INSTITUTO POLITECNICO NACIONAL

ESCUELA SUPERIOR DE INGENIERÍA MECÁNICA Y ELÉCTRICA
UNIDAD PROFESIONAL "ADOLFO LOPEZ MATEOS
SECCIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

"IMPLEMENTACIÓN DE UN SISTEMA DE VIDEO LLAMADAS SEGURAS SOBRE UNA PBX-ASTERISK"

TESIS
QUE PARA OBTENER EL GRADO DE
MAESTRO EN CINECIAS EN INGENIERÍA DE
TELECOMUNICACIONES

PRESENTA
ING. RAFAEL SORIA VARGAS

DIRECTORES DE TESIS
DR. MARCO ANTONIO ACEVEDO MOSQUEDA
DRA. MARIA ELENA ACEVEDO MOSQUEDA

. DEDICATORIAS Y AGRADECIMIENTOS

A mis padres por ser esa motivación poderosa y única,

A mi abuelo Augusto donde quiera que esté,

Y a todas aquellas personas que están lejos de casa y aun así no se detienen y siguen subiendo montañas.

La vida está hecha de pequeñas cosas que son simplemente las herramientas para defenderte, innovar, a veces improvisar para poder lograr tus sueños. Por muy individual que parezca la realización de esta tesis de maestría, no lo es. Existen muchas personas que pusieron su empeño para que esto fuera posible. Personas que nunca dudaron de mí y mantuvieron la llama de ese candil que hoy se convirtió en el fuego devorador de metas que calientan mis deseos de triunfar. Agradezco a mis padres por la educación que me dieron. A mi madre por estar siempre presente estando lejos o cerca sin importar las distancias. A mi padre por ser el génesis del espíritu de superación. Quiero agradecer además a todo el colectivo de la Sección de Estudios de Posgrados e Investigación de la Escuela Superior de Ingeniería Mecánica y Eléctrica del Instituto Politécnico Nacional. En especial al colectivo de la Maestría en Ciencias en Ingeniería de Telecomunicaciones. A mi tutor Marco Antonio Acevedo Mosqueda por ofrecerme la oportunidad de poder completar esta etapa profesional. A Marlén Cisneros por estar siempre dispuesta a ayudarme y hacerlo en cada momento que lo necesité. A Laurita por tantas molestias de mi parte. A cada uno de los profesores con los que tomé materia. A Lulú por tenerle paciencia a este extranjero despistado y bien perdido en todos los trámites de inscripción. A la Lic. Emelia Velázquez por tanta ayuda con los trámites migratorios. En general a todos aquellos compañeros que de una forma u otra aportaron al menos un ápice para que esto fuera posible.

Por otro lado quisiera agradecer al grupo Ceniteq del I3A en la Universidad de Zaragoza, España. Fue una experiencia inolvidable y motivadora el compartir con ustedes cada momento aún más lejos de casa. Agradezco la cooperación y la amistad brindada de todos: Julián Fernández Navajas, José María Saldaña, Idelkys Quintana, Luis Sequeira, Carmen Delgado, Carmen Rodríguez, Miguel Eguizabal y Sonda Bousnina. Agradecimientos especiales a Julián quien considero también mi tutor y que he decidido incluirlo en el titulado de la tesis, sin embargo no aparece por motivos de logística que me fallaron a la hora de inscribir la tesis en la Sección de Estudios de Posgrados e Investigación. No obstante quiero que quede escrito en

el cuerpo de los agradecimientos que Julián tiene todo el mérito para aparecer junto a Marco Antonio. Gracias por tu apoyo, conocimiento, comprensión, sobre todo por brindarme la oportunidad de pisar niveles tan elevados como los que se manejan en el viejo continente.

Finalmente y no menos importante quisiera dejar plasmado en este texto el agradecimiento a mi otra familia en México, a Paty y a Mariana, quienes con todo su amor, dedicación y apoyo han estado conmigo incondicionalmente en todo momento. Paty, has sido una madre para mí, tus consejos, tus buenas intenciones, tus desvelos y tus deseos de ayudarme en todo, son el reflejo de esa personalidad tan linda que posees. Marian, gracias por regalarme la ilusión, gracias por tu comprensión, tu incondicionalidad, tu lealtad y por tu amor. Eres parte de este triunfo y espero sin duda alguna que este sea el primero de muchos tantos que tendremos.

En general muchas gracias a todos.

Rafael Soria Vargas

. RESUMEN

En el mundo de la VoIP existen muchas empresas dedicadas a este tipo de servicios. Sin embargo no todas implementan la seguridad en sus productos. Este proyecto de investigación está enfocado al desarrollo de una solución de Telefonía de VoIP de modo que se agreguen características importantes como la seguridad mediante la encriptación de las tramas de multimedia que viajan por internet. Desde el punto de vista de sus componentes, se trata de una plataforma conformada por un servidor con sistema operativo en Linux donde corre un software llamado Asterisk. El servidor será capaz de brindar llamadas de voz, llamadas de video donde todos estos servicios viajarán por la red de redes de manera cifrada. Desde el punto de vista de telecomunicaciones se trata del empleo de protocolos bien definidos, tanto de señalización como de envío de datos de multimedia, como el Protocolo de Inicio de Sesión (SIP) y el Protocolo de Tiempo Real Cifrado (SRTP).

.ABSTRACT

In the world of VoIP there are many companies dedicated to these services. However, much of them, have not implemented security in their products. This research project is focused on the development of a VoIP telephony solution. Important features like the security by encrypting multimedia frames are added while them traveling over the Internet. From the viewpoint of its components, it is a server machine with Operative System Linux which it have running a software called Asterisk. The server will be able to provide voice calls, video calls, where all these services travel through the network of networks in encrypted form. From the point of view of telecommunications is employed well-defined protocols, such as Session Initiation Protocol (SIP) and Real Time Protocol Encryption (SRTP).

.Justificación

A partir de la creación de un grupo de investigación con sede en la Sección de Estudios de Posgrados e Investigación de la Escuela Superior de Ingeniería Mecánica y Eléctrica del Instituto Politécnico Nacional de México, conformado por ingenieros y licenciados del sector de las telecomunicaciones y con la finalidad de la realización de un Sistema de VoIP Seguro, se plantea una solución intermedia que sea capaz de realizar llamadas de VoIP Seguras empleando un servidor propio del grupo de investigación.

La versión anterior a este nuevo proyecto logró la realización de llamadas seguras por internet con las limitaciones de no poder realizar llamadas de video cifradas. El anterior proyecto tampoco tenía implementado ningún sistema de protección contra ataques de hackers para impedir el acceso a los servicios de VoIP y al control de la administración del servidor mediante Secure Shell (SSH).

Por tal razón surge la idea de suplir estas carencias en un nuevo trabajo de investigación mediante la implementación un sistema más completo y con mayor robustez.

. OBJETIVO GENERAL

Implementar un Sistema de Video Llamadas Seguras utilizando una PBX-Asterisk y demostrar su funcionamiento.

. OBJETIVOS ESPECÍFICOS

- Instalar en el servidor la versión de Centos 6.
- Instalar en el servidor el software Asterisk la versión versión 11.
- Instalar en el servidor el protocolo SRTP.
- Instalar en el servidor los códecs de audio: Speex, GSM, Opus, G.711 y G.722. y el códec de video Vp8.
- Utilizar protocolos como SIP y SRTP para el cliente.
- Utilizar en el cliente los códecs de audio como: Speex, GSM, Opus, G.711 y G.722. Y el códec de video Vp8.
- Estudiar y seleccionar la versión de Asterisk adecuada y con largo soporte por parte de los fabricantes.
- Estudiar y seleccionar el software del cliente para la utilizar los servicios a desarrollar.
- Instalar los parches de adecuados de SRTP y de códecs a utilizar.
- Programar las extensiones y el sistema de marcación a utilizar.
- Programar los archivos de configuración para la implementación de la video llamada.
- Programar archivos del sistema anti ataques.
- Realizar pruebas para validar los servicios programados.
- Medir ancho de banda consumidos por diferentes códec.

. Índice

. Lista de Figuras

.Lista de Tablas

. INTRODUCCIÓN

Problema a resolver

Es necesario el desarrollo de un Sistema de Video Llamadas Seguras utilizando una PBX-Asterisk que permita el disfrute global de servicios de video llamadas y video conferencias seguras, debido a que la anterior versión de este proyecto tenía la limitante de poder realizar sólo servicios de voz segura. Esto implica que no se estaban explotando algunas potencialidades atractivas de la PBX-Asterisk y que este nuevo proyecto soluciona.

Estado del Arte

Bajo la revisión de varios artículos publicados por la IEEE relacionados con el tema de este trabajo de investigación se encontraron algunos antecedentes que destacan la popularidad por la implementación de Sistemas de Voz sobre IP. Sin embargo, teniendo en cuenta que la revisión fue hecha sobre artículos entre los años 2012 y 2014 y aceptando que solamente se revisaron algunos artículos relacionados, no se encontró ninguno que hablara sobre la seguridad en los servicios mencionados. En la Tabla 1 figuran los trabajos relacionados con el tema incluyendo una breve descripción de los mismos.

Tabla 1 Estado del Arte

Año	Universidad, País	Autor	Título	Descripción
2014	School of Electrical Engineering and Computer Science National University of Sciences and Technology Islamabad, Pakistan	Ubaid Ur Rehman	**Security Analysis of VoIP Architecture for Identifying SIP Vulnerabilities [1]**	Vulnerabilidades del Protocolo de Inicio de Sesión
2014	University of Saskatchewan Saskatoon, Canada	**Richard K. Lomotey and Ralph Deters**	**Intrusion Prevention in Asterisk-based Telephony System [2]**	Prevención de ataques a Servidores Asterisk con salida a la red pública PSTN
2014	Culture and Research (ACECR), Tehran, Iran	Mohammad Hasanzadeh, Hossein Hamidi, Habibolah Asghari	**Plaintext transmission over Session Initiation Protocol [3]**	Transmisión de mensajería instantánea empleando el protocolo SIP

2013	University of Plymouth, UK	Lingfen Sun	**Guide to Voice and Video over IP [4]**	Implementación de servicios de voz y video
2013	Centro universitario Aligarh Uttar Pradesh, India	Priyanka Gupta	**GSM and PSTN Gateway for Asterisk EPBX [5]**	Convergencia de la red de VoIP con la PSTN
2012	Aligarh Muslim University, India	Priyanka Gupta	**SIP Server Security with TLS: Relative Performance Evaluation [6]**	Protección de datos de señalización con Transport Layer Security (TLS)

Los artículos abordan varios temas en general. Sobre seguridad se encontró uno que figuraba sobre las vulnerabilidades del Protocolo de Inicio de Sesión titulado: **Security Analysis of VoIP Architecture for Identifying SIP Vulnerabilities** escrito por Ubaid Ur Rehman de la School of Electrical Engineering and Computer Science National University of Sciences and Technology Islamabad de Pakistan en 2014. También sobre seguridad pero este destinado a la detección de ataques para vulnerar el sistema de telefonía de Asterisk que accede a la red pública PSTN se revisó uno titulado: **Intrusion Prevention in Asterisk-based Telephony System,** este es un artículo publicado en IEEE International Conference on Mobile Services en el año 2014 y escrito por Richard K. Lomotey y Ralph Deters provenientes de Department of Computer Science University of Saskatchewan Saskatoon, Canada**.** Sobre los servicios de mensajería instantánea se encontró un artículo que se

enfocaba en la su implementación, sin embargo no se hablaba de como cifrar los datos de textos al ser enviados de un usuario a otro. Otros se enfocaban sobre cómo implementar los servicios de voz y video uno de estos escritos y que constituye un libro es el titulado **Guide to Voice and Video over IP** escrito por Lingfen Sun de la School of Computing and Mathematics de la University of Plymouth en UK en 2013. Incluso existen otros autores que hablan sobre la implementación de los servicios de voz sobre IP y cómo hacer converger la red de datos de voz con la red de telefonía tradicional PSTN, ejemplo es el artículo titulado: **GSM and PSTN Gateway for Asterisk EPBX** escrito por Priyanka Gupta del Departamento de Ingeniería Computación del Centro universitario Aligarh Uttar Pradesh en India en 2013. Lo más cercano a lo investigado en este proyecto fue encontrado en el artículo titulado: **SIP Server Security with TLS: Relative Performance Evaluation** por Priyanka Gupta del Department of Computer Engineering de la Aligarh Muslim University en India quien habló de un sistema de voz sobre IP donde se protegen los datos de la señalización de llamada utilizando el protocolo Transport Layer Security (TLS) 2012.

Objeto de Estudio

El objeto de este trabajo es el estudio de los diferentes productos y características de Asterisk que pueden ser implementados en un servidor con Linux, así como el estudio de los protocolos de señalización y transporte de Voz sobre IP.

Campo de Acción

Dentro de los diferentes servicios implementables en la PBX-Asterisk el proyecto se centra en el estudio específico de las llamadas de video y las llamadas de video conferencias cifradas.

Capítulo 1.

“Introducción a la VoIP, Asterisk y a sus Protocolos”

1.0. Introducción

La Telefonía IP es una expresión del auge alcanzado por el Protocolo Internet (IP), como eje del desarrollo de las Telecomunicaciones, justificado por la ubicuidad de IP, que con el desarrollo de la informática abarca desde escenarios domésticos hasta ambientes de pequeñas, medianas y grandes empresas. Es un servicio de voz que a diferencia de la telefonía convencional de la Red Telefónica Pública Conmutada (PSTN), realiza el transporte de media sobre redes de conmutación de paquetes [7]. El constante desarrollo y crecimiento de las redes de voz y de datos, hacen inminentes la búsqueda de la convergencia de estas dos grandes redes en un mismo medio de trasporte. Diversos factores como Internet, redes digitales, protocolos de comunicación, etc., hacen posible la integración de los servicios provenientes de estas redes haciendo uso de las nuevas tecnologías. En este capítulo se evidenciarán las ventajas de la utilización de VoIP en el entorno empresarial, además de dar una definición de los términos de VoIP y Telefonía IP los cuales se utilizarán en este trabajo, indistintamente. Se hará énfasis en los dispositivos terminales de la red IP y en las capacidades de las IP-PBXs.

1.1. La telefonía IP y sus ventajas para el entorno empresarial.

La telefonía IP o VoIP se define como la capacidad de hacer llamadas telefónicas sobre redes de datos basadas en IP, con un estándar de Calidad de Servicio (QoS) y una relación de costo / beneficio superior.[8] El concepto original es relativamente simple: se trata de transformar la voz en "paquetes de información" manejables por una red IP con la utilización de otros protocolos de comunicación, como SIP (Session Initiation Protocol), H.323, RTP (Real Time Transport Protocol), RTCP (Real Time Control Protocol), que permiten la trasmisión de flujos de datos multimedia y el envío de señalización de control. VoIP abre un espacio muy importante dentro del universo que es Internet. Es la posibilidad de

estar comunicados a costos más bajos dentro de las empresas y fuera de ellas, es la puerta de entrada de nuevos servicios y es la forma de combinar una página de presentación Web con la atención en vivo y en directo desde un centro de llamadas, entre muchas otras prestaciones [8].

