Ministerio de Informática y Comunicaciones) está analizando el autorizo del uso de la *VoIP* en Cuba. Hasta el momento, la corporación CIMEX s.a. y el BFI están autorizados a implantar la *VoIP* para su trabajo interno. CIMEX ya ha obtenido beneficios de la aplicación de esta tecnología y ahora está incursionando en la prestigiosa solución de telefonía *IP* de *Cisco*. Sin embargo, por razones económicas, esta no podrá ser asumida por todas las empresas que en el país se les autorice el uso de la *VoIP*, por lo que se deben investigar softwares libres como *Asterisk*, que permiten, a un costo mínimo, contar con una *PBX IP*. Empresas como CIMEX pudieran valorar la integración de *Asterisk* con *Cisco CallManager Expres* (*PBXs IP* de *Cisco*), en aras de obtener nuevas funcionalidades a menores costos.

### 1.9.2 Experiencias con sistemas de *VoIP* en ETECSA

En el año 2003 se realizó una prueba de campo en el segmento nacional de la red por ETECSA, dicha prueba fue desarrollada por el Grupo de Investigación y Desarrollo de la Unidad de Negocios de la Red, utilizando equipamiento de *VocalTec*.

### 1.9.2.1 Aspectos sobre la prueba

- La prueba estuvo prevista para que enlazara tres ciudades, pero finalmente se realizó entre: La Ciudad de La Habana y Pinar del Río. En ambos sitios se utilizaron pasarelas *VGW* 120 de *VocalTec* enlazadas a las respectivas centrales telefónicas. Por el lado de la red de paquetes, cada pasarela se conectó mediante un acceso *Frame Relay* a 512 Kbps a la Red Cubadata.
- La Plataforma del Sistema es *H.323*, por lo que se suministró un *Gatekeeper* (*VGK*, por sus siglas en inglés). El sistema de Gestión alojado en el *Gatekeeper* permitía conocer el funcionamiento de todos los elementos de red, así como otros datos de las llamadas en curso.
- Se realizaron exitosamente llamadas Teléfono a Teléfono, *PC* a teléfono y de fax. Los resultados de las pruebas de calidad fueron excelentes en cuanto a demora, *jitter* y la evaluación del servicio percibida por los usuarios. No obstante, el tráfico conducido por el sistema fue muy inferior al tráfico crítico que pudiese deteriorar la calidad.

En general el desarrollo de la prueba de campo denota una tecnología madura y que introduce ahorros de ancho de banda, permitiendo elevar su rendimiento hasta en más de cinco veces, comparado con las tecnologías tradicionales. Ver ***Anexo2 figura 1*** donde se muestra la Red para prueba de campo de Telefonía IP con equipamiento *VocalTec* en ETECSA**.**

Esta solución es propia de etapas anteriores y en ningún caso ETECSA debe perder de vista que la Telefonía *IP,* o cualquiera de las variantes de transmitir voz sobre redes de paquetes, constituyen aspectos medulares, sobre los cuales se debe establecer una clara estrategia, que garantice la migración hacia las redes de próxima generación.

### 1.9.2.2 Otras experiencias de ETECSA

En la Gerencia Casa de Software *SIGTA*, de ETECSA, se realizaron un conjunto de pruebas para comprobar que *Asterisk* es capaz de conmutar llamadas telefónicas, que puede servir como pasarela entre varios protocolos *VoIP* y que permite la manipulación de correo de voz. Durante las pruebas se realizaron llamadas a usuarios que no respondieron antes de un tiempo establecido, a otros que sí lo hicieron y a otros que estaban ocupados; demostrando el buen comportamiento de *Asterisk* como operadora automática y servidor de correo de voz. Se permitió

la selección de idioma, por lo que se escucharon los mensajes tanto en inglés como en español. Se realizó un tratamiento especial con las llamadas fuera del horario laboral, en días feriados, a principios de año y a extensiones inexistentes. Se comunicaron clientes con protocolos *SIP* y *H.323* entre sí, sirviendo *Asterisk* como pasarela de *VoIP*. La calidad de audio percibida fue buena.

## 1.10 Ámbito Internacional

A escala mundial la tecnología de voz sobre *IP* ha tenido un auge considerable. Existe un conjunto de empresas que están desarrollando este tipo de tecnologías por las ventajas que brinda la misma. Entre las empresas más importantes se encuentran las siguientes:

### 1.10.1 *China Telecom Corporation*

Es el mayor proveedor de servicios de telecomunicaciones de China y se está convirtiendo en uno de los operadores más agresivos en el mundo de la Telefonía por *Internet*. *ITXC*, *VocalTec* y *China Telecom*, alcanzaron dieciséis ciudades chinas en sólo catorce días, lo cual pudo realizarse debido a que *ITXC* enrutó las llamadas sobre *Internet*. *China Telecom*, y *Clarent*(*TM*) *Corporation*, líder mundial en soluciones de Telefonía *IP*, anunciaron el completamiento de la segunda fase del *backbone* de la red de Telefonía *IP* en el país asiático que cubre todo el país.

### 1.10.2 VocalTec Communications

*VocalTec*. Empresa líder en esta tecnología, se apoya en su plataforma *VocalTec Ensemble Architecture* (*VEA,* por sus siglas en inglés), la cual facilita a los proveedores de servicios y a las empresas el disfrute de flexibilidad y manejo eficiente de los costos de las redes *IP*.

**Servicios soportados por VEA.**

| | |
|---|---|
| Teléfono a Teléfono. | Tarjeta de llamada prepagada. |
| *PC* a Teléfono. | Redes Privadas Virtuales (*VPN*). |
| Clearinghouse. | Web a Teléfono. |
| Conexión a Redes *SS7*. | Reemplazamiento de Enlace. |

Ver ***Anexo2 figura 2*** donde se muestra la solución *VocalTec*.

### 1.10.3 Criterios de otras firmas en el mundo

Otras firmas especializadas en el estudio del mercado, han emitido sus opiniones con respecto a la tendencia de la telefonía *IP* [8]. Entre ellas se encuentran:

- ***Phillips Tarifica Ltd*****,** plantea que el 43 % del tráfico internacional de usuarios fijos y móviles, será de Telefonía *IP* para el 2003, con 236 millones de usuarios.[9]
- ***Forrester Research,*** ha señalado que en el 2004 los *ITSP* incidirán en los ingresos incumbentes a los operadores por un monto de $ 3 billones en la Larga Distancia (*LD*,por sus siglas en inglés) doméstica de los Estados Unidos y $2 billones en gastos, así como ahorraran a los usuarios finales $ 1 billón de dólares. Las Tasas de Crecimiento en el mercado de la telefonía *IP* tendrían un ritmo de 212 % en 1999, de 122 % en el 2000, y de 114 % en el 2001. [9]
- ***Analysys*** **,**(firma de investigaciones del Reino Unido), estima que la telefonía *IP* se tragará para el 2003, el 36 % del mercado de las telecomunicaciones.[9]
- ***Cowen & Co***, (1998), ha señalado que el costo promedio por minutos de llamadas internacionales y el volumen de estas en Estados Unidos se han transformado con el paso del tiempo [9]. La ***tabla 1.2*** muestra el comportamiento del costo y volumen de las Llamadas Internacionales.

| Año | Costo por minuto | Cantidad de minutos |
|---|---|---|
| 1990 | $ 1.16 | |
| 1996 | Alrededor de $1.00 USD | 33,.3 billones |
| 1998 | $ 0.89 | 90 billones |
| 2000 | Alrededor de $ 0.83 | 105,2 billones |

Tabla 1.2. Comportamiento del costo y volumen de las Llamadas Internacionales en Estados Unidos.

***Vint Cerf****,* es la persona más comúnmente llamado el "padre de *Internet*"*,* actualmente trabaja en *Google* como su Vicepresidente Mundial, afirma que:

"Para el 2010, la mitad de la población mundial pudiera ser capaz de acceder a *Internet*, si las tasas presentes de crecimiento continúan constantes. Para ese tiempo, algunos estimados de dispositivos conectados de todos los tipos alcanzarían los 35 billones (cerca de 6 dispositivos por persona en el planeta). Esto no puede ser considerado demente si se ilustra con el hecho de que en el 2000, una persona con una *laptop*, un asistente digital personal y un teléfono *celular* podría ya tener tres dispositivos sobre *Internet*".[10]

### 1.11 Empresas y software que implementan *VoIP*

En el mundo *Cisco*, *Avaya*, *3Com* y *Skype* son los que más se destacan en las soluciones *PBXs IP* puras, mientras que *Nortel Networks*, *Alcatel* y *Siemens* resaltan en la tendencia híbrida. Por otra parte algunos PBXs *IP* con software libre son: *Asterisk*, *SIPX* y *Yate*. Se explicarán algunos de estos *software* para que se comprendan las soluciones que brindan con el objetivo de ver su funcionamiento.

#### 1.11.1 *SKYPE*

Es un programa (una red de telefonía entre pares por *Internet*) que te permite llamar gratis a cualquier otro usuario de *Skype*, en cualquier parte del mundo. Fundada por los suecos *Niklas Zennström* y *Janus Friis.*

##### 1.11.1.1 Ventajas que presenta

*Skype* es fácil y rápido de instalar, permite llamadas telefónicas gratuitas a otros usuarios de Skype en cualquier parte del mundo, funciona con todos los cortafuegos, *NAT* y *router* sin necesidad de reconfigurar. Las llamadas con *Skype* tienen una alta calidad de sonido y son altamente seguras. *Skype* funciona en la mayoría de los equipos: Windows, Mac *OS X*, *Linux* y *Pocket Out*. Además cifra de forma automática todo lo correspondiente a la conversación a modo de seguridad.

### 1.11.2 *Asterisk*

*Asterisk* es un software de código abierto, patrocinado por *Digium*, que funciona como *PBX IP*. Es distribuido libremente. *Asterisk* está diseñado para ofrecer una interfaz entre cualquier parte del *software* o *hardware* telefónico y cualquier aplicación de telefonía, de forma transparente y consistente. Principalmente se ejecuta sobre *Linux*, aunque también puede correr sobre otros sistemas basados en *Unix* como *FreeBSD*, y sobre *Windows*. [1]. *Asterisk* es económico, fácil de instalar y seguro. Puede integrarse con la telefonía tradicional y permite conectar gran variedad de teléfonos, tanto por *software* como por *hardware*.

### 1.11.3 Soluciones de *Alcatel*

Dentro del marco de los servicios de voz y multimedia, *Alcatel* ha desarrollado diversas soluciones que van desde *VoIP* hasta la provisión de nuevos servicios de Próxima Generación. A continuación se muestran algunas soluciones de *Alcatel* [11] aplicables en el ámbito de la red de Cuba.

- Telefonía *IP*. Es una aplicación que corre en la plataforma *Alcatel* 5020 *Softswitch*, proporcionando la inteligencia que se requiere para establecer conexiones de voz entre abonados *IP* nativos que utilizan dispositivos de comunicación basados en *IP*, tales como teléfonos software sobre *PC*, Asistentes Digitales Personales (*PDA*), o teléfonos *IP* sobre *SIP* / *H.323* y entre abonados *IP* nativos y abonados *RTPC* tradicionales conectados a la red *IP* a través de pasarelas *IP*. Proporciona servicio tanto al segmento residencial, como empresarial. Los primeros se benefician fundamentalmente de los servicios de voz tradicionales y de los servicios multimedia. El mercado empresarial se beneficia además con una amplia variedad de servicios gestionados de *VoIP* / *VPN* (Red Privada Virtual) que ofrece el paquete de aplicación *IPT*. Los proveedores de servicios estarán especialmente interesados en la solución *VPN-Wholesale* integrada en la Aplicación *IPT* Alcatel 5020 *Softswitch*.
- Desvío del tráfico de larga distancia. El desvío de larga distancia (LDB) es una aplicación que corre sobre la plataforma *Alcatel* 5020 *Softswitch*. Esta aplicación, ofrece una solución competitiva para el transporte de tráfico de voz sobre una red basada en *IP*, constituyendo

una alternativa para evitar las redes *RTPC* de larga distancia / interurbanas y desplegar servicios *VoIP* rentables. Las llamadas de voz, originadas en la red *RTPC* local, se entregan a la red *IP* por medio de pasarelas de enlace *IP* controladas por la aplicación *LDB* del *Alcatel* 5020 *Softswitch*. Ver ***Anexo2 figura 3*** donde se muestra la estructura piloto de *ALCATEL* para *VoIP*.

### 1.11.4 Solución de Cisco para telefonía *IP*

La solución para telefonía *IP* de *Cisco*, está integrada en una arquitectura muy sólida denominada *AVVID* (*Architecture for Voice, Video and Integrated Data*; Arquitectura para Voz, Vídeo y Datos Integrados, por sus siglas en inglés). *CCME* es una de las aplicaciones que la componen, la cual provee funcionalidades para el establecimiento y fin de llamadas, así como enrutamiento en redes de telefonía *IP*, y está diseñada para atender hasta 240 usuarios. CCME es una licencia opcional de *Cisco IOS* (*Internetwork Operating System*, Sistema Operativo para Equipos de Redes, por sus siglas en inglés). Se instala en los encaminadores de acceso multiservicio de *Cisco*. Utiliza el protocolo propietario *SCCP* (*Skinny Call Control Protocol*, Protocolo de Control de Llamada Ligero, por sus siglas en inglés) [13]. En la Dirección de Informática y Telecomunicaciones (*DataCIMEX*), de la Corporación CIMEX s.a., se realizaron un conjunto de pruebas para demostrar que *CCME* es capaz de conmutar llamadas telefónicas. Las pruebas realizadas confirmaron que *CCME* responde a la seriedad y experiencia de Cisco. CCME ofrece un sólido juego de características telefónicas para la *PYME* y proporciona capacidades de valor añadido, a través de *XML* (*Extensible Markup Language*, Lenguaje Extensible de Marcas, por sus siglas en inglés). Está respaldada por un soporte y una documentación de excelencia.

### 1.11.5 Arquitectura común de voz implementada por CISCO [8].

*Cisco Systems* introdujo mejoras en *software* y *hardware* para su línea de productos de acceso de múltiples servicios, que le permite a los proveedores de servicio y a los clientes corporativos desarrollar infraestructuras de red a gran escala basados en *VoIP*, integrando voz, vídeo y datos.

En *software*, las nuevas características ofrecen *VoFR* (*Voice Over Frame Relay*, Voz sobre Frame Relay, por sus siglas en inglés) en los ruteadores de acceso de múltiples servicios *Cisco* 2600, *Cisco* 3600, *Cisco* 7200 y en los concentradores de acceso de múltiples servicios *Cisco MC (Controlador Multipunto)* que permiten al usuario voz conmutada y evadir los *PBXs* a través de múltiples circuitos permanentes virtuales, con base en el número telefónico marcado. La red de voz sobre *IP* es confiable y escalable con posibilidad de integrar con facilidad locaciones internacionales. Las interfaces soportan *VoFR* o *VoIP*, haciendo posible las conexiones a los *PBXs* con *Basic Rate Interface* (BRI, por sus siglas en inglés), así como con las tradicionales de telefonía. Ver ***Anexo2 figura 4*** donde se presenta la arquitectura de *CISCO*.

### 1.11.6 Telefonía *IP* de *3Com*

*3Com* se basa en una arquitectura abierta de tres niveles de pasarelas, controladores de acceso y servidores de *backend*[12] interconectados mediante protocolos abiertos basados en normas. La arquitectura modular de *3Com* presenta *Apis* estándar en cada nivel a fin de brindarle a los operadores flexibilidad para personalizar el sistema, facilitando la diferenciación de servicios y la integración de las "mejores" aplicaciones de oficina. Este sistema modular, soporta la Telefonía sobre *IP* de teléfono a teléfono y de *PC* a teléfono. Sobre la base de la plataforma de acceso Total *Control Multiservice Access Plataform* de *3Com*, el sistema de *VoIP* acepta protocolos como *ITU T.120* y *H.323v2*. Utiliza la codificación de voz *G.711*, *G.723.1* y *G.729a*. Este desarrollo representa el próximo paso lógico para una plataforma diseñada para servicios múltiples de voz, fax y vídeo. Ver ***Anexo2 figura 5*** donde se presenta la arquitectura de *3Com*.

## 1.12 Protocolos utilizados en las Redes de *VoIP*

Un protocolo es un conjunto de normas o reglas que utilizan los dispositivos de red para comunicarse. En la actualidad en las redes de *VoIP* se utilizan diversos protocolos que son de gran importancia para el funcionamiento de la misma y para lograr una mejor interrelación; entre

[12] La parte de un programa que procesa las tareas para las cuales ha sido diseñado. Si nos referimos al backend de un sitio web, hablamos del área de administración en donde se ingresa, edita y organiza el contenido. Tomado de: www.emprende.org/component/option,com_rd_glossary/task,showcat/catid,69/Itemid,67/

los protocolos usados están: Megaco (También conocido como *H.248*) y *MGCP* (Protocolos de control), *Skinny Client Control Protocol* (Protocolo propiedad de *Cisco)*, MiNet (Protocolo propiedad de Mite), CorNet-IP (Protocolo propiedad de *Siemens*), Skype (Protocolo propiedad peer-to-peer utilizado en la aplicación Skype), Cliconnect (Proveedor de Servicio *VoIP* Cliconnect), Jajah (Protocolo propiedad peer-to-peer utilizado en los teléfonos-web Jajah *SIP)*, *IAX* y compatibles. En este trabajo se explicarán los protocolos *H.323* y *SIP* que son los más usados y por su gran importancia para la comunicación de voz sobre redes *IP* (*VoIP*).

## 1.13 Conclusiones

Los sistemas de señalización para el transporte de voz han evolucionado desde las redes basadas en conmutación de circuitos a redes basadas en conmutación de paquetes. Con esta evolución han aparecido diferentes estándares con el objetivo de resolver problemas de direccionamiento, control de admisión, interconexión con redes existentes, intercambio de capacidades, etc. En este capítulo se ha realizado un estudio de los temas y conceptos asociados a la *VoIP* así como una breve descripción de algunos *softwares* y proveedores que implementan esta tecnología, tanto en el ámbito nacional como internacional.

# CAPÍTULO 2: PROCEDIMIENTO PARA EL PROCESAMIENTO DE LOS PROTOCOLOS *SIP* Y *H.323,* USADOS EN LA COMUNICACIÓN DE VOZ SOBRE REDES *IP* (*VOIP*)

## 2.1 Introducción

En el Capítulo I, se analizaron algunos de los factores que han propiciado un acelerado desarrollo de la voz sobre *IP,* teniendo en cuenta la existencia de una red telefónica extensamente difundida por todo el mundo, tanto nacional como Internacional. En el presente Capítulo, se pretende hacer un estudio profundo de los principales protocolos relacionados con la Telefonía *IP* y otros que aseguran el funcionamiento de las mismas, además se pretende demostrar su composición por capas y por grupos según sus funciones, entre las que se encuentran, el transporte de medios, la señalización y el control de pasarelas.

## 2.2 Estructura del procedimiento

Para la realización del procedimiento se hace necesario estudiar en profundidad las características de los diferentes protocolos que permiten el buen funcionamiento de las redes, incluyendo los protocolos que son utilizados en las comunicaciones de voz sobre redes *IP* (*VoIP*), específicamente los protocolos *SIP* y *H.323*, así como su funcionamiento en las redes, con el objetivo de lograr un mayor entendimiento que nos permitirá la creación de un procedimiento para el procesamiento de los datos relacionados con los protocolos empleados en dichas comunicaciones. Para la realización de dicho procedimiento es importante desarrollar un aspecto fundamental:

## 2.3 Aspecto esencial a desarrollar para la realización del procedimiento

- Estudio profundo del funcionamiento de las redes.

### 2.3.1 Estudio del funcionamiento de las redes

Para poder entender con claridad cómo es el funcionamiento de las redes, se hace necesario el uso de un modelo en capas. Apoyándonos en este modelo para el análisis de las redes se podrá hacer un extenso estudio de todo lo relacionado con ellas, cómo funcionan las diferentes capas del modelo, así como la interacción que existe entra ellas, cómo funciona cada protocolo a nivel de capa, cómo están relacionados entre sí y sus características fundamentales.

#### 2.3.1.1 Beneficios del uso de un modelo en capas

Para visualizar la interacción entre varios protocolos, es común emplear un modelo en capas. Éste muestra el funcionamiento de los protocolos que se produce dentro de cada capa, así como la interacción de las capas sobre y debajo de él. Existen beneficios de su utilización para describir los protocolos, como son:

- Muestra información de los protocolos que operan en una capa específica.
- Poseen información que van a poner en práctica y una interfaz definida según las capas por encima y por debajo.
- Evita que los cambios en la tecnología o en las capacidades de una capa afecten otras capas superiores e inferiores.
- Proporciona un lenguaje común para describir las funciones y capacidades de red.

#### 2.3.1.2 Modelos de protocolo y referencia

Existen dos tipos básicos de modelos de Redes de Comunicación: modelo de protocolo y modelo de referencia. Un modelo de protocolo proporciona un modelo que coincide fielmente con la estructura de una *suite* de protocolo en particular. El conjunto jerárquico de protocolos relacionados en una *suite* representa típicamente toda la funcionalidad requerida para interconectar la red humana con la red de datos. El modelo *TCP/IP*[13] es un modelo de protocolos porque describe las funciones que se producen en cada capa de los protocolos dentro del

[13] TCP/IP: Familia de protocolos que hacen posible la interconexión y tráfico de red en Internet. Tomado de: www.um.es/dilengua/glosario.html

conjunto *TCP/IP.* En nuestro trabajo se utilizará el modelo *TCP/IP* por la ventajas que presenta, además de ser el mas usado en el mundo, solo se utilizara el modelo OSI para algunas explicaciones especificas.

Un modelo de referencia proporciona una referencia común para mantener consistencia en todos los tipos de protocolos y servicios de red. El propósito principal de un modelo de referencia es asistir en la comprensión más clara de las funciones y los procesos involucrados.

#### 2.3.1.3 Modelo *TCP/IP*

El modelo se creó a principios de la década de los setenta y se conoce con el nombre de modelo de *Internet*. Define cuatro categorías de funciones que deben tener lugar para que las comunicaciones sean exitosas. La arquitectura de la *suite* de protocolos *TCP/IP* sigue la estructura de este modelo. Por esto, es común que al modelo de *Internet* se lo conozca como modelo *TCP/IP*. La mayoría de los modelos de protocolos describen un *stack*[14] de protocolos específicos del proveedor. Sin embargo, puesto que el modelo *TCP/IP* es un estándar abierto, una compañía no controla la definición del modelo. Las definiciones del estándar y los protocolos *TCP/IP* se definen en un conjunto de documentos disponibles al público. Estos documentos se denominan solicitudes de comentarios (*RFCS*, por sus siglas en inglés), contienen las especificaciones formales de los protocolos de comunicación de datos y los recursos que describen el uso de los protocolos. Se utiliza el modelo *TCP/IP* y no el *OSI* porque hoy en día es más utilizado ver ***figura 1.***

[14] Stack: La Pila de protocolos para Servicios Web es una colección de protocolos para redes de datos. Tomado de: es.wikipedia.org/wiki/Web_Services_Protocol_Stack - 20k

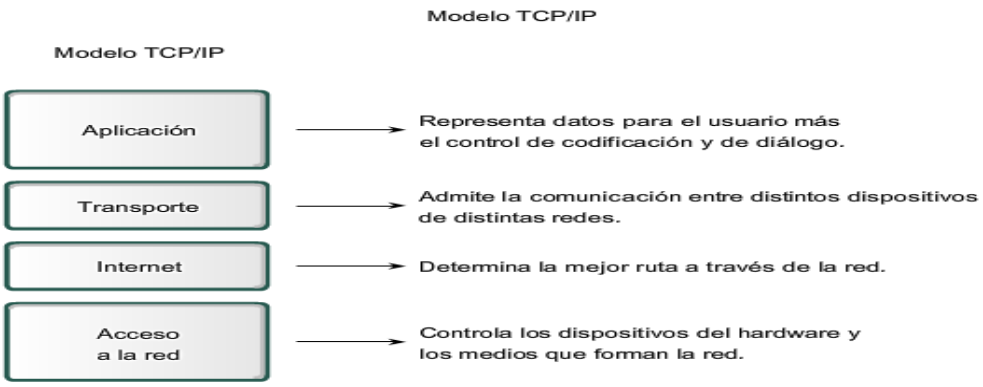



Figura1. Modelo *TCP/IP*.

### 2.3.1.4 Protocolos de *TCP/IP*

Para comprender este modelo es necesario comprender algunas de las características de cada uno de los protocolos que interactúan en cada capa, a continuación se explicará el conjunto de protocolos que implementa TCP/IP

### 2.3.1.5 Protocolo Ethernet

El comité de estándares para Redes Metropolitanas y Locales del Instituto de Ingenieros Eléctricos y Electrónicos (*IEEE*, por sus siglas en inglés) publicó los estándares para las *LAN* (*Redes de área local*, por sus siglas en inglés). Estos estándares comienzan con el número 802. El estándar para *Ethernet* es el 802.3. El *IEEE* quería asegurar que sus estándares fueran compatibles con los del modelo *OSI* de la Organización Internacional para la Estandarización (*ISO*, por sus siglas en inglés). Para garantizar la compatibilidad, los estándares *IEEE* 802.3 debían cubrir las necesidades de la Capa 1 y de las porciones inferiores de la Capa 2 del modelo *OSI*.

*Ethernet* opera en las dos capas inferiores del modelo *OSI*: la capa de enlace de datos y la capa física. Ver ***anexo 3 figura1*** donde se muestra las capas donde interviene el protocolo Ethernet.

El modelo ofrece una referencia que muestra con qué puede relacionarse *Ethernet*, aunque en realidad se implementa sólo en la mitad inferior de la capa de Enlace de Datos, que se conoce

como subcapa Control de Acceso al Medio (*Media Access Control, MAC*, por sus siglas en inglés), y la capa Física en su totalidad.

*Ethernet* en la capa 1 implica señales, *streams* de *bits* que se transportan en los medios, componentes físicos que transmiten las señales a los medios y distintas topologías. La capa 1 de *Ethernet* tiene un papel clave en la comunicación que se produce entre los dispositivos, pero cada una de estas funciones tiene limitaciones.

Como lo muestra el ***anexo 3 figura2*****,** Ethernet en la capa 2 se ocupa de estas limitaciones. Las subcapas de enlace de datos contribuyen significativamente a la compatibilidad de tecnología y la comunicación con la computadora. La subcapa *MAC* se ocupa de los componentes físicos que se utilizarán para comunicar la información y prepara los datos para transmitirlos a través de los medios.

El éxito de *Ethernet* se debe a los siguientes factores:

- ✓ Simplicidad y facilidad de mantenimiento.
- ✓ Capacidad para incorporar nuevas tecnologías.
- ✓ Confiabilidad.
- ✓ Bajo costo de instalación y de actualización.

**Dispositivos que implementa *Ethernet*.**

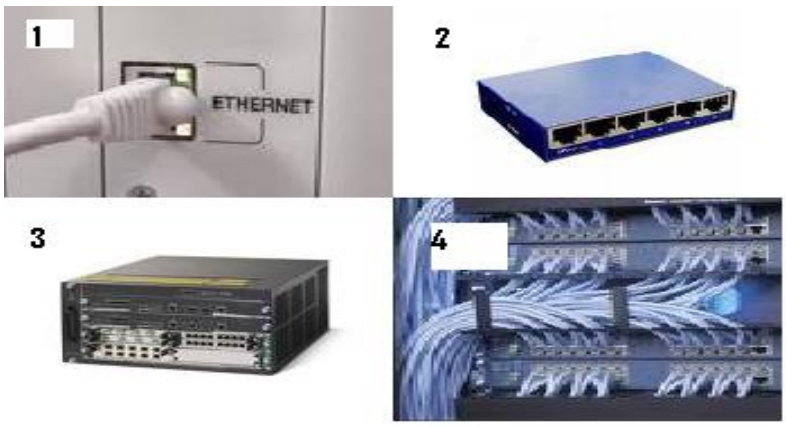



Figura2. Dispositivos físicos que implementa Ethernet, 1 cable UTP, 2 switch, 3 router y 4 patch panel.

Los medios físicos originales de cable coaxial grueso y fino se reemplazaron por categorías iniciales de cables *UTP*. En comparación con los cables coaxiales, los cables *UTP* eran más fáciles de utilizar, más livianos y menos costosos.

El aumento del rendimiento de la red es significativo cuando la velocidad de transmisión potencial aumenta de 100 Mbps a 1 Gbps y más. La actualización a *Ethernet* de 1 Gbps no siempre implica que la infraestructura de red de cables existente debe reemplazarse por completo. Algunos equipos y cableados de redes modernas bien diseñadas e instaladas podrían trabajar a mayores velocidades con sólo una actualización mínima. Esta capacidad tiene el beneficio de reducir el costo total de propiedad de la red.

#### 2.3.1.6 Protocolo de Internet (IP)

La capa Red o capa 3 del modelo *TCP/IP*, provee servicios para intercambiar secciones de datos individuales a través de la red entre dispositivos finales identificados. Para realizar este transporte de un extremo a otro, la capa Red emplea procesos como: el direccionamiento, encapsulamiento, enrutamiento y desencapsulamiento. Los protocolos especifican la estructura y el procesamiento del paquete utilizado para llevar los datos desde un *host* hasta otro *host*. Operar ignorando los datos de aplicación llevados en cada paquete permite a la capa Red llevar paquetes para múltiples tipos de comunicaciones entre *hosts* múltiples.

##### 2.3.1.6.1 Direccionamiento

Primero, la capa Red debe proveer un mecanismo para direccionar estos dispositivos finales. Si las secciones individuales de datos deben dirigirse a un dispositivo final, este dispositivo debe tener una dirección única. En una red *IPv4*, cuando se agrega esta dirección a un dispositivo, al dispositivo se lo denomina host.

##### 2.3.1.6.2 Encapsulación

Segundo, la capa Red debe proveer encapsulación. Los dispositivos no deben ser identificados sólo con una dirección; las secciones individuales, las *PDU* de la capa Red deben, además,

contener estas direcciones. Durante el proceso de encapsulación, la capa 3 recibe la *PDU* de la capa 4 y agrega un encabezado o etiqueta de capa 3 para crear la *PDU* de la capa 3. Cuando nos referimos a la capa Red, denominamos paquete a esta *PDU*. Cuando se crea un paquete, el encabezado debe contener, entre otra información, la dirección del host hacia el cual se lo está enviando.

A esta dirección se la conoce como dirección de destino. El encabezado de la capa 3 también contiene la dirección del *host* de origen. A esta dirección se la llama dirección de origen. Después de que la capa Red completa el proceso de encapsulación, el paquete es enviado a la capa Enlace de Datos que ha de prepararse para el transporte a través de los medios.

#### 2.3.1.6.3 Enrutamiento

Luego, la capa Red debe proveer los servicios para dirigir estos paquetes a su *host* destino. Los *host* de origen y destino no siempre están conectados a la misma red. En realidad, el paquete podría recorrer muchas redes diferentes. A lo largo de la ruta, cada paquete debe ser guiado a través de la red para que llegue a su destino final. Los dispositivos intermediarios que conectan las redes son los *routers*. La función del *router* es seleccionar las rutas y dirigir paquetes hacia su destino. A este proceso se lo conoce como enrutamiento. Durante este proceso a través de una *internetwork*, el paquete puede recorrer muchos dispositivos intermediarios. A cada ruta que toma un paquete para llegar al próximo dispositivo se la llama salto. A medida que el paquete es enviado, su contenido (la *PDU* de la Capa de transporte) permanece intacto hasta que llega al *host* destino.

#### 2.3.1.6.4 Desencapsulamiento

Finalmente, el paquete llega al *host* destino y es procesado en la capa 3. El *host* examina la dirección de destino para verificar que el paquete fue direccionado a ese dispositivo. Si la dirección es correcta, el paquete es desencapsulado por la capa Red y la *PDU* de la capa 4 contenida en el paquete pasa hasta el servicio adecuado en la capa Transporte.

### 2.3.1.7 Protocolos de la capa Red

Los protocolos implementados en la capa Red que llevan datos del usuario son:

- Versión 4 del Protocolo de Internet (*IPv4*, por sus siglas en inglés).
- Versión 6 del Protocolo de Internet (*IPv6*, por sus siglas en inglés).

Los servicios de la capa Red implementados por el conjunto de protocolos *TCP/IP* son el Protocolo de Internet (*IP*, por sus siglas en inglés). La versión 4 de *IP* (*IPv4*, por sus siglas en inglés) es la versión de *IP* más ampliamente utilizada. Es uno de los protocolos de capa 3 que se utiliza para llevar datos de usuario a través de *Internet* y será tratado en este trabajo investigativo. El *IPv6* opera junto con *IPv4* y puede en un futuro reemplazarlo.

#### 2.3.1.7.1 Características básicas de *IPv4 (Ver anexo3 figura 3)*

✓ **Sin conexión:**

Los paquetes *IP* se envían sin notificar al *host* final que están llegando. No requiere un intercambio inicial de información de control para establecer una conexión de extremo a extremo antes de que los paquetes sean enviados, ni requiere campos adicionales en el encabezado de la *PDU* para mantener esta conexión. Este proceso reduce en gran medida la sobrecarga del *IP*.

La entrega del paquete sin conexión puede hacer que los paquetes lleguen a destino fuera de secuencia. Si los paquetes que no funcionan o están perdidos crean problemas para la aplicación que usa los datos, luego los servicios de las capas superiores tendrán que resolver estas cuestiones.

✓ **Servicio de mejor intento (no confiable)**

*IP* a menudo se lo considera un protocolo no confiable. No confiable en este contexto no significa que el *IP* funciona adecuadamente algunas veces y no funciona bien en otras oportunidades. Tampoco significa que no es adecuado como protocolo de comunicaciones de datos. No confiable significa simplemente que *IP* no tiene la capacidad de administrar ni recuperar paquetes

no entregados o corruptos. Como los protocolos en otras capas pueden administrar la confiabilidad, se le permite a *IP* funcionar con mucha eficiencia en la capa Red.

- ✓ **Independiente de los medios**

Es responsabilidad de la capa de Enlace de Datos de *OSI* tomar un paquete *IP* y prepararlo para transmitirlo por el medio de comunicación. Esto significa que el transporte de paquetes *IP* no está limitado a un medio en particular. Cualquier paquete *IP* individual puede ser comunicado eléctricamente por cable, como señales ópticas por fibra, o sin cables como las señales de radio.
*IPv4* encapsula o empaqueta el datagrama o segmento de la capa de Transporte para que la red pueda entregarlo a su *host* de destino. La encapsulación de *IPv4* permanece en su lugar desde el momento en que el paquete deja la capa de Red del *host* de origen hasta que llega a la capa de Red del *host* de destino.
El proceso de encapsular datos por capas permite que los servicios en las diferentes capas se desarrollen y escalen sin afectar otras capas. Esto significa que los segmentos de la capa Transporte pueden ser empaquetados fácilmente por los protocolos de la capa Red existentes, como *IPv4* e *IPv6*, o por cualquier protocolo nuevo que pueda desarrollarse en el futuro.

### 2.3.1.8 Protocolo de Control de Transmisión (TCP)

Cuando hablamos de *TCP* nos estamos refiriendo al acrónimo de (*Tansmission Control Protocol*, por sus siglas en inglés). Es el protocolo de transporte que administra las conversaciones individuales entre servidores *web* y clientes *web*. *TCP* divide en pequeñas partes nombradas segmentos, los mensajes *http* para enviarlos al cliente de destino. Además se encarga de controlar el tamaño y los intervalos a los que se intercambian los mensajes entre el servidor y el cliente. Es un protocolo orientado a la conexión, descrito en la *RFC 793*. *TCP* incurre en el uso adicional de recursos para agregar funciones. Las funciones adicionales especificadas por *TCP* están en el mismo orden de entrega, son de entrega confiable y de control de flujo. Los segmentos de *TCP* son de 20 *bytes* de carga en el encabezado, que encapsulan los datos de la capa de aplicación. El protocolo *TCP* es utilizado en aplicaciones de transferencia de archivos,

correo electrónico y exploradores *web.* Es importante señalar que se usan los términos cliente y servidor como referencia.

#### 2.3.1.9 Protocolo *UDP*: Comunicación con baja sobrecarga

*UDP* es un protocolo simple que provee las funciones básicas de la capa Transporte. Genera mucho menos sobrecarga que *TCP*, ya que no es orientado a la conexión y no cuenta con los sofisticados mecanismos de retransmisión, secuenciación y control del flujo. Esto no significa que las aplicaciones que utilizan *UDP* no sean confiables. Sólo quiere decir que estas funciones no son contempladas por el protocolo de la capa Transporte y deben implementarse aparte, si fuera necesario.

Algunas aplicaciones como los juegos en línea o *VoIP* pueden tolerar alguna pérdida de datos. Si estas aplicaciones utilizaran *TCP*, experimentarían largas demoras, ya que *TCP* detecta la pérdida de datos y los retransmite. Estas demoras serían más perjudiciales para la aplicación que las pequeñas pérdidas de datos. Algunas aplicaciones como *DNS*, simplemente reintentan enviar la solicitud si no obtienen respuesta y, por lo tanto, no necesitan *TCP* para garantizar la entrega del mensaje. La baja sobrecarga de *UDP* lo hace deseable para dichas aplicaciones.

##### 2.3.1.9.1 Procesos del cliente *UDP*

Como en *TCP*, la comunicación cliente/servidor se inicia por una aplicación cliente que solicita datos de un proceso del servidor. El proceso de cliente *UDP* selecciona al azar un número de puerto del rango dinámico de números de puerto y lo utiliza como puerto de origen para la conversación. El puerto de destino por lo general será el número de puerto bien conocido o registrado asignado al proceso del servidor.

Los números de puerto de origen seleccionados al azar colaboran con la seguridad. Si existe un patrón predecible para la selección del puerto de destino, un intruso puede simular el acceso a un cliente de manera más sencilla intentando conectarse al número de puerto que tenga mayor posibilidad de estar abierto.

Ya que no se crean sesiones con *UDP*, tan pronto como los datos están listos para ser enviados y los puertos estén identificados, *UDP* puede formar el datagrama y enviarlo a la capa Red para direccionamiento y envío a la red. Cabe recordar que una vez que el cliente ha elegido los puertos de origen y destino, estos mismos puertos se utilizarán en el encabezado de todos los datagramas que se utilicen en la transacción. Para la devolución de datos del servidor al cliente, se invierten los números de puerto de origen y destino en el encabezado del datagrama.

### 2.3.1.10 Transporte de Medios, Señalización, así como Control y Calidad del Servicio relacionado con *VoIP*

Una vez que se explicaron los protocolos fundamentales que implementa *TCP/IP* queda explicar los protocolos que utiliza la telefonía *IP.*

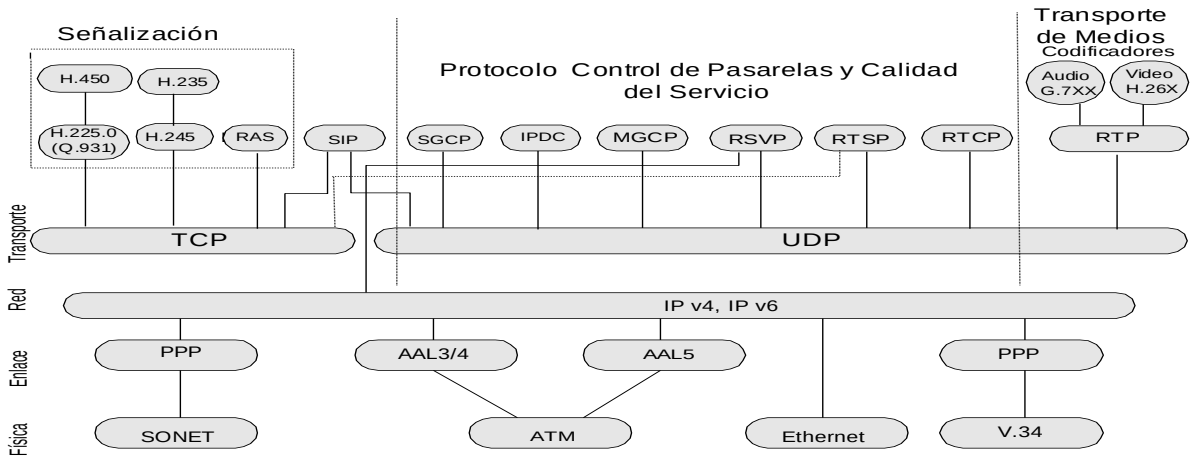



Figura 3a**.** Pila de Protocolos relacionados con la Telefonía IP y su disposición por Capas

### 2.3.1.11 Protocolo de transporte en Tiempo Real (*RTP*)

Este protocolo soporta el transporte de medios extremo a extremo en tiempo real como son voz, vídeo, audio y vídeo y servicios multimedios con múltiples participantes. Define el formato en el cual deben ser empaquetadas las tramas de voz y ofrece en su encabezado información necesaria para la reconstrucción del flujo original. Durante la conferencia, pudiera darse el caso

que se requiera cambiar el *códec*[15], es decir, el estándar para la codificación pudiera requerir por ejemplo, menos ancho de banda, lo cual, propiciaría la entrada de un nuevo participante en la conferencia o simplemente reaccionar ante la indicación de congestión de la red.

*RTP*, por sus siglas en inglés, posibilita transporte de datos síncronos a través de una red de paquetes, utilizando típicamente *UDP* para aprovechar su multiplexado de puertos y servicios de comprobación de errores. En este caso puede transportarse por paquetes *IP multicast*, o sea, un flujo generado por un origen puede alcanzar varios destinos.

Como las redes pueden perder y retrasar los paquetes, el encabezamiento de *RTP* proporciona información para sincronizar el audio y el vídeo, debido a que los paquetes pueden sufrir demoras diferentes, por lo cual también es necesario compensar el *jitter*, determinar si los paquetes se han perdido o han llegado fuera de orden, para lo que se vale de la indicación de tiempo y número de secuencia. Además informa sobre el tipo de datos transportados.

#### 2.3.1.12 Protocolo de Control de Transporte en Tiempo Real (*RTCP*)

Utilizado para el control de *RTP*. El funcionamiento de este protocolo se basa en la transmisión periódica de paquetes de control *RTCP*, por sus siglas en inglés, por cada participante en una sesión *RTP* al resto de los participantes. En comunicaciones individuales (*unicast*, por sus siglas en inglés) no es necesario pero es útil. En difusión, es imprescindible para conocer como reciben los distintos destinatarios y homogeneizar la calidad. El período de envío de los paquetes *RTCP* se adapta al número de fuentes para evitar una avalancha de tráfico de control.

#### 2.3.1.13 Señalización

Los protocolos de señalización constituyen el corazón de la Telefonía *IP*, dado que estos se encargan de establecer, controlar y terminar llamadas. Entre los protocolos para estos fines se destaca la Recomendación *H.323* de la *UIT* (Unión Internacional de Telecomunicaciones, por siglas en inglés), cuya primera versión aparece en 1996 con el nombre de "Sistemas y Equipos

[15] Compresor/descompresor. Es el software que comprime y luego interpreta (descomprime) sonido y video. MP3, AAC y Atrac son códecs. Tomado de: www.lanacion.com.ar/nota.asp

Videotelefónicos para redes de área local que proporcionan una calidad de servicio no garantizada", la segunda surge en Enero de 1998, mientras que la tercera versión es de Septiembre de 1999 conocida como "Sistemas de Comunicaciones Multimedios basados en Paquetes". Otro protocolo de señalización es SIP (Session Initialization Protocol, por sus siglas en inglés) propuesto por *IETF (*Fuerza de Trabajo de Ingenieros de *Internet,* por sus siglas en inglés).

### 2.3.1.14 Protocolo *H.323* de la *UIT-T*

*H.323* fue diseñado con un objetivo principal: proveer a los usuarios con tele-conferencias que tienen capacidades de voz, video y datos sobre redes de conmutación de paquetes. *H.323* y la convergencia de voz, video y datos permiten a los proveedores de servicios prestar esta clase de facilidades a los usuarios, de tal forma que se reducen costos mientras mejora el desempeño para el usuario.

Este estándar según se observa en el *anexo3 figura 4*, es una familia de recomendaciones que describen la arquitectura y operación de un sistema de comunicaciones multimedios sobre una red de paquetes sin una calidad de servicio garantiza, no es específico para *IP*, es posible su utilización sobre *IPX*[16]*/SPX*[17] o *ATM*. Trabaja en diferentes redes como *LAN*, *WAN* e *Internet*.

Las entidades *H.323* pueden proporcionar comunicaciones de audio, vídeo y /o datos en tiempo real. El soporte de audio es obligatorio, datos y vídeo son opcionales. El presente trabajo centra la atención en las aplicaciones de audio.

Las entidades *H.323* pueden utilizarse en configuraciones punto a punto, punto a multipunto (difusión), según la Recomendación *H.332.* Los terminales *H.323* pueden interoperar de la manera mostrada en el ***anexo3 figura 5***.

[16] Internetwork Packet Exchange. Protocolo de nivel 3 de Novell similar al XNS y al IP que se utilizan en redes NetWare. Tomado de: www.solint.com.mx/glosario.htm

[17] **SPX** (***Sequenced Packet Exchange - Intercambio de Paquetes Secuenciados***) es un antiguo protocolo de red de Novell perteneciente al sistema operativo NetWare utilizado para controlar la entrega de datos a través de una red de área local. Tomado de: es.wikipedia.org/wiki/SPX

*H.323* define principalmente las necesidades de señalización para el establecimiento de llamadas y conferencias, escoge los *códecs* comunes, etc. El núcleo de *RTP/RTCP* es aún utilizado para transportar flujos sincronizados y obtener una realimentación de la calidad de la red. **El modelo por capas de *H.323*, se observa en el *anexo3 figura 6*.**

#### 2.3.1.14.1 Componentes y canales de *H.323*

H.323 establece los estándares para la compresión y descompresión de audio y vídeo, asegurando que los equipos de distintos fabricantes se intercomuniquen. Así, los usuarios no se tienen que preocupar de cómo el equipo receptor actúa, siempre y cuando cumpla este estándar. Por ejemplo, la gestión del ancho de banda disponible para evitar que la *LAN* se colapse con la comunicación de audio y vídeo también está contemplada en el estándar, esto se realiza limitando el número de conexiones simultáneas.

Este estándar define un amplio conjunto de características y funciones, algunas son necesarias y otras opcionales. El *H.323* define mucho más que las funciones, este estándar define los siguientes componentes más relevantes:

- Terminales.
- Pasarelas o *Gateways*.
- Controlador de Acceso o *Gatekeeper*.
- Controlador Multipunto (MC, por sus siglas en inglés).
- Procesador Multipunto (MP, por sus siglas en inglés).
- Unidad de Control Multipunto (MCU, por sus siglas en inglés).

**Terminales:** Un terminal *H.323* es un extremo de la red que proporciona comunicaciones bidireccionales en tiempo real con otro terminal *H.323*, *Gateway* o unidad de control multipunto (MCU, por sus siglas en inglés). Esta comunicación consta de señales de control, indicaciones, audio, imagen en color en movimiento y/o datos entre los dos terminales. Conforme a la especificación, un terminal *H.323* puede proporcionar sólo voz, voz y datos, voz y vídeo, o voz, datos y vídeo.

Un terminal *H.323* consta de las interfaces del equipo de usuario, el *códec* de video, el *códec* de audio, el equipo telemático, la capa *H.225*, las funciones de control del sistema y la interfaz con la red por paquetes. Para la codificación del audio existen diversos estándares normados por la *UIT-T* (Sector de Normalización de las Telecomunicaciones de la *UIT*, por sus siglas en inglés), que van desde los 64 *kbps* hasta 5.3 *kbps*.

El ***anexo 3 figura 7*** muestra los bloques que integran un terminal *H.323*, que están dentro y fuera del alcance de la Recomendación. A continuación se verán los primeros.

- ✓ ***Códec de vídeo*** (Recomendación *H.261*, etc.): codifica el vídeo a partir de la fuente (es decir, una cámara) para transmisión y decodifica el código de vídeo recibido, que es la salida hacia una presentación visual del vídeo; opcionalmente podrá utilizar *H.263*.
- ✓ ***Códec de audio*** (Recomendación *G.711*, etc.): que codifica la señal de audio del micrófono para transmisión y decodifica el código de audio recibido que es la salida hacia el altavoz. Si se dispone audio *G.723.1*, el *códec* será capaz de codificar y decodificar con arreglo al modo de 5,3 *kbit/s* o al modo de 6,3 *kbit/s.*
- ✓ ***Canal de datos:*** soporta aplicaciones telemáticas como pizarras electrónicas, transferencia de imágenes fijas, intercambio de ficheros, acceso a bases de datos, conferencias audiográficas, etc. La aplicación de datos normalizada para conferencia audiográfica en tiempo real es la Recomendación *T.120*. Se pueden utilizar otras aplicaciones y protocolos mediante la negociación de la Recomendación *H.245*.
- ✓ ***Retardo del Trayecto de Recepción***: Incluye el retardo añadido a las tramas para mantener la sincronización y tener en cuenta la fluctuación de las llegadas de paquetes. No suele usarse en la transmisión sino en recepción, para añadir el retardo necesario en el trayecto de audio para lograr la sincronización con el movimiento de los labios en una videoconferencia.
- ✓ ***Unidad de control del sistema*** (Recomendaciones *H.245* y *H.225.0*): que proporciona la señalización para un funcionamiento adecuado del terminal *H.323*. Permite el control de la llamada, el intercambio de capacidad, la señalización de instrucciones e indicaciones y facilita mensajes de apertura y descripción completa del contenido de los canales lógicos.

Está formada por tres bloques principales: Función de control *H.245*, función de señalización *RAS y* función de señalización de llamada *H.225*.

- ***Canal de control H.245:*** Es utilizado para la función de control de la Recomendación *H.245*, conduciendo mensajes de control de extremo a extremo, los cuales rigen el funcionamiento de la entidad *H.323*, incluyendo el intercambio de capacidades, apertura y cierre de canales lógicos, peticiones de modo preferido, mensajes de control de flujo e instrucciones e indicaciones generales. La señalización *H.245* se establece entre dos puntos extremos, un punto extremo y un Controlador Multipunto (*MC*, por sus siglas en inglés*)* o un punto extremo y un controlador de acceso. El punto extremo establecerá un único canal de control *H.245* en cada sentido para cada llamada en la que él participe, utilizando los mensajes y procedimientos de la Recomendación *H.245*. Como un terminal, una Unidad de Control Multipunto (*MCU*, por sus siglas en inglés), una pasarela o un controlador de acceso pueden soportar muchas llamadas, entonces podrán existir muchos canales de control *H.245*. Estos se llevan por el canal lógico 0, el cual está permanentemente abierto desde el establecimiento del canal de control *H.245* hasta la terminación de este canal.
- ***El canal de señalización RAS*** es independiente del canal de señalización de llamada y del canal de control *H.245*. Los procedimientos de apertura de canal lógico *H.245* no se utilizan para establecer el canal de señalización *RAS*. Cuando no hay controlador de acceso, no se utiliza el canal de señalización *RAS*. Cuando este existe, el canal se abre entre el punto extremo y el controlador de acceso. El canal de señalización *RAS* se abre antes de que se establezca cualquier otro canal entre puntos extremos *H.323* y se emplea para transportar mensajes utilizados en los procesos de descubrimiento del controlador de acceso y registro de punto extremo que asocian una dirección de alias con su dirección de transporte de canal de señalización de llamada. El canal *RAS* es no confiable. La función de señalización *RAS* utiliza mensajes de la Recomendación *H.225.0* para llevar a cabo los procedimientos de registro, admisiones, cambios de

anchura de banda, situación y desligamiento entre puntos extremos y controladores de acceso.

- ***Canal de señalización de llamada H.225:*** Se emplea para transportar mensajes de control de la llamada de la Recomendación *H.225.0*. Es un canal fiable. Si no existe controlador de acceso, los mensajes de señalización de llamada se pasan directamente entre los puntos extremos llamante y llamado utilizando las direcciones de transporte de señalización de llamada; se supone que el llamante conoce la dirección de transporte de señalización de llamada (del llamado) y puede comunicar directamente. Cuando existe controlador de acceso, el intercambio de mensajes de admisión inicial tiene lugar entre el llamante y el controlador de acceso utilizando la dirección de transporte de canal de *RAS* del controlador de acceso. Durante el intercambio de mensajes de admisión inicial, el controlador de acceso indica en el mensaje *ACF* si la señalización de llamada se envía directamente al otro punto extremo o se encamina a través de él. El canal de señalización de llamada puede transportar diversas llamadas concurrentes. La Recomendación *H.225.0* especifica los mensajes *Q.931* obligatorios que se utilizan para señalización de llamada en la Recomendación *H.323*. Además canales lógicos de información de vídeo, audio y datos son establecidos en correspondencia con la Recomendación *H.245*. Son unidireccionales en cada sentido de transmisión, en el caso de datos pueden ser bidireccionales. Es posible la transmisión de cualquier número de canales lógicos de cada tipo de medios, excepto en el caso del canal de control *H.245*, que será uno por llamada.

✓ ***Capa H.225.0*** (Recomendación *H.225.0*): realiza el formato de los trenes de vídeo, audio, datos y control transmitidos en mensajes de salida hacia la interfaz de la red y recupera los trenes de éstos medios recibidos de los mensajes que han sido introducidos desde la interfaz de la red. Además, lleva a cabo la alineación de trama lógica, la numeración secuencial, la detección de errores y la corrección de los mismos según conviene a cada tipo de medio.

- ✓ ***Interfaz de red de paquetes:*** es específica en cada implementación. Debe proveer los servicios descritos en la recomendación *H.225*. Esto significa que el servicio extremo a extremo fiable (por ejemplo, *TCP*) es obligatorio para el canal de control *H.245*, los canales de datos y el canal de señalización de llamada.
- ✓ El servicio de extremo a extremo no fiable (*UDP*, *IPX*) es obligatorio para los canales de audio, los canales de video y el canal de *RAS,* por sus siglas en inglés. Estos servicios pueden ser dúplex o simplex y de *unicast* o *multicast* dependiendo de la aplicación, las capacidades de los terminales y la configuración de la red.

**Pasarela o Gateway**: Conecta una red *H.323* y otra red diferente. Por ejemplo una pasarela puede brindar comunicación entre una red *H.323* y un terminal de la red pública telefónica, convirtiendo la señalización proveniente de la red externa al formato manejado por la red *H.323* y viceversa.

La pasarela proporcionará la conversión adecuada entre formatos de transmisión (por ejemplo, *H.225.0* a/de *H.221*) y entre procedimientos de comunicaciones (por ejemplo, *H.245* a/de *H.242*). Además llevará a cabo el establecimiento y la liberación de la llamada en el lado de la red de conmutación de circuitos. La pasarela, realiza la conversión entre formatos y trenes de control, audio vídeo y/o datos, para su entrega a los distintos terminales en virtud de las diferentes recomendaciones que cumplen los mismos. Por lo general, la finalidad de la pasarela (cuando no funciona como una *MCU*), consiste en reflejar transparentemente las características de un punto extremo de red a un punto extremo de la red de conmutación de circuitos y a la inversa.

Entre las funciones que cumple una pasarela *H.323*, están la función de terminal o *MCU H.323*, de terminal o *MCU* de la red de conmutación de circuitos y la función de conversión.

Una pasarela puede conectarse a través de la red de conmutación de circuitos con otras, para facilitar la comunicación entre terminales *H.323* que no están en la misma red. Los equipos que proporcionan la interconexión transparente entre las redes sin utilizar protocolos de la serie *H* (tales como los encaminadores y las unidades de marcación de entrada a distancia) no son pasarelas.

**Controlador de Acceso o *Gatekeeper*:** El *gatekeeper* es una entidad que proporciona la traducción de direcciones y el control de acceso a la red de los terminales H.323, gateways y *MCUs*. Es considerado el cerebro de la red *H.323*, pero puede no estar presente. Si está presente es el punto central de todas las llamadas en la red, aunque está separado lógicamente de los puntos extremos, su implementación física puede coexistir con un terminal, *MCU*, pasarela, *MC* u otro dispositivo de *LAN* no *H.323*.

El controlador de acceso deberá prestar los siguientes servicios:

1. ***Conversión de dirección:*** Conversión de dirección de alias a dirección de transporte, utilizando un cuadro de conversión actualizado mediante los mensajes *RAS*. Son posibles otros métodos de actualización de dicho cuadro.
2. ***Control de admisiones:*** La autorización del acceso puede basarse en la autorización de la llamada, en la anchura de banda o en algún otro criterio que se deja a decisión del fabricante. El *gatekeeper* puede rechazar aquellas llamadas procedentes de un terminal por ausencia de autorización a terminales o *gateways* particulares de acceso restringido. También puede ser una función nula que admita todas las peticiones.
3. ***Control y gestión de ancho de banda:*** Para controlar el número de terminales *H.323* a los que se permite el acceso simultáneo a la red, así como el rechazo de llamadas tanto entrantes como salientes para las que no se disponga de suficiente ancho de banda o porque se haya sobrepasado el nivel establecido, de manera tal que se garantice ancho de banda suficiente para las aplicaciones de datos sobre la *LAN*. Esta función puede ser nula.
4. ***Gestión de zona:*** Lleva a cabo el registro y la admisión de los terminales y *gateways* de su zona. Conoce en cada momento la situación de los *gateways* existentes en su zona. El controlador de acceso proporcionará las funciones anteriores para terminales, *MCU* y pasarelas que se hayan registrado en él, o sea que estén dentro de su zona.

Funciones opcional del controlador de acceso:

- ***Señalización de control de llamada***– El controlador de acceso puede optar por

completar la señalización de la llamada con los puntos extremos y puede procesar él mismo la señalización de la llamada. De manera alternativa, puede encaminar los puntos extremos para que conecten el canal de señalización de llamada directamente el uno al otro.

- ***Autorización de llamada*** – Utilizando la señalización *H.225.0*, el controlador de acceso puede rechazar llamadas procedentes de un terminal por ausencia de autorización. Pueden ser motivos de rechazo, entre otros, el acceso restringido hacia/desde terminales o pasarelas particulares y el acceso restringido durante determinados periodos de tiempo.
- ***Gestión de llamada*** – Por ejemplo, el controlador de acceso puede mantener una lista de llamadas *H.323* en curso. Esta información puede ser necesaria para indicar que un terminal llamado está ocupado y proporcionar información para la función de gestión de ancho de banda.

**Controlador multipunto (*MC*):** Proporciona funciones de control para soportar conferencias entre tres o más puntos extremos de una conferencia multipunto. El *MC* lleva a cabo el intercambio de capacidades con cada uno de los puntos extremos y les envía un conjunto de capacidades indicando los modos de funcionamiento en los que pueden transmitir. Puede revisar el conjunto de capacidades que envía a los terminales como consecuencia de la incorporación/abandono de terminales a la conferencia, o por otros motivos.
Como parte del establecimiento de una conferencia multipunto, un punto extremo quedará conectado a un *MC* en su canal de control *H.245*. El *MC* puede estar situado dentro de un controlador de acceso, una pasarela, un terminal, o una *MCU* (contiene siempre un *MC*).

**Procesador multipunto (*MP*):** Es un componente de *H.323* de *hardware* y *software* especializado, mezcla, conmuta y procesa audio, vídeo y / o flujo de datos para los participantes de una conferencia multipunto, de tal forma que los procesadores del terminal no sean pesadamente utilizados. El procesador multipunto puede procesar un flujo medio único o flujos medio múltiples, dependiendo de la conferencia soportada.

El *MP* recibe trenes de audio, vídeo y/o datos de los puntos extremos que participan en una conferencia multipunto centralizado o híbrida. El *MP* procesa estos trenes de medios y los devuelve a los puntos extremos. Puede procesar uno o más tipos de trenes de medios.
Un *MP* que procese audio deberá preparar N salidas de audio a partir de M entradas de audio conmutando, mezclando o combinando ambas cosas. La mezcla de audio requiere la decodificación del audio de entrada en señales lineales (*MIC* o analógicas), efectuando una combinación lineal de las señales y recodificando el resultado en el formato de audio apropiado.

**Unidad de control multipunto (*MCU*):** Es un punto extremo en la red que da soporte a conferencias multipunto y deberá estar formada por un *MC* y ninguno, uno o varios *MP*. La *MCU* utiliza los mensajes y procedimientos *H.245* para implementar características similares a las que figuran en la Recomendación *H.243*.
Una *MCU* típica, que soporta conferencias multipunto centralizadas, consta de un *MC* y de un *MP* de audio, vídeo y datos. Para conferencias descentralizadas consta de un *MC* y de un *MP* de datos que soporte la Recomendación *T.120*. Se basa en el procesamiento descentralizado de audio y vídeo.

### 2.3.1.14.2 Pila de protocolos para H.323

*H.323* para su funcionamiento utiliza un conjunto de protocolos, es por ello que es conocido como un paraguas de protocolos. Los más significativos para *H.323* serán explicados a continuación:

- ***RTP/RTCP (Real-Time Transport Protocol / Real-Time Transport Control Protocol***, por sus siglas en inglés***):*** Protocolos de transporte en tiempo real que proporcionan servicios de entrega punto a punto de datos.
- ***RAS (Registration, Admission and Status***, por sus siglas en inglés***):*** Sirve para registrar, control de admisión, control del ancho de banda, estado y desconexión de los participantes.
- ***H225.0:*** Protocolo de control de llamada que permite establecer una conexión y una desconexión.

- ***H.245:*** Protocolo de control usado en el establecimiento y control de una llamada. Presenta las siguientes funcionalidades:
  1. Intercambio de capacidades: Los terminales definen los *códecs* de los que disponen y se lo comunican al otro extremo de la comunicación.
  2. Apertura y cierre de canales lógicos: Los canales de audio y video *H.323* son punto a punto y unidireccionales. Por lo tanto, en función de las capacidades negociadas, se tendrán que crear como mínimo dos de estos canales. Esto es responsabilidad de *H.245*.
  3. Control de flujo cuando ocurre algún tipo de problema.
  4. Multitud de otras pequeñas funciones.

- ***Q.931: (Digital Subscriber Signalling***, por sus siglas en inglés***)*** este protocolo se define para la señalización de accesos *RDSI* básico.
- ***RSVP (Resource ReSerVation Protocol,*** por sus siglas en inglés***):*** Protocolo de reserva de recursos en la red para cada flujo de información de usuario.
- ***T.120:*** La recomendación T.120 define un conjunto de protocolos para conferencia de datos.

Entre los códecs que recomienda usar la norma *H.323* se encuentran principalmente:

- ***G.711:*** De los múltiples *códecs* de audio que pueden implementar los terminales *H.323*, este es el único obligatorio. Usa modulación por pulsos codificados (***PCM,*** por sus siglas en inglés) para conseguir tasas de bits de 56 *Kbps* y 64 *Kbps*
- *H.261y H.263:* Los dos *códecs* de video que propone la recomendación *H.323,* sin embargo, se pueden usar otros.

### 2.3.1.15 Protocolo de Inicialización de Sesión (*SIP*).

*SIP* es un protocolo de control (señalización) de la capa de aplicación para crear, modificar y terminar sesiones o llamadas con uno o más participantes; esas sesiones incluyen conferencias de multimedios por Internet, aprendizaje a distancia, llamadas telefónicas por Internet y

aplicaciones similares. Los miembros en una sesión pueden comunicar mediante multicast o mediante una malla de relaciones unicast o una combinación de estas. El iniciador no necesariamente tiene que ser miembro de la sesión a la cual está invitando

Las invitaciones *SIP* utilizadas para crear sesiones transportan las descripciones de la sesión, las cuales permiten a los participantes ponerse de acuerdo en un conjunto de tipos de medios compatibles. *SIP* soporta movilidad de los usuarios, mediante *proxys*[18] y redireccionamiento de las solicitudes a las localizaciones actual de los usuarios. *SIP* no se encuentra atado a ningún protocolo de control de conferencia particular. Está diseñado independientemente del protocolo de transporte de la capa inferior y puede extenderse con capacidades adicionales. *SIP* soporta cinco facetas de establecimiento y terminación de comunicaciones multimedia, estas son:

1. ***Localización de usuario***: Se refiere a la determinación del sistema final a ser utilizado para la comunicación.
2. ***Habilidades del usuario***. Determinación de los medios y los parámetros de los mismos para ser utilizados en la comunicación.
3. ***Disponibilidad del usuario***: Determinación de la voluntad de la parte llamada a entrar en la comunicación.
4. ***Establecimiento de la llamada***: Establecimiento de los parámetros de las partes llamada y llamante.
5. ***Manipulación de la llamada***. Incluye transferencia y terminación de las llamadas.

Las pasarelas telefónicas pueden también utilizar *SIP* para conectar *Internet* con la Red Telefónica Conmutada Pública y para establecer llamadas entre ellas.

*SIP* forma parte de una arquitectura de control y datos multimedia propuesta por la *IETF* y que tiene en cuenta otros protocolos como *RSVP* (*RFC 2205*) para reservar recursos de red, *RTP(Protocolo de Transporte de Tiempo Real,* por sus siglas en inglés*)* (*RFC 1889*) para el

[18] Servidor alternativo que se sitúa entre el cliente y el servidor habitual o real, cuya función es interceptar las peticiones enviadas por el cliente al servidor para actuar en su lugar. Tomado de: www.informatica-pc.net/glosario/glosario_p.php

transporte de datos en tiempo real y provisión de realimentación sobre la *QoS*[19] con la utilización de *RTCP*, así como *RTSP* (*RFC 2326*), que es un protocolo para el control de la entrega de flujos de medios, el protocolo de anuncio de sesión (*SAP,* por sus siglas en inglés*)* para anunciar sesiones multimedios utilizando *multicast* y el protocolo de descripción de sesión (*SDP,* por sus siglas en inglés*)* (*RFC 2327*) para la descripción de sesiones multimedios. Sin embargo, el funcionamiento y operación de *SIP* no depende de ninguno de estos protocolos.

*SIP* puede también ser utilizado en conjunto con otros protocolos de señalización y establecimiento de llamadas. En ese modo, un sistema final utiliza intercambios *SIP* para determinar la dirección del sistema final y el protocolo desde una dirección independiente del mismo. Por ejemplo, *SIP* pudiera ser utilizado para determinar qué parte puede ser alcanzada mediante *H.323*, para obtener la pasarela *H.245* y la dirección de usuario y entonces utilizar *H.225.0* para establecer la llamada. En otro ejemplo, *SIP* pudiera determinar que la llamada es alcanzable mediante la Red Telefónica Conmutada Pública e indicar el número a ser llamado, posiblemente sugiriendo una pasarela entre Internet y dicha red.

*SIP* no ofrece servicios de control de conferencia, ni cómo gestionarlas, pero puede ser utilizado para introducir protocolos de control de conferencias. *SIP* no asigna direcciones *multicast*.

*SIP* puede invitar a usuarios a sesiones con y sin reservación de recursos, no reserva recursos, pero puede transmitir a los sistemas invitados la información necesaria para hacer esto.

**Visión de la Operación de SIP** Los llamadores y llamados se identifican por las direcciones *SIP*. Cuando se realiza una llamada *SIP*, un llamador primero localiza el servidor apropiado y envía una solicitud *SIP*. La operación *SIP* más común es la invitación. En lugar de alcanzar directamente el llamado deseado, una solicitud *SIP* debe ser redirigida o puede disparar una cadena de nuevas solicitudes *SIP* mediante los *proxys*. Los usuarios pueden registrar su(s) localización(es) en servidores *SIP*.

[19] **QoS** o **Calidad de Servicio** (*Quality of Service*, en inglés) son las tecnologías que garantizan la transmisión de cierta cantidad de datos en un tiempo dado (*throughput*). Tomado de: es.wikipedia.org/wiki/QoS

*SIP* es un protocolo basado en comandos de texto, utilizando *ISO 10646*[20]. Esto facilita la implementación en lenguajes tales como *Java*, *Tcl* y *Perl*, que permiten depurar, flexibilizar y extender el protocolo *SIP*. El protocolo se utiliza más para inicializar conferencias multimedios que para datos. La cabecera adicional debida a la utilización de un protocolo basado en texto no es significativa.

#### 2.3.1.15.1 Entidades *SIP*

*SIP* define dos componentes, el cliente y el servidor de red. El software que interactúa con el cliente se conoce como Agente Usuario

***Agente Usuario* (*UA***, por sus siglas en inglés**).** Aplicación que contiene un Cliente Agente Usuario (*UAC*) y a un Servidor Agente Usuario (*UAS*).

***Cliente Agente Usuario* (*UAC***, por sus siglas en inglés**).** Aplicación cliente que inicia una solicitud *SIP*.

***Servidor Agente Usuario* (*UAS***, por sus siglas en inglés**).** Aplicación Servidora que contacta el usuario cuando una solicitud *SIP* es recibida y retorna una respuesta en representación del usuario. La respuesta acepta, rechaza o redirecciona la solicitud.

***Servidor Proxy***. Un programa intermediario que actúa como servidor y como cliente para propósitos de hacer solicitudes en representación de otros clientes. Las solicitudes podrán realizarse también a otros servidores.

***Servidor de Redireccionamiento*.** Acepta la solicitud *SIP*, la asocia con una o más direcciones nuevas y retorna estas al cliente. No inicia solicitudes *SIP*.

***Servidor de Registro*.** Acepta solicitudes de registro, actualizándose sobre la localización corriente de los usuarios, por lo que también se conoce como servidor de localización.

Además han de tenerse en cuenta algunas definiciones importantes como:

[20] El estándar internacional **ISO/IEC 10646** define el **Conjunto de Caracteres Universal** (denominado también en inglés como: **Universal Character Set** - UCS) como un sistema codificación de caracteres en varios octetos. Tomado de: es.wikipedia.org/wiki/ISO_10646

- ***Sesión multimedios***. Es un conjunto de emisores y receptores multimedios y los flujos de datos que fluyen desde los emisores hasta los receptores. Una conferencia multimedios es un ejemplo de una sesión multimedios. En una sesión pueden existir una o más sesiones *RTP*.
- ***Conferencia*.** Una sesión multimedios, identificada por una descripción de sesión común. Puede tener cero o más miembros, incluye conferencia *multicast*, conferencia completamente mallada y una "llamada telefónica" de dos partes, así como combinaciones de las variantes anteriores. Cualquier número de llamadas pueden utilizarse para crear una conferencia.