Las principales ventajas de la telefonía IP, orientadas a un entorno empresarial, comienzan al permitir que en una empresa exista una única red basada en la tecnología IP que integra la voz y los datos, a diferencia de la solución tradicional en la que subsisten dos redes separadas. La simplificación de la infraestructura de comunicaciones en una empresa, la integración de las diferentes redes telefónicas en un sistema unificado de telefonía, con un sistema de gestión centralizada, llamadas internas gratuitas, plan de numeración integrado y optimización de las líneas de comunicación, así como el acceso a funcionalidades avanzadas como buzones de correos de voz, contestadoras automáticas, distribuciones automáticas de llamadas, son otras de las ventajas que solidifican la idea de su implementación. Los ahorros en costo que pueden traer la telefonía IP para una empresa son considerables si se analiza que es posible aprovechar el cableado proveniente de una red de datos como Ethernet, de esta manera se pudieran reducir los gastos referidos a instalación y montaje.

1.2. Elementos de la Telefonía VoIP

Los elementos implicados en la telefonía IP son aquellos por medio de los cuales es posible la implementación de la tecnología. Estos elementos se refieren a teléfonos IP, teléfonos USB, los adaptadores analógicos, pequeñas centrales telefónicas así como variedades de Softphone. Además cabe destacar que todos estos elementos estarán en acción sobre una infraestructura IP o una red IP, la cual sería el medio de trasporte para la señalización de las llamadas, para la trasmisión de voz y de video. Esta red debe tener en cuenta en sus condiciones

de diseño cierta calidad de servicio para que tanto la voz como el video puedan ser trasmitidos con una calidad adecuada.

1.2.1. Teléfonos IP

Un teléfono de VoIP o teléfono IP es un equipo especialmente diseñado para conectarse a una red de VoIP. Los teléfonos IP pueden implementar uno o varios protocolos de señalización de voz sobre IP. Básicamente son teléfonos normales de apariencia tradicional con la diferencia de que poseen conexión mediante el conector de red RJ45 [9]. La figura 1 muestra ejemplos de teléfonos IP.

CISCO CP-9971-W-CAM-K9

CISCO 9971-C-CAM

CISCO CP-8945-K9

Figura 1 Ejemplo de teléfonos IP con sus respectivos modelos [10]

1.2.2. Teléfonos USB.

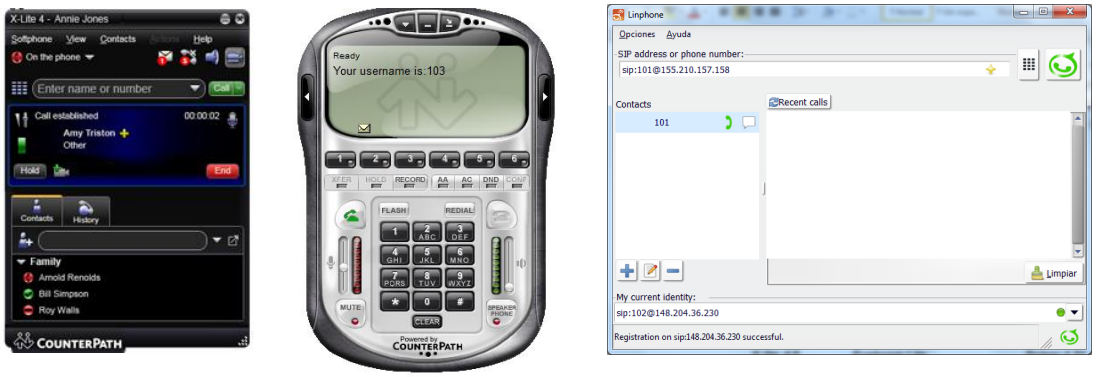
Los teléfonos USB son bien parecidos a los teléfonos IP, con la diferencia que estos intercambian señalización de control y flujos de audio por el puerto USB. Por lo general estos teléfonos son más baratos que los teléfonos IP dependiendo del fabricante. La Figura 2 muestra algunos de los teléfonos USB que hay en el mercado actualmente

SPA514G SPA302D-G7 Cisco 6901

Figura 2 Ejemplo de teléfonos IP con sus respectivos modelos [11]

1.2.3. Softphones y Aplicaciones móviles

Una alternativa al uso de equipos dedicados (físicos) de VoIP es el uso de programas para emularlos. Estos programas se conocen como softphone y funcionan en cualquier ordenador personal. El único requerimiento es tener una tarjeta de sonido en funcionamiento. Estos tienen el aspecto de un teléfono real, pero no dejan de ser programas que corren sobre un sistema operativo [9]. Existen muchísimos ejemplos en internet de softphones como son el ejemplo de Zoiper, Linphone, Xlite, EyeBeam entre otros. La figura 3 muestra ejemplos de algunos de los softphones.



X-lite 4.0 **Eyebeam Lite** **Linphone 3.7.0**

Figura 3 Ejemplo de Softphones y Aplicaciones Móviles.

1.2.4. PBX Private (Automatic) Branch Exchange

El uso más común de una PBX es compartir una o varias líneas de enlace de una central telefónica local con un grupo de usuarios. Una PBX se emplaza entre los enlaces telefónicos de una central local y los teléfonos (terminales de voz). La PBX tiene la propiedad de ser capaz de redirigir las llamadas entrantes a uno o varios teléfonos. De la misma forma que un enrutador en Internet es responsable de dirigir los paquetes de un origen a su destino, una PBX es responsable de dirigir “llamadas telefónicas”. [9]

Las IP-PBXs son pequeñas centrales telefónicas para VoIP, por medio de ellas se establece, se mantiene y se liberan las sesiones o llamadas que hacen los subscriptores entre sí, mediante el uso de protocolos de señalización como SIP, IAX2 y H.323. Una IP-PBX no sólo permite compartir un conjunto de líneas con un grupo de usuarios sino que también ofrece la posibilidad de crear servicios de valor añadido como transferencia de llamadas, llamadas en conferencia, correo de voz o servicios basados en respuesta de voz interactiva (IVR).[9]

Existen varios tipos de IP-PBX, algunas libres de costos. Entre las IP-PBX libres se encuentra la Asterisk quien trabaja sobre Linux y utiliza entre sus protocolos de señalización IAX2. Se pueden enumerar varios fabricantes de IP-PBX como son el caso de NCH Software, Ackerman, Alcatel, Altigen, Artisoft, Hitachi, Nortel, entre otros más. La mayoría de las IP-PBX de estos fabricantes no son libres. [12]

1.2.4.1. Servidor Asterisk

Asterisk como se había mencionado es una aplicación para establecer, mantener, controlar y gestionar comunicaciones de cualquier tipo, ya sean analógicas, digitales o VoIP. Es compatible con la mayoría de los protocolos existentes de VoIP, por lo que es una plataforma muy completa para este tipo de comunicaciones. Según su sitio oficial, Asterisk es un framework libre y de código abierto para la creación de aplicaciones de comunicación y con el patrocinio de Digium. Una de las ventajas más interesantes es su posibilidad como sistema híbrido, ya que permite gestionar comunicaciones telefónicas tradicionales (analógicas, digitales, móviles.) como comunicaciones IP mediante el uso de los protocolos estándar de VoIP. Tiene la potencialidad de convertir una computadora normal en un Servidor de Comunicaciones o Sistema de IP-PBX. Otras de las ventajas más significativas son las de poder hacer de Gateways de VoIP, Servidor de Video Conferencia y de Correo de Voz, Servidor de Centros de Llamadas entre otros. Posee una alta capacidad para ser programada, permitiendo realizar labores que hasta el día de hoy lo llevaban realizando sistemas extremadamente costosos y complicados. Gracias a Asterisk, esta misma labor se realiza de una forma más económica lo que fomenta el uso de sistemas libres como Linux y estándares abiertos como SIP, H323 o IAX. [13]

En la **Tabla 2** se pueden ver algunas de las versiones con las subversiones estables de Asterisk.

Tabla 2 Versiones de Asterisk [13]

Versiones	**Subversiones**
Versión 12	Asterisk Versión 12.2.0 Estable
Versión 11 LTS	Asterisk Versión 11.9.0 Estable
Versión 1.8 LTS	Asterisk Versión 1.8.6.0 Estable
Versión 1.6	Asterisk Versión 1.6.0.28 Estable (Descontinuada) Asterisk Versión 1.6.1.25 Estable (Descontinuada) Asterisk Versión 1.6.2.20 Estable
Versión 1.4 LTS	Asterisk Versión 1.4.42 Estable (Descontinuada) Asterisk Addons Versión 1.4.13 Estable (Descontinuada)
Versión 1.2 y 1.0	Estas versiones se consideran paralizadas y no se continuarán manteniendo.

1.2.4.2. Ventajas del Sistema Asterisk

Funcionalidad

Asterisk dispone de todas las funcionalidades de las grandes centralitas propietarias (Cisco, Avaya, Alcatel, Siemens, etc). Desde las más básicas como desvíos,

capturas, transferencias y multi-conferencias, hasta las más avanzadas como Buzones de voz, IVR, CTI, ACD. [20]

Escalabilidad

El sistema puede dar servicio desde 10 usuarios en una sede de una pequeña empresa, hasta 10.000 de una multinacional repartidos en múltiples sedes. [14]

Competitividad en coste

No solo por ser un sistema de código abierto (Open Source) sino gracias a su arquitectura hardware: utiliza plataforma servidor estándar (de propósito no específico) y tarjetas PCI para los interfaces de telefonía, que por la competencia del mercado se han ido abaratando progresivamente. [14]

Interoperabilidad y Flexibilidad

Asterisk ha incorporado la mayoría de estándares de telefonía del mercado, tanto los tradicionales (TDM) con el soporte de puertos de interfaz analógicos (FXS y FXO) y RDSI (básicos y primarios), como los de telefonía IP (SIP, H.323, MGCP, SCCP/Skinny). Eso le permite conectarse a las redes públicas de telefonía tradicional e integrarse fácilmente con centralitas tradicionales (no IP) y otras centralitas IP. [14]

Funciones Básicas

Asterisk puede funcionar como cualquier centralita tradicional, e incorpora todas sus funcionalidades. Enumeramos las más importantes:

- Conexión con líneas de telefonía tradicional, mediante interfaces tipo analógico (FXO) para líneas de teléfono fijo o bien móvil y RDSI (BRI o PRI).
- Soporte de extensiones analógicas, bien para terminales telefónicos analógicos, terminales DECT o bien equipos de fax.
- Soporte de líneas (trunks) IP: SIP, H323 o IAX.
- Soporte de extensiones IP: SIP, SCCP, MGCP, H323 o IAX
- Música en Espera basada en archivos MP3 y similar.
- Funciones básicas de usuario:
    - Transferencias (directa o consultiva)
- Desvíos
- Capturas (de grupo o de extensión)
- Conferencia múltiple
- Aparcamiento de llamadas (Call parking)
- Llamada directa a extensión
- Retrollamada – Callback (llamada automática cuando disponible)1.
- Paging – Megafonía a través del altavoz del teléfono2
- DND [14]

Funciones avanzadas

El sistema incorpora asimismo muchísimas funcionalidades avanzadas que tendrían un elevado coste en sistemas tradicionales propietarios. Enumeramos sólo los más importantes:

- Buzón de Voz: sistema de contestador automático personalizado por usuario. Se integra con el sistema de directorio (LDAP) y con el email.

- Sistema de Audioconferencias: Sistema que permite la conexión remota de diferentes usuarios que quieren mantener una reunión virtual y suministra la correcta gestión y control de los usuarios que se incorporan a ella.

- IVR Operadora Automática: Sistema automatizado de respuesta que permite redirigir las llamadas entrantes en función de las opciones seleccionadas por el llamante.

    - Informes detallados de llamadas (CDR): Detalle de llamadas realizadas/recibidas por extensión, para imputación de costes departamentales, por cliente o incluso para facturación.

- ACD: Sistema Automático de Distribución de Llamadas entrantes. Pensado para Centros de Llamadas para atención comercial o soporte técnico.

- CTI: Integración con sistemas de gestión comercial o de atención al cliente (CRM).

- IPCC (IP Contact Center): Integración con sistemas avanzados de gestión de centros de llamadas, vía soluciones abiertas o propietarias.[14]

1.3. Esquema general de una red VoIP.

La Telefonía IP puede ser implementada en diferentes escenarios. Bien en una empresa de cualquier objeto o en una escuela, universidad o institución que tengan la necesidad de comunicarse unos con otros. De manera general se pudiera configurar una PBX como por ejemplo Asterisk, en una red de una empresa, donde en esta PBX estarían almacenadas cada una de las extensiones de los participantes. A esta pequeña central se podrían subscribir tanto un Softphone, un

teléfono IP o un teléfono USB como los mostrados anteriormente. Como se ve en la figura no importa que equipo terminal se tenga, ni donde se encuentre siempre y cuando se tenga acceso a una conexión a internet. Esta constituirá el medio perfecto y esencial para que la comunicación sea efectiva. Describiendo la figura 4 se puede apreciar el conjunto de clientes, ya sean teléfonos físicos o aplicaciones sobre un sistema operativo, un servidor que constituye la PBX, en este caso Asterisk y una conexión a internet. Con estos elementos implicados se puede diseñar un Sistema de VoIP que sea funcional en las distintas redes que en la actualidad se conocen, como es: WIFI, Fast-Ethernet y Redes de Datos como 3G, 4G o LTE. En el servidor se pueden implementar diversidad de servicios de VoIP en dependencia de la necesidad o del problema a resolver. Servicios como llamadas de voz, llamadas de video, envío de texto en tiempo real, son algunos de los principales. En posteriores capítulos se abordará más detalladamente el esquema descrito.

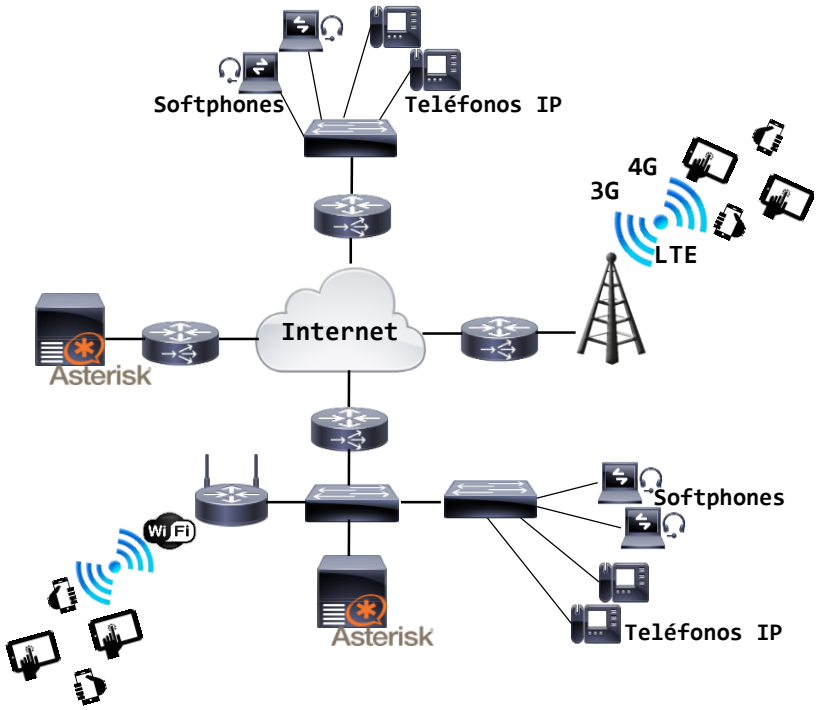


Figura 4 Red VoIP

1.4. Protocolos

Son muchos los protocolos que intervienen en la comunicación de VoIP. Por lo general se habla de protocolos de señalización como SIP, H.323 e IAX2 y de protocolos de transporte de tiempo real como RTP y SRTP en caso de comunicación cifrada. Realmente son muchos más los implicados sin embargo continuación solo se pretende analizar algunos protocolos de los más importantes.