**Arquitectura *SIP*.**

*El* ***anexo3 figura 8****,* muestra la arquitectura de *SIP*, donde son reconocidos varios de los elementos definidos antes como *servidor de registro*, *servidor proxy* y *servidor de redireccionamiento*. Se dice también que sólo existen dos tipos de servidores, los dos últimos mencionados pues los servidores de registro o localización típicamente aparecen junto a los anteriores dándoles apoyo.

**2.3.1.15.2 Mensajes.**

*SIP* posee dos tipos de mensajes: solicitud y respuesta. En el primer tipo la solicitud es desde el cliente al servidor, los mensajes de respuestas son en sentido contrario. Los mensajes están compuestos de una línea de aviso, uno o más campos de encabezamiento, y una línea vacía, es decir, una línea sin nada que preceda a un *CRLF* que indica la terminación de los campos de encabezamiento. Luego podrá encontrarse un cuerpo de mensaje que es opcional. Los encabezamientos que describen el cuerpo del mensaje son denominados como entidades. La ***figura 3*** muestra la estructura genérica de un Mensaje *SIP*.

| Línea de Inicio | Encabezamiento del Mensaje | CRLF | Cuerpo del Mensaje |
|---|---|---|---|

El formato del mensaje de solicitud es el siguiente:

Solicitud = Línea de Solicitud

(Encabezamiento General/Encabezamiento Solicitud/Encabezamiento Entidad)

CRLF


Cuerpo del Mensaje

Figura 3b. Arquitectura SIP

**2.3.1.15.3 Métodos**

Los métodos que no son soportados por un *proxy* o un servidor de redirección son tratados como si fueran un método *OPTIONS* y reexpedidos correspondientemente. Aquellos que no son soportados por un servidor agente usuario o servidor registrador retorna una respuesta 501 (Not Implemented).

Los métodos son:

- ***INVITE*****:** Invita a un usuario o servicio a participar en una sesión. El cuerpo del mensaje contiene una descripción de la sesión a la cual el llamado está siendo invitado. El llamador indica el tipo de medios que es capaz de recibir y posiblemente los medios que está deseando enviar y sus parámetros tales como destino de red. Una respuesta exitosa debe indicar qué medio el llamado desea recibir y posiblemente los que está enviando.
- ***ACK*****:** La solicitud *ACK* confirma que el cliente ha recibido una respuesta final para una solicitud *INVITE*. Respuestas *2XX* son reconocimientos de los agentes usuarios clientes, todas las otras respuestas son recibidas por el primer *proxy* o agente usuario cliente
- ***OPTIONS*****:** El servidor está siendo interrogado para sus capacidades. Un servidor que considera que puede contactar al usuario, tal como un agente usuario, puede responder a esta solicitud con un juego de capacidades. Un agente usuario llamado puede retornar un estado que refleje cómo este hubiera respondido a una invitación. Ejemplo 600 (Ocupado). Tal como un servidor debe retornar un encabezamiento *Allow* que indica los métodos que este soporta, los *proxys* y servidores de redirección simplemente reenvían la solicitud sin indicar sus capacidades.
- ***BYE*****:** El cliente agente usuario utiliza *BYE* para indicar al servidor que desea liberar la llamada. Puede provenir del llamador o del llamado.

- ***CANCEL*:** Cancela una solicitud pendiente, pero no afecta una solicitud completada (cuando el servidor ha retornado una respuesta de estado final)
- ***REGISTER*:** Un cliente utiliza este método para registrar la dirección listada en el campo *To header* con un servidor *SIP*. Un agente usuario puede registrar con un servidor local en el inicio enviando una solicitud *REGISTER* a todos las direcciones *multicast* de los servidores *SIP*.
- ***Request-URI*.** Es una *SIP URL* o una *URI* general. Indica el usuario o servicio al cual esta solicitud está siendo direccionada. A diferencia del campo *To*, la *Solicitud-URI* debe ser reescrita por los *proxys*.

Típicamente, la *UAC* fija la *Solicitud-URI* y *To* a la misma *SIP URL*, asumiendo que permanezca sin cambiar por un largo período de tiempo. Sin embargo, si la *UAC* ha depositado una trayectoria más directa para el llamado, ejemplo desde el campo del encabezamiento *Contact* de una respuesta a una solicitud previa, el campo *To* pudiera contener aún la dirección pública de término largo, mientras la *Solicitud-URI* pudiera colocar la dirección depositada.

**Versión *SIP*.** Mensajes de solicitud y de respuesta incluyen la sesión *SIP* en uso.

**Mensaje de Respuesta**.

Luego de recibir e interpretar un mensaje de solicitud, el receptor responde con un mensaje de respuesta *SIP*, con el siguiente formato:

Respuesta = Línea de Estado (Encabezamiento General, Encabezamiento Respuesta, Encabezamiento Entidad)
CRLF
Cuerpo del mensaje

**Línea de Estado.** Es la primera línea del Mensaje de Respuesta, como se observa en la ***figura 4,*** consiste en la versión del protocolo, seguida por un Código de Estado y sus frases textuales asociadas con cada elemento separado por espacio.

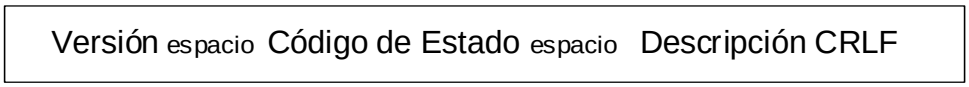



Figura 4. Línea de Estado correspondiente a un Mensaje de Respuesta.

El Código de Estado es un entero de tres dígitos que indica el resultado de los intentos para entender y satisfacer la solicitud, el primero de los cuales proporciona la clase. En la ***Tabla 2.1***, se muestran las seis Clases de Códigos de Estados existentes. En la ***Tabla 2.2*** se presentan algunos ejemplos concretos. En Descripción se dan cortas descripciones del Código de Estado.

| Clase | Propósito |
|---|---|
| 1XX | Informativo. Solicitud recibida, continuación del proceso |
| 2XX | Exitosa. La acción fue satisfactoriamente recibida, entendida y aceptada |
| 3XX | Redirección. Próxima acción necesitada para completar la solicitud |
| 4XX | Error del Cliente. La solicitud con sintaxis errónea o no puede ser completada en este servidor |
| 5XX | Error del Servidor. El servidor falló al completar una aparentemente válida solicitud. |
| 6XX | Fallo Global. La solicitud no pudo ser atendida en ningún servidor |

Tabla 2.1**.** Clases de Códigos de Estado

| Clase | Cód.Est. | Descripción |
|---|---|---|
| Informativas | 100 | Intentando |
| | 180 | imbre |
| | 182 | En cola |
| Exitosa | 200 | OK |

Tabla 2.2. Códigos de Estado Informativo y Exitoso

**Invitación *SIP*.**

Una invitación *SIP* satisfactoria consiste de dos solicitudes, *INVITE* donde se invita al llamado a unirse a una conferencia o a una conversación, cuando el llamado acepta, entonces el llamador confirma con *ACK*. Cuando este último no desea participar más en la llamada entonces envía una solicitud *BYE*.

La solicitud *INVITE* contiene una descripción de la sesión, de manera que la parte llamada recibe información suficiente para unirse a la sesión. La solicitud está en dependencia del tipo de servidor que participe en la llamada. Si se trata de un servidor *proxy*, es este quien realiza la solicitud una vez que la parte llamadora le indique que desea establecer la llamada, pero si es un servidor de direccionamiento luego de algunas operaciones devuelve al usuario la información para que de forma directa se encargue de contactar el terminal deseado.

Los servidores realizan operaciones internas para obtener información más precisa acerca de la dirección donde localizar al usuario llamado, esto se realiza consultando el servidor de registro.

Para actualizar un servidor de registro el cliente lanza una solicitud de registro en un mensaje de multidifusión a la dirección *sip.mcast* por toda conocida. En el caso de un servidor *proxy* los pasos a seguir para el establecimiento de la llamada son representados en el ***anexo 3 figura 9 y 10.***

## 2.4 Procedimiento para el procesamiento de los protocolos *H.323* y *SIP*

Una vez que se dio una explicación del funcionamiento de las redes y las características de los protocolos que funcionan en ellas, estamos en condiciones de realizar un procedimiento para el análisis de los protocolos ***H.323*** y ***SIP***. Para llegar al análisis de los mismos es necesario comprender la relación que existe entre los protocolos ***Ethernet***, ***IP*** y ***TCP*** o ***UDP***, además de conocer cómo es el establecimiento y funcionamiento de las sesiones creadas por la capa de transporte.

Es importante señalar que el protocolo *H.323* utiliza tanto *TCP* como *UDP.* Como es un protocolo de comunicación necesita de *TCP* para el establecimiento de sesiones entre terminales y de *UDP* para el intercambio de datos. Mientras que el protocolo *SIP* sólo utiliza *UDP*.

La importancia de este análisis radica en la información que recoge, la cual será de gran utilidad para entender cómo funcionan los protocolos. Además servirá como guía a todo aquel que le interese el tema, es decir todo el que esté interesado en saber cómo funcionan los protocolos *H.323* y *SIP* y no tenga ningún conocimiento de los mismos.

El procedimiento comienza haciendo una captura real de un paquete que viaja por la red para empezar a analizarlo y ver toda la información que trae consigo. Precisamente por ahí, es por donde comenzará nuestro procedimiento.

### 2.4.1 Obtener paquete de red.

El primer paso del procedimiento radica en la captura de un paquete que viaja por la red, el cual contiene parte de la información que se quiere enviar. Un paquete es uno de los bloques en que se dividen, en el nivel de Red, la información a enviar. En los sistemas de comunicación resulta

interesante dividir la información a enviar en bloques más pequeños. Esto simplifica el control de la comunicación, las comprobaciones de errores, la gestión de encaminamiento, etc.

Los paquetes pueden estar formados por una cabecera, una parte de datos y una cola. En la cabecera estarán los campos que pueda necesitar el protocolo de nivel de red, en la cola, si la hubiere, se ubica normalmente algún mecanismo de comprobación de errores. En las redes de datagramas no suele haber cola, porque no se comprueban errores, quedando esta tarea para el nivel de transporte.

Esta captura del paquete es realizada a nivel de la capa Física donde se encuentra implementado el protocolo ***Ethernet***.

Una vez capturado el paquete se puede verificar cuál es el protocolo de la capa Internet que está contenido en el mismo. Esta información está dada gracias a la trama ***Ethernet***.

### 2.4.2 Verificar si es *IP*

En este paso lo importante es conocer si es el protocolo *IP* el que está en la trama *Ethernet*. Para esto se tiene que hacer un análisis de la misma. El encabezado de *Ethernet* tiene varias secciones de información que el protocolo *Ethernet* utiliza. Cada sección de la trama se denomina campo. Hay dos estilos de tramas de *Ethernet*: el *IEEE* 802.3 (original) y el *IEEE* 802.3 revisado (*Ethernet*).

Las diferencias entre los estilos de tramas son mínimas. La diferencia más significativa entre el *IEEE* 802.3 (original) y el *IEEE* 802.3 revisado es el agregado de un delimitador de inicio de trama (*SFD*, por sus siglas en inglés) y un pequeño cambio en el campo Tipo que incluye la Longitud, tal como se muestra en la ***figura7***.


Figura 4. Comparación del tamaño del campo y las estructuras de trama de *Ethernet* y 802.3

### 2.4.2.1 Descripción de los campos de la trama *Ethernet* (*IEEE* 802.3)

- **Preámbulo** (compuesto por 7 *bytes*) y **Delimitador de inicio de trama** (compuesto por 1byte), se utilizan para la sincronización entre los dispositivos de envío y de recepción. Estos ocho primeros *bytes* de la trama se utilizan para captar la atención de los nodos receptores. Básicamente, los primeros bytes le indican al receptor que se prepare para recibir una trama nueva.
- **Dirección de destino** (dirección MAC de destino, compuesto por 6 *bytes*) es el identificador del receptor deseado.
- **Dirección de origen** (dirección MAC de origen, compuesto por 6 bytes) identifica la NIC o interfaz que origina la trama. En decir es el identificador del emisor.
- **Longitud/Tipo** (compuesto por 2 *bytes*) define la longitud exacta del campo Datos de la trama. En este campo debe ingresarse una longitud o un tipo. Sin embargo, sólo uno u otro podrá utilizarse en una determinada implementación. Si el objetivo del campo es designar un tipo, el campo Tipo describe qué protocolo se implementa.

  Dicho campo solo aparecerá como Longitud en las versiones anteriores del *IEEE* y como Tipo en la versión *DIX*. Ambos usos del campo se combinaron en una versión posterior del *IEEE*, ya que eran comunes. El campo Tipo de la *Ethernet* II se incorporó a la actual

definición de trama del 802.3. La *Ethernet II* es el formato de trama de *Ethernet* que se utiliza en redes *TCP/IP*.

Cuando un nodo recibe una trama, debe analizar el campo **Longitud/Tipo** para determinar qué protocolo de capa superior está presente. El protocolo que lleva los contenidos codificados en el campo Datos se identifica con un número, de esta manera podemos saber cuál es. Varios números que identifican a los Protocolos son:

- ✓ 0800 >>>>>>>>>Protocolo IP
- ✓ 0806 >>>>>>>>>Protocolo ARP
- ✓ 0600 >>>>>>>>>Protocolo XNS
- ✓ 814C >>>>>>>>>Protocolo SNMP
- ✓ 8137 >>>>>>>>>Protocolo IPX
- ✓ 8138 >>>>>>>>>Protocolo NOVELL
- ✓ 86DD >>>>>>>>Protocolo IPNG

- **Datos y Relleno** (de 46 a 1500 bytes) contienen los datos encapsulados de una capa superior, que es una *PDU* de Capa 3 genérica o, con mayor frecuencia, un paquete *IPv4*. Las tramas deben tener al menos 64 *bytes* de longitud. Si es encapsulado un paquete pequeño, el *Pad* se utiliza para aumentar el tamaño de la trama hasta alcanzar este tamaño mínimo.
- **Secuencia de verificación de trama (*FCS*)** (compuesto por 4 *bytes*) se utiliza para detectar errores en la trama. Utiliza una comprobación cíclica de redundancia (CRC). El dispositivo emisor incluye los resultados de una *CRC* en el campo *FCS* de la trama. El dispositivo receptor recibe la trama y genera una *CRC* (basada en los contenidos de la trama recibida) para detectar errores. Si los cálculos coinciden, significa que no se produjo ningún error. Los cálculos que no coinciden indican que los datos cambiaron y, por consiguiente, se descarta la trama. Un cambio en los datos podría ser resultado de una interrupción de las señales eléctricas que representan los bits.

Después de haber visto una descripción de cada campo de la trama *Ethernet*, vamos a centrarnos específicamente en el campo ***Longitud/Tipo***. Según el valor que contenga el campo,

será el protocolo que traiga consigo. Si el valor presente es el 0800 eso indica que el protocolo que implementa es *IP*, el cual estamos verificando. Actualmente el más usado es el *IPv4*, aunque en un futuro el *IPv6* (que no es objetivo del presente trabajo) podrá sustituirlo. En caso que no sea ese el valor entonces es otro el protocolo que implementa que no sería objetivo en este trabajo y se pasaría a la captura de un nuevo paquete. Como el campo ***Longitud/Tipo*** es el más importante de la trama *Ethernet*, la información que brinda se podría almacenar en una base de datos.

### 2.4.3 Preparar los datos para la sesión (Análisis del protocolo *IPv4*).

Una vez que se sabe que el protocolo que está implementado en la trama *Ethernet* es *IP*, entonces se procede al análisis del mismo, para saber cuál es el protocolo de transporte que trae consigo.

El protocolo *IP* esta implementado en la capa Internet del modelo *TCP/IP* la cual provee servicios para intercambiar secciones de datos individuales a través de la red entre dispositivos finales identificados. Su objetivo es lograr que los paquetes sean direccionados de una forma correcta.

El Protocolo de *Internet* provee sólo las funciones necesarias para enviar paquetes desde un origen a un destino a través de un sistema interconectado de redes. Estos paquetes presentan una cabecera y sus datos correspondientes, aunque la cabecera y los datos son diferentes uno de otros viajan juntas como uno solo en la red. La ***figura5*** muestra una cabecera *IP* con sus campos correspondientes.

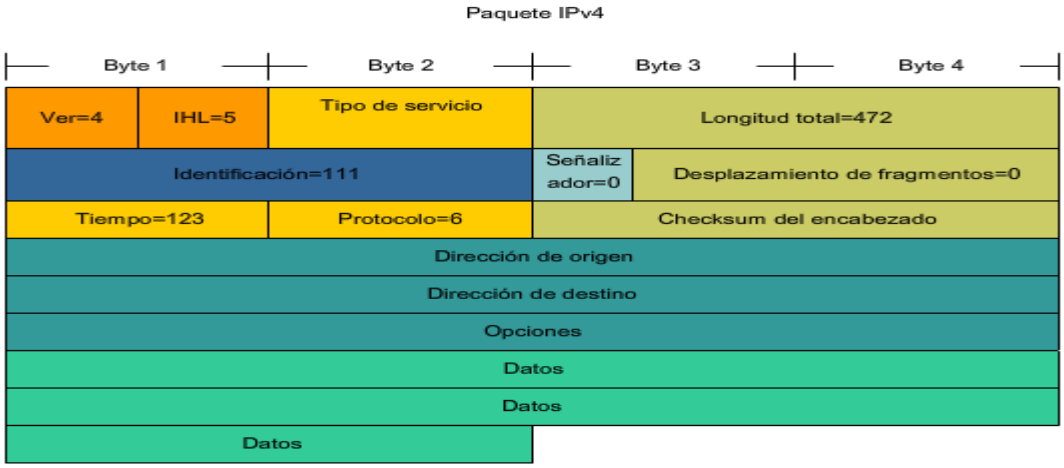



Figura 5. Estructuras del paquete IP.

### 2.4.3.1 Descripción de los campos del encabezado del paquete *IPv4*

Un protocolo *IPv4*, define campos diferentes en el encabezado del paquete. Estos campos contienen valores binarios que los servicios *IPv4* de 8 *bits*, toman como referencia a medida que envían paquetes a través de la red. Los campos más importantes o claves de la cabecera son:

- **Dirección *IP* origen:** El campo de Dirección *IP* origen contiene un valor binario de 32 *bits* que representa la dirección de host de capa de red de origen del paquete.
- **Dirección *IP* destino:** El campo de Dirección *IP* destino contiene un valor binario de 32 *bits* que representa la dirección de *host* de capa de red de destino del paquete.
- **Tiempo de existencia (*TTL, por sus siglas en inglés*):** El tiempo de vida (*TTL*, por sus siglas en inglés) es un valor binario de 8 bits que indica el tiempo remanente de "vida" del paquete. El valor *TTL* disminuye al menos en uno cada vez que el paquete es procesado por un *router* (es decir, en cada salto). Cuando el valor se vuelve cero, el router descarta o elimina el paquete y es eliminado del flujo de datos de la red. Este mecanismo evita que los paquetes que no pueden llegar a destino sean enviados indefinidamente entre los *routers*.
- **Tipo de servicio (*ToS*, por sus siglas en inglés):** El campo de tipo de servicio contiene un valor binario de 8 bits que se usa para determinar la prioridad de cada paquete. Este valor permite aplicar un mecanismo de Calidad del Servicio (*QoS*) a paquetes de alta

prioridad, como aquellos que llevan datos de voz en telefonía. El router que procesa los paquetes puede ser configurado para decidir qué paquete es enviado primero basado en el valor del Tipo de servicio.

- **Protocolo:** Este valor binario de 8 *bits* indica el tipo de relleno de carga que el paquete traslada, o sea saber qué protocolo es el indicado dependiendo del número que se tenga. El campo de protocolo permite a la capa Internet pasar los datos al protocolo apropiado de la capa superior. Los valores de ejemplo son: 01 que representa al protocolo *ICMP* (por sus siglas en inglés), 06 a *TCP* (por sus siglas en inglés), 17 a *UDP* (por sus siglas en inglés), etc.
- **Desplazamiento del fragmento:** Cuando se produce una fragmentación, el paquete *IPv4* utiliza el campo Desplazamiento de fragmento y el señalizador *MF* en el encabezado *IP* para reconstruir el paquete cuando llega al host destino. El campo de desplazamiento del fragmento identifica el orden en el cual ubicar el fragmento del paquete en la reconstrucción.

Claro está que estos no son los únicos campos que conforman la cabecera ***IPv4*** aunque son los más importantes. Existen otros campos los cuales se verán a continuación:

**Otros Campos que conforman el encabezado *IPv4*.**

- **Versión**: Contiene el número *IP* de la versión (4).
- **Longitud del encabezado (*IHL***, por sus siglas en inglés). Especifica el tamaño del encabezado del paquete.
- **Longitud del Paquete**: Este campo muestra el tamaño completo del paquete, incluyendo el encabezado y los datos, en *bytes*.
- **Identificación**: Este campo es principalmente utilizado para identificar únicamente fragmentos de un paquete *IP* original.

- **Checksum del encabezado**: El campo de *checksum*[21] se utiliza para controlar errores del encabezado del paquete.
- **Opciones**: Existen medidas para campos adicionales en el encabezado *IPv4* para proveer otros servicios pero éstos son rara vez utilizados.

De los campos más importantes antes mencionados, se podrían almacenar en una base de datos los valores de algunos de ellos como son: Dirección *IP* origen, Dirección *IP* destino, Tiempo de existencia, Protocolo, Longitud del Paquete, Identificación y Longitud del encabezado.

Hemos visto los campos más importantes que conforman la cabecera del protocolo ***IPv4***. Entre ellos hay un campo, que dependiendo de la información que contenga, será el protocolo que trae consigo. El campo es el denominado Protocolo. Existen dos protocolos de transporte que son los más usados para la transferencia de datos en la red, ellos son ***TCP*** y ***UDP***.

### 2.4.4 Verificar tipo de protocolo para el transporte de los datos (*TCP* y *UDP*)

Los dos protocolos más comunes de la capa Transporte del conjunto de protocolos *TCP/IP* son el Protocolo de control de transmisión (***TCP***) y el Protocolos de datagramas de usuario (***UDP***). Ambos protocolos gestionan la comunicación de múltiples aplicaciones. Las diferencias entre ellos son las funciones específicas que cada uno implementa. Como son los protocolos más usados para el transporte de datos, en la mayoría de los paquetes *IP* en el campo Protocolo puede venir identificado cualquiera de los dos. Si dicho campo el valor que contiene es el 06, quiere decir que el protocolo que viene a continuación es el ***TCP***, si en cambio, el valor es 17, el que continua es el protocolo ***UDP***. Si no es ninguno de los valores antes mencionados, entonces es otro el protocolo usado para transportar los datos. En este trabajo los protocolos de transporte de datos que se tendrán en cuenta serán ***TCP*** y ***UDP***.

[21] Una **suma de verificación** o **checksum** es una forma de control de redundancia, una medida muy simple para proteger la integridad de datos, verificando que no hayan sido corruptos. Tomado de: es.wikipedia.org/wiki/Checksum

### 2.4.4.1 Protocolo de transporte de datos *UDP*.

Una vez verificado el campo protocolo de la cabecera ***IPv4*** y que este campo tenga como contenido el número 17, se sabe que el protocolo que se usara para el transporte de los datos es *UDP*, este es un protocolo simple, sin conexión, descrito en la *RFC* 768. Cuenta con la ventaja de proveer la entrega de datos sin utilizar muchos recursos. Las porciones de comunicación en *UDP* se llaman datagramas. Este protocolo de la capa de Transporte envía estos datagramas como "mejor intento", el cual se vio anteriormente. Entre las aplicaciones que utilizan *UDP* se incluyen, sistema de nombres de dominios (*DNS*), streaming de vídeo y Voz sobre IP (*VoIP*). En la ***figura 6*** se muestra la cabecera del protocolo *UDP* con el objetivo de explicar en qué consiste cada uno de sus campos.


Figura6. Datagrama *UDP*.

#### 2.4.4.1.1 Descripción de los campos del datagrama *UDP*

- **Números de Puerto de Origen y Destino:** Estos números, junto con las direcciones *IP* definen el inicio y el punto final de la comunicación. El número del puerto de origen, puede tener valor cero si no se usa. El número del puerto de destino solo tiene sentido en el contexto de un datagrama *UDP* y un a dirección *IP* en particular. El número de puerto de origen es un campo de 16 *bits* al igual que puerto de destino tiene la misma longitud.
  - **Puerto de origen:** Indica el puerto del proceso que envía. Este es el puerto que se direcciona en las respuestas.
  - **Puerto destino:** Especifica el puerto del proceso destino en el *host* de destino.
- **Longitud del Mensaje:** Este campo tiene una longitud de 16 *bits* y contiene el número total de octetos que forman el datagrama, incluida la cabecera.

- ***Checksum*:** El uso del *checksum* es opcional, y este campo debe ponerse a cero si no es utilizado. Mientras que el checksum del datagrama *IP* solo tiene en cuenta la cabecera del mensaje, el *UDP* tiene su propio checksum para garantizar la integridad de los datos. La longitud de este campo es de 16 *bits*, y está formado por la suma de los campos del *UDP*, y algunos campos del *IP*.

Una vez que se ha explicado cómo está compuesta una cabecera *UDP*, se debe conocer como es el funcionamiento del mismo usando la arquitectura cliente servidor, ya que a diferencia del protocolo *TCP*, *UDP* es un protocolo no orientado a la conexión.

#### 2.4.4.1.2 Procesos del cliente *UDP*

La comunicación cliente/servidor se inicia por una aplicación cliente que solicita datos de un proceso del servidor. El proceso de cliente *UDP* selecciona al azar un número de puerto del rango dinámico de números de puerto y lo utiliza como puerto de origen para la conversación. El puerto de destino por lo general será el número de puerto bien conocido o registrado asignado al proceso del servidor.

Los números de puerto de origen seleccionados al azar colaboran con la seguridad. Si existe un patrón predecible para la selección del puerto de destino, un intruso puede simular el acceso a un cliente de manera más sencilla intentando conectarse al número de puerto que tenga mayor posibilidad de estar abierto.

Ya que no se crean sesiones con *UDP*, tan pronto como los datos están listos para ser enviados y los puertos estén identificados, *UDP* puede formar el datagrama y enviarlo a la capa de Red para direccionamiento y envío a la red. Cabe recordar que una vez que el cliente ha elegido los puertos de origen y destino, estos mismos puertos se utilizarán en el encabezado de todos los datagramas que se utilicen en la transacción. Para la devolución de datos del servidor al cliente, se invierten los números de puerto de origen y destino en el encabezado del datagrama. Datos que se deben guardar en una Base de Datos, con el objetivo de procesarlos. A continuación se verá un ejemplo en la ***figura 7*** que ilustra cómo se realiza una solicitud de un cliente a un servidor y éste da una respuesta inmediata con el objetivo de satisfacer esa petición.

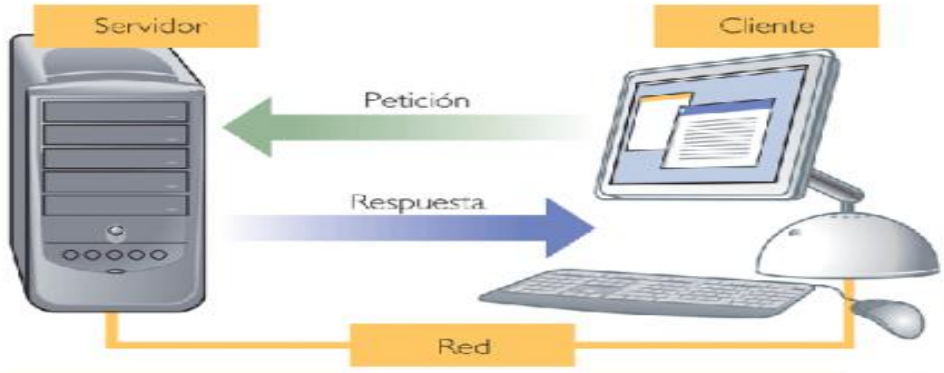



Figura 7. Cliente-Servidor UDP.

Entre los campos del datagrama *UDP*, el que define qué protocolo de capa de aplicación está presente es Puerto destino. Existe una serie de puertos que determinan varios servicios, en nuestro trabajo los protocolos que se explicarán son *H.323* y *SIP* cuyos valores de puertos que los identifican son 1720 y 5060 respectivamente.

Si el valor que presenta el campo Puerto destino cuando un cliente hace una solicitud al servidor es 5060 estamos en presencia de una petición de activación del servicio de Voz sobre *IP*, que es atendido por el protocolo de Iniciación de Sesiones (*SIP),* que se detallara a continuación.

#### 2.4.4.1.3 Análisis del protocolo *SIP*

*SIP* es un protocolo textual que usa una semántica semejante a la del protocolo *HTTP* y utiliza para el establecimiento de la sesión al protocolo *UDP;* es un protocolo de la capa de aplicación. Los *UAC* (agente usuarios cliente, por sus siglas en inglés) realizan las peticiones y los *UAS* (agente usuarios servidores, por sus siglas en inglés) retornan respuestas a las peticiones de los clientes. *SIP* define la comunicación a través de dos tipos de mensajes. Las solicitudes (métodos) y las respuestas (códigos de estado) emplean el formato de mensaje genérico establecido en el *RFC* 2822, que consiste en una línea inicial seguida de uno o más campos de cabecera , una línea vacía que indica el final de las cabeceras, y por último, el cuerpo del mensaje que es opcional.

Para entender cómo es el funcionamiento del protocolo *SIP* nos apoyaremos en una llamada donde intervienen varias transacciones *SIP*. Una transacción *SIP* se realiza mediante un intercambio de mensajes entre un cliente y un servidor. Consta de varias peticiones y respuestas y para agruparlas en la misma transacción está el parámetro *CSeq.* Una transacción se compone por la influencia de varios métodos o tipos de mensajes. Para entender mejor los métodos se explicarán a continuación.

**Métodos *SIP*:**

Las peticiones *SIP* son caracterizadas por la línea inicial del mensaje, llamada Request-Line, que contiene el nombre del método, el identificador del destinatario de la petición (Request-URI) y la versión del protocolo *SIP*. Existen seis métodos básicos *SIP* (definidos en RFC 254) que describen las peticiones de los clientes estos son usados con el objetivo de lograr una eficiente comunicación, los cuales se explicaron anteriormente ellos son ***INVITE, ACK, OPTION, BYE, CANCEL, REGISTER***.

Para conocer el funcionamiento de los mensajes o métodos *SIP* se hace necesario conocer al detalle una llamada *SIP* usando teléfonos IP. En una llamada *SIP* existen varias fases o pasos donde intervienen diferentes protocolos en cada una de ellas. Cada paso es fundamental para el funcionamiento del inicio de sesión.

**2.4.4.1.4 Pasos para realizar una llamada *SIP***

**Paso1:** Registrar los usuarios.

**Paso2:** Una vez registrado los usuarios se establece la llamada.

**Paso3:** Envió de datos de Voz.

**Paso4:** Finalización de la sesión.

Se muestra una figura de una llamada *SIP* con el objetivo de visualizar el flujo de mensajes existente entre ambas entidades (Usuario A y E).

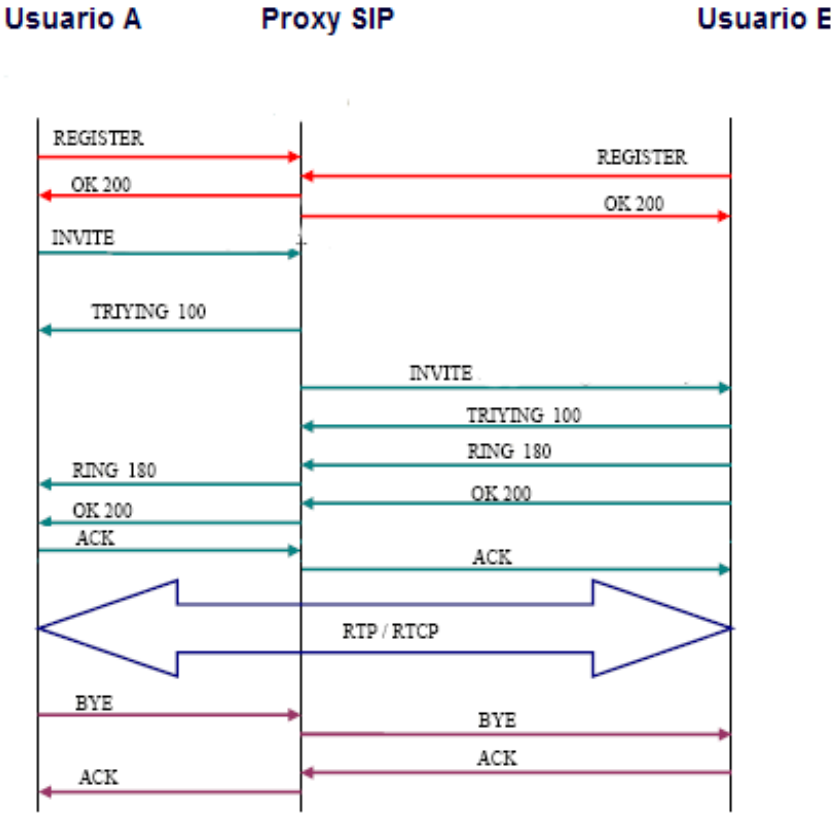



Figura 8. Llamada *SIP*, métodos que participan en ella

**Paso1: Registrar los usuarios**

Los usuarios deben registrarse para poder ser encontrados por otros usuarios, usando el método *REGISTER.* En este caso, los terminales envían una petición *REGISTER* donde los campos ***from*** y ***to*** corresponden al usuario registrado. El servidor *Proxy*, que actúa como *REGISTER*, consulta si el usuario puede ser autenticado y envía un mensaje de *OK* en caso positivo; o sea, para que dos usuarios puedan comunicarse deben primeramente registrarse pero este no es el único tipo de mensaje existen muchos más.

**Captura de los diferentes métodos utilizados por *SIP***

Se usó el *WireShark* como herramienta de análisis de los protocolos y de los mensajes que en él intervienen, específicamente en el capítulos 3 en las pruebas de laboratorio se muestra la captura

de diferentes tramas y dentro de éstas se señala el método que se utilizó y los *bits* que lo representan para un mayor análisis del método.

**Paso2: Una vez registrados los usuarios, se establece la llamada.**

Una vez que los usuarios se registraron y el servidor les manda un mensaje de ok, se establece una ***sesión***. Esta sesión consiste en una petición *INVITE* del usuario al servidor proxy. Inmediatamente, el *proxy* envía un *TRYING* 100 para parar las retransmisiones porque está haciendo una llamada y reenvía la petición al usuario B. El usuario B envía un *Ringing* 180 cuando el teléfono empieza a sonar y también es reenviado por el proxy hacia el usuario A. Por último, el *OK* 200 corresponde a aceptar la llamada y el usuario B descuelga el teléfono se utiliza el *ACK* con el objetivo de verificar que acepta la llamada de un usuario a otro, es una forma de verificar que se quiere aceptar la llamada.

**Paso3: Envió de datos de Voz.**

Una vez que el usuario A realiza la llamada al usuario B y toda la llamada se realiza de manera eficiente, el proceso que se desarrolla es el envío de datos de voz, este envío se hace mediante el protocolo de transporte *RTP* con los parámetros (puertos, direcciones, *codecs*, etc.) establecidos en la negociación mediante el protocolo *SDP* y además se utiliza el protocolo *RTCP.* Estos protocolos son importantes ya que *SIP* los utiliza con el objetivo de lograr mejoras en su funcionamiento, pero *SIP* para su funcionamiento no depende de ellos.