1.4.1. Protocolo de inicio de sesión (SIP). Características

SIP es un protocolo diseñado por la IETF. Su desarrollo está orientado a la integración con aplicaciones y servicios de Internet. SIP es un protocolo de señalización, de la capa de aplicación, para crear, modificar y terminar sesiones multimedias o llamadas, con uno o más participantes, sobre una red de conmutación de paquetes.[15] Entre algunas de sus funciones esenciales se encuentran la de localización de usuario, determinación de las habilidades del usuario, determinación de la disponibilidad del usuario, establecimiento de la llamada y manipulación de la llamada.[16] SIP como protocolo está basado en texto y utiliza una codificación Utf-8, tiene una sintaxis similar a la de HTTP.[17] La forma de identificar a una entidad SIP es similar a la empleada para definir una cuenta de correo electrónico. A esta forma se le denomina URI. El URI de SIP es de la forma *sip:usuario@doMinio*, por ejemplo: *sip:103@voip.ipn.cu*.

Tiene definidos métodos y códigos que están contenidos en su RFC 3261. Los métodos corresponden a las acciones que desean realizar, ya sea de invitación, registro, cancelación o terminación de una sesión. El anexo A- 1 agrupa algunos de los métodos que tiene SIP. Los códigos son números enteros de tres dígitos que se genera como el resultado de una petición. El anexo A- 2 agrupa algunos de los códigos que utiliza el protocolo SIP.

1.4.1.1. SIP y protocolos que lo acompañan

SIP en sus inicios no se creó para trasmitir voz en tiempo real por lo que para desempeñar sus funciones se ayuda de otros protocolos como son: SDP y RTP con RTCP. RTP es usado para la indicación de la secuencia, el sincronismo entre los medios, la identificación de la carga, de la trama y de la fuente. Permite difusión (multicast), cifrado, así como realimentación de calidad de servicio (QoS) mediante el empleo de RTCP. Este último se utiliza principalmente para detectar situaciones de congestión de la red, pérdidas, retardos, etc. y tomar, en su caso, acciones correctoras; proporcionando información adicional a los participantes para adaptar las fuentes al estado de la red. SDP se emplea para comunicar las capacidades y propiedades deseadas entre las partes envueltas en la comunicación. Es un protocolo muy versátil, que puede ser empleado con diferentes estándares de señalización. La descripción de las sesiones es representada en texto plano en forma de una lista de variables con sus valores y parámetros [7]

La Figura 5 pone en evidencia la pila de protocolos que usa SIP para realizar sus funciones, así como la distinción entre protocolos de señalización y de trasporte de audio y video en tiempo real.

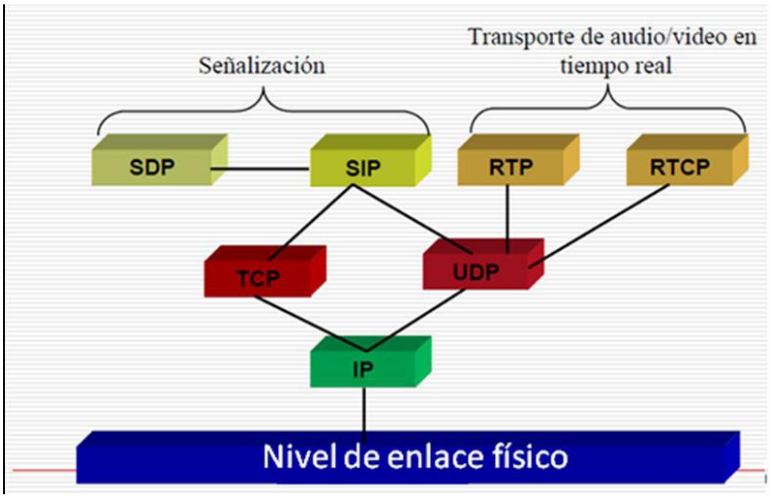


Figura 5 Pila de protocolos de SIP

1.4.1.2. Elementos que participan en una sesión.

Los elementos que participan en una sesión SIP se pueden agrupar en Agentes de Usuarios (UA) y Servidores de Red. La Figura 6 pone de manifiesto la relación entre cada una de las entidades SIP.

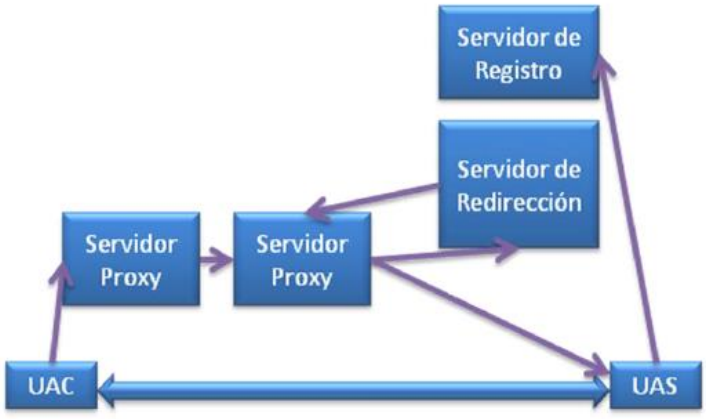


Figura 6 Entidades SIP.

Los UA son aplicaciones que se encuentran en terminales SIP, teléfonos, PC. Un UA está conformado por el agente de usuario servidor (UAS) y el agente de usuario cliente (UAC). Los UAC originan las solicitudes SIP (asociados al extremo que origina la llamada) y los UAS responden a estas solicitudes, es decir, originan respuestas SIP (asociados al extremo que recibe la llamada). Los UA deben implementar el transporte tanto sobre TCP como sobre UDP [18]. La Figura 7 pone de manifiesto el funcionamiento de un agente de usuario.

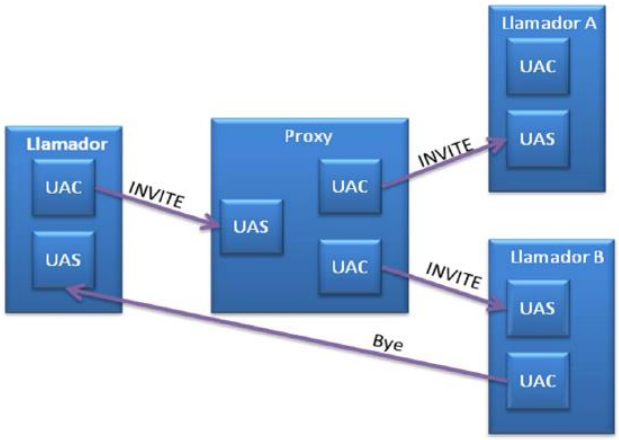


Figura 7 Funcionamiento de un Agente de Usuario.

Los UAC y UAS pueden, por sí solos y sin los servidores de red, ser capaces de soportar una comunicación básica (modelo de llamada básico, directamente entre terminales). No obstante, la potencialidad de SIP se aprovecha con el empleo de los servidores de red. Los servidores de red se clasifican, desde un punto de vista lógico, de la manera siguiente: Servidores de redirección, Servidores proxy y Servidores de registro. [7]

1.4.1.3. Mensajes SIP

SIP utiliza una serie de mensajes para señalizar las sesiones. Los mensajes SIP se pueden dividir en mensajes de solicitud o peticiones y mensajes de respuestas

[17]. De modo general un mensaje SIP se conforma de una línea inicial, el encabezado del mensaje, una línea en blanco (opcional) y el cuerpo del mensaje.

Línea de inicio.

La línea de inicio en caso de una petición SIP, contiene la versión del protocolo SIP, el método y direcciones involucradas en la sesión. Se usan para iniciar una acción o para información. En caso de que sea una respuesta la línea de inicio contiene la versión del protocolo, el código y un campo llamado Reason Phrase. Se usan para confirmar que una petición fue recibida y procesada y contiene el estado del procesamiento [19].

Caso de Petición SIP

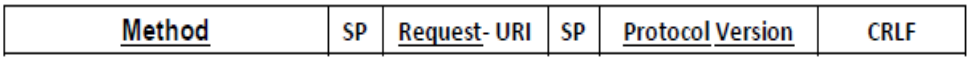


Figura 8 Línea inicial. Petición SIP

- El campo Método indica el tipo de solicitud, corresponde a la acción que desea realizar. Consultar anexo A- 1
- El campo Request-URI corresponde a una URI que indica el usuario o servicio al cual va dirigida la petición.
- El campo Versión del Protocolo indica la versión del protocolo SIP que se está usando en el instante de la petición SIP.

Caso de Respuesta SIP:

Protocol Version	SP	Status-Code	SP	Reason-Pharse	CRLF

Figura 9 Línea inicial. Respuesta SIP

- El campo Status-Code es un entero de tres dígitos que se genera como el resultado de una petición. El primer dígito define la clase de la respuesta. Consultar anexo A- 2
- El campo Reason-Phrase representa una descripción corta y textual del Status-Code.

Cabecera

- Las Cabeceras contienen información, en forma de texto, relacionada con la llamada. En ellas se especifican el origen de la llamada, la trayectoria del mensaje, tipo y largo del cuerpo del mensaje, entre otras características. Pueden ser agrupadas de la manera siguiente:
- cabeceras generales: aplicadas tanto a los mensajes de peticiones como a los de respuesta.
- Cabeceras de entidad: definen información sobre el cuerpo del mensaje. Si el cuerpo no está presente informa sobre los recursos identificados por la petición.
- Cabeceras de solicitud: actúan como modificadores de solicitud. Permiten que el cliente pase información adicional sobre la solicitud o sobre sí mismo.
- Cabeceras de respuesta: permiten al servidor agregar información adicional sobre la respuesta cuando no hay lugar en la línea de inicio.

En el anexo A- 9 se muestran los cuatros grupos de cabeceras y los campos que las componen.

Cuerpo del mensaje

El cuerpo del mensaje o carga útil es quien lleva la información. El cuerpo es opcional, sin embargo muchas veces es utilizado para describir las sesiones multimedia. El contenido del cuerpo del mensaje sólo es de interés para los UA, no para los servidores de red, pues éstos para encaminar los mensajes SIP sólo necesitan conocer los contenidos de la *línea de solicitud* o de la *línea de estado*, según el caso, y de las cabeceras [20].

En los anexos A- 3 y A- 4 se pueden ver ejemplos de paquetes SIP y en A- 5, A- 6, A-7 se puede apreciar el uso de métodos y códigos para el registro, establecimiento, terminación y cancelación de una llamada, utilizando el protocolo SIP.

1.4.2. Protocolo de Tiempo Real (RTP)

El protocolo RTP surge debido a las insuficiencias que tenía el protocolo TCP para realizar ciertas funciones como el envío de información en tiempo real. El otro protocolo existente para entonces era el UDP quien carencia de confirmación sobre los datos del contenido y además no comprendía conceptos como el de calidad de servicio (QoS).

El objetivo del protocolo RTP es el de proporcionar el transporte de extremo a extremo para aplicaciones con requisitos de tiempo real en redes unicast o multicast [21]:

- Videoconferencia.
- Difusión de audio/video.
- Simulaciones.

RTP más que ser un protocolo es un modelo de protocolos que contiene un principio de diseño novedoso, en el que se incluyen diferentes protocolos, está fuertemente ligado a la aplicación. RTP se encarga esencialmente del transporte de datos, mientras que el encargado de tener control sobre la transferencia de dichos datos es el RTCP (Real-Time Control Protocol).

Este protocolo proporciona servicios de una red end-to-end para transmisión de datos en tiempo real, RTP es un protocolo independiente de transporte y de red aunque es a menudo utilizado sobre UDP.

Formato de Paquete RTP

Los datos de media para una sesión son transmitidos como una serie de paquetes. Una serie de paquetes de datos que se originan de una fuente particular es referida como una corriente RTP. Cada paquete de datos RTP en una corriente contiene dos partes, un encabezado estructurado y los datos reales (la carga útil del paquete) [21]. La figura 10 muestra la información.

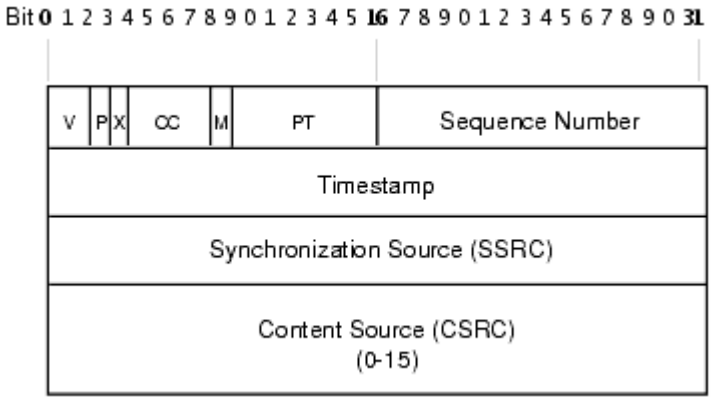


Figura 10 Cabecera de RTP [21]

El encabezado de un paquete de datos RTP contiene:

- RTP número de versión (V - versión number): 2 bits. La versión definida por la especificación actual es 2.
- Relleno (P - Padding): 1 bit. Si el bit del relleno está colocado, hay uno o más bytes al final del paquete que no es parte de la carga útil. El byte más último en el paquete indica el número de bytes de relleno. El relleno es usado por algunos algoritmos de encriptación.
- La extensión (X - Extensión): 1 bit. Si el bit de extensión está colocado, entonces el encabezado fijo es seguido por una extensión del encabezado. Este mecanismo de la extensión posibilita implementaciones para añadir información al encabezado RTP.
- Conteo CSRC (CC): 4 bits. El número de identificadores CSRC que sigue el encabezado fijo. Si la cuenta CSRC es cero, entonces la fuente de sincronización es la fuente de la carga útil.
- El marcador (M - Marker): 1 bit. Un bit de marcador definido por el perfil particular de media.
- La carga útil Type (PT): 7 bits. Un índice en una tabla de los perfiles de media que describe el formato de carga útil. Los mapeos de carga útil para audio y video están especificados en el RFC 1890.
- El número de Secuencia: 16 bits. Un único número de paquete que identifica la posición de este en la secuencia de paquetes. El número del paquete es incrementado en uno para cada paquete enviado.
- Timestamp: 32 bits. Refleja el instante de muestreo del primer byte en la carga útil. Varios paquetes consecutivos pueden tener el mismo timestamp si son lógicamente generados en el mismo tiempo - por ejemplo, si son todo parte del mismo frame de video.
- SSRC: 32 bits. Identifica la fuente de sincronización. Si la cuenta CSRC es cero, entonces la fuente de carga útil es la fuente de sincronización.

Si la cuenta CSRC es poco cero, entonces el SSRC identifica el mixer (mezclador).