**Paso4: Finalización de la sesión.**

Ya establecida la llamada de manera correcta y que se transmitan los datos de audio y video, lo que queda es finalizar la sesión o sesiones que se han creado, esta finalización se lleva a cabo con una única petición *BYE* enviada al *Proxy* y posteriormente reenviada al usuario B. Este usuario contesta con un mensaje *OK* 200 para confirmar que se ha recibido el mensaje final correctamente, usando además el *ACK* para confirmar que se desea terminar.

De esta manera ha quedado plasmado cómo es el funcionamiento del protocolo *SIP*. Es importante guardar en una base de datos los campos método, vía, id de la llamada, *from*, *to*, por la información valiosa que contienen.

Recordemos que el protocolo que se analizó fue *UDP* porque en al campo Protocolo de la cabecera *IP* tenía el valor 17, en caso de que el valor contenido en dicho campo sea 06 entonces el protocolo de transporte a analizar es *TCP*, que se verá a continuación.

#### 2.4.4.2 Protocolo de transporte de datos *TCP*.

Una vez verificado que es el protocolo *TCP*, o sea que el valor del campo Protocolo de la cabecera *IP* es 06, procedemos al análisis del mismo.

Con el uso del protocolo *TCP*, las aplicaciones pueden comunicarse en forma segura, gracias al sistema de acuse de recibo del protocolo *TCP,* independientemente de las capas inferiores. Esto significa que los *routers* (que funcionan en la capa de Internet) sólo tienen que enviar los datos en forma de datagramas, sin preocuparse por el monitoreo de datos porque esta función la cumple la capa de transporte o más específicamente, el protocolo *TCP*.

Durante una comunicación usando el protocolo *TCP*, las dos máquinas deben establecer una conexión, ya que el protocolo *TCP* es un protocolo orientado a la conexión. La máquina emisora (la que solicita la conexión) se llama cliente, y la máquina receptora se llama servidor. Por eso es que decimos que estamos en un entorno Cliente-Servidor. Las máquinas de dicho entorno se comunican en modo en línea, es decir, que la comunicación se realiza en ambas direcciones.

El segmento *TCP* contiene un conjunto de campos que recogen información valiosa para transportar los datos y está formado de la siguiente manera:

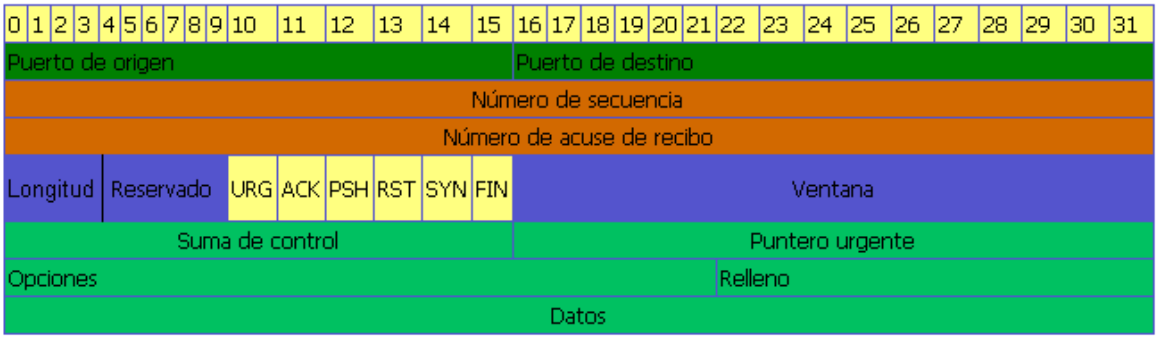



Figura 9. Segmente TCP con sus campos.

#### 2.4.4.2.1 Descripción de los campos del segmento *TCP*.

- **Puerto de origen** (16 *bits*): Puerto relacionado con la aplicación en curso en la máquina origen
- **Puerto de destino** (16 *bits*): Puerto relacionado con la aplicación en curso en la máquina destino
- **Número de secuencia** (32 *bits*): Cuando el indicador *SYN* está fijado en 0, el número de secuencia es el de la primera palabra del segmento actual. Cuando *SYN* está fijado en 1, el número de secuencia es igual al número de secuencia inicial utilizado para sincronizar los números de secuencia (*ISN*).
- **Número de acuse de recibo** (32 *bits*): El número de acuse de recibo, también llamado número de descargo se relaciona con el número (secuencia) del último segmento esperado y no el número del último segmento recibido.
- **Longitud** (4 *bits*): Esto permite ubicar el inicio de los datos en el paquete. Aquí, el margen es fundamental porque el campo opción es de tamaño variable.
- **Reservado** (6 *bits*): Un campo que actualmente no está en uso pero se proporciona para el uso futuro.
- **Indicadores** (6x1 *bit*): Los indicadores representan información adicional:
  - **URG**: Si este indicador está fijado en 1, el paquete se debe procesar en forma urgente.
  - **ACK**: Si este indicador está fijado en 1, el paquete es un acuse de recibo.

- **PSH** (PUSH): Si este indicador está fijado en 1, el paquete opera de acuerdo con el método PUSH.
- **RST**: Si este indicador está fijado en 1, se restablece la conexión.
- **SYN**: El indicador SYN de TCP indica un pedido para establecer una conexión.
- **FIN**: Si este indicador está fijado en 1, se interrumpe la conexión.

- **Ventana** (16 bits): Campo que permite saber la cantidad de *bytes* que el receptor desea recibir sin acuse de recibo.
- **Suma de control** (CRC): La suma de control se realiza tomando la suma del campo de datos del encabezado para poder verificar la integridad del encabezado.
- **Puntero urgente** (16 *bits*): Indica el número de secuencia después del cual la información se torna urgente.
- **Opciones** (tamaño variable): Diversas opciones.
- **Relleno:** Espacio restante después de que las opciones se rellenan con ceros para tener una longitud que sea múltiplo de 32 bits.

Hemos visto los diferentes campos que componen el segmento ***TCP*** y la información que recogen. A continuación veremos cómo es el funcionamiento del protocolo. Como *TCP* es un protocolo orientado a la conexión, antes de la comunicación entre las dos maquinas, es necesario establecer una conexión entre ambas, que no es más que una sesión *TCP*, la cual será analizada.

Antes de pasar al análisis de la sesión *TCP*, es importante conocer la longitud de los datos que trae el paquete, que se calcula de la siguiente manera:

Longitud de los Datos = Longitud Total – (Longitud del paquete IP + Longitud del paquete TCP o Longitud del paquete UDP).

**Nota: A la longitud del paquete *IP* se le suma la longitud de un protocolo u otro dependiendo de cuál sea el protocolo de transporte que trae el paquete.**

Longitud Total: es un campo del paquete *IP*.

Longitud del paquete *IP*: en la mayoría de los casos es 20 *bytes*, se calcula multiplicando el contenido del campo **Longitud del encabezado (IHL) * 4,** IHL es un campo de la cabecera *IP*.

Longitud del paquete *TCP*: campo del encabezado *TCP*.
Longitud del paquete *UDP*: campo del encabezado *UDP*.
Una vez calculada la longitud si el valor que contiene no es cero, quiere decir que es un paquete que contiene datos lo cual indica que la sesión ya fue creada, de lo contrario puede ser un paquete de sincronismo (*SYN*), o de aceptación (*ACK*), para establecer una sesión, o de finalización de sesión (*FIN*), o de cualquier otro indicador. Una vez analizado esto procedemos al análisis de la sesión.

#### 2.4.4.2.2 Análisis de la sesión *TCP*.

El establecimiento de una sesión entre las aplicaciones es responsabilidad de la capa de Transporte ya que puede brindar esta orientación a la conexión. Una sesión no es más que una conexión que prepara a las aplicaciones para que se comuniquen entre sí antes de que se transmitan los datos. Los parámetros fundamentales que definen a una sesión son: dirección *IP* origen y destino, el puerto origen y destino y el identificador, que no es más que el número de secuencia del paquete. Dentro de estas sesiones, se pueden tratar de cerca los datos para la comunicación entre dos aplicaciones. Los propios mecanismos de conexión y de sesión habilitan la función de confiabilidad de *TCP*. Ahora veremos cómo se establece una conexión.

##### 2.4.4.2.2.1 Establecimiento de la conexión *TCP*

Cuando se comunican dos *host* utilizando *TCP*, antes de que puedan intercambiarse los datos se establece una conexión. Una vez finalizada la comunicación se cierra la sesión y la comunicación finaliza.
Para establecer la conexión los *hosts* realizan un intercambio de señales de tres vías. Los *bits* de control en el encabezado *TCP* (campo **Indicadores**) indican el progreso y estado de la conexión. Se realiza un intercambio de tres vías porque:

- Establece que el dispositivo de destino esté presente en la red.
- Verifica que el dispositivo de destino tenga un servicio activo y esté aceptando las peticiones en el número de puerto de destino que el cliente que lo inicia intente usar para la sesión.

- Informa al dispositivo de destino que el cliente de origen intenta establecer una sesión de comunicación en ese número de puerto.

En conexiones *TCP*, el *host* que brinde el servicio como cliente inicia la sesión al servidor. Los tres pasos para el establecimiento de una conexión *TCP* son:

1. El cliente que inicia la conexión envía un segmento que contiene un valor de secuencia inicial, que actúa como solicitud para el servidor para comenzar una sesión de comunicación.
2. El servidor responde con un segmento que contiene un valor de reconocimiento igual al valor de secuencia recibido más 1, además de su propio valor de secuencia de sincronización. El valor es uno mayor que el número de secuencia porque el *ACK* es siempre el próximo *Byte* u Octeto esperado. Este valor de reconocimiento permite al cliente unir la respuesta al segmento original que fue enviado al servidor.
3. El cliente que inicia la conexión responde con un valor de reconocimiento igual al valor de secuencia que recibió más uno, esto completa el proceso de establecimiento de la conexión.

Para entender el proceso de enlace de tres vías, es importante observar los valores que intercambian los dos *hosts*. Dentro del encabezado del segmento *TCP*, existen seis campos de 1 *bit* que contienen información de control utilizada para gestionar los procesos de *TCP*. Estos campos son: ***URG, ACK, PSH, RST, SYN, FIN,*** que se vieron anteriormente. Además nos apoyaremos en un ejemplo que explica cómo se establece una conexión.

Ejemplo de establecimiento de una sesión ***TCP***:

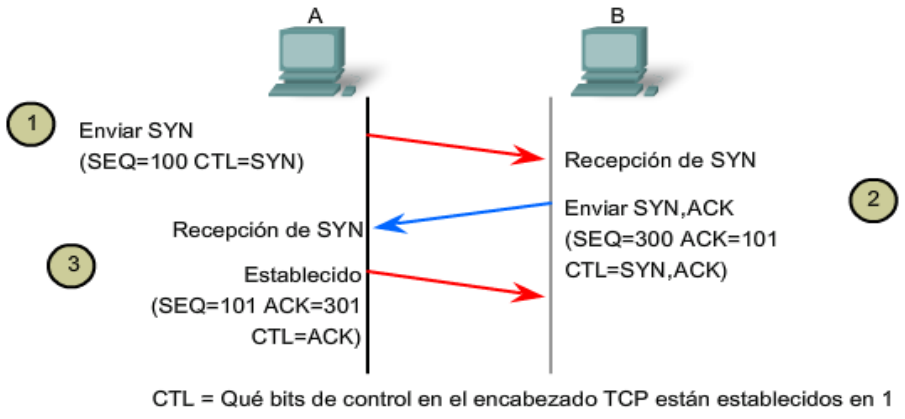



Figura 10. Establecimiento de una conexión TCP.

El funcionamiento detallado del protocolo *TCP* de enlace de tres vías quedaría de la siguiente forma:

**Paso 1:** Un cliente *TCP* comienza el enlace de tres vías enviando un segmento con el señalizador de control *SYN* (Sincronizar números de secuencia, por sus siglas en inglés) establecido, indicando un valor inicial en el campo de número de secuencia del encabezado. Este valor inicial para el número de secuencia, conocido como número de secuencia inicial (*ISN*, por sus siglas en inglés), se elige de manera aleatoria y se utiliza para comenzar a rastrear el flujo de datos desde el cliente al servidor para esta sesión. El *ISN* en el encabezado de cada segmento se incrementa en uno por cada *byte* de datos enviados desde el cliente hacia el servidor mientras continúa el intercambio de datos.

**Paso 2:** El servidor *TCP* necesita reconocer la recepción del segmento *SYN* del cliente para establecer la sesión de cliente a servidor. Para hacerlo, el servidor envía un segmento al cliente con el señalizador *ACK* establecido indicando que el número de acuse de recibo es significativo. Con este señalizador establecido en el segmento, el cliente interpreta esto como acuse de recibo de que el servidor ha recibido el *SYN* del cliente *TCP*.

El valor del número de campo del acuse de recibo es igual al número de secuencia inicial del cliente más 1. Esto establece una sesión desde el cliente al servidor. El señalizador *ACK* permanecerá establecido para mantener el equilibrio de la sesión. Cabe recordar que la conversación entre el cliente y el servidor está compuesta en realidad por dos sesiones de una vía: una del cliente al servidor y la otra del servidor al cliente. En este segundo paso del enlace de tres vías, el servidor debe iniciar la respuesta del servidor al cliente. Para comenzar esta sesión, el servidor utiliza el señalizador *SYN* de la misma manera en que lo hizo el cliente. Establece el señalizador de control *SYN* en el encabezado para establecer una sesión del servidor al cliente. El señalizador *SYN* indica que el valor inicial del campo de número de secuencia se encuentra en el encabezado. Este valor se utilizará para rastrear el flujo de datos en esta sesión del servidor al cliente.

**Paso 3:** Por último, el cliente *TCP* responde con un segmento que contiene un *ACK* que actúa como respuesta al *SYN* de *TCP* enviado por el servidor. No existen datos de usuario en este

segmento. El valor del campo número de acuse de recibo contiene uno más que el número de secuencia inicial recibido del servidor. Una vez establecidas ambas sesiones entre el cliente y el servidor, todos los segmentos adicionales que se intercambien en la comunicación tendrán establecido el señalizador *ACK*.

De esta manera queda establecida la sesión *TCP*. Es importante guardar en una base de datos los campos Puerto de origen, Puerto de destino, Número de secuencia y Longitud ya que recogen información importante del protocolo *TCP*. Los campos del segmento *TCP* que definen cual sería el protocolo de la capa de Aplicación son Puerto de origen y Puerto de destino. El número de puerto indica cual es la aplicación que se debe ejecutar en el servidor dependiendo de la petición que hace el cliente, o sea, el servicio que debe brindar el servidor.

Existen muchos números de puertos entre los que están:

- Puerto 20 >>>>>>>>> *FTP* -- Datos
- Puerto 21 >>>>>>>>> FTP -- Control
- Puerto 23 >>>>>>>>> Telnet
- Puerto 53 >>>>>>>>> *Domain Name System* (DNS)
- Puerto 80 >>>>>>>>> *HTTP*
- Puerto 109 >>>>>>>>> *POP2*
- Puerto 110 >>>>>>>>> *POP3*
- Puerto 443 >>>>>>>>> *HTTPS*

Para la realización de este trabajo los números de puertos que nos interesan son el 1720 que reconoce al protocolo *H.323* y el 5060 que reconoce al protocolo *SIP*.

Cuando un cliente envía una petición al servidor solicitando el servicio de comunicación de voz sobre *IP*, en el campo Puerto de destino del segmento *TCP* el valor que se encuentra es el 1720, mientras que en el Puerto de origen hay un valor aleatorio. De esta manera el servidor sabe qué servicio debe activar; el servicio de comunicación de voz sobre *IP*, utilizando el protocolo *H.323* el cual analizaremos a continuación.

#### 2.4.4.2.2.2 Análisis del protocolo *H.323*

*H.323* hace uso de los procedimientos de señalización de los canales lógicos contenidos en *H.245*, en los que el contenido de cada uno de los canales se define cuando se abre. Estos procedimientos se proporcionan para fijar las prestaciones del emisor y receptor, el establecimiento de la llamada, intercambio de información, terminación de la llamada y como se codifica y decodifica. Por ejemplo, cuando se origina una llamada telefónica sobre Internet, los dos terminales deben negociar cual de los dos ejerce el control, de manera tal que sólo uno de ellos origine los mensajes especiales de control. Un punto importante es que se deben determinar las capacidades de los sistemas, de forma que no se permita la transmisión de datos si no pueden ser gestionados por el receptor.

La función de señalización se basa en la recomendación *H.225* [25], que especifica el uso y soporte de mensajes de señalización *Q.931/Q932*. Las llamadas son enviadas sobre *TCP* por el puerto 1720. Sobre este puerto se inician los mensajes de control de llamada *Q.931* entre dos terminales para la conexión, mantenimiento y desconexión de llamadas.

**Mensajes más comunes de *Q.931/Q.932* usados como mensajes de señalización *H.323*:**

- ***Setup:*** Mensaje que se envía para iniciar una llamada *H.323*, para establecer una conexión con una entidad *H.323*. Entre la información que contiene el mensaje se encuentra la dirección *IP*, puerto y alias del llamador o la dirección *IP* y puerto del llamado.
- ***Call Proceeding:*** Es en mensaje que envía el *Gatekeeper* a un terminal advirtiendo del intento de establecer una llamada una vez analizado el número llamado.
- ***Alerting:*** Es el mensaje que indica el inicio de la fase de generación de tono.
- ***Connect:*** Es en mensaje que indica el comienzo de la conexión.
- ***Release Complete:*** Mensaje enviado por el terminal para iniciar la desconexión.
- ***Facility:*** Es un mensaje de la norma *Q.932* usado como petición o reconocimiento de un servicio suplementario.

**Función de control *H.245***

EL canal de control *H.245* es un conjunto de mensajes de *Notación de Sintaxis Abstracta.1* (*ASN.1*, por sus siglas en inglés*)* usados para el establecimiento y control de una llamada. Unas de las características que se intercambian más relevantes son:

- ***MasterSlaveDetermination (MSD***, por sus siglas en inglés***).*** Este mensaje es usado para prevenir conflictos entre dos terminales que quieren iniciar la comunicación. Decide quién actuará de *Máster (maestro)* y quién de *Slave (esclavo)*.
- ***TerminalCapabilitySet (TCS***, por sus siglas en inglés***).*** Es el mensaje de intercambio de capacidades soportadas por los terminales que intervienen en una llamada.
- ***OpenLogicalChannel (OLC***, por sus siglas en inglés***).*** Mensaje enviado para abrir el canal lógico de información, contiene información para permitir la recepción y codificación de los datos. Contiene la información del tipo de datos que serán transportados.
- ***CloseLogicalChannel (CLC***, por sus siglas en inglés***).*** Mensaje enviado para cerrar el canal lógico de información.

Para entender con claridad cómo funcionan estos mensajes y de qué manera son utilizados, es necesario analizar detalladamente una llamada. En una llamada *H.323* existen varias fases e intervienen diferentes protocolos en cada una de ellas, como se muestra en la ***figura 11***. Dichas fases son:

**2.4.4.2.2.2.1 Fases que intervienen en una llamada *H.323***

1. Establecimiento.
2. Señalización de Control.
3. Audio.
4. Desconexión.

Ejemplo de una llamada ***H.323*** donde se muestran las Fases, protocolos y los diferentes mensajes que intervienen.

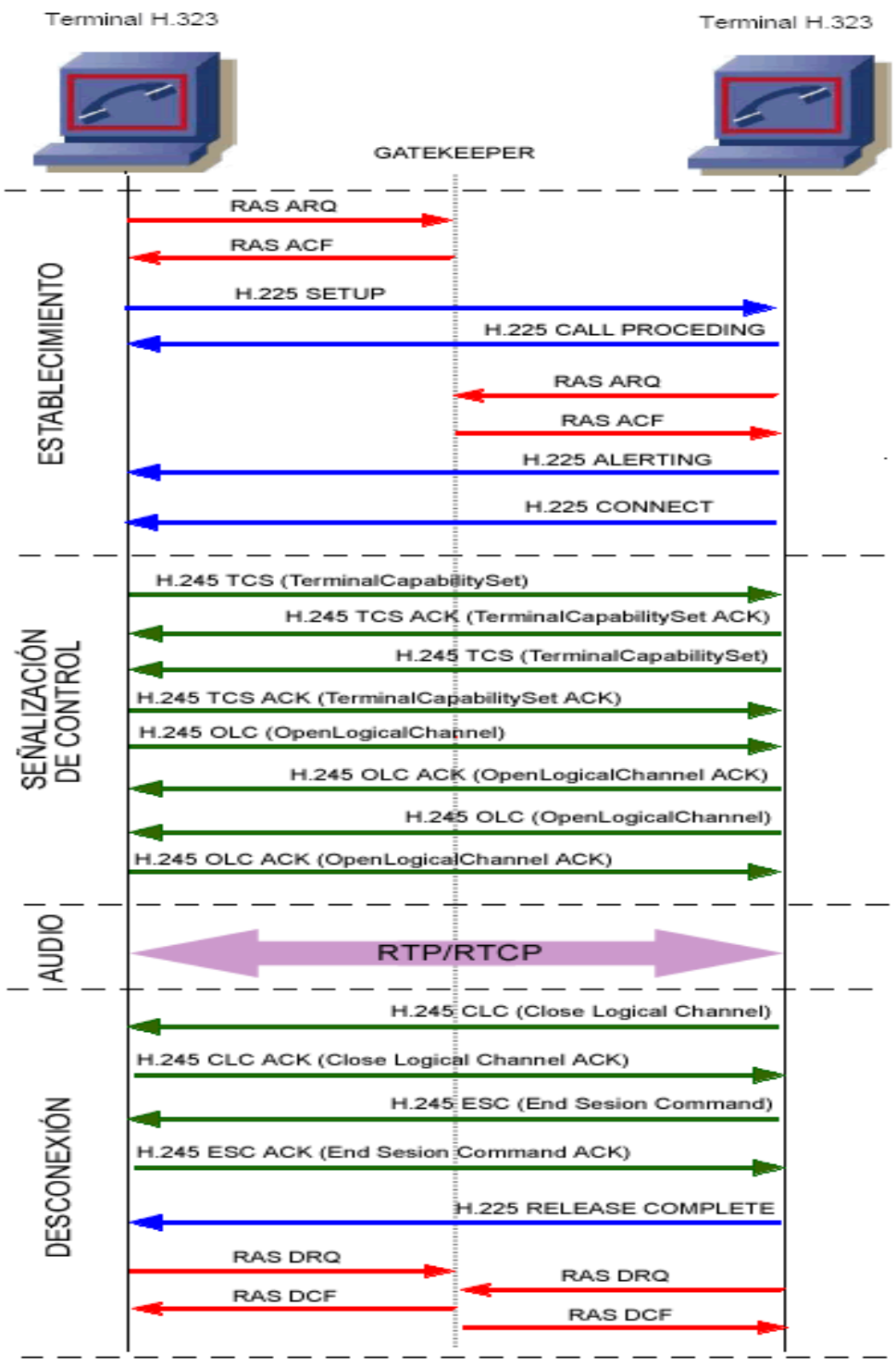

Terminal H.323
Terminal H.323
GATEKEEPER
ESTABLECIMIENTO
RAS ARQ
RAS ACF
H.225 SETUP
H.225 CALL PROCEDING
RAS ARQ
RAS ACF
H.225 ALERTING
H.225 CONNECT
SEÑALIZACIÓN DE CONTROL
H.245 TCS (TerminalCapabilitySet)
H.245 TCS ACK (TerminalCapabilitySet ACK)
H.245 TCS (TerminalCapabilitySet)
H.245 TCS ACK (TerminalCapabilitySet ACK)
H.245 OLC (OpenLogicalChannel)
H.245 OLC ACK (OpenLogicalChannel ACK)
H.245 OLC (OpenLogicalChannel)
H.245 OLC ACK (OpenLogicalChannel ACK)
AUDIO
RTP/RTCP
DESCONEXIÓN
H.245 CLC (Close Logical Channel)
H.245 CLC ACK (Close Logical Channel ACK)
H.245 ESC (End Sesion Command)
H.245 ESC ACK (End Sesion Command ACK)
H.225 RELEASE COMPLETE
RAS DRQ
RAS DRQ
RAS DCF
RAS DCF

Figura 11. Fases, protocolos y mensajes de una llamada H.323.

**1. Establecimiento.**

- Esta fase puede ser iniciada por cualquiera de los terminales que vallan a intervenir en la llamada. Lo primero que se observa es que uno de los terminales se registra en el *gatekeeper* utilizando el protocolo de *Registro, admisión y estado* (*RAS,* por sus siglas en inglés) con los mensajes *ARQ*[22] y *ACF*[23].
- Posteriormente utilizando el protocolo *H.225* (que se utiliza para el establecimiento y liberación de la llamada) se manda un mensaje de ***SETUP*** para iniciar una llamada *H.323*. Entre la información que contiene el mensaje se encuentra la dirección *IP*, puerto y alias del llamador o la dirección IP y puerto del llamado.

Haciendo una captura de paquetes con la herramienta *WireShark* podemos ver el mensaje ***SETUP*** con la información que contiene. En el Panel de paquetes está marcado el paquete donde se encuentra el mensaje, en el Panel de detalles se puede ver claramente el mensaje y la información que contiene y en el Panel de paquetes capturados en byte, se encuentra la información del mensaje en *byte*.

**Nota:** En el capítulo 3 en las pruebas de laboratorio se verán cada uno de estos mensajes capturados con la herramienta. Es importante saber que la distribución por paneles antes mencionada, es la misma para todos los mensajes. Esto es válido para las 4 fases. Para saber que es un Panel consultar el capítulo 3.

- Como respuesta al mensaje ***SETUP***, el terminal llamado contesta con un ***CALL PROCEEDING*** advirtiendo del intento de establecer una llamada.
- Después de esto el segundo terminal tiene que registrarse en el *gatekeeper* utilizando el protocolo *RAS* de manera similar al primer terminal.
- Posteriormente el segundo terminal envía un mensaje de ***ALERTING***, que indica el inicio de la fase de generación de tono.

[22] Técnica utilizada para la corrección de errores. Si durante la transmisión ocurre un error, el dispositivo receptor solicita al emisor la retransmisión del mensaje. Tomado de: www.dednet.net/institucion/itba/cursos/000183/demo/@glosario.html

[23] Atributo de archivo de configuración, un archivo que puede crear, opcionalmente, que le permite seleccionar los métodos y las específicas de control de errores. Este archivo debe tener la extensión. Acf. Tomado de: docs.hp.com/en/64/ll/llglos.htm

- Para terminar la fase de establecimiento el segundo terminal envía al primero un mensaje ***CONNECT*** para indicar el comienzo de la conexión.

**2. Señalización de Control.**

En esta fase se abre una negociación mediante el protocolo *H.245* (que se encarga del control de conferencia). El intercambio de los mensajes (petición y respuesta) entre los dos terminales establece quién será el que actuará como *Máster* y quién como *Slave*, además de intercambiar las capacidades de los participantes y *códecs* de audio y video a utilizar. Como punto final de esta negociación se abre el canal de comunicación (direcciones *IP*, puerto).

Los principales mensajes *H.245* que se utilizan en esta fase son:

- ***TerminalCapabilitySet (TCS).*** Es el mensaje de intercambio de capacidades soportadas por los terminales que intervienen en una llamada.
- ***OpenLogicalChannel (OLC).*** Es el mensaje para abrir el canal lógico de información que contiene información para permitir la recepción y codificación de los datos. Contiene la información del tipo de datos que será transportado.

Estos mensajes son utilizados de la siguiente manera:

- El segundo terminal envía el mensaje ***TerminalCapabilitySet*** ver ***Anexo 4 figura 24*** al terminal uno, para verificar la capacidad que soporta y el mensaje ***MasterSlaveDetermination*** para establecer quién actuará de *Máster* y quién de *Slave*.
- De la misma forma el terminal uno envía los mismos mensajes al segundo terminal y adiciona dos mensajes como respuestas a los enviados por el segundo terminal, ellos son el ***TerminalCapabilitySetAck*** y el ***MasterSlaveDeterminationAck.***
- De igual manera el terminal dos envía los mensajes ***TerminalCapabilitySetAck*** y ***MasterSlaveDeterminationAck*** como respuesta al terminal uno.
- Posteriormente se procede a la apertura del canal lógico de información, para ello se utiliza el mensaje ***OpenLogicalChannel***. El terminal uno envía este mensaje al terminal y viceversa.

- Para finalizar la fase de señalización de control se envía el mensaje ***OpenLogicalChannelAck***, como respuesta a los menajes enviados anteriormente por ambos terminales. Este mensaje también es enviado en ambos sentidos.

### 3. Audio.

Llegado a esta fase los terminales se encuentran en condiciones para comenzar el intercambio de los datos, ya sea audio o video. Para el intercambio de información en tiempo real intervienen los protocolos ***RTP/RTCP***, donde los paquetes son enviados con el protocolo de transporte *UDP*, que se explicó anteriormente. Ambos protocolos de transporte en tiempo real serán explicados a continuación.

### Protocolo *RTP*

*RTP* se utiliza con *RTCP* (*Protocolo de Control de Transporte en Tiempo Real*, por sus siglas en inglés). Cada paquete *RTP* lleva un número de secuencia y una indicación de tiempo, que pueden ser utilizadas en dependencia de la aplicación.

La carga útil de cada paquete *RTP*, es la información en tiempo real contenida en el paquete, su formato es completamente libre y debe ser definido por la aplicación o el perfil de *RTP* en uso. Para distinguir un formato particular, el encabezamiento de cada paquete *RTP* contiene un identificador de tipo de carga útil (*PT*, por sus siglas en inglés). ***La tabla 2.3*** ilustra los identificadores en uso en *H.225* para algunos *códecs* estándar.

### El paquete *RTP*

<table>
<tr><td colspan="2">V=2</td><td>P</td><td>X</td><td colspan="2">CC</td><td>M</td><td colspan="2">Tipo de Carga Util</td><td colspan="6">Número de Secuencia</td></tr>
<tr><td colspan="14">Indicación de Tiempo</td></tr>
<tr><td colspan="14">SSRC (32 bits)</td></tr>
<tr><td colspan="14">CSRC<br>CSRC 1<br>...<br>CSRC n</td></tr>
<tr><td colspan="8">Dependiente del Perfil</td><td colspan="6">Tamaño</td></tr>
<tr><td colspan="14">Datos</td></tr>
<tr><td>0</td><td>2</td><td>4</td><td>6</td><td>8</td><td>10</td><td>12</td><td>14</td><td>16</td><td>18</td><td>20</td><td>24</td><td>26</td><td>28</td><td>30</td></tr>
</table>

Tabla 2.3 Un paquete RTP.

Todos los campos hasta la lista *CSRC* están siempre presentes en un paquete *RTP*.

- Dos *bits* indican la versión (*V*, por su sigla en inglés). La actual es la 2(10).
- Relleno (*P*, por su sigla en inglés). Un *bit* que indica si el paquete contiene bytes de relleno al final que no forman parte de los datos.
- Extensión (*X*, por su sigla en inglés). Indica la extensión después de *CSRC* del encabezado fijo. Las extensiones utilizan el siguiente formato:

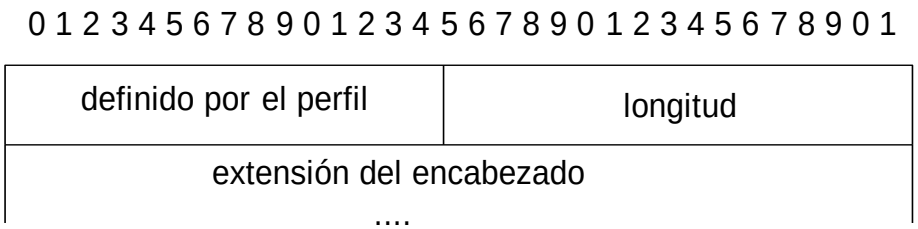



Figura 12. Formato de la extensión del encabezado

- Contador *CSRC* (*CC*, por sus siglas en inglés). Cuatro *bits* que indican cuantos identificadores *CSRC* siguen al encabezamiento fijo.
- Marcador (*M*, por su sigla en inglés). Un *bit* utilizado para aplicaciones específicas, definido por el perfil *RTP*. Por ejemplo, H225.0 pone en "1" dicho *bit* para indicar que la codificación de audio soporta supresión de silencio.
- Tipo de carga útil (*PT*, por sus siglas en inglés): Siete *bits* que informan sobre algunos tipos de carga estáticos definidos en la *RFC 1889*.
- Número Secuencial (16 *bits*). Comienza con un valor aleatorio y se incrementa con cada paquete *RTP*. Permite detectar pérdida de paquetes.
- Indicación de Tiempo (*Timestamp*, por sus siglas en inglés). La frecuencia de reloj se define para cada tipo de carga útil. Ejemplo la carga útil de H.261, utiliza un reloj de 90 kHz para la indicación de tiempo. Para la mayoría de los *códecs* (G.711, G.723.1, G729, etc.) el reloj de frecuencia de *RTP* es puesto a 8000 Hz.
- *SSRC* (*Synchronization Source Identifier*, por sus siglas en inglés): Identifica fuente de sincronismo.
- *CSRC* (*Contributing Source Identifier*, por sus siglas en inglés): De 0 a 15 elementos de 32 *bits* cada uno. Cuando un flujo *RTP* es resultado de la mezcla de varios flujos *RTP*

contribuyentes, en el campo *CSRC*, se coloca la lista de los *SSRC* de cada contribuyente. El flujo resultante tiene su propia *SSRC*. La cantidad de *CSRC* que pueden aparecer se expresa en el campo *CC*.

**Protocolo *RTCP***

Como habíamos visto anteriormente dicho protocolo es utilizado con *RTP* ya que ofrece información sobre la calidad de la transmisión y sobre la identidad de los participantes. En este caso a *RTP* se le asigna un puerto par y a *RTCP* el siguiente puerto (impar). *RTCP* realiza las siguientes funciones:

1. Brindar realimentación acerca de la calidad de distribución de los datos para lo que se relaciona con funciones de control de flujo y congestión. Esta realimentación se utiliza de forma directa para el control adaptativo de la codificación, enviando reportes a todos los participantes, que permiten que uno que esté observando un problema pueda determinar si este es global o local y en caso necesario modificar sus transmisiones. Cada paquete contiene información del remitente y/o receptor y transportan estadísticas útiles a la aplicación (cantidad de *jitter*, promedio de paquetes perdidos, etc.).
2. Identificar la fuente *RTP*. *RTCP* lleva un identificador de nivel de transporte para una fuente *RTP*, a este identificador se le denomina nombre canónico (*CNAME*, por sus siglas en inglés). Esto es de gran importancia porque dentro de la sesión puede ocurrir algún problema que haga cambiar el *SSRC* y los receptores requieren del *CNAME* para no perder de vista a los participantes.
3. Control del envío de transmisión. Para que *RTP* pueda aceptar un gran número de participantes en la sesión, se limita el tráfico de control a un 5 % (como máximo) del ancho de banda de la sesión global. Como cada participante envía paquetes de control a todos los demás, cada uno puede conocer el número total de participantes y utilizar este número para calcular la velocidad de envío de los paquetes *RTCP*. Este propósito debe ser compartido por todos los participantes. *RTP* estipula que los transmisores activos obtienen un cuarto del referido porciento ya que alguna de las informaciones por ellos enviadas (por ejemplo *CNAME*

utilizada para sincronización) es muy importante para todos los receptores y los reportes de los transmisores *RTCP* necesitan ser muy sensibles. La parte restante del ancho de banda destinada a los paquetes de control es compartida entre los receptores. Aún para sesiones pequeñas, las razones más rápidas a las cuales un participante puede enviar reportes *RTCP* es de uno cada cinco segundos. La razón de envío es aleatorizada por un factor de 0.5 a 1.5 para evitar no deseada sincronización entre reportes.