- CSRC: 32 bits cada uno. Identifica las fuentes contribuyentes para la carga útil. El número de fuentes contribuyentes está indicado por el campo de la cuenta CSRC; Allí puede haber más de 16 fuentes contribuyentes. Si hay fuentes contribuyentes múltiples, entonces la carga útil son los datos mezclados de esas fuentes. [21]

Paquetes de control.

Además de los datos de media para una sesión, los datos de control (RTCP) son enviados periódicamente para todos los participantes en la sesión. Los paquetes RTCP pueden contener información acerca de la calidad de servicio (QoS) para los participantes de sesión, información acerca de la fuente de media siendo transmitidos en el puerto de datos, y las estadísticas relacionadas con a los datos que les han sido transmitidas hasta ahora. Application-specific (aplicación en específico). [21]

Todos los participantes en una sesión envían a RTCP los paquetes. Un participante que recientemente ha enviado paquetes de datos emite un reporte del remitente. El reporte del remitente (SR) contiene que el número total de paquetes y bytes enviados así como también la información que puede usarse para sincronizar media streams de sesiones diferentes. [21]

1.4.3. Protocolo de Tiempo Real Cifrado (SRTP)

El protocolo SRTP está definido según la RFC 3711 como un perfil del protocolo RTP o extensión del protocolo de tiempo real antes descrito que agrega confidencialidad, autenticación de mensajes y protección contra la réplica. Es ideal

para proteger el tráfico de voz sobre IP, ya que puede ser utilizado en combinación con la compresión de cabecera y no tiene ningún efecto sobre la Calidad del servicio IP [22]. Su función es interceptar paquetes RTP y reenviarlos en el lado del emisor como paquetes SRTP o paquetes RTP cifrados. Asimismo en el lado del receptor su función es recibir paquetes cifrados o paquetes SRTP y convertirlos en los paquetes RTP equivalentes para colocarlos en la pila de protocolos de TCP/IP.

Con el SRTP se crea un flujo de clave única para cada paquete RTP, por lo tanto es casi imposible recuperar el flujo original. También proporciona protección a la réplica que es sin duda importante para datos multimedia. De lo contrario, le sería posible a un atacante llevar a cabo manipulaciones de los datos. En una aplicación de voz, la frase "sí" podría ser sustituido por "no" si la protección de réplica no estuviese presente [22].

El algoritmo de cifrado por defecto es Advanced Encryption Standard (AES) con una clave de 128 bits y está definido en RFC 3711. La figura 11 muestra el formato de los paquetes SRTP.

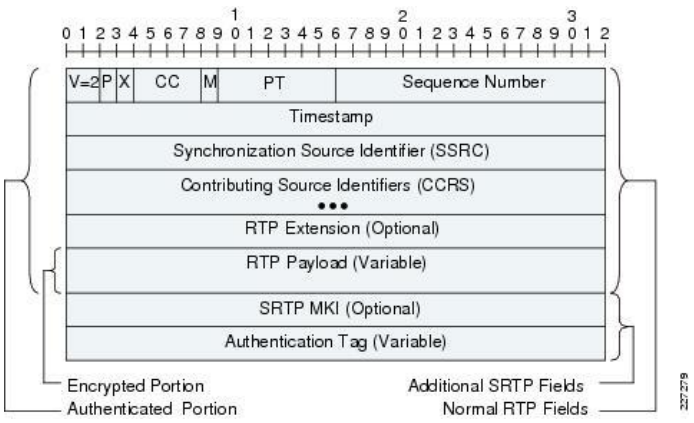


Figura 11 Formato de paquetes SRTP [22]

Como se puede apreciar en la figura 11 el formato de los paquetes SRTP son una extensión del formato de los paquetes RTP con la diferencia de los campos RTP Payload, SRTO MKI y Authentication Tag.

- RTP Payload: Constituyen los datos RTP que han sido encriptados.
- SRTP MKI (Master Key Identifier): Es un campo opcional de longitud configurable, que se utiliza para identificar la llave maestra de la sesión de llave o llaves que fueron utilizadas para autenticar y/o encriptar un paquete en un contexto criptográfico.
- Authentication Tag: Es un campo recomendable de longitud configurable que se utiliza para realizar la autenticación de los mensajes de datos para la cabecera RTP y la carga útil del paquete en particular.[23]

Capítulo 2.

“Instalación de los componentes necesarios para la realización de video llamada segura. Configuración de Sistema de Seguridad anti ataques”

2.0. Introducción

La video-llamada es uno de los servicios de mayor aceptación a nivel internacional. Este tipo de servicios permiten que los usuarios puedan realizar llamadas donde los extremos son capaces de verse, conversar e interactuar, simulando la realidad de la intercomunicación humana a un nivel muy alto. Además, el poseer sistemas que permitan este tipo de servicios seguros puede ahorrar tiempo, dinero y dar confianza a los usuarios. La tranquilidad de tener un sistema seguro capaz de llevar la comunicación a cualquier sitio del mundo dónde la administración de los servicios son gestionados a conveniencia, es una de las principales ventajas del sistema propuesto en el trabajo de maestría. En el siguiente capítulo tratará sobre cómo se puede construir un sistema de video-llamadas a partir de la instalación de softwares que corren sobre LINUX. Además se pondrá una breve tutorial de la utilización de Linphone como aplicación cliente y complemento para los servicios diseñado. Finalizando el capítulo se evidenciarán algunas configuraciones útiles para evitar los ataques de piratas cibernéticos.

2.1. Instalación de Centos 6.x

Para poder implementar un Sistema de Llamadas, desde el más simple como la llamada de voz hasta los más complejos como los que tienen cifrados de datos, video-llamadas, multi-conferencias, etc., se debe investigar cuál de los sistemas existentes es el más conveniente para realizar la implementación de las aplicaciones y servicios. El seleccionado para el desarrollo de este proyecto de investigación fue Asterisk, del cual se estuvo hablando en el capítulo inicial. La versión de Asterisk utilizada es la 11.4 y se montó sobre una distribución del sistema operativo de Linux llamada CENTOS en la versión 6.x.

¿Por qué CENTOS?

El Sistema Operativo CENTOS que significa Sistema Operativo Empresarial Comunitario (Community Enterprise Operating System) es un sistema operativo en base a la estructura de Linux. Deriva de Red Hat Enterprise Server (servidor empresarial Red Hat) y fue desarrollado para uso gratuito por la organización Centos Project. Centos ofrece mucha más estabilidad operacional a sus usuarios que otros sistemas de Linux distribuidos libremente. Comparado con otros sistemas operativos basados en Linux, Centos sólo ejecuta las versiones más básicas y estables de programas, reduciendo el riesgo de bloqueos del sistema. En el lado negativo, tiene como resultado un menor grado de funcionalidad comparado con versiones de software avanzados compatibles con otros sistemas de Linux; pero para las funcionalidades requeridas es más que suficiente. [24]

En cuanto a la velocidad Centos puede operar mucho más rápido que los sistemas operativos basados en Linux porque sólo ejecuta las versiones básicas de software. De esa manera, el procesador que ejecuta el sistema Centos no se sobre carga intentando ejecutar muchas aplicaciones diferentes. En cuanto a la confiabilidad, Centos puede ejecutar una computadora mucho tiempo sin requerir ningunas actualizaciones del sistema adicionales. Las actualizaciones de hardware para Centos son desarrolladas para ser concurrentes con las actualizaciones del sistema Red Hat Enterprise Linux en el que se basa. El ciclo de soporte de actualización para Centos es de aproximadamente cinco años; otros sistemas basados en Linux tienen ciclos de soporte más cortos, de tres años hasta aproximadamente 18 meses. En la figura 12 puede verse como el Centos 6 que se ha escogido tiene un final de vida hasta el 30 de nov de 2020.

Version	Minor release	CD and DVD ISO Images	Packages	Release Email	Release Notes	End-Of-Life
CentOS-7	7 (1503)	x86_64	RPMs	CentOS	CentOS RHEL	30 June 2024
CentOS-6	6.7	i386 x86_64	RPMs	CentOS	CentOS RHEL	30 Nov 2020
CentOS-5	5.11	i386 x86_64	RPMs	CentOS	CentOS RHEL	31 Mar 2017**

Figura 12 Distribución de Centos 6 y vida útil. [24]

2.2. Instalación de Asterisk

Una vez que se ha instalado el sistema Centos se procede a actualizar cada uno de los programas que vienen con esta distribución. El objetivo de dicha actualización es hacer menos vulnerable a ataques de piratas cibernético. La comunidad de Linux se encarga de poner parches de seguridad a cada una de las aplicaciones que corre sobre sus sistemas operativos y para evitar estos huecos inseguros pues actualizamos utilizando el sencillo comando:

[root@localhost certified-asterisk-11.6-cert4]# yum -y update

Automáticamente se actualizarán todos los paquetes y programas instalados en el Centos.

Instaladas las dependencias necesarias nuestro sistema está listo para instalar la central telefónica que brindara los servicios propuestos.

Lo primero que se hace es ubicarse en la carpeta de descargas del sistema mediante el comando:

[root@localhost certified-asterisk-11.6-cert4]# cd /usr/src/

[root@localhost src]#

Una vez ubicados en la carpeta utilizando la aplicación de wget que se instaló en las dependencias al inicio descargamos Asterisk de su sitio oficial. La versión que se descargará será la 11.6 debido al análisis que se ha hecho de que es estable y que tiene un largo tiempo de soporte por parte de los desarrolladores.

Con el siguiente comando se puede descargar la versión de Asterisk deseada:

[root@localhostcertified-asterisk-11.6-cert4]#wget http://downloads.asterisk.org/pub/telephony/certified-asterisk/certified-asterisk-11.6-current.tar.gz

Descargado el archivo que se escogió como se encuentra comprimido se debe descomprimir para luego ser compilado. Esto se hace utilizando el comando tar.

[root@localhost certified-asterisk-11.6-cert4]# tar -zxvf certified-asterisk-11.6-current.tar.gz

Luego entramos a la carpeta de Asterisk que debemos instalar y lo hacemos usando el comando cd

[root@localhost certified-asterisk-11.6-cert4]# cd certified-asterisk-11.6-cert4

Dentro de esta carpeta se debe bajar un parche de instalación del códec VP8. Dicho códec es un códec de video ampliamente utilizado y que será empleado para realizar las llamadas de video. Destacar que el códec Vp8 es el último códec de video de On2 Technologies diseñado para reemplazar a su antecesor, VP7. Google liberó el códec VP8 como código abierto. Dicho código fuente está disponible a la comunidad del software libre. [25]

Para instalar el parche se lanzó el siguiente comando:

[root@localhost certified-asterisk-11.6-cert4]# patch -p1 -i asterisk-11.5.0_opus+vp8.diff

Una vez que se ha lanzado el comando queda instalado el códec de Vp8

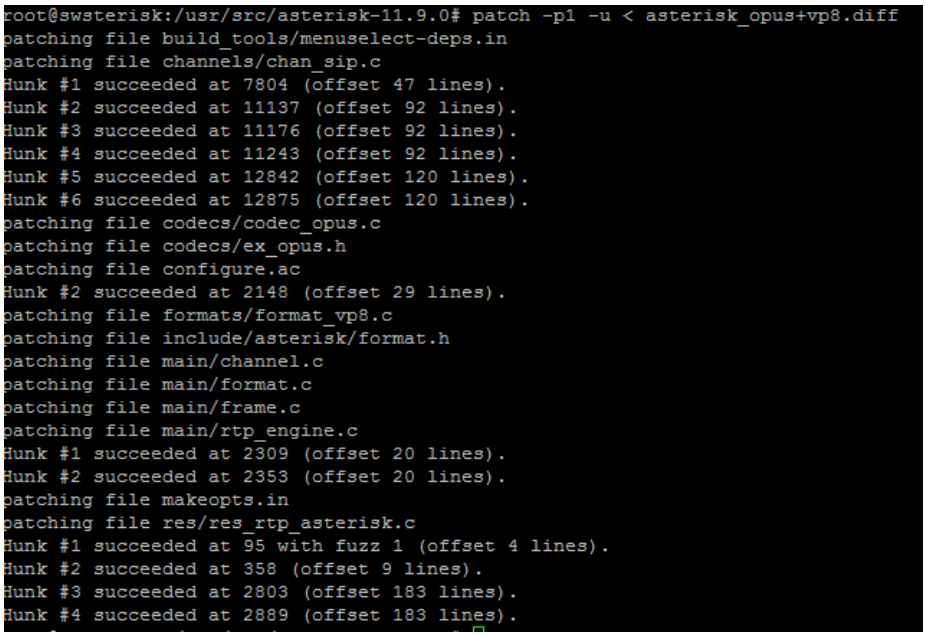


Figura 13 Instalación de parche de Vp8

El parche de Vp8 es necesario instalar porque hasta la actualidad ninguna de las versiones de Asterisk tienen en su núcleo de códec este códec de video. Se espera que en próximas versiones o parches de Asterisk sea incluido puesto que el Vp8 por las capacidades que tiene está remplazando los códecs tradicionalmente usados. La figura 13 muestra la aplicación del parche Vp8

Hasta ahora lo que se ha hecho es preparar los componentes que se necesitan para la correcta instalación de Asterisk y para la implementación de los servicios deseados. En este punto ya se pueden compilar los archivos de Asterisk. Para esto se debe lanzar el comando. /configure como se muestra a continuación.

[root@localhost certified-asterisk-11.6-cert4]# ./configure

La figura muestra el estado final de la compilación

Figura 14 Compilación de Asterisk

Con la compilación hecha se debe lanzar el menú de selección de los componentes a instalar en Asterisk, dicho menú mostrará y dejara configurar los parámetros deseados. La figura 14 muestra la compilación.

El siguiente comando hace mostrar el menú antes descrito

[root@localhost certified-asterisk-11.6-cert4]# make menuselect

La figura 15 muestra el menú de selección de opciones de Asterisk

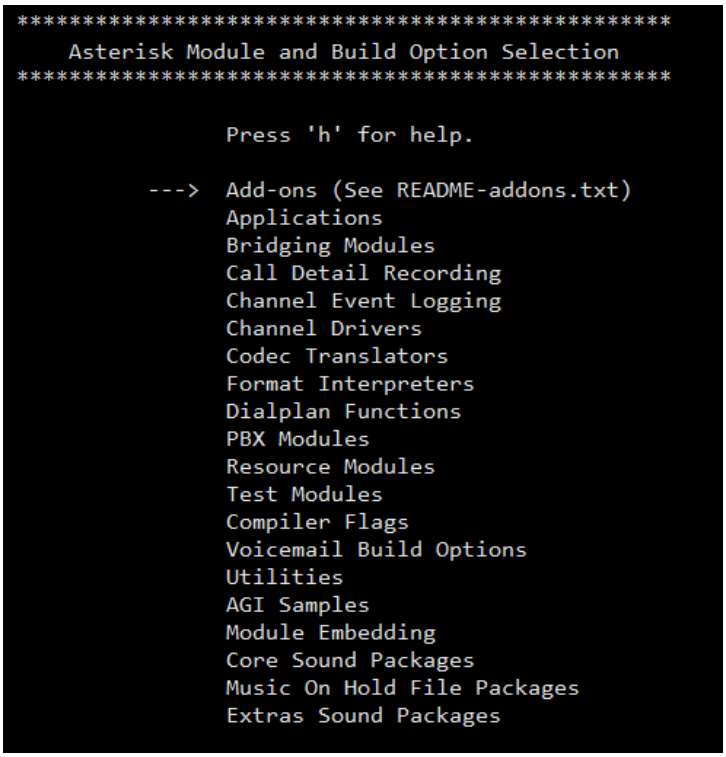


Figura 15 Menú de Selección de Opciones Asterisk

Lanzado el menú se debe verificar que se encuentran los seleccionados los siguientes componentes:

-SRTP

-VP8

-Speex

La Figura 16 muestra lo anterior.

El SRTP se usara para cifrar los datos del usuario mientras son transmitidos por internet y los códecs VP8 y speex serán los códecs de video y de audio respectivamente que se emplearan para las llamadas.

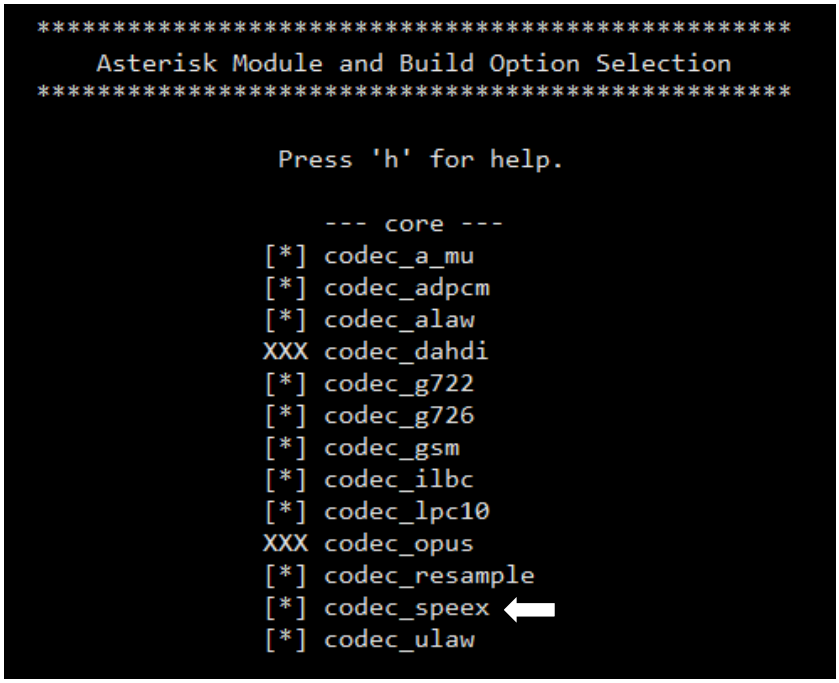


Figura 16 Códec Speex en Asterisk

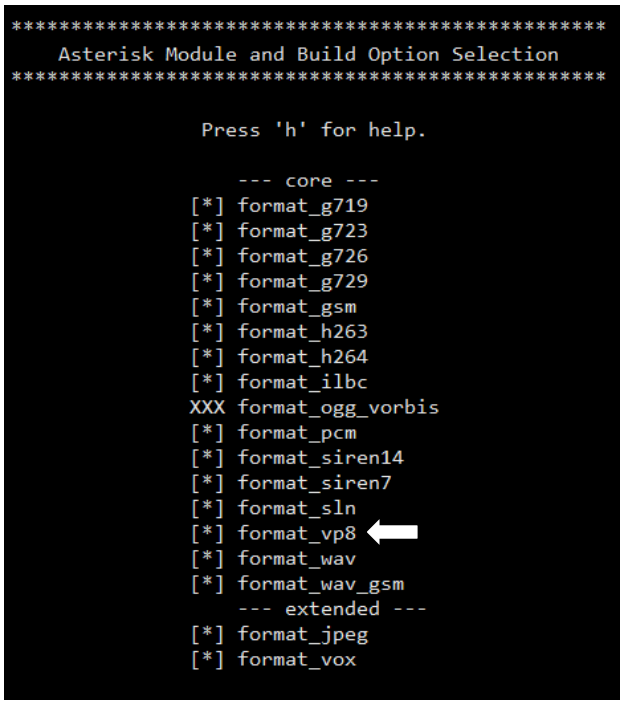


Figura 17 Códec de VP8 en Asterisk

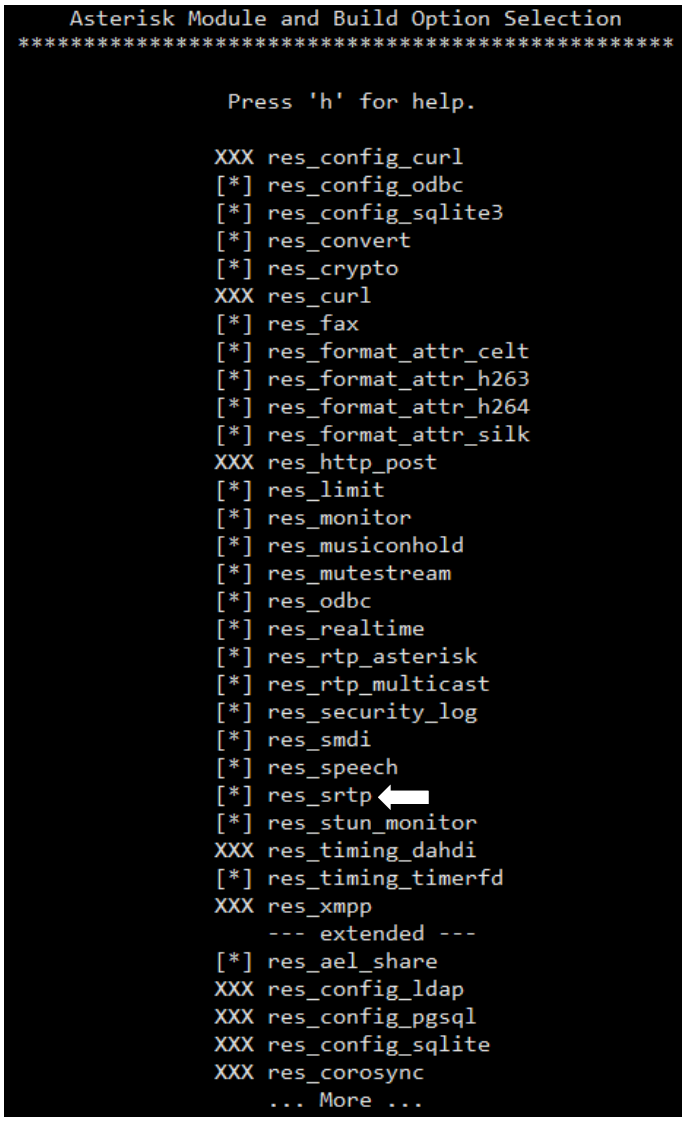


Figura 18 SRTP en Asterisk

Las Figuras 16, 17 y 18 muestran la selección de los codecs speex, Vp8 y del protocolo SRTP respectivamente.

Luego con todo listo se lanza make y luego el make install para finalizar la instalación. Los siguientes comandos muestran el procedimiento

[root@localhost certified-asterisk-11.6-cert4]# make

[root@localhost certified-asterisk-11.6-cert4]# make install

Las figuras 19 y 20 muestran las pantallas de Asterisk para los comandos make y make install

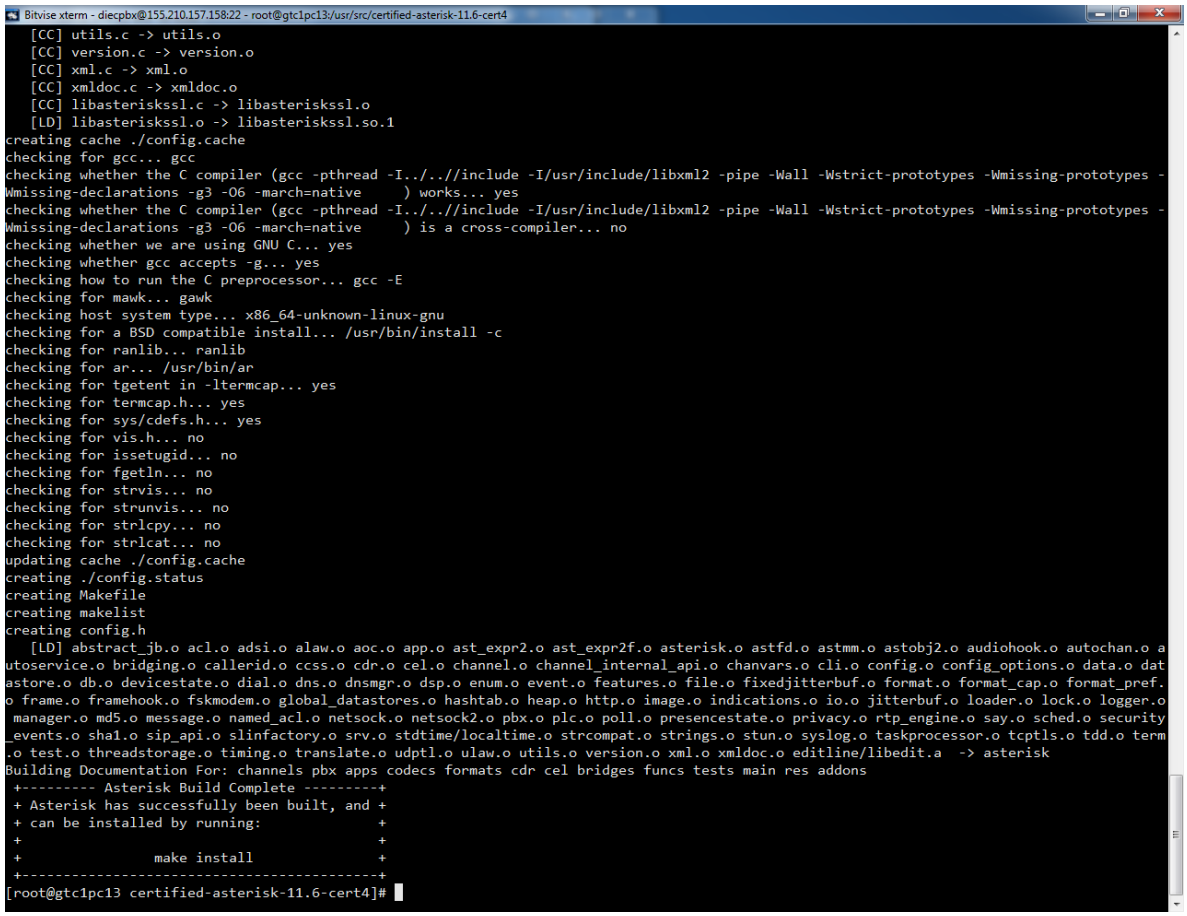


Figura 19 Pantalla de make

Como se puede ver el propio servidor Asterisk ha sugerido que sea lanzado el make install. Esto puede dar una medida que el proceso de instalación de la plataforma de comunicación está siendo instalada correctamente.

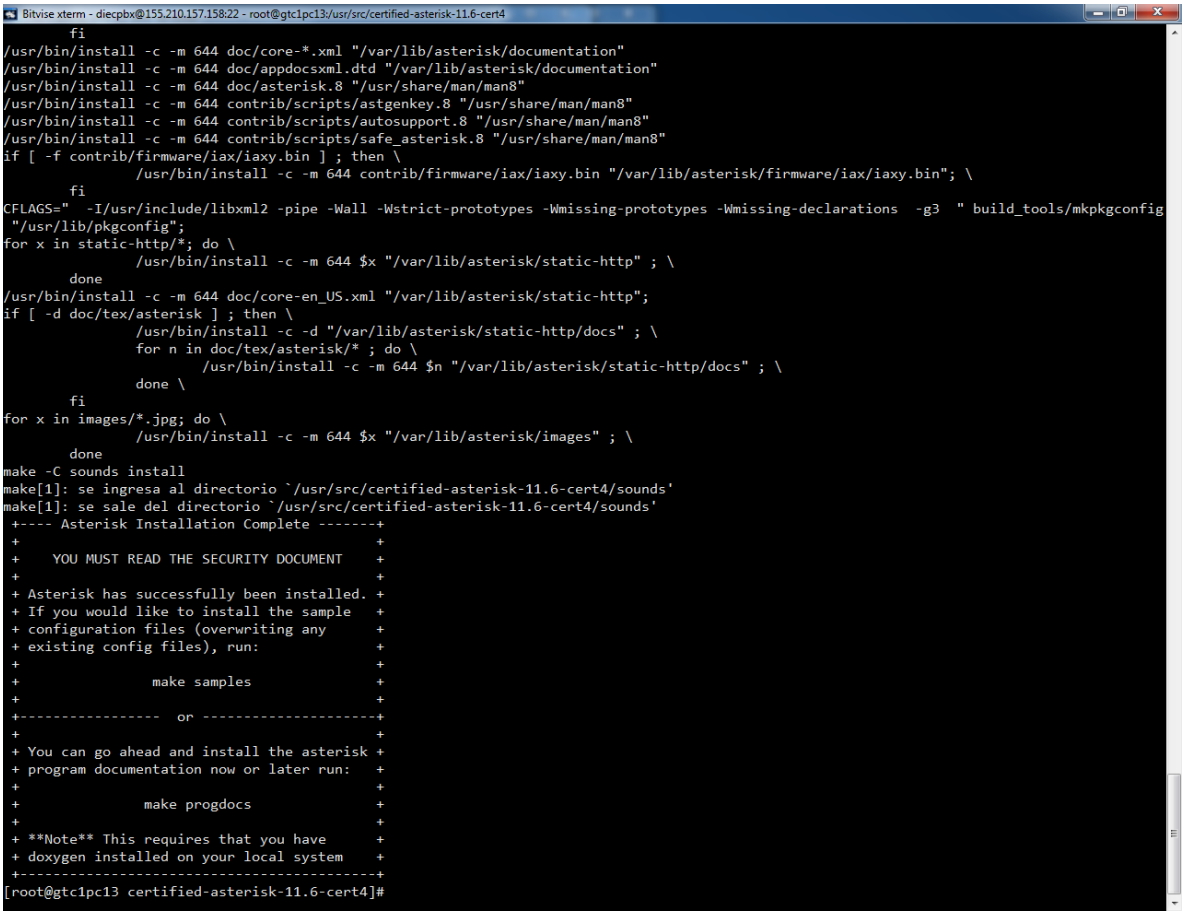


Figura 20 Pantalla de make install

Hasta este punto el Asterisk ya ha sido instalado y como sugerencia lanza un cartel de si se desea instalar los ejemplos que trae por defectos. Estos ejemplos o muestra de los archivos de configuración de Asterisk son muy importantes a la hora de configurar un servidor de este tipo, ya que mediante ellos se puede encontrar la ayuda necesaria para construir una rutina de configuración o ya se para rectificar alguna existente.

Para instalar los ejemplos sugeridos se debe lanzar el comando make samples y al ejecutarlo Asterisk tendrá en su núcleo de configuración los archivos de muestra.