4. Llevar información de control de sesión mínima a todos los participantes. Por ejemplo para mostrar en pantalla quién está participando en ese momento o quién entra o sale de la sesión, lo cual es de gran utilidad cuando el número de participantes varía constantemente pues entran o salen sin restricción alguna.

**Paquetes *RTCP***

Existen cinco tipos de paquetes *RTCP*, cada uno de ellos posee un formato similar al de un paquete *RTP*. Comúnmente estos paquetes son enviados en forma concatenada sin espaciamiento entre ellos dando lugar a paquetes compuestos que son manipulados por la capa de transporte como un sólo paquete.

En un paquete compuesto, el primero de ellos debe ser un *SR*, por sus siglas en inglés *ó RR*, por sus siglas en inglés*.* El límite, en cuanto a cantidad paquetes, será puesto por los protocolos de las capas inferiores.

En la ***figura 13*** se observa un ejemplo de paquete compuesto formado por tres paquetes, uno corresponde con un *SR*, para que sea válido, el segundo es un *SDES*, por sus siglas en inglés, el cual lleva el *CNAME* que es imprescindible y por último el paquete *BYE*. El primer campo nombrado *R* es un valor entero de 32 *bits* que aparece sólo si el paquete se ha encriptado.

| | Paquete | Paquete | Paquete |
|---|---|---|---|
| **R** | **SR** | **SDES** | **BYE** |

Figura 13. Ejemplo de paquete compuesto RTCP

*RTCP* permite el crecimiento de una sesión desde unos pocos participantes hasta cientos de forma automática. No obstante, al crecimiento del número de participantes en la sesión, en aplicaciones de voz, el tráfico de datos permanece aproximadamente constante pues generalmente no hablan más de dos personas a la vez. Por el contrario, con el crecimiento del número de participantes, el número de paquetes de control crece de forma lineal.

**4. Desconexión.**

- Esta fase puede ser empezada por cualquiera de los participantes activos en la comunicación, iniciando el proceso de finalización de llamada mediante el mensaje ***CloseLogicalChannel*** para cerrar el canal lógico de información antes abierto y el mensaje ***EndSessionComand*** para cerrar la sesión de comandos de *H.245*. El terminal llamado entonces mandaría dos mensajes como respuestas, el ***CloseLogicalChannel Ack*** y el ***EndSessionComand Ack***. El envío de estos mensajes no es al azar, primero se envía el mensaje ***CloseLogicalChannel***, del cual se recibe respuesta y posteriormente el mensaje ***EndSessionComand***, del cual también se recibe respuesta.
- Posteriormente utilizando el protocolo *H.225* se cierra la conexión con el mensaje ***RELEASE COMPLETE***.
- Para finalizar son liberados los registros con el *gatekeeper* utilizando mensajes del protocolo *RAS*.

Una vez terminada la llamada *H.323* se procede a la terminación de la sesión creada por *TCP* anteriormente. Es importante guardar en una base de datos información que posee el protocolo. Los campos que se proponen son: tipo de mensaje, donde se guardaría el mensaje que fue capturado, el campo protocolo de *H.323*, que guardaría el protocolo al que pertenece el mensaje capturado y el campo datos, que almacenaría la información que contiene el protocolo.

### 2.4.4.2.2.3 Terminación de la conexión *TCP*

Una vez terminada la comunicación o el intercambio de datos y se procede a cerrar la conexión se debe establecer el señalizador de control *FIN* (Finalizar) en el encabezado del segmento TCP. Como se muestra en la ***figura 14.***

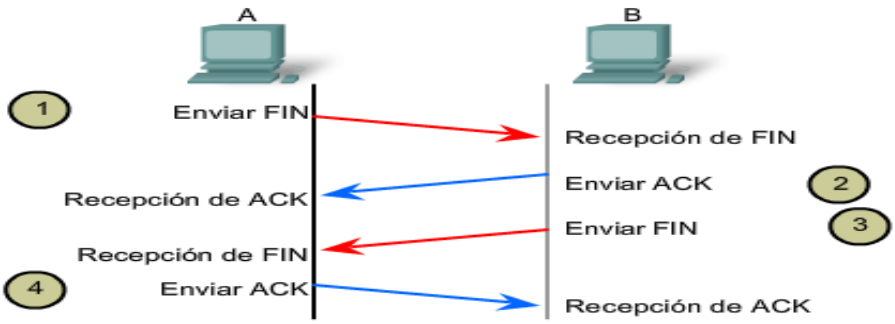



Figura 14. Finalización de una conexión TCP.

Para finalizar todas las sesiones *TCP* de una vía, se utiliza un enlace de dos vías, que consta de un segmento *FIN* y un segmento *ACK*. Por lo tanto, para terminar una conversación simple admitida por *TCP*, se requieren cuatro intercambios para finalizar ambas sesiones.

**Nota:** Los términos cliente y servidor se han utilizados para nombrar los *host* que intervienen en el proceso, pero la finalización del mismo puede ser iniciada por cualquiera de los dos *hosts* que completen la sesión:

1. Cuando el cliente no tiene más datos para enviar al servidor, envía un segmento con el señalizador *FIN* establecido.
2. El servidor envía un *ACK* para acusar recibo de *FIN* y terminar la sesión del cliente al servidor.
3. El servidor envía un *FIN* al cliente para finalizar la sesión del servidor al cliente.
4. El cliente responde con un *ACK* para dar acuse de recibo de *FIN* desde el servidor.

Cuando la finalización de sesión del cliente no tiene más datos para transferir, establece el señalizador *FIN* en el encabezado de un segmento. Luego, el servidor finaliza la conexión y envía un segmento normal que contiene datos con el señalizador *ACK* establecido utilizando el número de acuse de recibo, confirmando así que se han recibido todos los *bytes* de datos. Cuando se produce el acuse de recibo de todos los segmentos, se cierran las sesiones.

Una vez que se ha explicado teóricamente el procedimiento queda representarlo en una figura con vista a una mejor comprensión y organización del mismo.

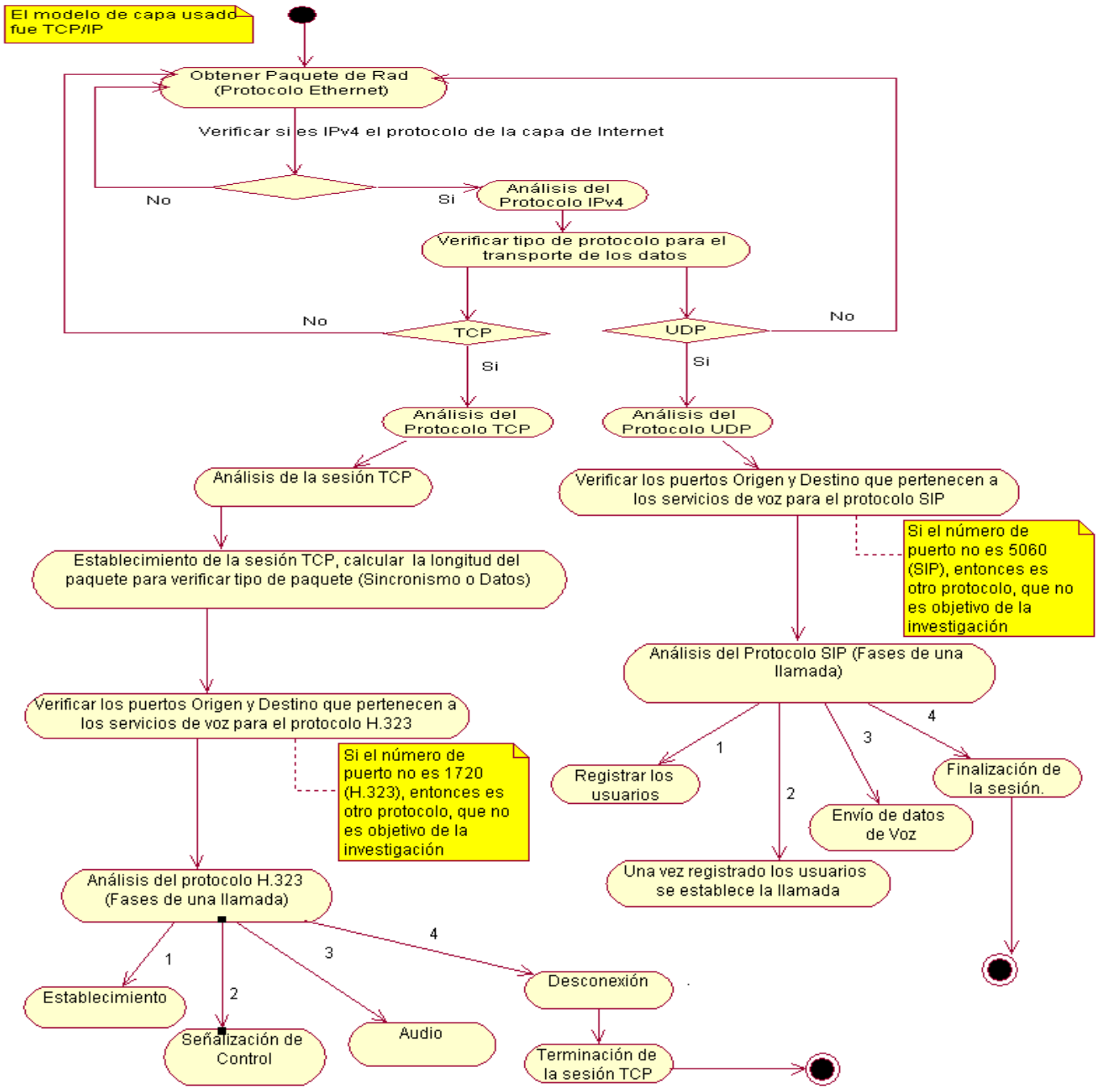



Figura 14.1. Diagrama del procedimiento.

## 2.5 Conclusiones

En el presente capítulo se cumplió con el objetivo del trabajo que era la creación de un procedimiento que permita el análisis de los protocolos usados en las comunicaciones de voz sobre redes *IP*. Para esto fue necesario realizar una caracterización de los protocolos presentes en el modelo *TCP/IP* hasta llegar a los de señalización de voz *H.323* y *SIP*. Todos los pasos del procedimiento tienen un orden lógico, los que deben ser respetados para lograr entender la información que brinda.

# CAPÍTULO 3: PRUEBAS DE LABORATORIO Y PROCESAMIENTO DE LOS DATOS RELACIONADOS CON LOS PROTOCOLOS *SIP* Y *H.323* Y FAMILIA *TCP/IP*

## 3.1 Introducción

En el capítulo 2, se explicó el funcionamiento de los protocolos empleados en las comunicaciones de voz sobre redes *IP* (*SIP* y *H323*), además se dio una solución del procedimiento que permite el análisis de dichos protocolos y se explicó en qué consistían sus partes fundamentales. En el presente capítulo, se pretende demostrar mediante pruebas de laboratorio la validez del procedimiento propuesto. Para esto se harán algunas capturas de los paquetes que viajan por la red, que traen consigo dichos protocolos lo cual permitirá analizarlos en detalles. Para las capturas antes mencionadas se utilizara el Analizador de Protocolos de Red "*Wireshark*", por las ventajas que brinda.

## 3.2 Pruebas de laboratorio

Para el proceso de adquisición de los protocolos se utilizó una herramienta que nos da la posibilidad de analizarlos en profundidad y en detalles, o sea ver e interpretar como viaja la información por la red a nivel de *byte*, esto permite saber qué es lo que está viajando por la red en un momento dado, además de analizar sus características.

Existe un conjunto de herramientas que se conocen como: Analizadores de Protocolos, que permiten capturar los paquetes que viajan por la red. Algunas de estas herramientas son: Red *SoftPerfect*, *VisualSniffer*, *IP Sniffer*, *Ethereal*, *Appsniffing*, *Expert Observer*, *Observer Suite*, *Observer Probes*, *SuperAgent, Windump, TCPDump* ó *dsniff*, entre otras.

Para realizar las pruebas de laboratorio con vista a demostrar la funcionalidad del procedimiento propuesto, se utilizó el Analizador de Protocolo de Red *Wireshark, **figura 15**.* Es una herramienta multiplataforma de análisis de red. A diferencia de los analizadores antes mencionados, funciona mostrando los datos a través de un entorno gráfico, de forma más amigable y entendible. Añade muchas opciones de organización y filtrado de información. Permite ver todo el tráfico que pasa a

través de una red (usualmente una red *Ethernet*, aunque es compatible con algunas otras) estableciendo la configuración en modo promiscuo. Permite examinar datos de una red viva o de un archivo de captura salvado en disco. Se puede analizar la información capturada, a través de los detalles y sumarios por cada paquete. *Wireshark* incluye un completo lenguaje para filtrar lo que queremos ver y la habilidad de mostrar el flujo reconstruido de una sesión de *TCP*.

Para capturar paquetes directamente de la interfaz de red, generalmente se necesitan permisos de ejecución especiales. Es por esta razón que *Wireshark* es ejecutado con permisos de Superusuario, tomando en cuenta la gran cantidad de analizadores de protocolo que posee, los cuales son ejecutados cuando un paquete llega a la interfaz.

*Wireshark* es *software* libre, y se ejecuta sobre la mayoría de sistemas operativos *Unix* y compatibles, incluyendo *Linux*, *Solaris*, *FreeBSD*, *NetBSD*, *OpenBSD*, y *Mac OS X*, así como en *Microsoft Windows*.

***Wireshark***

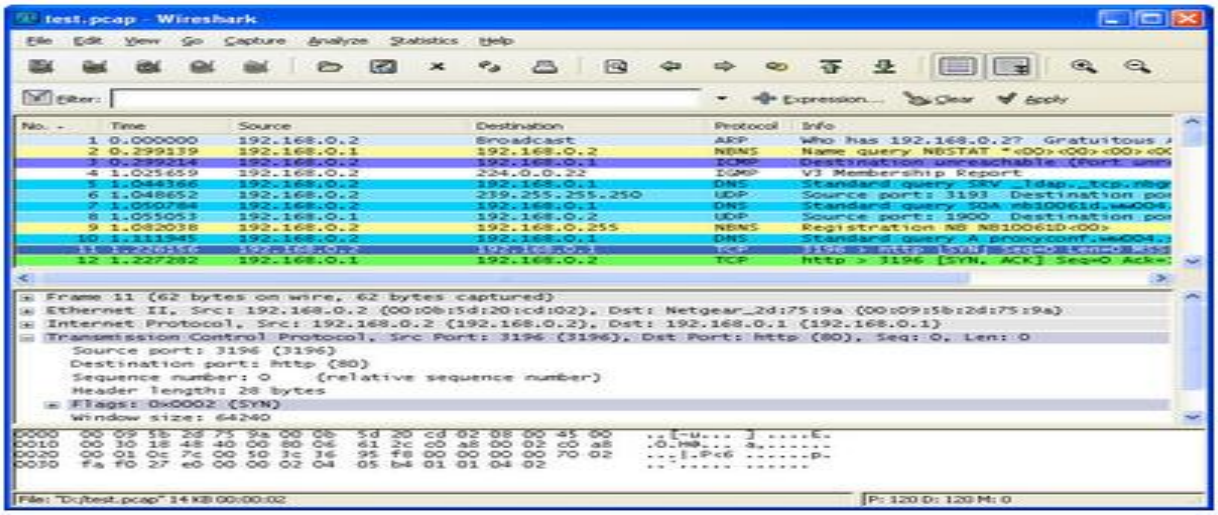



Figura 15. Analizador de Protocolo de Red ***Wireshark***

## 3.3 Descripción de las pruebas de laboratorio

En las pruebas de laboratorio se muestra cómo los protocolos son capturados por la herramienta escogida, donde se ven los campos del encabezado, así como los datos de las trazas. A continuación se detallará, apoyándonos en 4 pasos, como se hicieron las pruebas de laboratorio y como es el funcionamiento de la herramienta.

### 3.3.1 Paso 1: Conexión Cliente-Servidor

El primer paso es establecer desde un cliente que tiene un *software* que implemente la voz *IP* una conexión al servidor mediante el envío de mensajes, esta conexión no es mas que el establecimiento de una sesión, una vez lograda la sesión se procede a establecer una conversación o intercambio de voces entre dos aplicaciones, utilizando algún dispositivo de audio, ya sea un micrófono o altavoz.

### 3.3.2 Paso 2: Captura de paquetes utilizando la herramienta

Después de comenzada la comunicación estamos en condiciones de capturar los paquetes que se envían de un host a otro. Entre las principales funciones de la herramienta *Wireshark* está la de capturar paquetes que viajan por la red, para que los administradores o ingenieros de redes puedan realizar el análisis necesario para tener una red segura y estable. Es necesario para el proceso de captura de datos ser administrador o contar con estos privilegios y se debe identificar cuál es la interfaz que se quiere analizar. Para iniciar la captura de los paquetes se ejecuta la herramienta de la siguiente manera:

Haciendo doble clic en  se despliega una ventana donde se listan las interfaces locales disponibles para iniciar la captura de paquetes.

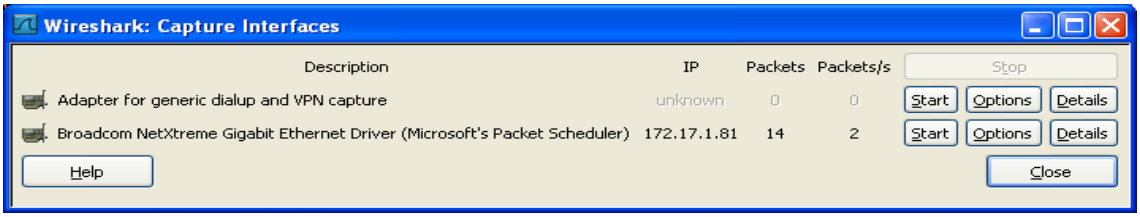


Figura 16. Para capturar una Interfaz

Tres botones se visualizan por cada interfaz

- *Start*, para iniciar
- *Options*, para configurar
- *Details*, proporciona información adicional de la interfaz como su descripción, estadísticas, etc.

**Detener/Reiniciar la captura de paquetes**

Para detener la captura de paquetes podemos aplicar una de las siguientes opciones:

- Haciendo uso del icono  desde el menú *Capture* o desde la barra de herramientas.
- Haciendo uso de *ctrl*+*E*.
- La captura puede ser detenida automáticamente, si una de las condiciones de parada definidas en las opciones de la interfaz se cumple, por ejemplo: si se excede cierta cantidad de paquetes.

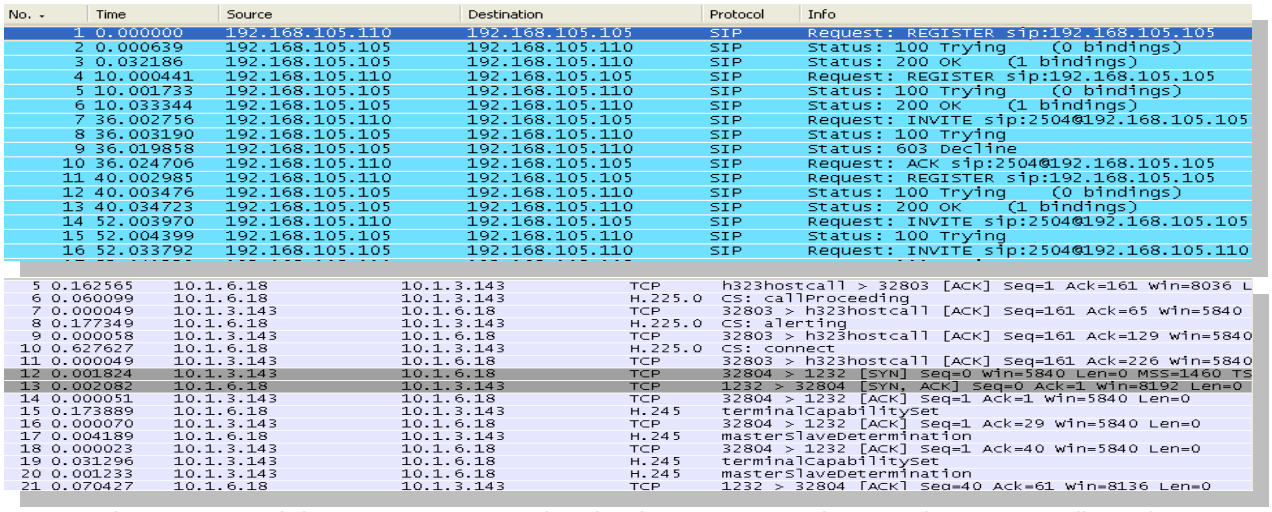
Para reiniciar el proceso de captura se debe seleccionar el icono  en la barra de herramientas o desde el menú *Capture*.

Después de haber capturado los paquetes, estos son listados en el panel de paquetes capturados, al seleccionar uno se despliega el contenido del paquete en los otros paneles que son panel de detalles de paquetes y panel en *bytes*. Expandiendo cualquier parte del árbol presentado en el panel de detalle del paquete, se puede seleccionar un campo en particular cuyo contenido se muestra resaltado en negritas en el panel de *bytes*. A continuación veremos cuales son cada uno de estos paneles.

### 3.3.2.1 Panel de paquetes capturados

La ***figura 17*** muestra el panel donde están los paquetes capturados**,** en este panel se despliega la lista de todos los paquetes que han sido capturados. Al hacer *clic* sobre algunos de estos se

despliega cierta información en los otros paneles o se muestra información específica de una traza, todo lo correspondiente al destino, origen, protocolo, además de los datos que presenta.



| No. | Time | Source | Destination | Protocol | Info |
|---|---|---|---|---|---|
| 1 | 0.000000 | 192.168.105.110 | 192.168.105.105 | SIP | Request: REGISTER sip:192.168.105.105 |
| 2 | 0.000639 | 192.168.105.105 | 192.168.105.110 | SIP | Status: 100 Trying (0 bindings) |
| 3 | 0.032186 | 192.168.105.105 | 192.168.105.110 | SIP | Status: 200 OK (1 bindings) |
| 4 | 10.000441 | 192.168.105.110 | 192.168.105.105 | SIP | Request: REGISTER sip:192.168.105.105 |
| 5 | 10.001733 | 192.168.105.105 | 192.168.105.110 | SIP | Status: 100 Trying (0 bindings) |
| 6 | 10.033344 | 192.168.105.105 | 192.168.105.110 | SIP | Status: 200 OK (1 bindings) |
| 7 | 36.002756 | 192.168.105.110 | 192.168.105.105 | SIP | Request: INVITE sip:2504@192.168.105.105 |
| 8 | 36.003190 | 192.168.105.105 | 192.168.105.110 | SIP | Status: 100 Trying |
| 9 | 36.019858 | 192.168.105.105 | 192.168.105.110 | SIP | Status: 603 Decline |
| 10 | 36.024706 | 192.168.105.110 | 192.168.105.105 | SIP | Request: ACK sip:2504@192.168.105.105 |
| 11 | 40.002985 | 192.168.105.110 | 192.168.105.105 | SIP | Request: REGISTER sip:192.168.105.105 |
| 12 | 40.003476 | 192.168.105.105 | 192.168.105.110 | SIP | Status: 100 Trying (0 bindings) |
| 13 | 40.034723 | 192.168.105.105 | 192.168.105.110 | SIP | Status: 200 OK (1 bindings) |
| 14 | 52.003970 | 192.168.105.110 | 192.168.105.105 | SIP | Request: INVITE sip:2504@192.168.105.105 |
| 15 | 52.004399 | 192.168.105.105 | 192.168.105.110 | SIP | Status: 100 Trying |
| 16 | 52.033792 | 192.168.105.105 | 192.168.105.110 | SIP | Request: INVITE sip:2504@192.168.105.110 |

| | | | | | |
|---|---|---|---|---|---|
| 5 | 0.162565 | 10.1.6.18 | 10.1.3.143 | TCP | h323hostcall > 32803 [ACK] Seq=1 Ack=161 Win=8036 L |
| 6 | 0.060099 | 10.1.6.18 | 10.1.3.143 | H.225.0 | CS: callProceeding |
| 7 | 0.000049 | 10.1.3.143 | 10.1.6.18 | TCP | 32803 > h323hostcall [ACK] Seq=161 Ack=65 Win=5840 |
| 8 | 0.177349 | 10.1.6.18 | 10.1.3.143 | H.225.0 | CS: alerting |
| 9 | 0.000058 | 10.1.3.143 | 10.1.6.18 | TCP | 32803 > h323hostcall [ACK] Seq=161 Ack=129 Win=5840 |
| 10 | 0.627627 | 10.1.6.18 | 10.1.3.143 | H.225.0 | CS: connect |
| 11 | 0.000049 | 10.1.3.143 | 10.1.6.18 | TCP | 32803 > h323hostcall [ACK] Seq=161 Ack=226 Win=5840 |
| 12 | 0.001824 | 10.1.3.143 | 10.1.6.18 | TCP | 32804 > 1232 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 TS |
| 13 | 0.002082 | 10.1.6.18 | 10.1.3.143 | TCP | 1232 > 32804 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 |
| 14 | 0.000051 | 10.1.3.143 | 10.1.6.18 | TCP | 32804 > 1232 [ACK] Seq=1 Ack=1 Win=5840 Len=0 |
| 15 | 0.173889 | 10.1.6.18 | 10.1.3.143 | H.245 | terminalCapabilitySet |
| 16 | 0.000070 | 10.1.3.143 | 10.1.6.18 | TCP | 32804 > 1232 [ACK] Seq=1 Ack=29 Win=5840 Len=0 |
| 17 | 0.004189 | 10.1.6.18 | 10.1.3.143 | H.245 | masterSlaveDetermination |
| 18 | 0.000023 | 10.1.3.143 | 10.1.6.18 | TCP | 32804 > 1232 [ACK] Seq=1 Ack=40 Win=5840 Len=0 |
| 19 | 0.031296 | 10.1.3.143 | 10.1.6.18 | H.245 | terminalCapabilitySet |
| 20 | 0.001233 | 10.1.3.143 | 10.1.6.18 | H.245 | masterSlaveDetermination |
| 21 | 0.070427 | 10.1.6.18 | 10.1.3.143 | TCP | 1232 > 32804 [ACK] Seq=40 Ack=61 Win=8136 Len=0 |

Figura17. Panel de paquetes capturados donde se muestra el protocolo ***SIP*** y una llamada ***H.323***, empleando los protocolos ***H.225.0*** y ***H.245*** para el establecimiento de una sesión.

### 3.3.2.2 Panel para detalles del paquete

Para que se puedan entender los datos específicos de los protocolos que vienen en los paquetes, existe el Panel para detalles del paquete. Aquí se encuentra la información detallada que contiene cada campo que compone al protocolo. Para acceder a la misma, se despliega la información detallada del paquete seleccionado en el panel de paquetes. Desplegando el protocolo que se quiera conocer, encontrará la información referente al mismo. Esto se muestra e la ***figura 17***. Donde se desplegó el protocolo *H.225.0* para conocer toda la información del mismo.

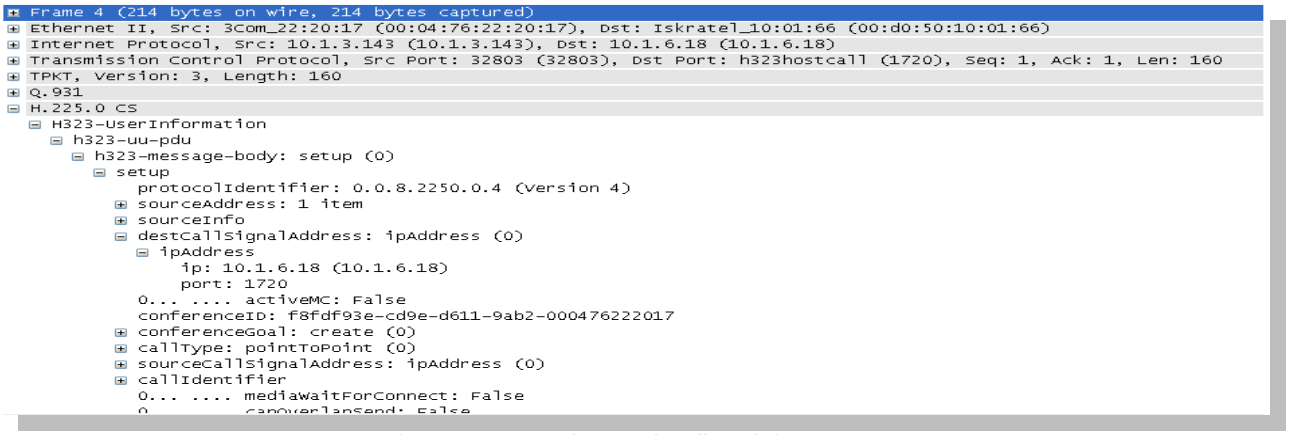



Figura 17. Panel para detalles del paquete.

### 3.3.2.3 Panel de paquetes capturados en *bytes*.

Existe además el Panel de paquetes capturados en *bytes*, en este panel se despliega el contenido del paquete en formato hexadecimal, así como la información contenida en el paquete seleccionado desde el Panel para detalles del paquete.

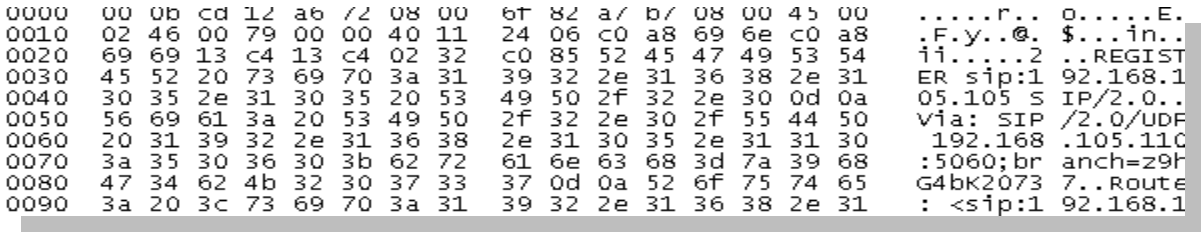



Figura 17. Panel de paquetes capturados en bytes.

Si después de una captura se desea filtrar los paquetes, la herramienta brinda esa posibilidad utilizando la Barra de herramienta para filtros.

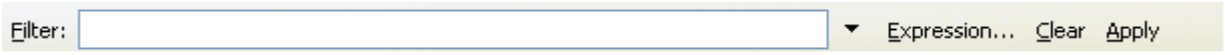



Figura 18. Barra de herramientas para filtros.

Aquí se especifica el filtro que se desea aplicar a los paquetes que están siendo capturados, o sea, buscar un tipo de protocolo específico. Si se pone *SIP* en el filtro se capturarían todas las tramas de dicho protocolo, como se muestra en la ***figura 17*** en el panel de paquetes capturados.

### 3.3.3 Paso 3: Comprensión de las tramas o paquetes.

Después de haber capturado un conjunto de paquetes utilizando el analizador potente "*Wireshark*", estamos en condiciones de analizar los datos del protocolo tanto de su cabecera como del cuerpo. Este análisis permitirá entender claramente el contenido de los campos de los diferentes protocolos que están en los paquetes. Además se podrá conocer las características de un paquete en detalles. A continuación veremos algunos ejemplos de paquetes capturados donde se muestran los protocolos ***Ethernet***, ***IP***, ***TCP***, ***UDP***, ***H.323*** y ***SIP***.

#### 3.3.3.1 Captura del protocolo *Ethernet*

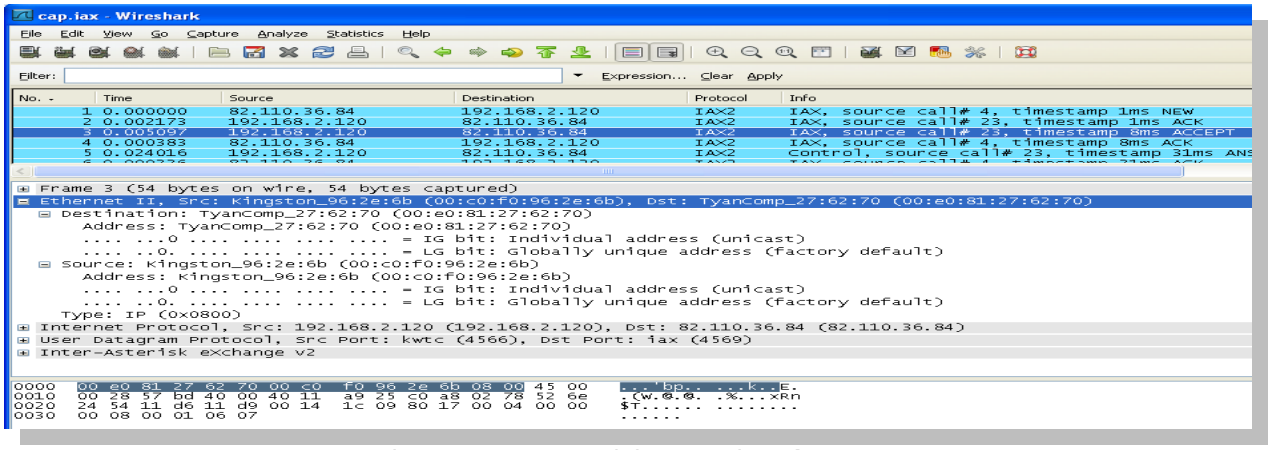



Figura 19. Captura del protocolo ***Ethernet***.

En la ***figura 19*** podemos observar los diferentes campos que componen la trama ***Ethernet*** con sus respectivos valores. En este caso en el campo Dirección de destino se encuentra la dirección compuesta por seis *byte* 00:e0:81:27:62:70 (en hexadecimal) y la dirección igualmente compuesta por seis *byte* 00:c0:f0:96:2e:6b (en hexadecimal) en el campo Dirección origen. Como habíamos visto anteriormente en el capítulo 2 el campo que indica cual es el protocolo que le sigue es el denominado Tipo (*Type*) que en este caso el valor que contiene es el 0800 el cual indica que los datos serán tratados por el protocolo *IP* de la capa de red. Además se muestra la información codificada en *byte* en el panel de paquetes capturados en *byte*.