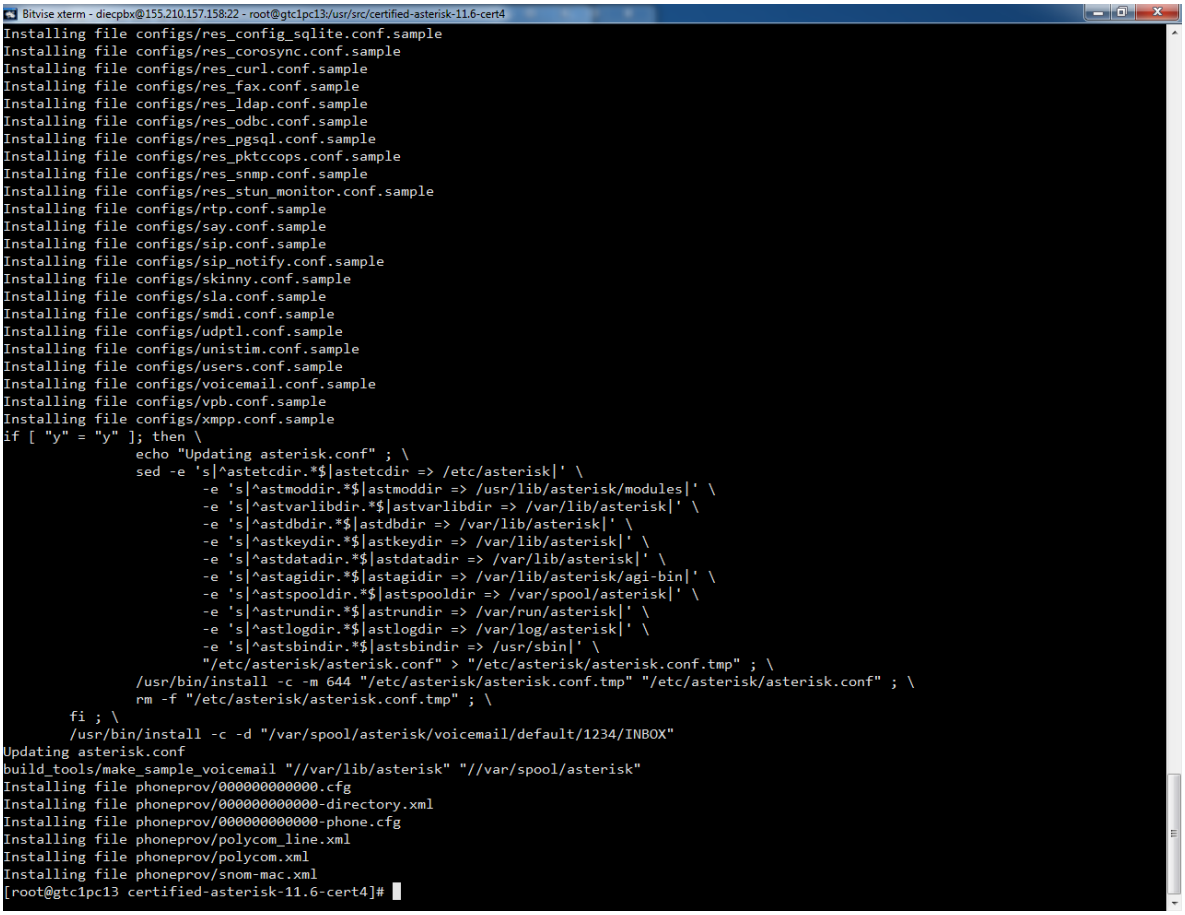


Figura 21 Instalación de ejemplos de configuración

La Figura 21 muestra el proceso de instalación de los ejemplos.

Hasta este punto Asterisk está listo para ser configurado y ser usado con cada uno de los servicios propuestos. A partir de este punto se mostrará cómo se configuran las extensiones y el plan de marcado para el establecimiento de las llamadas

Archivos sip.conf y extension.conf

Antes de configurar los archivos sip.conf y extension.conf se explicará en qué consisten cada uno de ellos.

El archivo sip.conf es un archivo que se encuentra en la carpeta Asterisk, exactamente ubicada en la dirección /etc/Asterisk/sip.conf. Dicho archivo es el encargado de contener e informarle al sistema de las extensiones que están configuradas, entiéndase por extensiones a los usuarios que podrán realizar llamadas. El sip.conf se divide por secciones, la sección general la cual contiene información general y que afecta a todo los usuarios contenidos en el archivo; otra sección o secciones más específicas que agrupan configuraciones de varias extensiones de modo que se pueda diferenciar las especificaciones de un grupo de usuarios con las especificaciones de otro grupo de usuarios y por último la sección propia de cada extensión, cuyas configuraciones sólo son competentes dentro del mismo usuario.

El archivo extension.conf es un archivo que también se encuentra en la carpeta de Asterisk, exactamente en la dirección /etc/Asterisk/extension.conf. Este archivo o fichero contiene lo que es el plan de marcación de las extensiones creadas en el sip.conf. Es donde se configura que es lo que harán las extensiones una vez que reciban acciones de comandos que indican marcar, colgar, enviar texto, etc. Dentro de este archivo además se configuran las llamadas a ciertas aplicaciones que contiene Asterisk y que son útiles para implementar ciertos servicios avanzados como son el buzón de voz, la conferencia múltiple entre usuarios, los distintos tipos de melodías al poner en espera una llamada o aplicaciones para envío de texto en tiempo real o lo que se conoce como chating en inglés. El archivo extension.conf también es estructurado por secciones al igual que el sip.conf, donde se tiene una sección general que afecta a todas las configuraciones, una sección más específica

y finalmente los grupos de configuración para cada grupo de usuarios creados en el sip.conf.

Configuración de sip.conf y extension.conf

Sip.conf

Para configurar un sip.conf debemos ir a la dirección donde Asterisk contiene este fichero: /etc/Asterisk/sip.conf

Con el comando: nano /etc/Asterisk/sip.conf se podrá abrir el fichero, modificarlo y guardarlo.

Lo primero que se hace para la configuración es crear un grupo de parámetros generales, o sea, la sección general. A continuación serán explicados los parámetros que se han configurado en el archivo sip.conf.

Localnet

El parámetro localnet indicará al servidor cuál es su red local y cuál es la máscara de subred. Todo esto con el objetivo de evitar problemas de resolución de direcciones de red o los llamados problemas NAT (Network Address Traslator).

Useragent

El parámetro useragent se le pone un nombre que sea sugerente pero que no de información sobre la versión de Asterisk que se está utilizando, esto es cambiado ya que el sistema brinda un nombre por defecto que indica datos que pueden ser usados para atacar el servidor.

Alwaysauthreject

La opción alwaysauthreject=yes es configurada en este estado debido a que en esta configuración se rechazarán los pedidos de autenticación fallidos utilizando nombres de extensiones válidas con la misma información que se rechazará un usuario inexistente. De esta forma se restringe la información que recibe un atacante para detectar nombres de extensiones utilizando técnicas de "fuerza bruta".

Allowguest

La opción allowguest=no es configurada en este valor para no permitir que usuarios no autenticados alcancen contextos que les permitan realizar llamadas.

Context

Además se puede configurar como opción que brinde seguridad un contexto de llamadas denominado default empleando la configuración context=default. Esta opción hace que los usuarios que se hayan podido registrar rompiendo la seguridad de las configuraciones anteriores, vayan a parar al contexto default del archivo extension.conf donde se fuerzan un colgar llamada. La mayoría de los atacantes hacen pruebas empleando contextos por defecto.

Srvlookup

Esta opción se pone en yes para hacer que Asterisk busque a través de una consulta DNS el servidor SIP correspondiente al dominio solicitado.

Udpbindaddr

Esta opción se pone en 0.0.0.0 para hacer que Asterisk escuche peticiones de UDP en todas las interfaces.

Transport

Esta opción configure sobre qué medio viajara la información y las opciones son tcp o udp. Se ha configurado en udp debido a que udp es un protocolo que es usado para servicios de tiempo real porque no tiene retransmisión y como la voz y el video son servicios de tiempo real y que la pérdida de una muestra o un frame es imperceptible, se configura de esta manera.

Port

Port es una opción que permite definir el puerto por donde viajara la información de señalización. Generalmente se deja en la opción por defecto que es el puerto 5060.

Rtptimeout

Este parámetro nos permite terminar la comunicación en el tiempo que deseemos, está puesto que a los 15 segundos de no haber flujo RTP o RTCP la comunicación sea suspendida.

Jbenable

Es un parámetro que habilita el buffer de jittter en el lado del receptor en los canales SIP.

Jbforce

Es un parámetro que Fuerza el uso de un jitterbuffer en el lado de recepción de un SIP. Por defecto está en no

Jbmaxsize

Es la máxima longitud del buffer del jiiter dado en milisegundos. Max length of the jitterbuffer in milliseconds.

Jbresyncthreshold

Es un parámetro que permite que saltos de tiempo en la trama una vez que el jiiterbuffer es re-sincronizado. Es un parámetro útil para mejorar la calidad de la llamada.

Videosupport

Es un parámetro que se configura si está permitido o no el uso de llamadas de videos en una comunicación.

Allow

Es una opción que habilitar o deshabilitar los códecs que interpreta o que tiene Asterisk en su núcleo. De este modo podemos permitir que determinados clientes usen determinados códecs conveniencia del administrador del servidor.

Canreinvite

Es una opción que restringe a los usuarios que están configurados en el sip.conf a permitir o no flujos RTP directamente entre ellos, sin que Asterisk tenga que estar en medio de la comunicación reenviando la información de uno a otro. [18]

Qualify

Este es un parámetro que permite restringir o no a que Asterisk monitorice las extensiones mediante el envío del campo SIP OPTION del protocolo SIP con el objetivo de ver si están activas o no. La petición SIP OPTIONS es el método usado para preguntar a un UA u otro servidor sobre sus capacidades y actual disponibilidad. La respuesta a la misma, evidencia el estado del UA y además lista en el campo Allow sus capacidades o métodos soportados. Cuando esta opción es puesta en no, Asterisk deja de monitorear el estado de la extensión y de esta forma omite las peticiones SIP OPTIONS hacia sus destinos. [19]

Host

Esta opción permite a los usuarios poderse conectar desde cualquier ip una vez que esta puesto en dinámico. Esto es una ventaja ya que permite que una extensión sea alcanzada desde cualquier ip.

Type

La opción type se usa para configurar la finalidad una extensión creada en el sip.conf. Es un campo que puede tener tres valores, peer, user y friend. Cuando está en peer permite hacer llamadas como peer y recibe llamadas como peer generalmente usada para enlaces entre dos centrales Asterisk los llamados trunks.

Cuando está en user sólo puede recibir llamadas como usuario, y no puede realizar llamadas. Cuando está configurado como friend entonces permite hacer llamadas como peer y recibe llamadas como usuario. Esta configuración es usada para las extensiones.

Secret

El campo secret es la contraseña de la extensión que se ha creado.

Encryption

Este campo es usado para definir si un usuario va a utilizar seguridad o no en la llamada. Solo resta decir que para crear una extensión se debe poner entre [número] el número de la extensión que se desea crear luego entre (sección) la sección del documento a la que pertenecen las extensiones y luego se configuran los campos context correspondientes al contexto indicado en el extension.conf y el campo secret con la contraseña. En el ejemplo que se pone a continuación, la contraseña coincide con la extensión, lo cual es una mala práctica pero se ha hecho sólo por motivos de simplicidad.

A continuación se pone un ejemplo de un archivo sip.conf:

[general]

;NAT OPTION
localnet=148.204.36.0/255.255.255.0

;Security OPTIONS
useragent=Asterisk PBX
alwaysauthreject=yes
allowguest=no
context=default

;Resto de configuración general
srvlookup=yes
udpbindaddr=0.0.0.0
transport=udp
port=5060
rtptimeout=15
jbenable=yes
jbforce=no
jbmaxsize=200
jbresyncthreshold=1200
videosupport=yes

[especifico]
disallow=all
allow=opus
allow=vp8
allow=speex
allow=gsm
allow=ulaw
allow=alaw
allow=g729
;allow=h263p
;allow=h263
canreinvite=no
qualify=yes
host=dynamic
type=friend

[101](específico)
secret=101
context=users
encryption=yes

[102](especifico)
secret=102
context=users
encryption=yes

Extension.conf

El extension.conf se encuentra en la carpeta Asterisk en la dirección /etc/asterisk/extension.conf. Para comprender la configuración del extension.conf se deben aclarar algunas de las opciones presentes. Se había dicho que este archivo de Asterisk tenía una estructura parecida a la del sip.conf, es decir, dividida en secciones. Consta con una sección general, una global y las secciones correspondientes a los contextos utilizados en el sip.conf.

En la sección general se configuran los siguientes campos:

Static

Indica si se ha de hacer caso a un comando "save dialplan" desde la consola. Por defecto es "yes". Funciona en conjunto con "writeprotect"

Writeprotect

Si writeprotect=no y static=yes se permite ejecutar un comando "save dialplan" desde la consola. El valor por defecto es no.

Autofallthrough

Si está activado y una extensión se queda sin algo que hacer termina la llamada con BUSY, CONGESTION o HANGUP. Si no está activada se queda esperando otra extensión.

Clearglobalvars

Si está activado se liberan las variables globales cuando se recargan las extensiones o se reinicia Asterisk.

En la sección o contexto Globals se definen las variables globales que se van a poder utilizar en el resto de los contextos.

Por ejemplo:

CONSOLE=Console/dsp

Esto indica que cuando se hace referencia a la variable CONSOLE se está llamando a /Console/dsp.

Después de general y globals vienen el resto de los contextos que serán usados para hacer los planes de marcación.

Como se puede ver existe un contexto llamado users, esto es un contexto que se ha creado para configurar el comportamiento de Asterisk. La línea exten => _1XX,1,Dial(SIP/${EXTEN}) lo que hace es permitir a cualquier extensión que comience por 1, es decir, 100, 101,102, etc, utilizar la aplicación marcar o Dial establecer comunicación con otra extensión. La extensión a la que se desea llamar será almacenada en la variable EXTEN. De esta forma Asterisk sabe a quién se debe dirigir la llamada. Además le sigue la línea exten => _1XX,2,SendText("Ahora no estoy disponible, llame luego"). Esta instrucción envía un mensaje de texto de "Ahora no estoy disponible, llame luego" cuando no se ha podido establecer la llamada. Destacar que esta instrucción tiene prioridad 2. Esto significa que solo será lanzada una vez que no se ha podido establecer la llamada. Siguiendo se encuentra la línea exten => _1XX,3,hangup que indica un colgar. La prioridad de dicha instrucción es 3, se ejecutará una vez que se ha completado el Dial o el SendText. Siempre se debe terminar un plan de marcación con esta instrucción debido a que existen ocasiones en las que la llamada se realiza y se mantiene activa gastando ancho de bando innecesariamente.

El resto de la configuración del plan de marcación son instrucciones para encuestar el servidor Asterisk y conocer si está funcionando bien. La instrucción exten => 111, 1, Playback (demo-congrats) permite que desde una extensión se marque el 111 y la aplicación Playback reproduce un demo o grabación de bienvenida que simula una llamada. Además se pude configurar otros demos como son el de la instrucción exten => 610,1,Playback(demo-echotest) que permiten a las extensiones marcar a la 610 y evaluar una prueba de eco.