### 3.3.3.2 Captura del protocolo IP

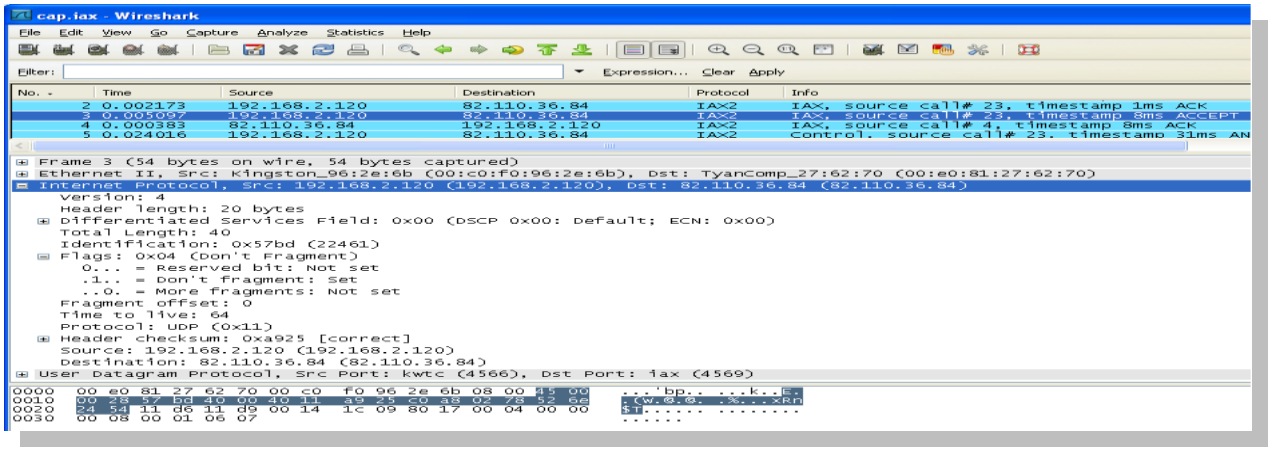



Figura 20. Captura del protocolo *IP*.

La ***figura 20*** muestra el paquete *IP* con sus respectivos valores en cada uno de sus campos. Los campos más importantes de dicho protocolo son Dirección *IP* origen que contiene el valor 192.168.2.120 (en decimal), Dirección *IP* destino con el valor 82.110.36.84 (en decimal) y el campo Protocolo que especifica cuál es el protocolo que traslada el paquete, los valores más comunes que puede tener este campo son 06 y 17 correspondientes a *TCP* y *UDP* respectivamente, ya que son los protocolos más usados para el transporte de datos. Además se muestra la información codificada en *byte* en el panel de paquetes capturados en *byte*.

### 3.3.3.3 Captura del protocolo *TCP*

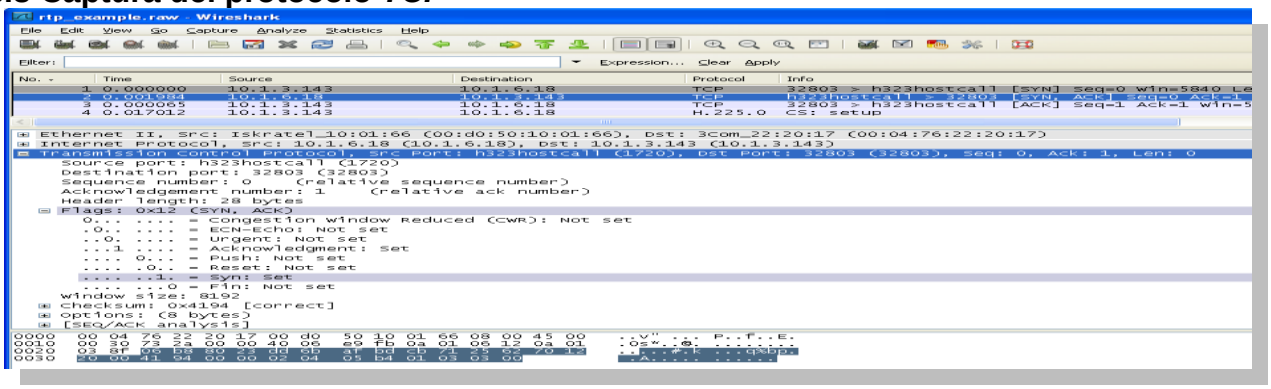



Figura 21. Captura del protocolo *TCP*.

Podemos ver el segmento *TCP* ***figura 21*** y los valores presentes en cada campo. Los campos más importantes del mismo son Puerto de origen que presenta el valor 1720, Puerto de destino con el valor 32803, que indica qué aplicación es la que se debe ejecutar en la máquina y los números de *SYN* y *ACK* indicados en la bandera (*flags*), importantes para el establecimiento de una sesión. Además se muestra la información codificada en *byte* en el panel de paquetes capturados en *byte*.

### 3.3.3.4 Captura del protocolo *UDP*

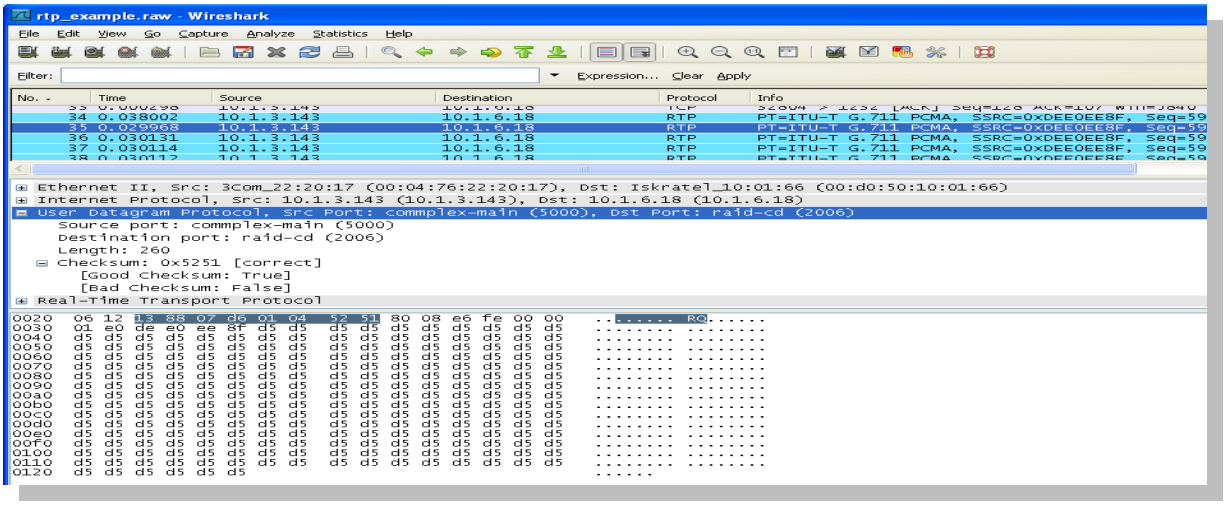



Figura 22. Captura del protocolo *UDP*.

En esta captura del protocolo se muestran los valores de lo campos que componen el datagrama *UDP* ***figura 22***. Los campos fundamentales son Puerto origen con el valor 5000, Puerto destino con el valor 2006 y la longitud del datagrama que en este caso presenta el valor 260. Además se muestra la información codificada en *byte* en el panel de paquetes capturados en *byte*.

Hasta el momento hemos visto los protocolos que facilitan la comunicación entre terminales y que traen consigo las porciones de información; *Ethernet*, *IP*, *TCP* y *UDP*. Ahora veremos los protocolos *H.323* y *SIP* que son objetivo de análisis en este trabajo. Además se verá toda la información referente a ellos y los diferentes mensajes que intervienen en el establecimiento de

una llamada *H.323* y *SIP*, los cuales fueron descritos en el procedimiento. Es importante acotar que para una llamada *H.323* existe un conjunto de mensajes y para una llamada *SIP* otros.

### 3.3.3.5 Captura del protocolo *H.323*, mensaje *SETUP*

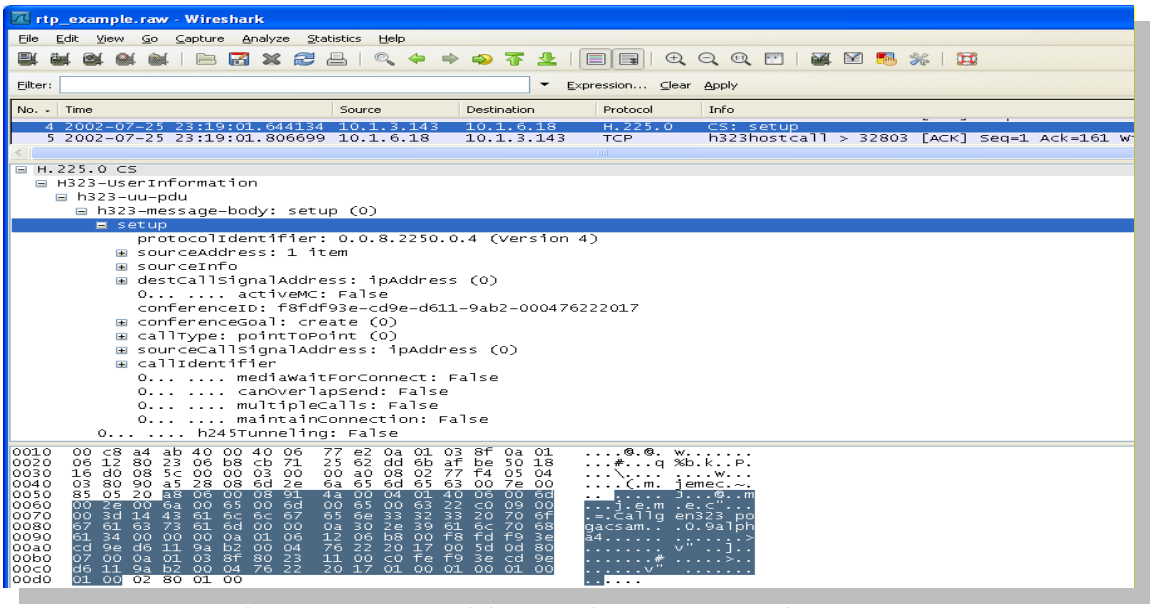



Figura 23. Captura del protocolo ***H.323***, mensaje ***SETUP***.

### 3.3.3.6 Captura del protocolo *H.323*, mensaje *CALL PROCEEDING*

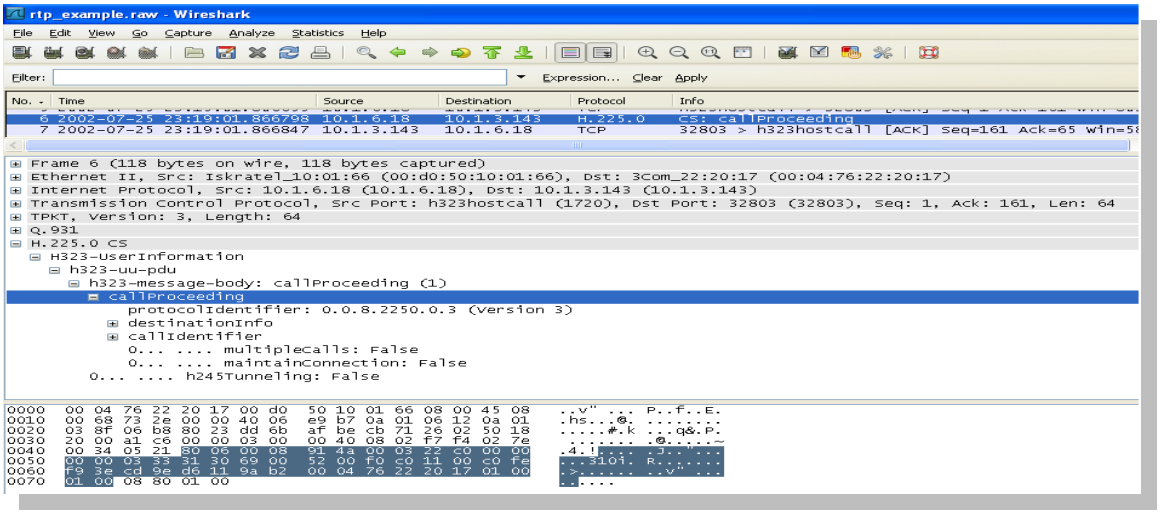



Figura24. Captura del paquete que contiene el mensaje ***CALL PROCEEDING***.

### 3.3.3.7 Captura del protocolo *H.323*, mensaje *ALERTING*

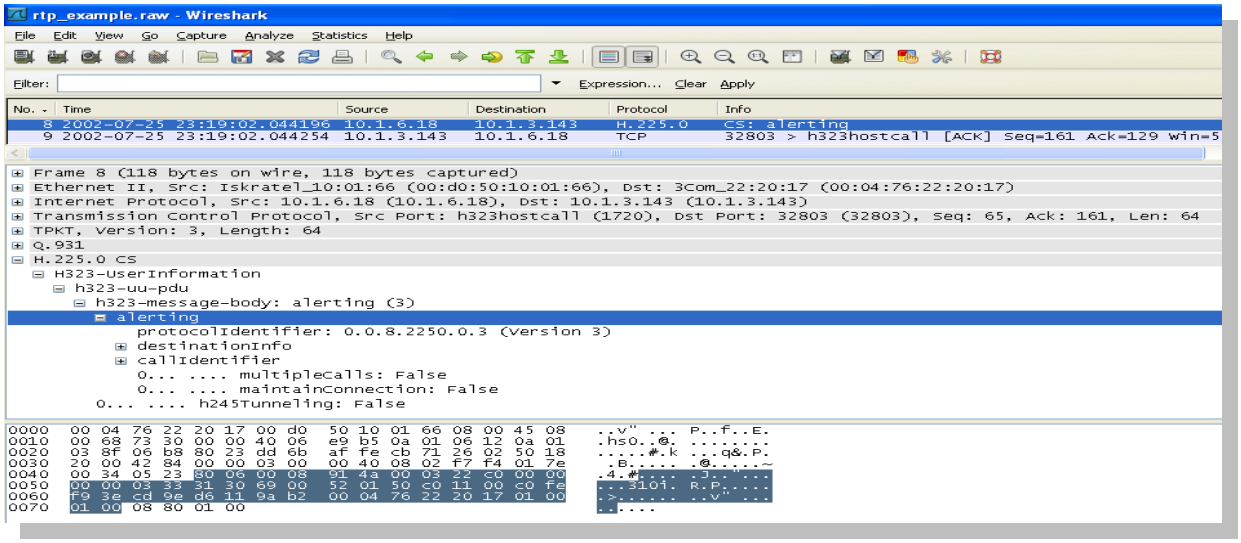



Figura25. Captura del paquete que contiene el mensaje ***ALERTING***.

### 3.3.3.8 Captura del protocolo *H.323*, mensaje *CONNECT*

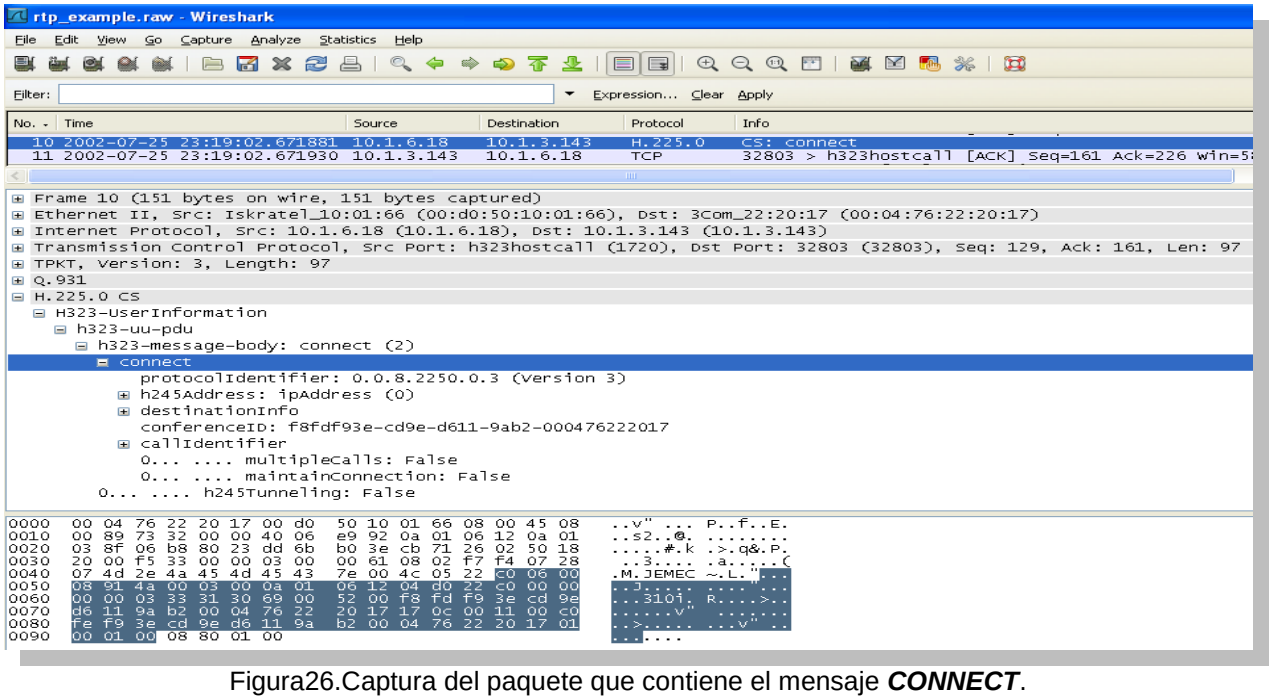



Figura26.Captura del paquete que contiene el mensaje ***CONNECT***.

### 3.3.3.9 Captura del protocolo *H.323*, mensaje *TerminalCapabilitySet*

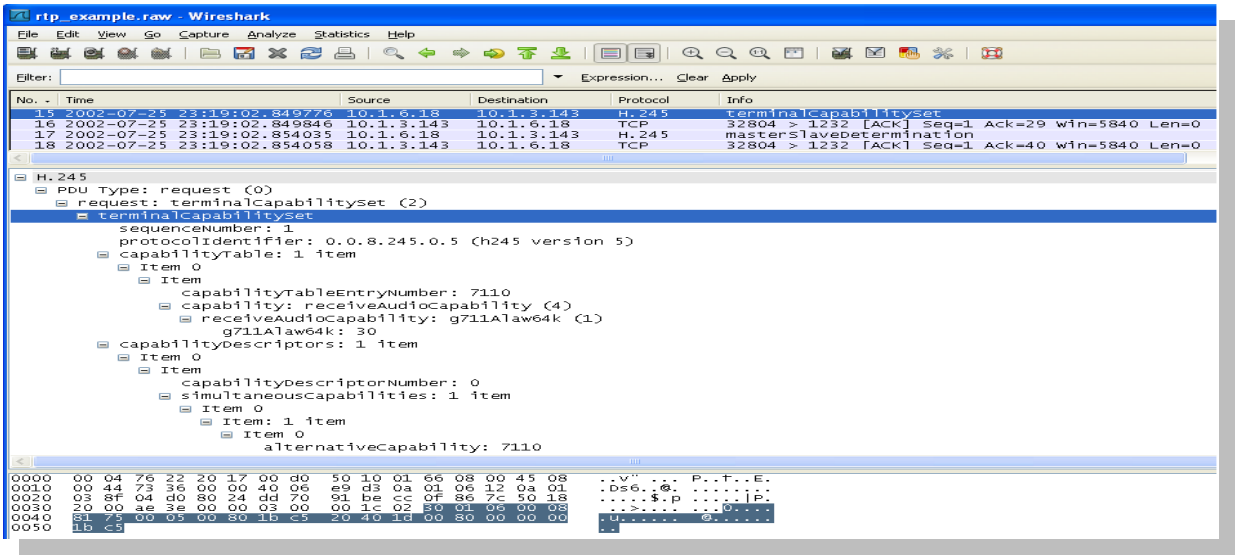



Figura 27.Captura del paquete que contiene el mensaje ***TerminalCapabilitySet***.

### 3.3.3.10 Captura del protocolo *H.323*, mensaje *MasterSlaveDetermination*

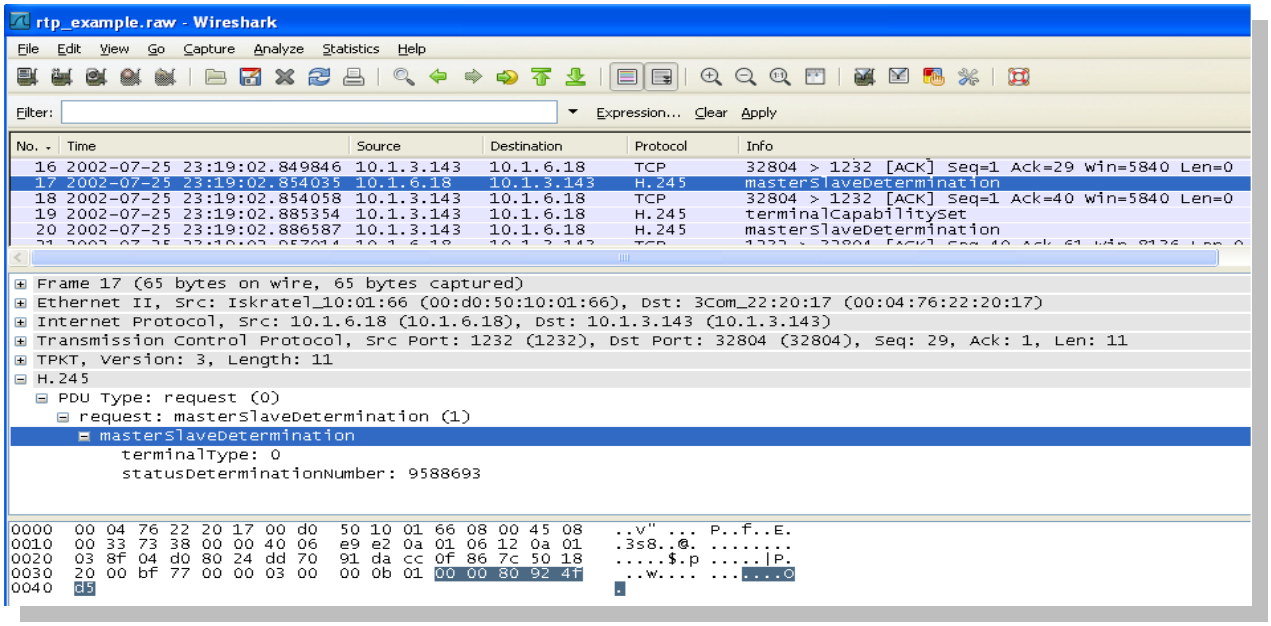



Figura 28. Captura del paquete que contiene el mensaje ***MasterSlaveDetermination***.

### 3.3.3.11 Captura del protocolo *H.323*, mensaje *TerminalCapabilitySet Ack*

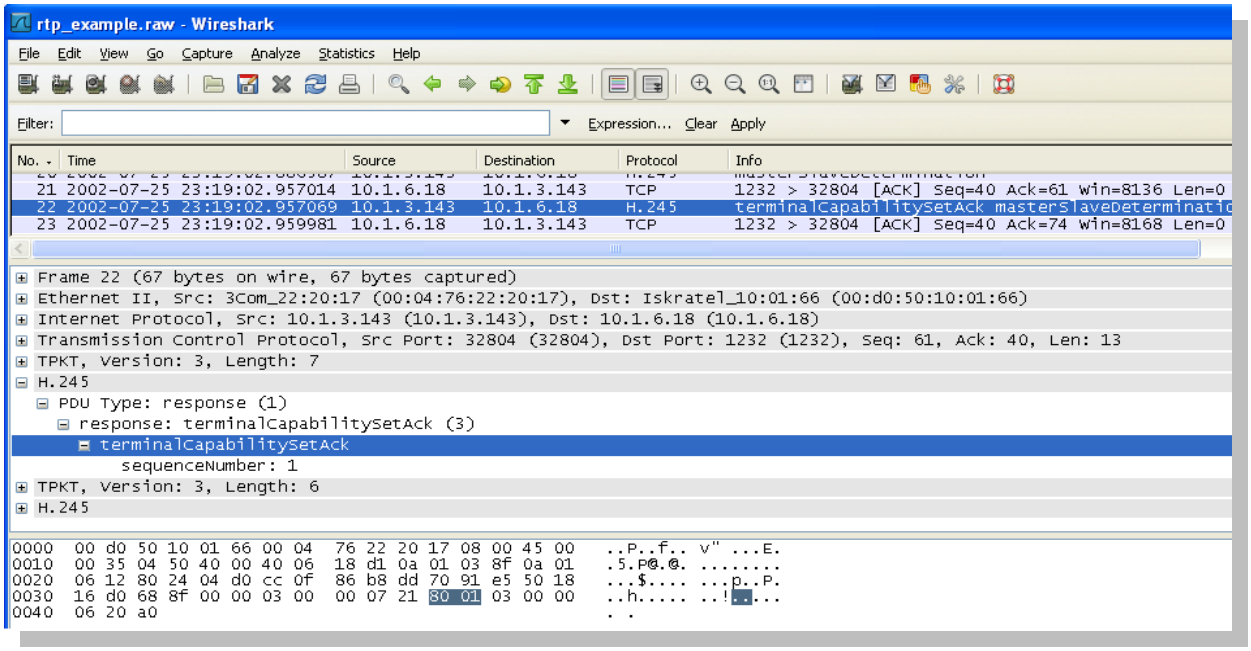



Figura 29.Captura del paquete que contiene el mensaje ***TerminalCapabilitySet Ack.***

### 3.3.3.12 Captura del protocolo *H.323*, mensaje *MasterSlaveDetermination Ack*

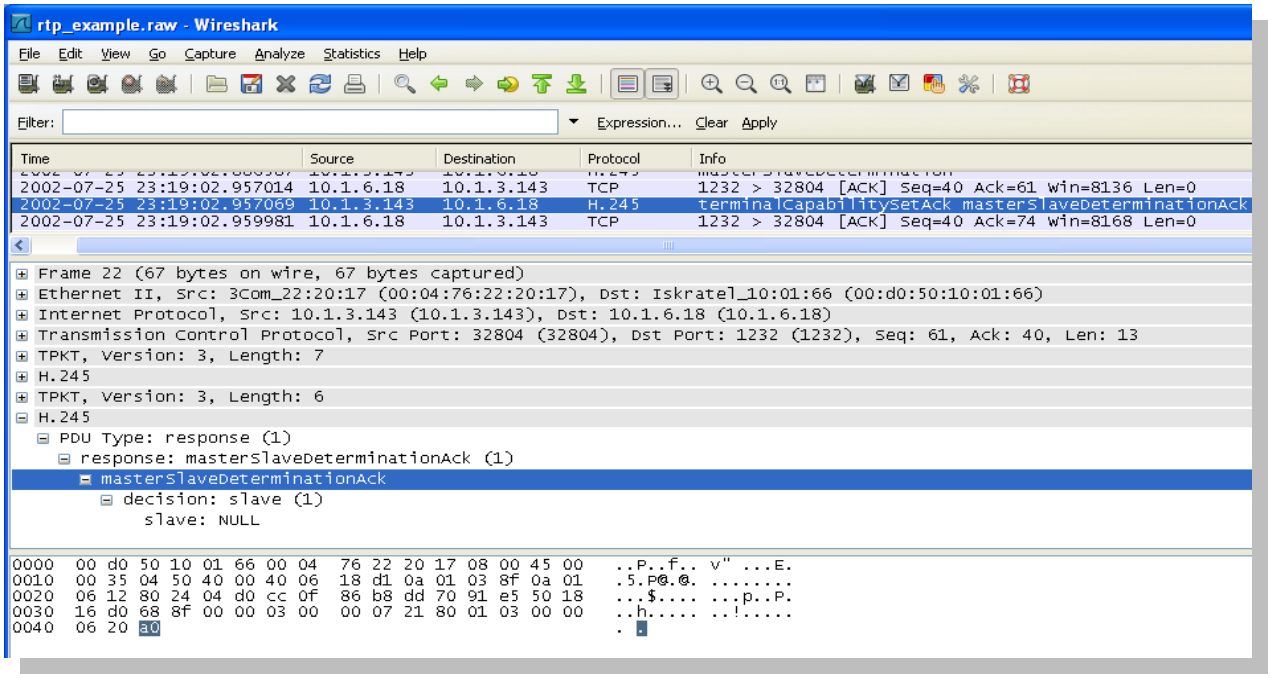



Figura 30.Captura del paquete que contiene el mensaje ***MasterSlaveDetermination Ack.***

### 3.3.3.13 Captura del protocolo *H.323*, mensaje *OpenLogicalChannel*

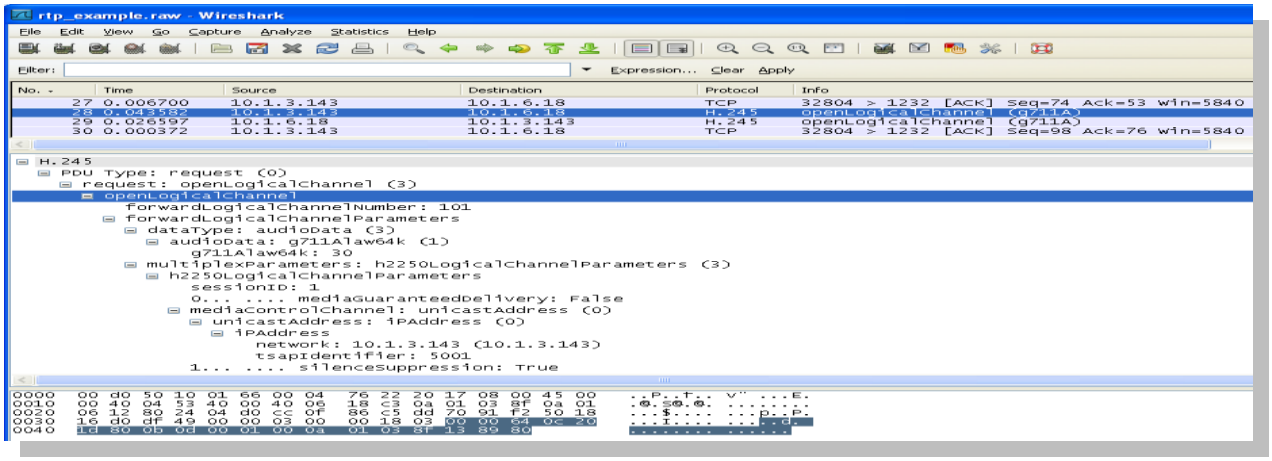



Figura 31.Captura del paquete que contiene el mensaje ***OpenLogicalChannel.***

### 3.3.3.14 Captura del protocolo *H.323*, mensaje *OpenLogicalChannel Ack*

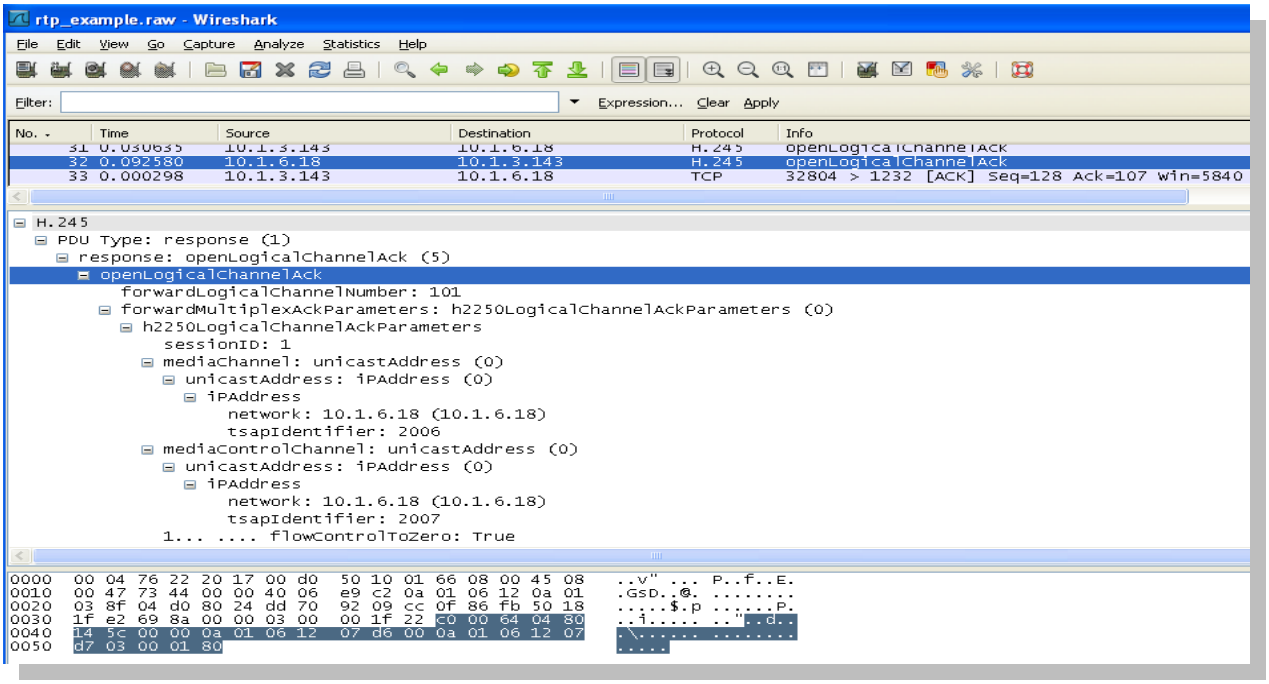



Figura 32. Captura del paquete que contiene el mensaje ***OpenLogicalChannel Ack.***

**Nota**: Este envío de mensajes se establece en ambos sentidos, antes de comenzar la conversación o intercambio de información entre los terminales, donde interviene el protocolo *RTP*.

### 3.3.3.15 Captura del paquete que contiene el protocolo *RTP*

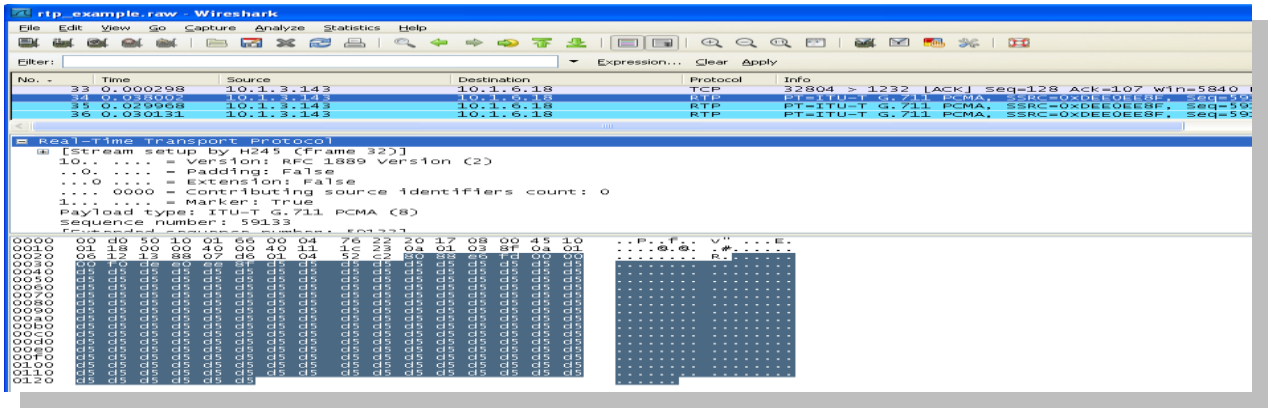



Figura 33. Captura del paquete que contiene el protocolo ***RTP***.

### 3.3.3.16 Captura del protocolo *SIP*, método *REGITER*

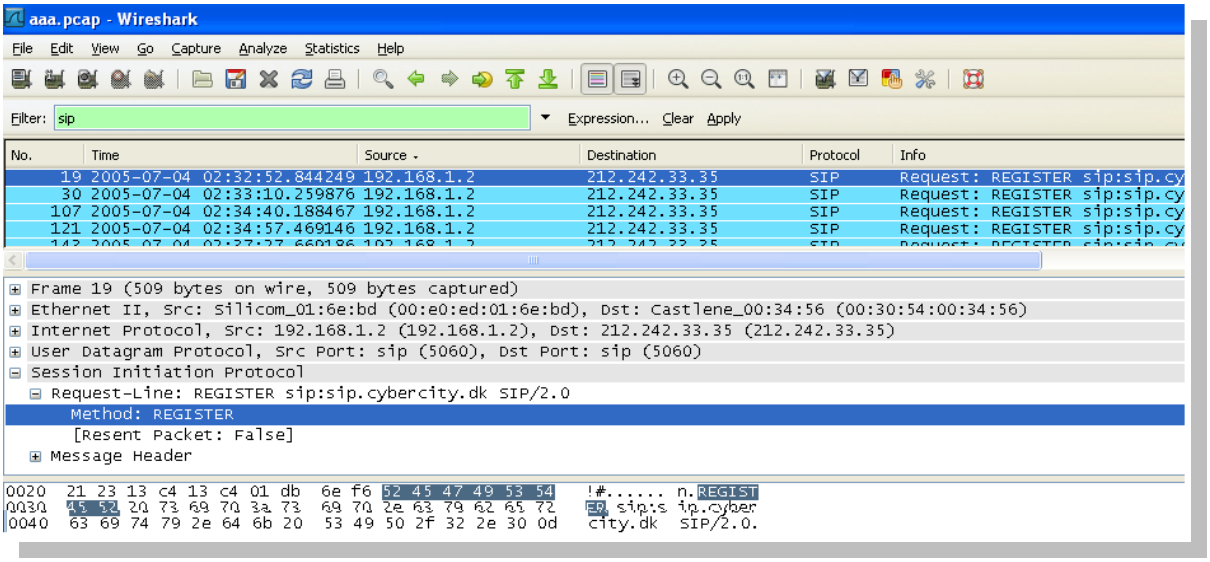



Figura 34. Captura de un paquete SIP usando el método ***REGITER***.

### 3.3.3.17 Captura del protocolo *SIP*, método *INVITE*

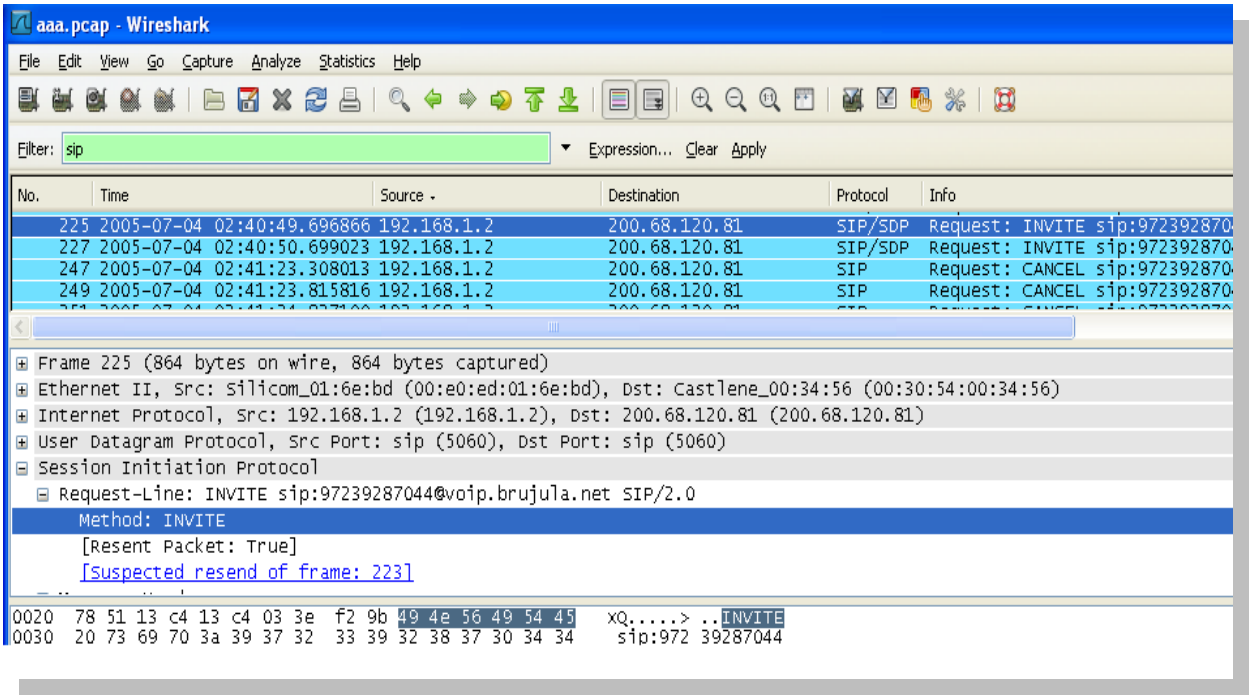



Figura 35. Captura de un paquete SIP usando el método ***INVITE***.

### 3.3.3.18 Captura del protocolo *SIP*, método *ACK*

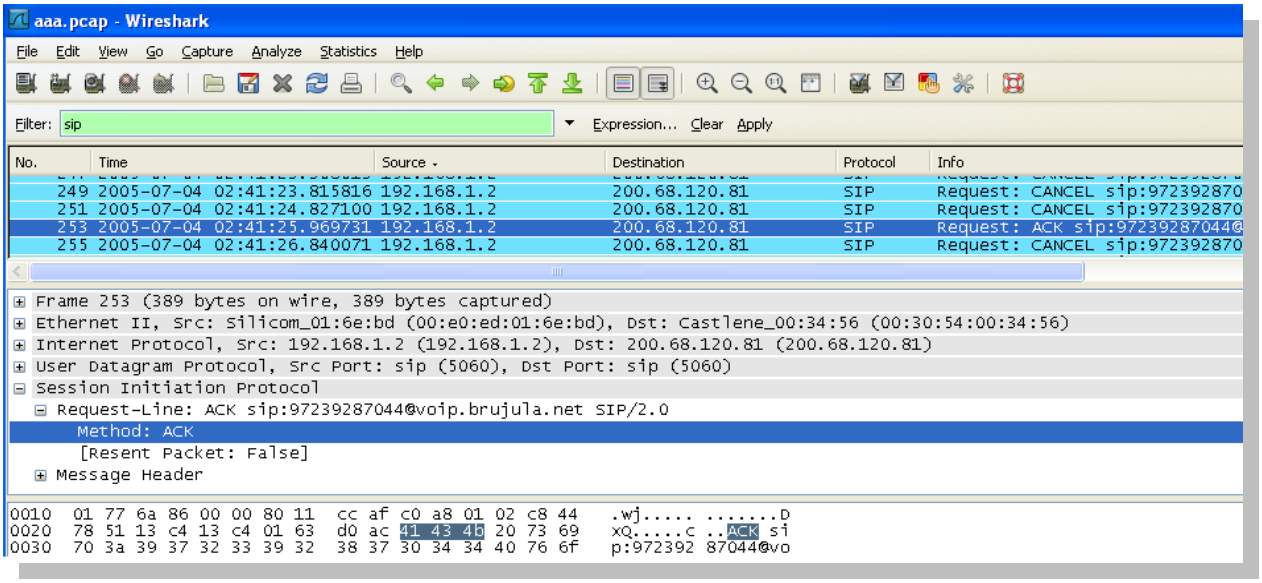



Figura 36. Captura de un paquete SIP usando el método ***ACK***.

### 3.3.3.19 Captura del protocolo *SIP*, método *BYE*

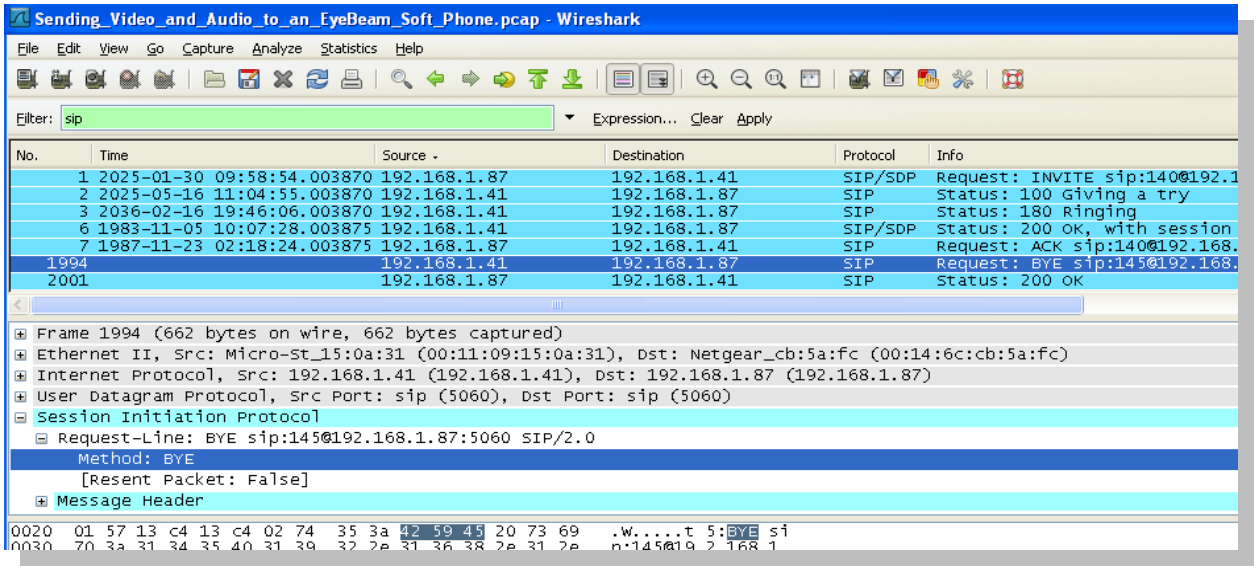



Figura 37. Captura de un paquete *SIP* usando el método ***BYE***.

### 3.3.3.20 Captura del protocolo *SIP*, método 200 *OK*

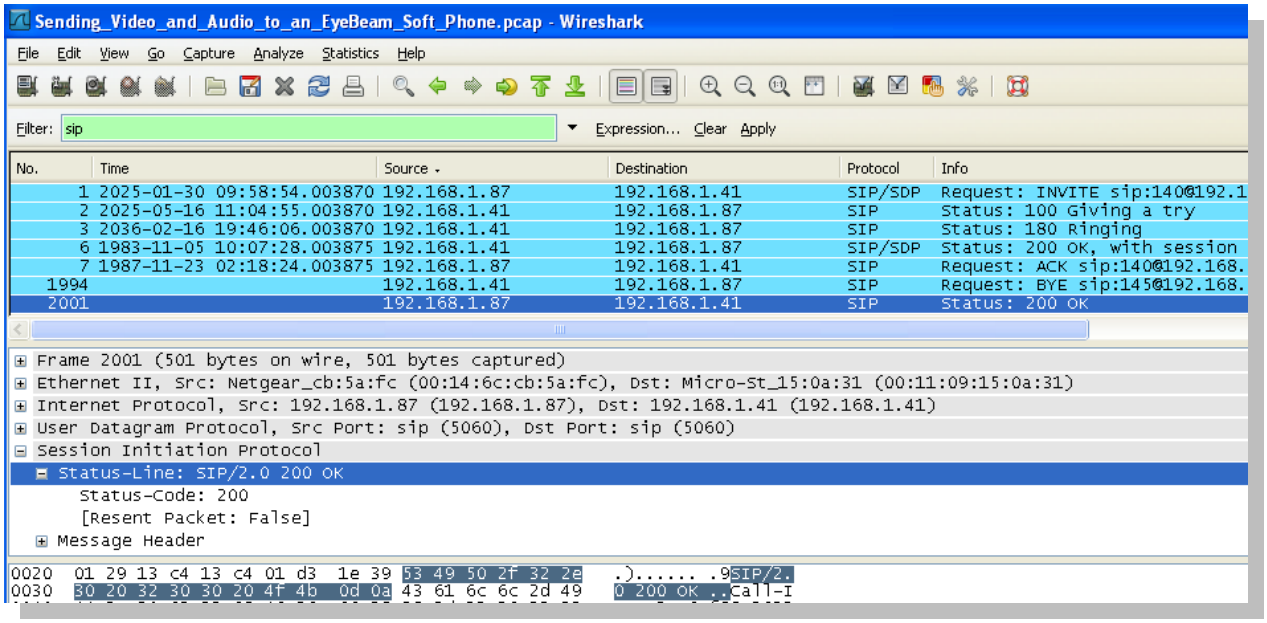



Figura 38. Captura de paquete SIP usando el método *200 OK*.

#### 3.3.3.21 Visualizando estadísticas

*Wireshark* proporciona un rango amplio de estadísticas de red a las cuales se puede acceder desde el menú *Statistics* que abarcan desde la información general de los paquetes capturados hasta las estadísticas específicas de un protocolo.

Estadísticas Generales

- *Summary*, la cantidad de paquetes capturados.
- *Protocol Hierarchy*, presenta las estadísticas para cada protocolo de forma jerárquica.
- *Conversations*, un caso particular es el tráfico entre una *IP* origen y una *IP* destino.
- *Endpoints*, muestra las estadísticas de los paquetes hacia y desde una dirección *IP*.
- *IO Graphs*, muestra las estadísticas en grafos.

Estadísticas específicas de los protocolos

- *Service Response Time* entre la solicitud (*request*) y la entrega *(response*) de algún protocolo existente.
- Entre otras.

Es importante tener presente que los números arrojados por estas estadísticas solo tendrán sentido si se tiene un conocimiento previo del protocolo, de lo contrario será compleja su comprensión.

### 3.3.4 Paso 4: Exportación de los datos.

Después de obtener y conocer los datos del protocolo especificado se necesita exportar o guardar la información en un fichero, para que esta sea procesada con un programa. Esta información se exporta de la siguiente forma **menú** -***File*** - ***Export*** – ***File-Lugar*****.** La ***figura 39*** es un ejemplo donde se ilustra de qué manera se guardan los datos usando la herramienta *Wireshark*.

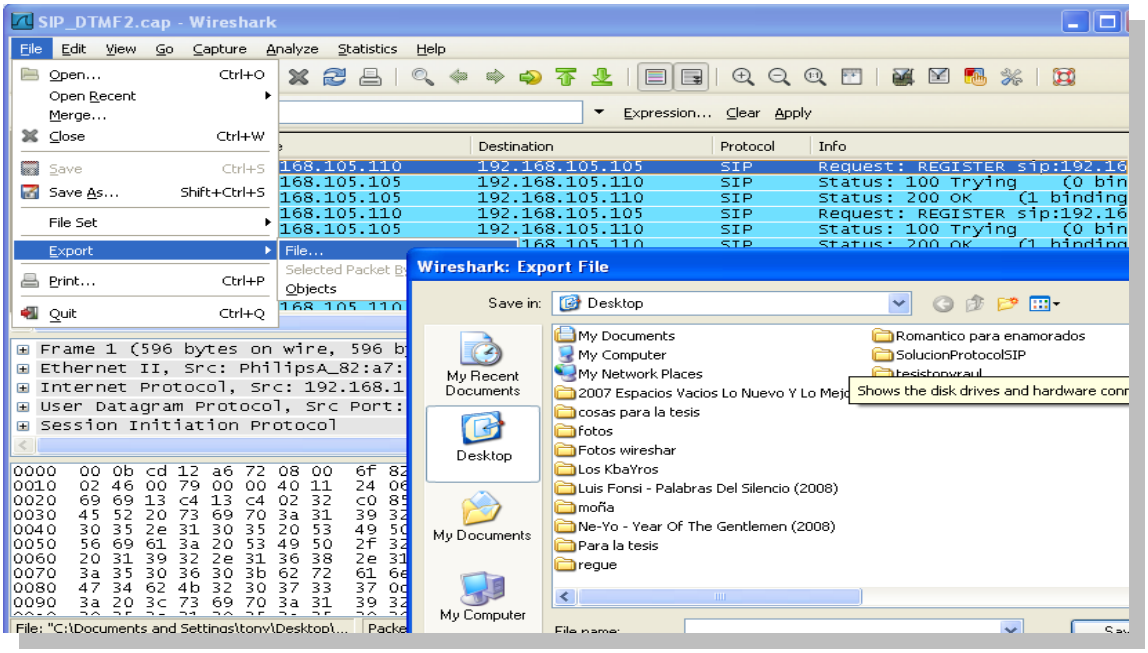



Figura 39. Como exportar los datos.

### 3.4 Propuesta de Base de Datos

Posterior a la captura de los protocolos, de analizar la información que traen los mismos y determinar qué datos se desea guardar, se puede almacenar dicha selección en una Base de Datos. Se realizó el diseño de una que consta de dos tablas, una para cada protocolo. Este diseño no es más que una propuesta, la cual servirá para almacenar la información que fue seleccionada de cada protocolo.

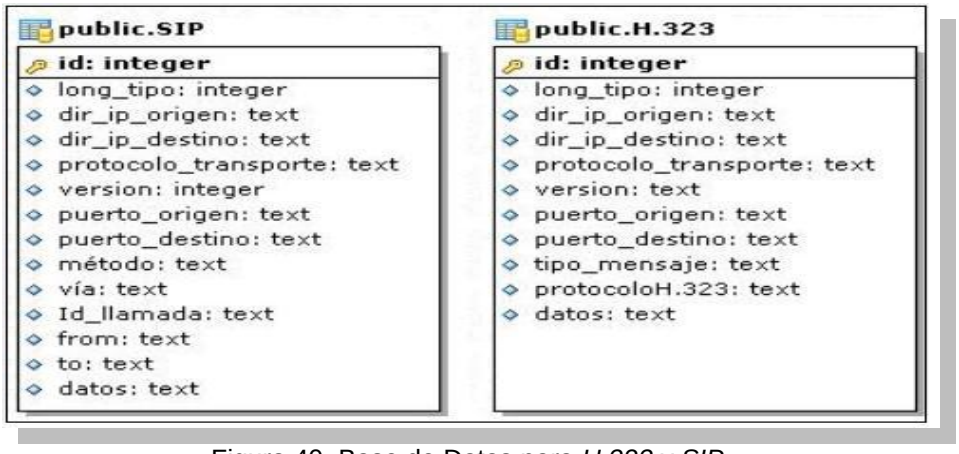



Figura 40. Base de Datos para *H.323* y *SIP*.

## 3.5 Conclusiones

En este capítulo se hicieron pruebas al procedimiento con el objetivo de demostrar la validez del mismo. Para la realización de las pruebas se utilizo una herramienta de captura de protocolos para mostrar los datos de los diferentes protocolos que se describen en el procedimiento. Como resultado principal se demostró la factibilidad y aceptación del procedimiento, pues al seguir los pasos que se plantearon, se verifico la relación de los datos capturados con lo que en el procedimiento se describe. Además como parte de este capítulo se hizo el diseño de una base de datos para el almacenamiento de los datos de los protocolos que se capturaron, de manera tal que se puedan procesar los mismos.

## CONCLUSIONES GENERALES

A medida que se fueron realizando las tareas se le dio cumplimiento al objetivo propuesto, hasta llegar a la realización de un procedimiento para el procesamiento de los protocolos usados en las comunicaciones de voz sobre redes *IP* (*SIP* y *H.323*).

Para la realización del mismo fue necesario hacer un estudio detallados de las redes, haciendo énfasis en el modelo *TCP/IP*, el cual sirvió de apoyo para comprender cómo es el funcionamiento de las mismas y de los protocolos que en ella están implementados. Con el objetivo de lograr un procedimiento óptimo. Es importante señalar que todos los pasos del procedimiento tienen un orden lógico, los que deben ser respetados para que cada lector o persona interesada en el tema logre entender con claridad la información que brinda.

Las pruebas realizadas estaban enfocadas en demostrar la funcionalidad del procedimiento. Al comparar los datos capturados siguiendo los pasos que el procedimiento describe, se verificó que dichos pasos son correctos y tiene un orden lógico. Además se propone el diseño de una base de datos, para almacenar los datos de los protocolos que se capturan, de manera tal que puedan ser procesados.

**RECOMENDACIONES**

Es importante recomendar una serie de acciones a realizar, por ello se propone:

- Desarrollar aplicaciones que permitan manejar los datos de los protocolos.
- Crear una Base de Datos que posibilite almacenar la información necesaria que contienen los protocolos analizados.
- Continuar el estudio de otros protocolos que se utilicen en las comunicaciones de voz sobre redes *IP* (sean estándares o no).
- Continuar el trabajo en grupos o equipos en el desarrollo de otros proyectos relacionados con las redes, dada la efectividad que se logra.
- Continuar profundizando en aspectos nuevos relacionados con los protocolo *H.323* y *SIP* que vayan surgiendo, para mantener actualizado el procedimiento descrito.

**BIBLIOGRAFÍA**

[1]. UIT. Informe del Secretario General sobre Telefonía IP. Foro Mundial de Políticas de Telecomunicaciones. Ginebra 7-9 de Marzo del 2001. 37 p.

[2]. ITU. IP TELEPHONY. Group of Experts - Technical Aspects. Geneva. 13-14 December 2001: 46 p. IP Tel 3/X-E.

[3]. Castañeda Segura, Rodolfo. "*Protocolos para voz IP*". Dirección de Telemática. CICESE, 2005.

[4]. V. Estay, C. Frez, "*Estudio y aplicación del transporte de voz sobre intranet IP"*, Memoria de Título, UTFSM, 2001.

[5]. Unitronics Comunicaciones*, "El estándar VoIP – Voz sobre IP"*, Madrid, 1998. <http://www.comunicaciones.unitronics.es/tecnologia/voip.htm>

[6]. DAI (Dittberner Associates, Inc. Worlwide Digital Switching Status and Forecast [en línea]. Noviembre 6, 2003. Disponible en: <http://www.dittberner.com/news/pr20031106.php>

[7]. Buchli, M.J.C.; De Vleeschauwer; et al. Calidad de las Llamadas transportadas sobre una red de acceso DSL. Revista de Telecomunicaciones de Alcatel. NGN: La Actual Próxima Generación. Segundo Trimestre del 2001: 111-115.

[8]. Ing. Beceiro García, Humberto. "*Telefonía IP*". Departamento de Telemática, Diplomado en Telemática. ISPJAE Facultad de Ingeniería Eléctrica, 2001.

[9]. Evslin, Mary. "Internet Telephony Industry". Serie de tres partes sobre la Industria, temas de QoS y cómo hacer mucho dinero". ITXC Seminar on Internet Telephony (Presentación), 2000.<http://seminar.techonline.com/itxc1/archive/welcome.htm>

[10]. Cerf, Vint. "Cerf's Up: Social, Economic and Regulatory Issues. "Internet in the next five to ten years", <http://www.worldcom.com>

[11]. Alcatel. Voice and Multimedia. Next Generation Services [en línea]. Sitio de Alcatel. Disponible en <http://www.alcatel.com/solutions/solutionsbyportafolio.jhtml?_DARGS=/>

[12]. Alcatel. Managed Data Services. Products in the Solution (Carrier Data and IP Plattforms) [en línea]. Sitio de Alcatel. Disponible en: <http://www.alcatel.com/solutions/productsinsolution.jhtml>

[13]. Systems, C., Cisco CallManager Express 3.2 System Administrator Guide. 2005.

[14]. RFC 1889. H.Shulzrinne S.Castner, R.Frederick, V.Jacobson. RTP: A transport protocol for real time protocol.

[15]. ITU-T Recommendation H.323: "Packet-based Multimedia Communications Systems", November 2000

[16]. CCP.I/doc.1111/01, "Redes de voz y protocolos de próxima generación"

[17]. Ignacio Moreno, Jose, Soto, Ignacio, Larrabeiti, David. "Protocolos de Señalización para el transporte de Voz sobre redes IP1". Departamento de Ingeniería Telemática. Universidad Carlos III de Madrid.

[18]. CCP.I/doc.1112/01, "Breve explicación del SIP"

[19]. CCP.I/doc.1344/01, "Redes de próxima generación – Normas de rendimiento/QoS".

[20]. Organización de los Estados Americanos, Comisión Interamericana de Telecomunicaciones. Carpeta Técnica: Redes de Próxima Generación [en línea]. Septiembre del 2003. Disponible en <http://citel.oas.org>

[21]. Beceiro García, Humberto, Gradaille Cosa, Ricardo, Gago Romero, José. "*MIGRACION DE LAS REDES DE TELECOMUNICACIONES, Consideraciones en el escenario cubano*". XV FORUM DE CIENCIA Y TECNICA GERENCIA TERRITORIAL DE ETECSA CIEGO DE AVILA, 2004.

[22]. RFC 2068. R. Fielding and others. Hypertext Transfer Protocol -- HTTP/1.1

[23]. RFC 2327. 2327 M. Handley, V. Jacobson. SDP: Session Description Protocol.

[24]. RFC 2396. T.Berners-Lee, R.Fielding, Uniform Resource Identifiers (URI): generic syntax.

[25]. Recomendación UIT-T H.225.0, versión 4, Protocolos de señalización de llamada y paquetización de trenes de medios para sistemas de comunicación multimedios por paquetes, 2000.

[26]. Olivier, Hersent; David, Gurle; Jean-Pierre, Petit. IP Telephony. Packect-based multimedia communications systems. Adisson-Wesley; 2000.

[27]. Alcatel. Alcatel 5620 Network Manager [en línea]. Sitio Web de Alcatel. Disponible en: <http://www.alcatel.com/products/productsbysubfamily.jthtml?subCartegory=Management52

0of20DATA/.>

[28]. Alcatel. IP Telephony Design Guide [en línea]. 2003. 11 p. Disponible en: <http://www.alcatel.com/enterprise.>

[29]. Alcatel. Alcatel 1000 MM E10. Media Gateway Controller. Architects of an Internet World. 2002. 8p.

[30]. Corporation, NetIQ. Checklist of VoIP Network Design Tips [en línea]. 2001. 2 p. Disponible en: <http://www.netiq.com/voip. >

[31]. España, Alcatel. Productos por Categoría: Soporte de Datos y plataformas IP [en línea]. Disponible en: <http://www.alcatel.es/Productos Alcatel/Builds Next> Generation2.htm.

[32]. Hersen, Olivier; Gurle, David; Pierre-Petit, Jean. “IP Telephony. Sistemas de comunicaciones Multimedia basadas en paquetes.” ADISSON_WESLEY, 2000.

[33]. UIT-T, Recomendación H.323. *"Sistemas de Comunicaciones Multimedia Basados en Paquetes"* Septiembre 1999.

[34]. (RFC H323) Versión traducida de http://www.packetizer.com/ipmc/h323/standards.html. [citado 3 Abril 2009]. Disponible en: <http://translate.google.com.cu/translate?hl=es&sl=en&u=http://www. packetizer.com/ipmc/h323/standards.html&ei=FDLWSZHzG97MlQfgmMyVDA&sa=X&oi=translate&resnum=1&ct=result&prev=/search%3Fq%3Drfc%2Bh323%26hl%3Des%26client%3Dfirefox-a%26channel%3Ds%26rls%3Dorg.mozilla:en-US:official%26sa%3DG>.

[35]. Analizador de Protocolo de Red SoftPerfect (SoftPerfect Network Protocol Analyzer) por SoftPerfect Research - reporte y descarga. [citado 7 Abril 2009]. Disponible en: <http://www.freedownloadmanager.org/es/downloads/Analizador_de_Protocolo_de_Red_de_SoftPerfect_8116_p/>.

[36]. cliente/servidor » imagenes cliente - servidor. [citado 2 Junio 2009]. Disponible en: <http://www.gratisblog.com/imagenes_cliente_servidor/i124521-imagenes_cliente_-_servidor.htm>.

[37]. Descarga gratis VisualSniffer. VisualSniffer es una potente herramienta de captura de paquetes y analizador de protocolo (rastreador -sniffer- de direcciones IP o rastreador de

paquetes [citado 7 Abril 2009]. Disponible en: <http://www.freedownloadscenter.com/es/Red_e_Internet/Herramientas_para_la_Gestion_de_Redes/VisualSniffer.html>.

[38]. ebierzo.com Cultura, internet desde el Bierzo | Términos informáticos (2.006). [citado 4 Mayo 2009]. Disponible en: <http://www.ebierzo.com/especiales/terminos-informaticos-2006>.

[39]. Ethereal (Wireshark) - Analizador de Protocolos de Red - MegaUNDER - La Revolución es Digital. [citado 7 Abril 2009]. Disponible en: <http://www.megaunder.com.ar/seguridad/143-ethereal-wireshark-analizador-de-protocolos-de-red.html>.

[40]. FORMATO MENSAJE UDP. [citado 22 Mayo 2009]. Disponible en: <http://pegaso.ls.fi.upm.es/~lmengual/anexo/sld011.htm>.

[41]. FORMATO SEGMENTO TCP. [citado 22 Mayo 2009]. Disponible en: <http://pegaso.ls.fi.upm.es/~lmengual/anexo/sld010.htm>.

[42]. H.323: Paquete basado en los sistemas de comunicaciones multimedia. [citado 3 Abril 2009]. Disponible en: <http://74.125.93.132/translate_c?hl=es&sl=en&u=http://www.itu.int/rec/T-REC-H.323-200606-I/en&prev=/search%3Fq%3Drfc%2Bh323%26hl%3Des%26client%3Dfirefox-a%26channel%3Ds%26rls%3Dorg.mozilla:en-US:official%26sa%3DG&usg=ALkJrhgiwOhbTR89ZX8YBBxU8Hl053ZyRA>.

[43]. Informática-PC Glosario P. [citado 4 Mayo 2009]. Disponible en: <http://www.informatica-pc.net/glosario/glosario_p.php>.

[44]. Ip-telephony Protocolos. [citado 17 Marzo 2009]. Disponible en: <http://www.scribd.com/doc/7353499/14-Iptelephony-Protocolos>.

[45].practica_capturaredes.pdf (application/pdf Object) wireshark. [citado 7 Abril 2009]. Disponible en: <http://centros4.pntic.mec.es/ies.luis.de.lucena/eds/practica_capturaredes.pdf>.

[46].Protocolo TCP. [citado 22 Mayo 2009]. Disponible en: <http://es.kioskea.net/contents/internet/tcp.php3>.

[47].RFC (SIP). [citado 3 Abril 2009]. Disponible en: <http://www.rfc-editor.org/rfc/rfc3261.txt>.

[48].RFC 4123 - Session Initiation Protocol (SIP)-H.323 Interworking Requisitos. [citado 3 Abril 2009]. Disponible en: <http://74.125.93.132/translate_c?hl=es&sl=en&u=http://www.rfc-archive.org/getrfc.php%3Frfc%3D4123&prev=/search%3Fq%3Drfc%2Bh323%26hl%3Des%26client%3Dfirefox-a%26channel%3Ds%26rls%3Dorg.mozilla:en-US:official%26sa%3DG&usg=ALkJrhjLuxte4E4TU-fyAzC60VJ8obOSkQ>.

[49]. RFC-es - Grupo de Traducción de RFC al español. [citado 3 Abril 2009]. Disponible en: <http://www.rfc-es.org/>.

[50]. RFC-ES Grupo de traducción al español de RFC - Documentos en proceso de traducción. [citado 3 Abril 2009]. Disponible en: <http://www.rfc-es.org/asignadas.php>.

[51]. RFC-ES Grupo de traducción al español de RFCs - Documentos Traducidos. [citado 3 Abril 2009]. Disponible en: <http://www.rfc-es.org/descargas.php>.

[52]. Sistemas de telecomunicaciones. Concepto de IP en las nuevas redes Integradas – Monografias.com. [citado 2 Junio 2009]. Disponible en: <http://www.monografias.com/trabajos33/telecomunicaciones/telecomunicaciones2.shtml>.

[53]. Versión traducida de http://compnetworking.about.com/cs/voicefaxoverip/g/bldef_h323.htm. [citado 3 Abril 2009]. Disponible en: <http://translate.google.com.cu/translate?hl=es&sl=en&u=http://compnetworking.about.com/cs/voicefaxoverip/g/bldef_h323.htm&ei=2z3WSeKxOKDglQe6voTVDA&sa=X&oi=translate&resnum=8&ct=result&prev=/search%3Fq%3Dh%2B323%2Bprotocol%26hl%3Des%26client%3Dfirefox-a%26channel%3Ds%26rls%3Dorg.mozilla:en-US:official%26sa%3DX>.

[54]. Versión traducida de http://www.rfc-editor.org/rfc/rfc3508.txt. [citado 3 Abril 2009]. Disponible en: <http://translate.google.com.cu/translate?hl=es&sl=en&u=http://www.rfc-editor.org/rfc/rfc3508.txt&ei=_kPWSdjPDZLWlQfi2rDkDA&sa=X&oi=translate&resnum=3&ct=result&prev=/search%3Fq%3Drfc%2Bde%2BH323%26hl%3Des%26client%3Dfirefox-a%26channel%3Ds%26rls%3Dorg.mozilla:en-US:official%26sa%3DG>.

[55]. VoIP Foro - H.323 Arquitectura H323: Protocolo VoIP. [citado 16 Mayo 2009]. Disponible en: <http://www.voipforo.com/H323/H323objetivo.php>.

[56]. VoIP Foro – H323 Componentes H.323: Gatekeeper, gateway, proxy. [citado 16 Mayo 2009]. Disponible en: <http://www.voipforo.com/H323/H323componentes.php>.

[57]. VoIP Foro - H323 Ejemplo comunicación H.323. [citado 16 Mayo 2009]. Disponible en: <http://www.voipforo.com/H323/H323ejemplo.php>.

[58]. VoIP Foro - H323 Protocolos H.323 - H.245, RAS, Q. 931, RTP, H225, RSVP. [citado 16 Mayo 2009]. Disponible en: <http://www.voipforo.com/H323/H323pilaprotocolos.php>.

[59]. VoIP Foro - H323 Señalización H.323: H.225, Q.931 y H.245. [citado 16 Mayo 2009]. Disponible en: <http://www.voipforo.com/H323/H323senalizacion.php>.

[60]. VoIP Foro - Protocolos Voz sobre IP: SIP y H.323. [citado 11 Mayo 2009]. Disponible en: <http://www.voipforo.com/protocolosvoip.php>.

[61]. Ortega, Ing. Israel. Voz sobre IP Esquemas de Funcionamiento. De Abril 2004. [citado 17 Marzo 2009]. Disponible en: <http://www.cudi.edu.mx/primavera_2004/presentaciones/israel_ortega.pdf>.