El ejemplo que se muestra es una copia exacta de la configuración que se hizo para hacer funcionar el sip.conf anterior y que permite realizar llamadas de video y de voz entre los usuarios.

```
[general]

static=yes
writeprotect=no
autofallthrough=no
clearglobalvars=no

[globals]

CONSOLE=console/dsp

[users]
exten => _1XX,1,Dial(SIP/${EXTEN})
exten => _1XX,2,SendText("Ahora no estoy disponible, llame luego") exten =>
_1XX,3,hangup

;Peers para pruebas
exten => 111,1,Playback(demo-congrats)
exten => 111,2,hangup
exten => 610,1,Playback(demo-echotest)
exten => 610,2,Echo
exten => 610,3,Playback(demo-echodone)
exten => 610,4,hangup
```

Hasta este punto se han explicado los archivos sip.conf y extension.conf. Esta configuración cargada en el servidor Asterisk permitirá realizar llamadas de voz y llamadas de video encriptadas.

2.3. Linphone aplicación cliente.

Linphone es un teléfono SIP de código abierto, disponible en entornos móviles, web y de escritorio (iOS, Android, Windows Phone 8, Linux, Windows, MAC OSX). Utiliza el protocolo SIP por lo que puede ser usado por cualquier operador SIP de VoIP. Linphone además cuenta con una librería denominada como Liblinphone quien implementa todas las funcionalidades de Linphone. Dicha librería es un potente SDK de vídeo VoIP SIP que cualquiera puede utilizar para agregar capacidades de audio o videollamada a una aplicación determinada que se desee diseñar. Proporciona una API de alto nivel para iniciar, recibir y terminar llamadas de audio y video. [28]

Linphone se puso en marcha en 2001. Fue la primera aplicación de código abierto que utilizara SIP en Linux. Por más de 10 años Linphone ha sido portador de soluciones para las principales plataformas existentes de escritorio, móviles y web:

- Escritorio de Windows en 2006
- iOS y Android en 2010
- Blackberry OS5-7 en 2011
- Windows Phone 8 en 2013
- Aplicaciones web en 2013 (Chrome, Mozilla Firefox, Microsoft Internet Explorer, Apple Safari) [24]

2.4. Configuración de Linphone Desktop de Windows.

Para la realización de las pruebas de funcionalidad del sistema, se utilizó linphone en dos versiones diferentes, la versión de Desktop de Windows y la versión de IOS en Iphone.

Para la configuración de la versión de escritorio es necesario descargar la aplicación desde el sitio de Linphone. Se puede descargar desde el siguiente enlace: http://www.linphone.org/releases/windows/Linphone-3.9.1-win32.exe

Una vez que se ha descargado e instalado, se puede configurar para su utilización.

La figura 22 muestra el asistente de configuración guiará a los usuarios con el correcto llenado de los datos para que la aplicación quede funcional.

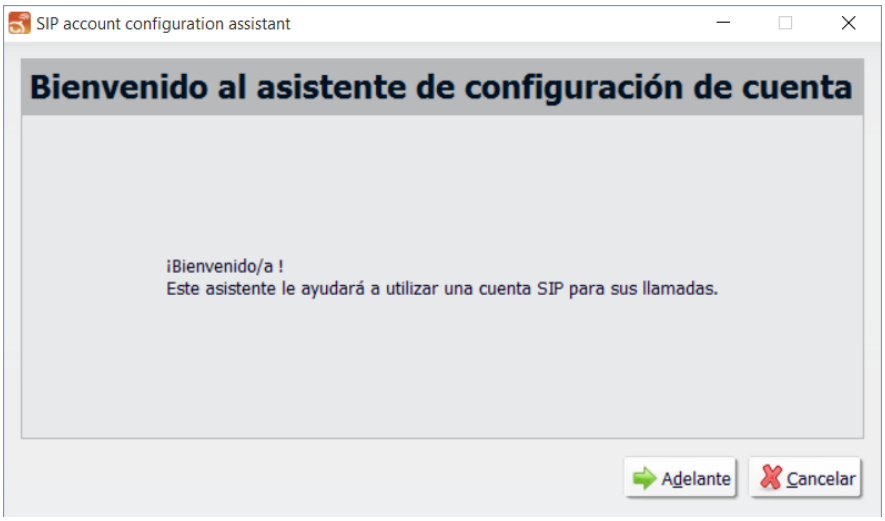


Figura 22 Asistente de configuración de Linphone.

La figura 23 muestra un menú de selección para la elección del servidor de VoIP que se desee. Como lo que se tiene es un servidor propio se ha seleccionado la opción que se puede ver en la imagen.

Figura 23 Selección del servidor de VoIP.

Una vez seleccionada la opción deseada se configuran los datos del servidor VoIP. Se tiene que proporcionar información sobre el usuario o extensión que se quiere utilizar, el password o contraseña correspondiente con la extensión y el dominio o IP del servidor de VoIP. Además aparece un campo de proxy que en el caso de esta tesis no fue utilizado nunca. La figura 24 muestra dicha explicación.

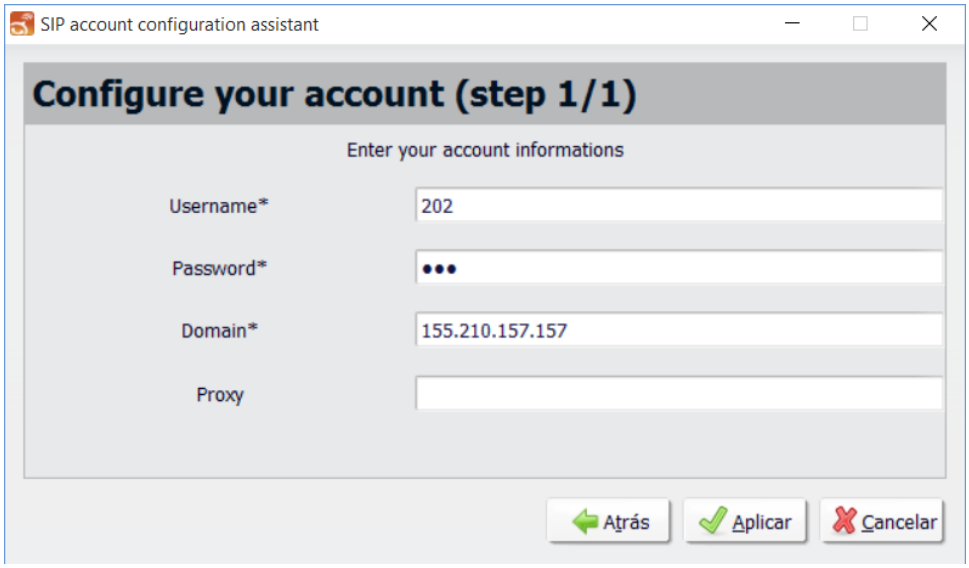


Figura 24 Configuración de la extensión.

Como se puede ver se ha configurado la extensión 202, se ha proporcionado la contraseña y además se ha puesto el IP del servidor 155.210.157.157 donde se encuentra un servidor Asterisk.

Cuando se ha completado la configuración la aplicación está casi lista para su utilización, faltarían algunos ajustes como los códecs utilizados, puertos de audio, de video, puertos de señalización y selección del tipo de seguridad. La figura 25 muestra la finalización de la configuración.

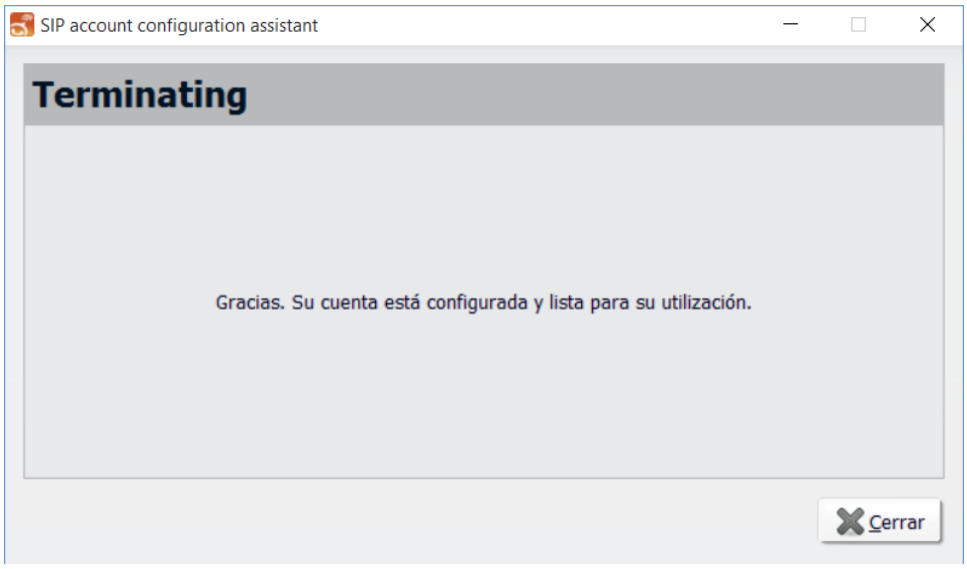


Figura 25 Finalización del proceso de configuración de la extensión.

Estando configurada la extensión, la aplicación debe decir que se ha podido registrar la extensión en el servidor. La figura 26 muestra lo anterior dicho.

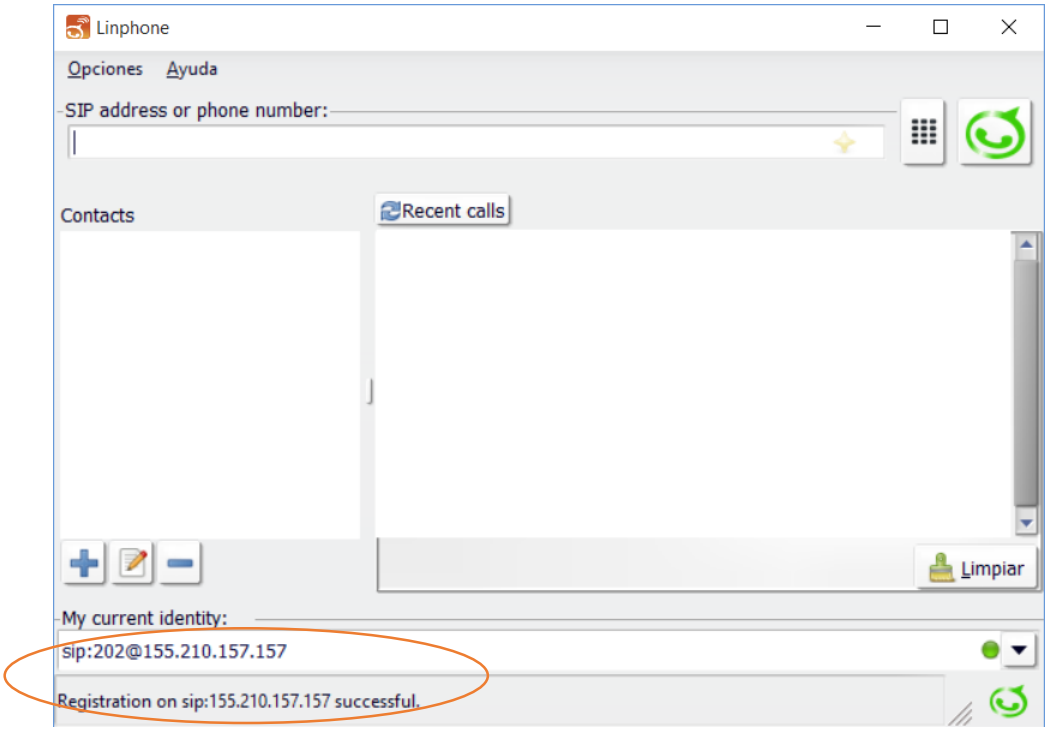


Figura 26 Proceso de registro correcto.

Para la configuración de la seguridad se necesita acceder a las opciones en el menú superior e ir a preferencias como se muestra en la figura 27.

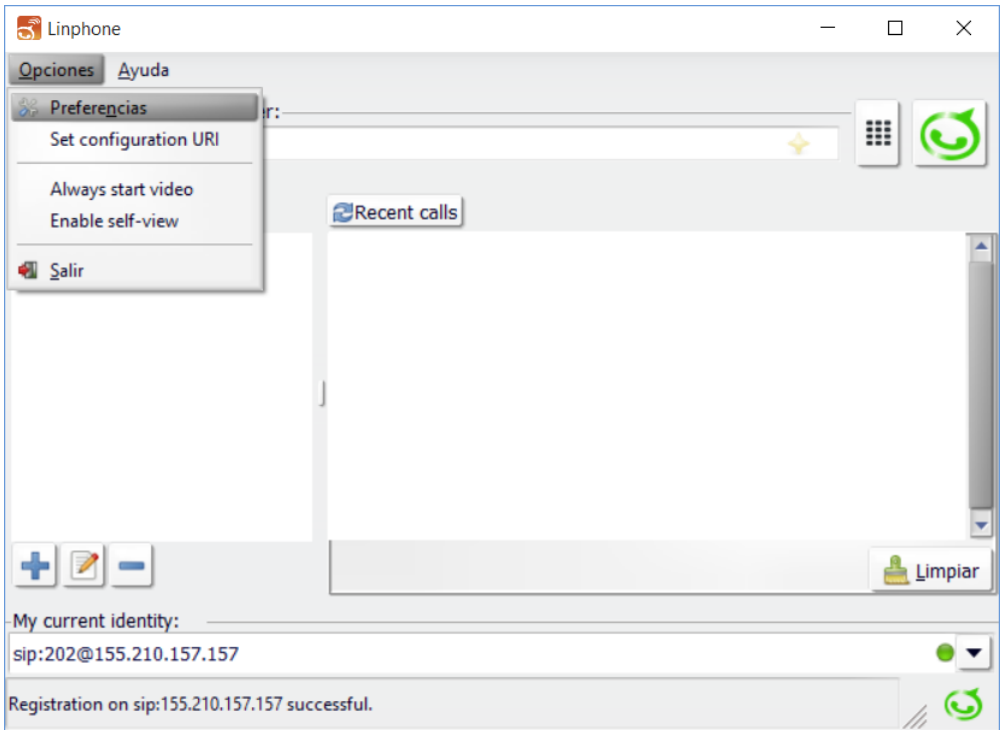


Figura 27 Acceso a las opciones de Linphone

Como se ve en la figura 28, Linphone da la posibilidad de utilizar seguridad con SRTP, ZRTP o no utilizarla. En el caso de esta tesis se ha utilizado la seguridad de SRTP que es la configurada en el servidor Asterisk.