# ANEXOS

## Anexo 1 Redes.

En la figura1 se muestra un ejemplo de como pasan de una red tradicional a una red telefónica, también se muestran los componentes de dichas redes:

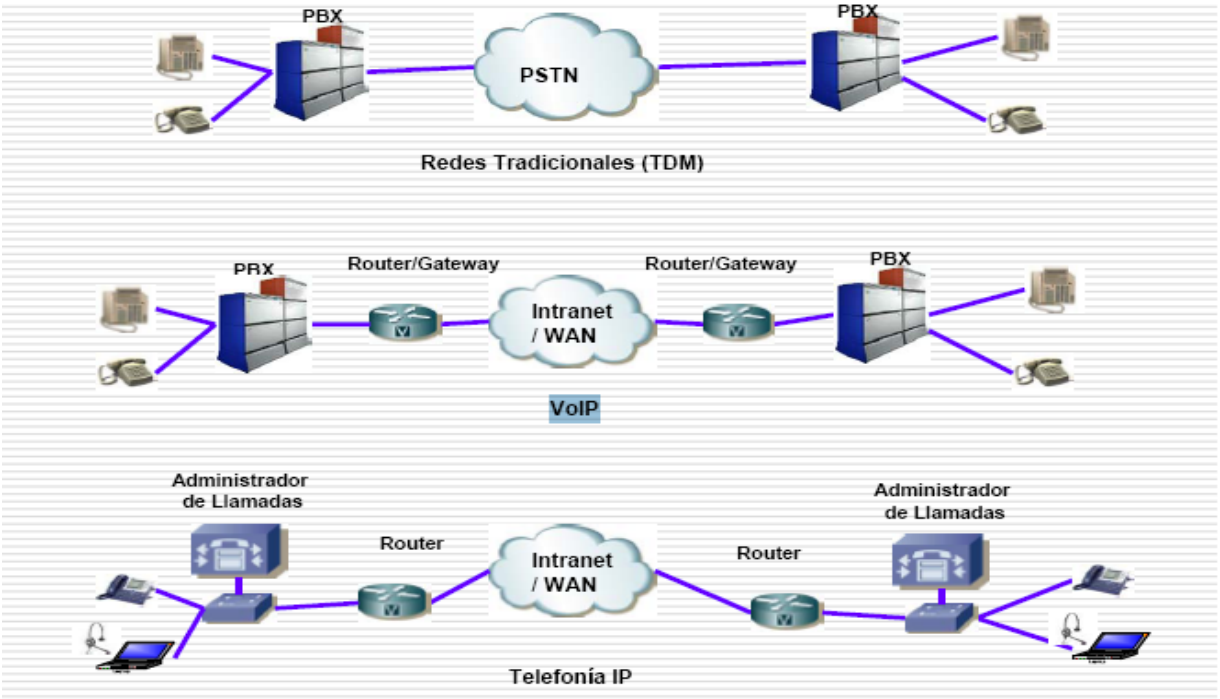



Figura 1. Evolución de las redes telefónicas hasta llegar a la telefonía IP.

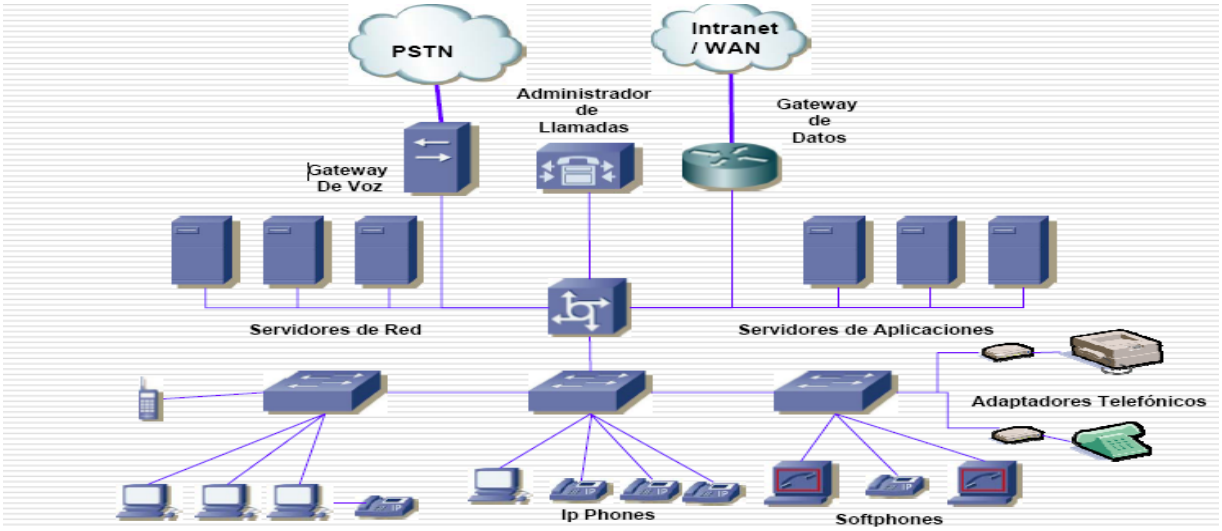



Figura 2. Arquitectura de sistema de Telefonía IP.

En la siguiente figura se muestra los elementos que componen una red de voz ip.

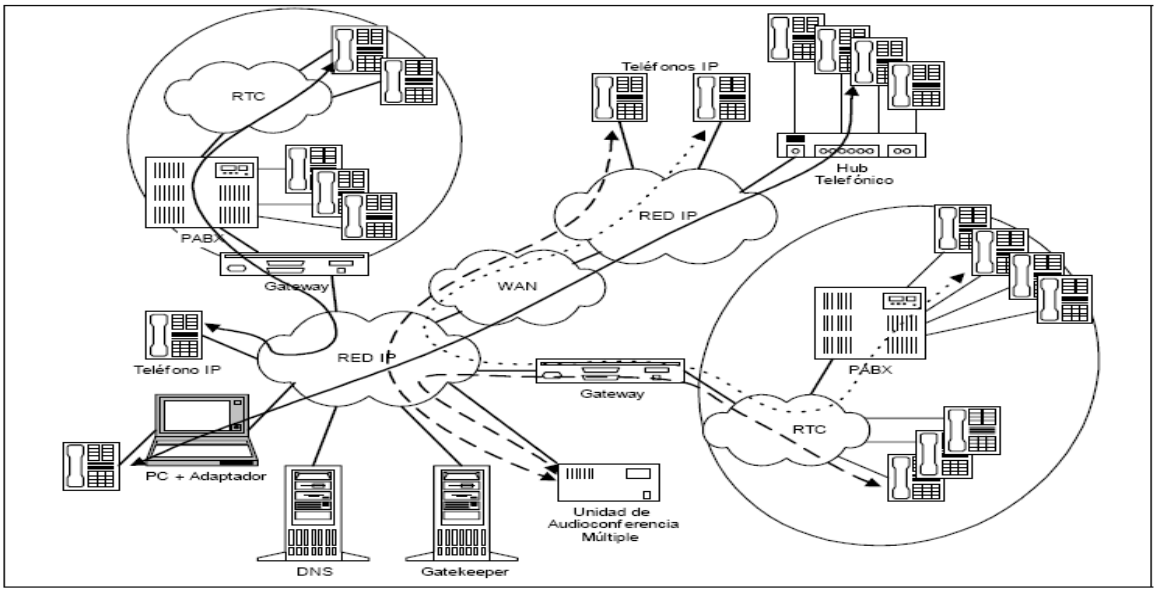



Figura 3. Elementos de una red VoIP

## Anexo 2. Prueba de campo y Soluciones

Prueba de campo que se realizo en ETECSA en el 2003 dicha prueba fue desarrollada por el Grupo de Investigación y Desarrollo de la Unidad de Negocios de la Red, utilizando equipamiento de VocalTec. La configuración del sistema instalado.

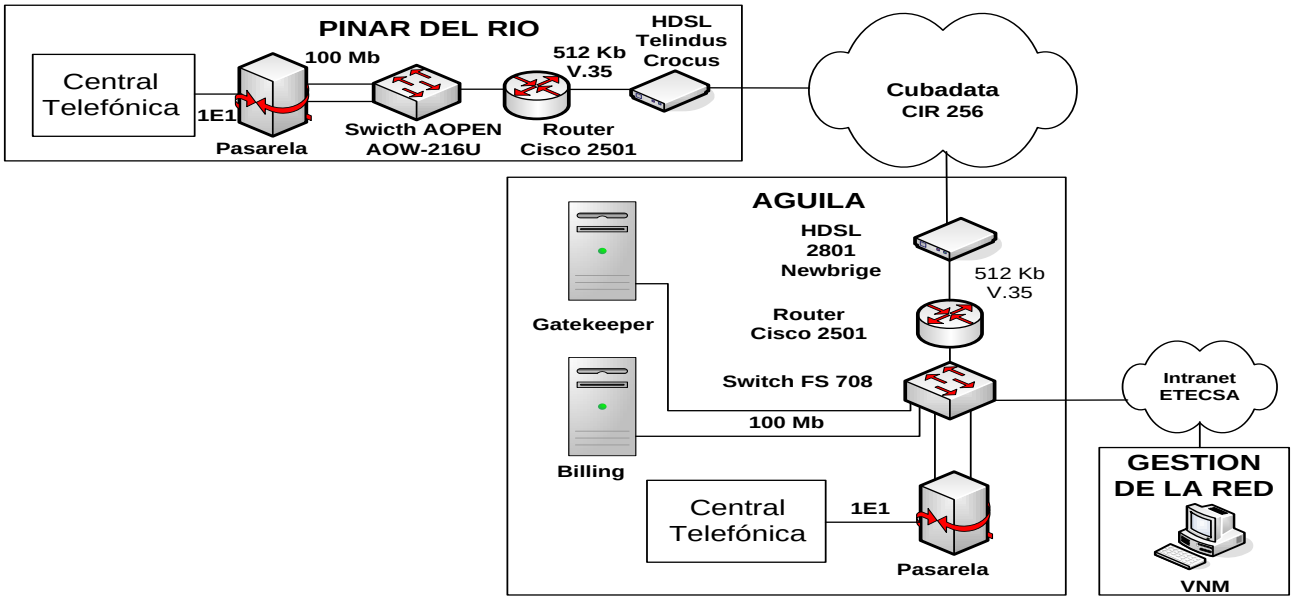



Figura 1. Red para prueba de campo de Telefonía IP con equipamiento VocalTec en ETECSA

## Solución de VocalTec (VEA)

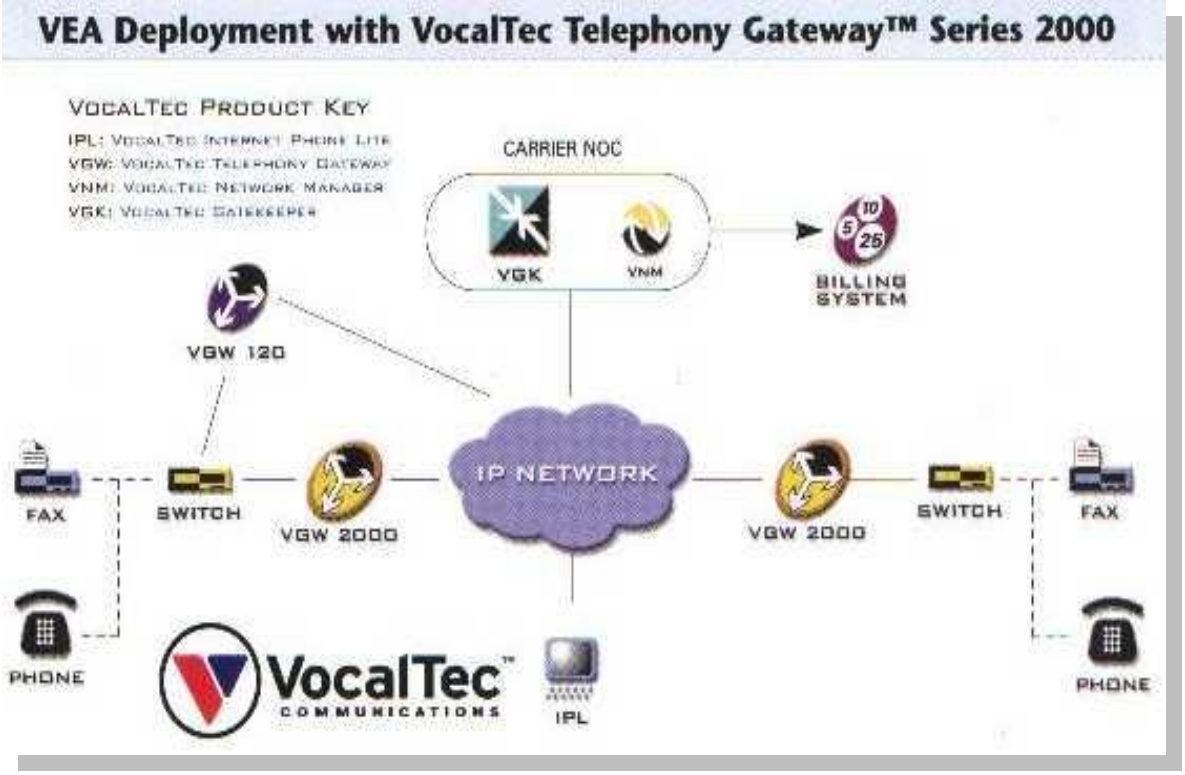



Figura 2. Soluciones de VocalTec (VEA)

## Estructura de la red piloto de ALCATEL para VoIP

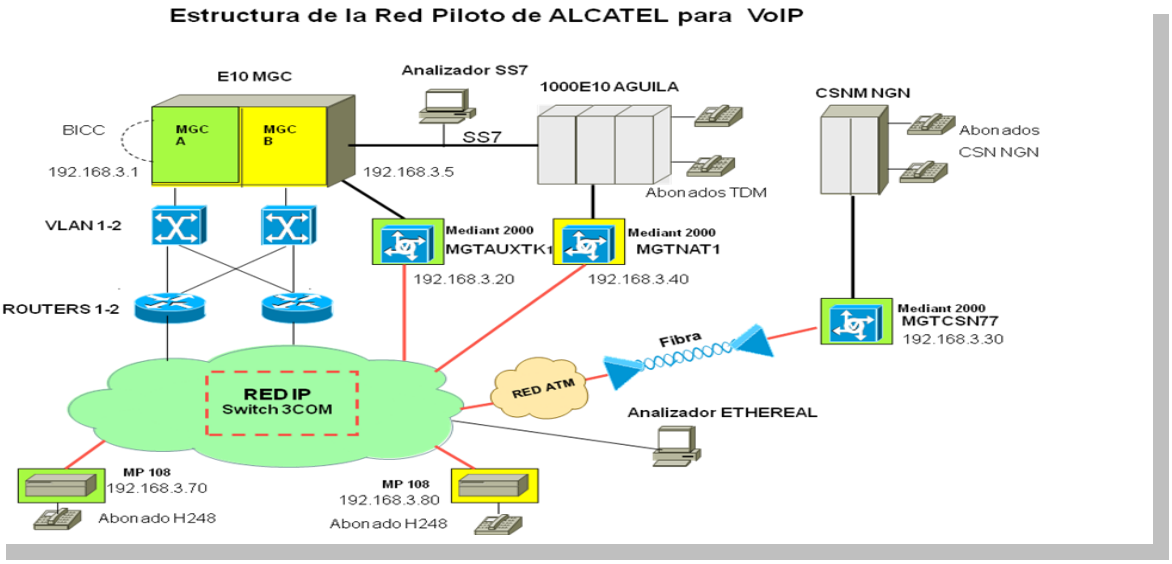



Figura 3. Estructura de la red piloto de ALCATEL para VoIP

**Arquitectura CISCO**

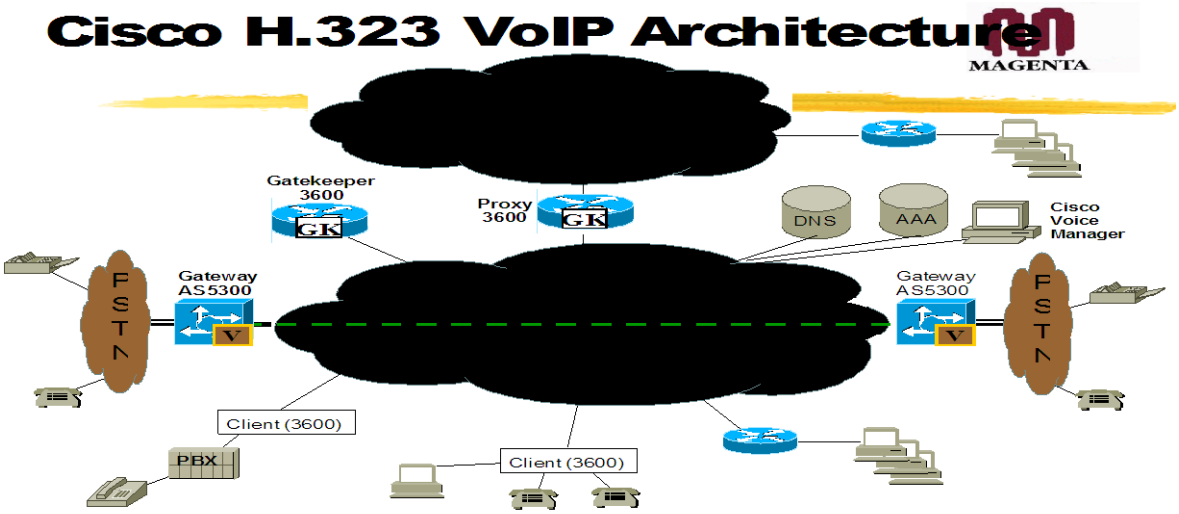



Figura 4. Arquitectura común de Cisco

**Arquitectura 3Com**


Figura 5. Arquitectura común de 3Com

## Anexo 3. Protocolos

### Modelo de Ethernet

Se muestra las capas donde interviene el protocolo Ethernet.

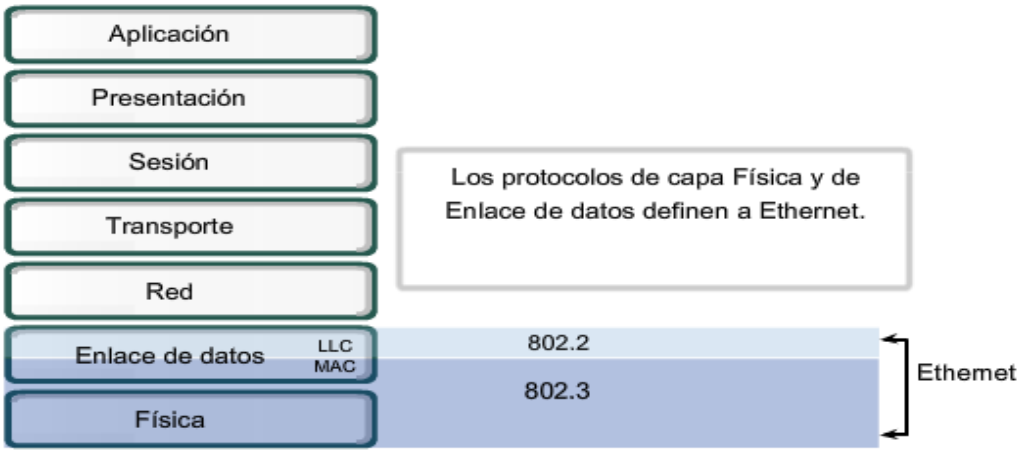



Figura1. Modelo de Ethernet.

### Limitación de la capa 1del modelo OSI

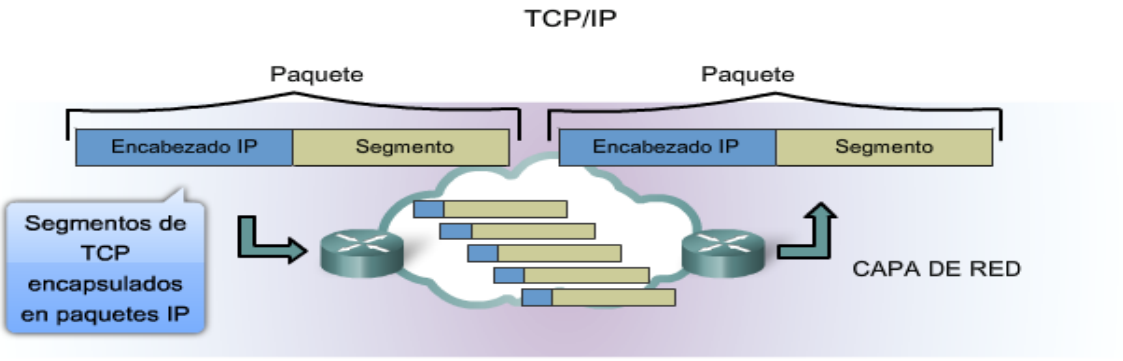
La Capa 1 de *Ethernet* tiene un papel clave en la comunicación que se produce entre los dispositivos, pero cada una de estas funciones tiene limitaciones.

Direcciones de la Capa 2 Limitaciones de la Capa 1

| Limitaciones de la Capa 1 | Funciones de la Capa 2 |
|---|---|
| No se puede comunicar con capas superiores | Se conecta con las capas superiores mediante control de enlace lógico (LLC) |
| No pueden identificar dispositivos | Utiliza esquemas de direccionamiento para identificar dispositivos |
| Sólo reconoce streams de bits | Utiliza tramas para organizar los bits en grupos |
| No puede determinar la fuente de la transmisión cuando transmiten múltiples dispositivos | Utiliza control de acceso al medio (MAC) para identificar fuentes de transmisión |

Figura2. Limitaciones del a capa 1, funciones capa 2 del modelo OSI.

## Características básicas de *IPv4*




Los paquetes IP fluyen a través de internetwork.

- Sin conexión: sin establecimiento de conexión en forma previa al envío de paquetes de datos.
- Mejor intento (no confiable): sin sobrecarga para garantizar la entrega de paquetes.
- Independiente de los medios: funciona en forma independiente de los medios que transportan los datos.

Figura11. Características de IPv4.

## Normas y protocolos de la Recomendación H.323 versión 3

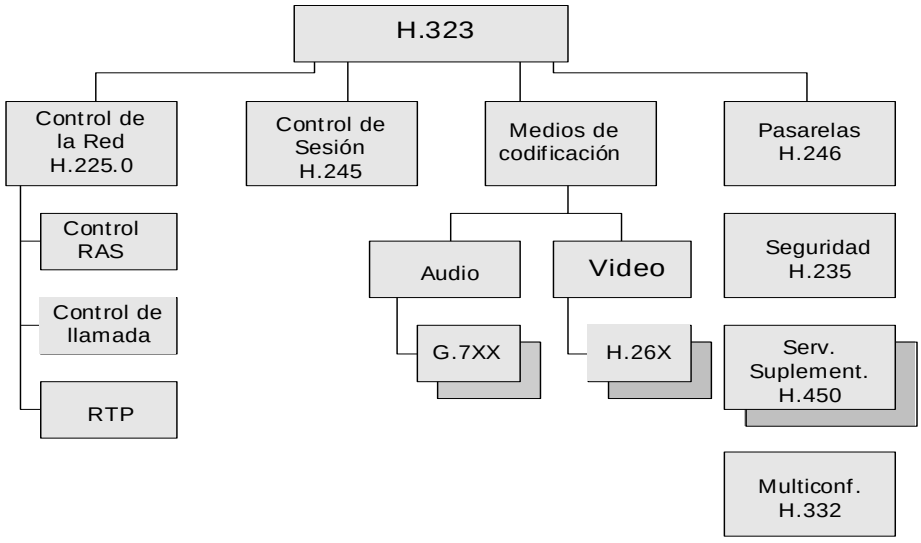



Figura 4. Normas y protocolos de la Recomendación H.323 versión 3

## Interoperabilidad de otros terminales con terminales H.323

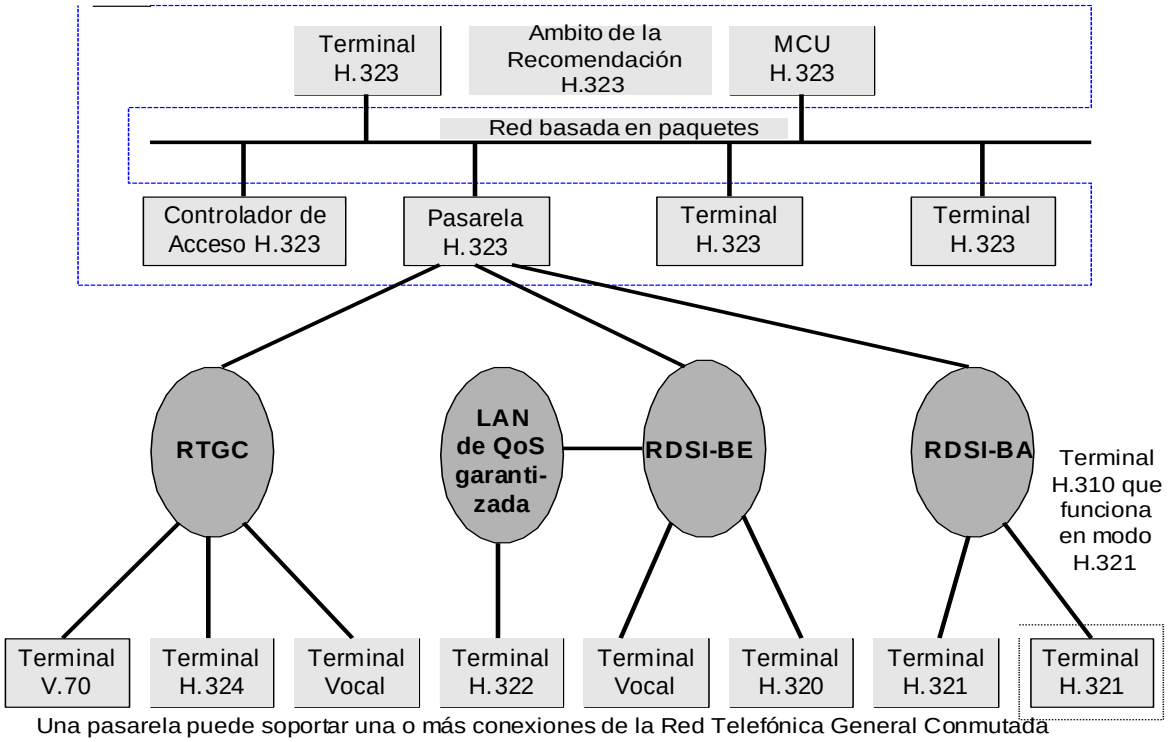



Figura 5. Interoperabilidad de otros terminales con terminales H.323

## Modelo en capas de H.323

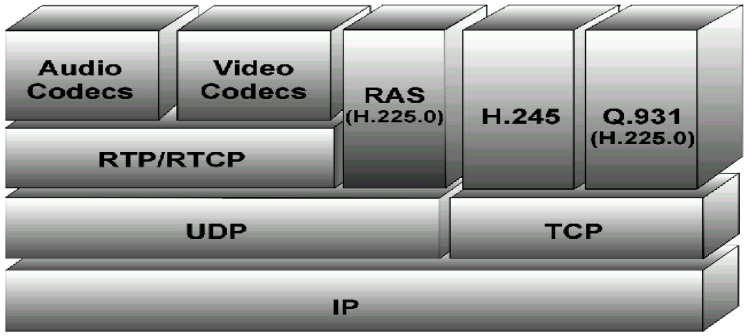



Figura 6. Modelo en capas de H.323

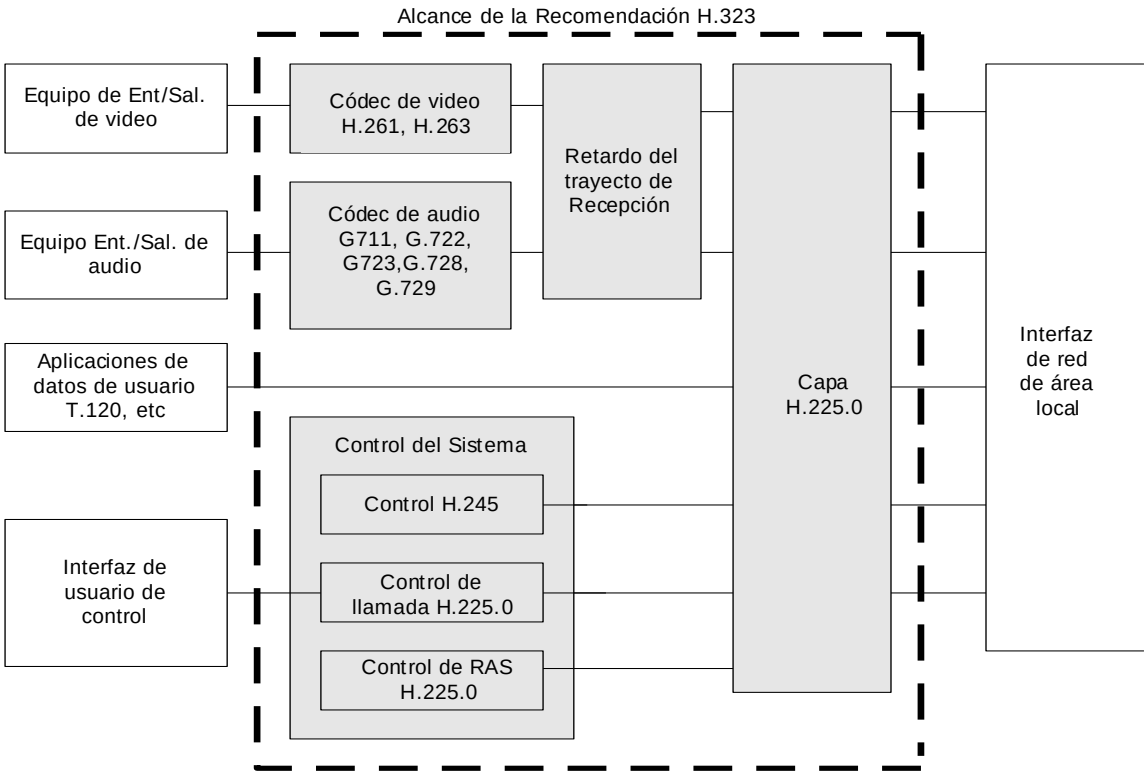



Figura7. Equipo Terminal H.323

## Arquitectura SIP

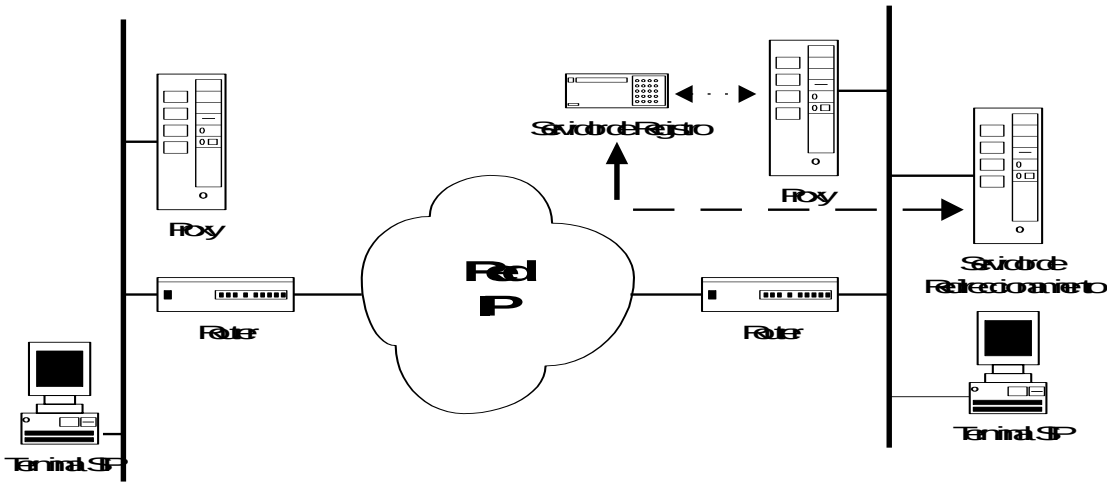



Figura 8. Arquitectura SIP

## Llamada SIP exitosa a través de un Servidor Proxy

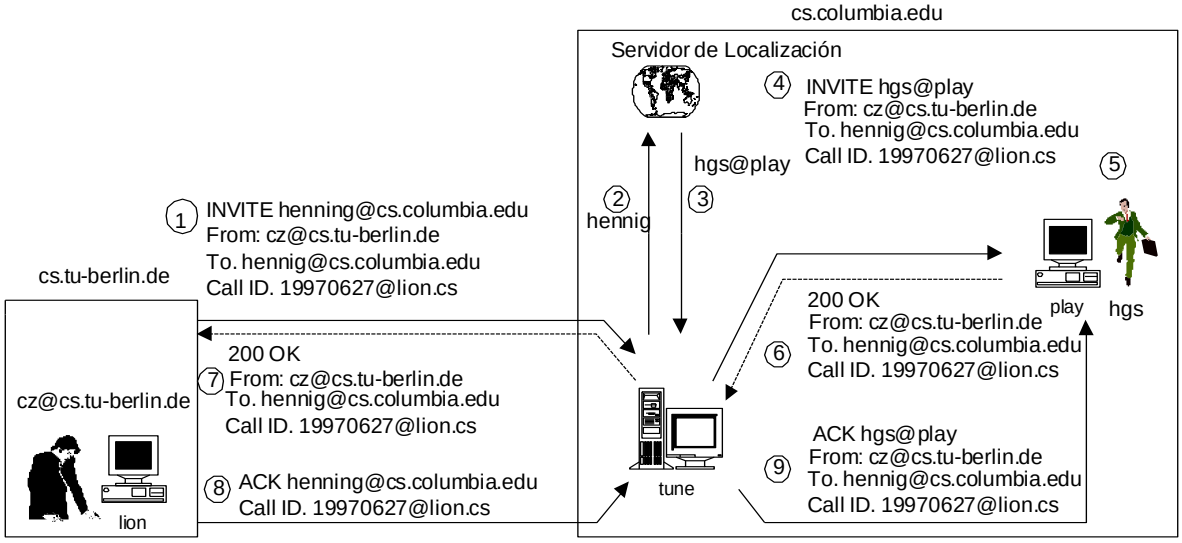



Figura 9.

Llamada SIP exitosa a través de un Servidor Proxy.

## Llamada SIP exitosa a través de un Servidor de Redireccionamiento.

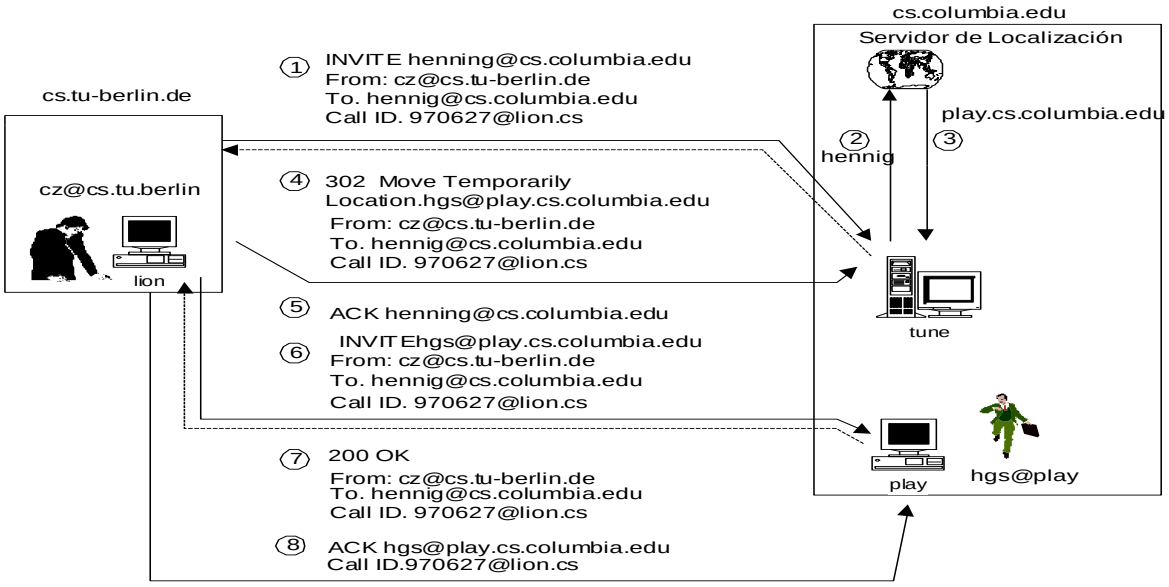



Figura10. Llamada SIP exitosa a través de un Servidor de Redireccionamiento.