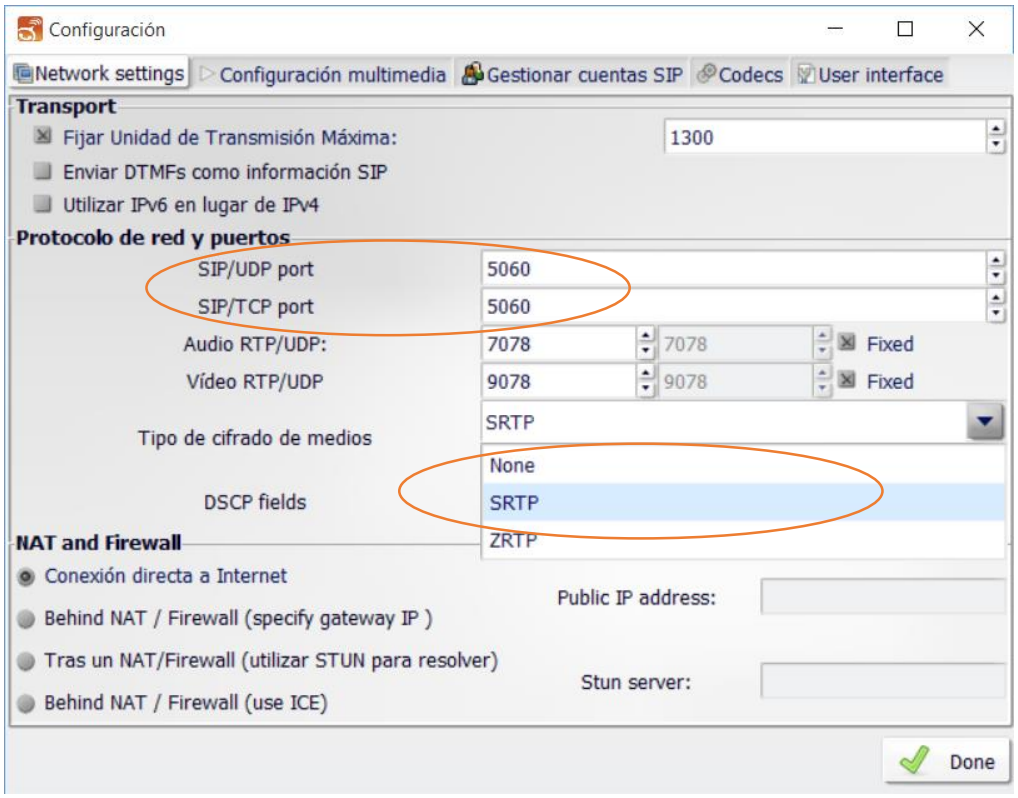


Figura 28 Selección de seguridad SRTP.

También se puede ver en la figura 28 la configuración de los puertos de audio, de video y de la señalización SIP. Se ha optado por mantener los puertos que SIP tiene como predeterminados.

La siguiente figura 29 muestra la configuración que tiene que ver con la selección de los códecs.

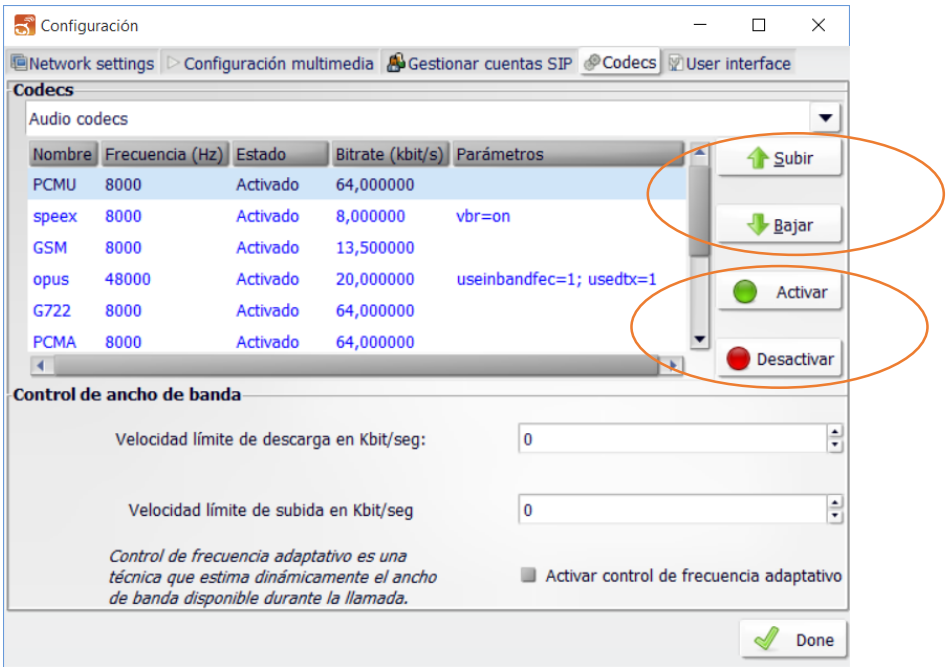


Figura 29 Configuración de códecs.

Se puede ver claramente presencia de los principales códecs que se utilizarán en las pruebas del próximo capítulo. Los códecs son: g711u o PCMU, g711a o PCMA, speex, GSM, opus y g722. Existe unos botones donde se puede dar prioridad a los códecs incluso activarlos o desactivarlos.

Una vez que se ha configurado el tipo de seguridad, los puertos de la señalización SIP y los códecs a utilizar la aplicación de Linphone está lista para ser usada.

2.5. Linphone Mobile IOS

Para la instalación de linphone en un teléfono móvil se deben hacer pasos similares a los que se describieron con anterioridad. No obstante se hará un breve tutorial de cómo se puede configurar este tipo de aplicaciones para que trabaje con el servidor Asterisk que se presenta.

Lo primero que se debe hacer es descargar la aplicación desde el appstore. Se pone el enlace para que pueda ser descargado sin problemas:

https://itunes.apple.com/en/app/linphone/id360065638?mt=8

Una vez que esta descargado e instalado en el teléfono móvil se puede pasar a configurarlo.

La figura 30 muestra el asistente de configuración predeterminado por la propia aplicación. Como se ve y muy similar a la versión de escritorio, se puede seleccionar el servidor desde donde se va a servir el cliente. En el caso de esta tesis como se cuenta con un servidor propio pues se selecciona la opción de tengo una cuenta SIP que sale circulada en la figura.

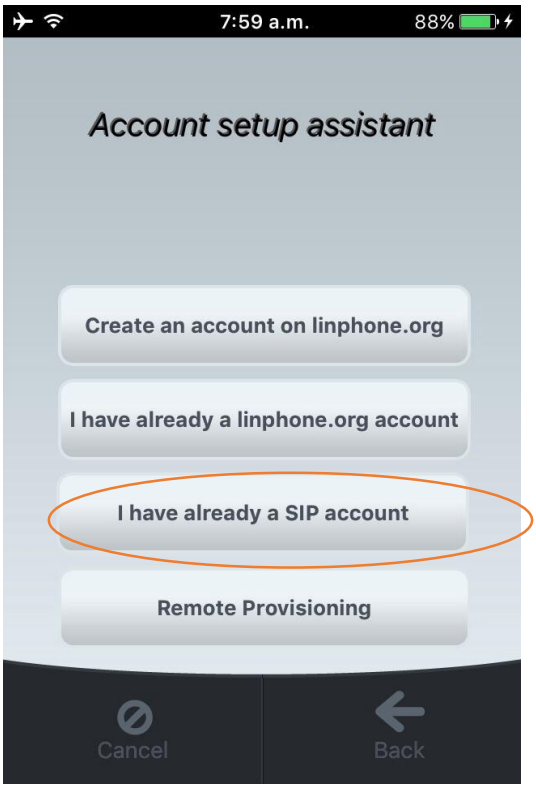


Figura 30 Asistente de Configuración.

Una vez que se ha seleccionado la opción deseada se puede pasar a brindarle a la aplicación los datos correspondientes a la extensión que ha sido asignada en el servidor para este cliente. La figura 31 muestra cómo se proporciona los datos de la extensión, el password o contraseña y el dominio o IP del servidor.

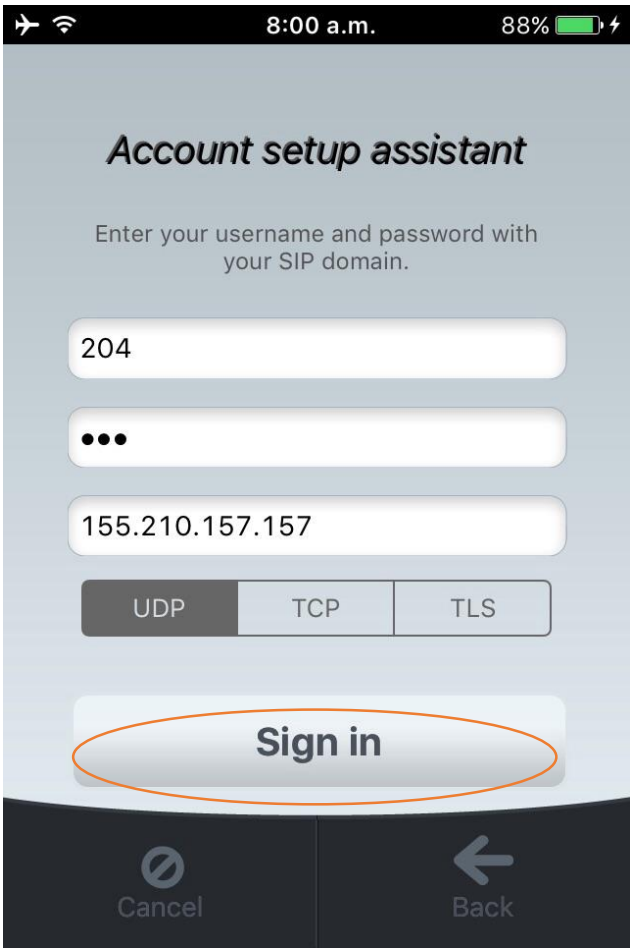


Figura 31 Configuración de una extensión en Linphone.

Como se puede ver se ha asignado la extensión 204, la contraseña y el IP 155.210.157.157. También se puede ver cómo se puede seleccionar en esta versión móvil, el tipo de transporte que se desea, UDP, TCP o TLS. En el caso de esta tesis se ha estado trabajando con UDP.

Una vez que se han suministrado los datos para la configuración de la extensión en el cliente se le da registrar en el botón visible de Sing in marcado con un óvalo rojo.

Si los datos son correctos y el servidor está funcionando bien debe salir que la extensión ha sido registrada satisfactoriamente como se muestra en la figura 32.

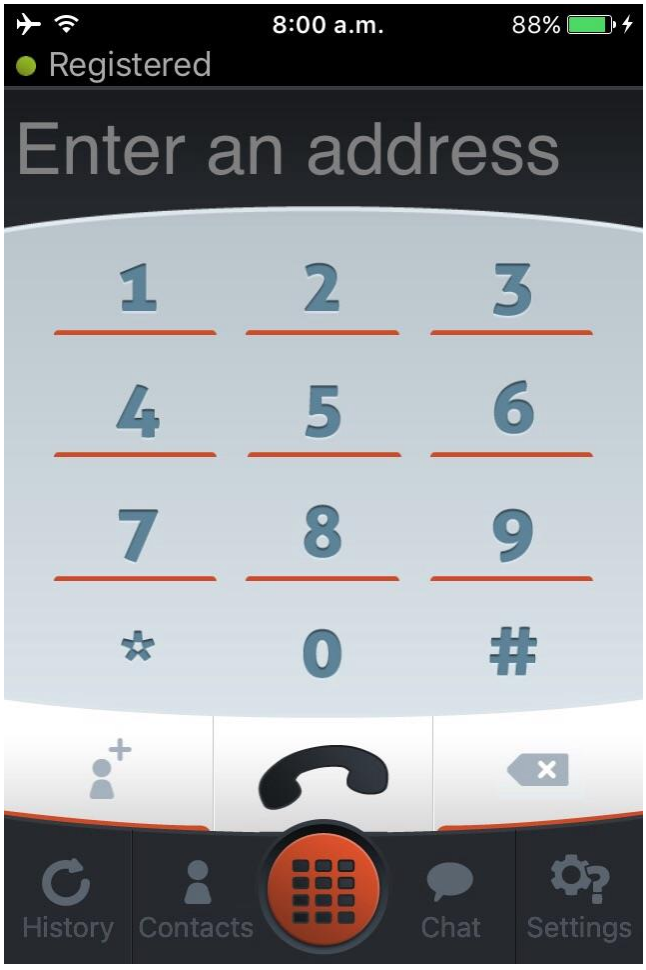


Figura 32 Proceso de registro correcto.

Hasta este punto la extensión asignada ha sido configurada y se ha registrado correctamente en el servidor. Entonces se puede pasar a realizar otras configuraciones como la de la seguridad, el tipo de códec deseado, etc.

Para acceder a la configuración se debe ir a Settings donde estarán las opciones deseadas para continuar. La figura 33 muestras dichas configuraciones.

Se puede ver claramente opciones dedicadas a habilitar o deshabilitar el video, configuraciones de audio, video, configuraciones relacionadas con la llamada, con la red y opciones avanzadas. Dentro de las opciones de audios están las determinadas para la selección de los códecs. La figura 34 muestra dicha configuración.

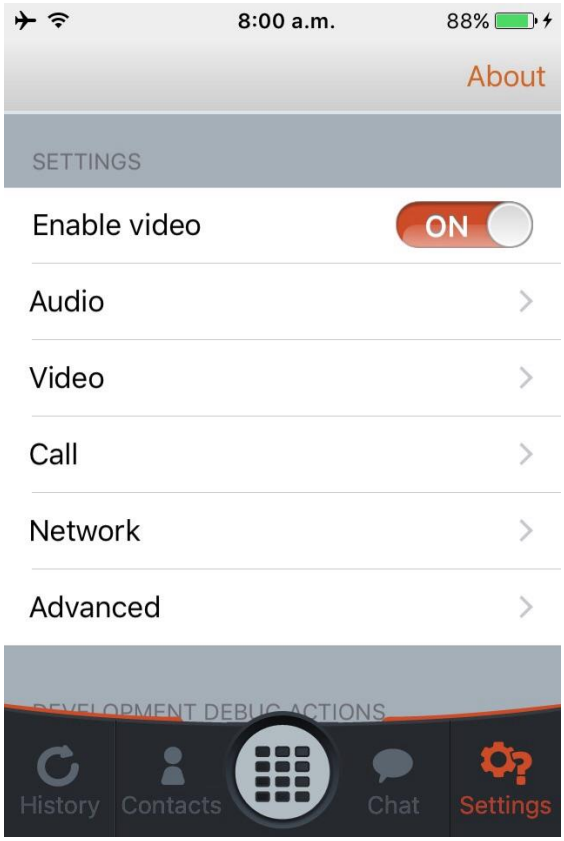


Figura 33 Configuración de Linphone.

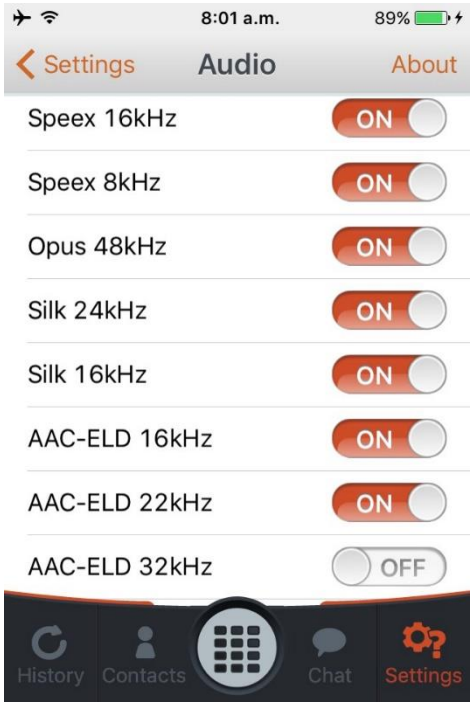


Figura 34 Configuración de audio.

Como se puede ver existen muchos de los códecs que son compatibles con la versión móvil de linphone. En este apartado de la configuración se pueden habilitar o deshabilitar a conveniencia del usuario.

Dentro de la configuración de video, Linphone solo es compatible como se puede ver en la figura 35, con el códec Vp8, pero es precisamente el códec que se ha manejado en la implementación del sistema de video llamadas seguras.

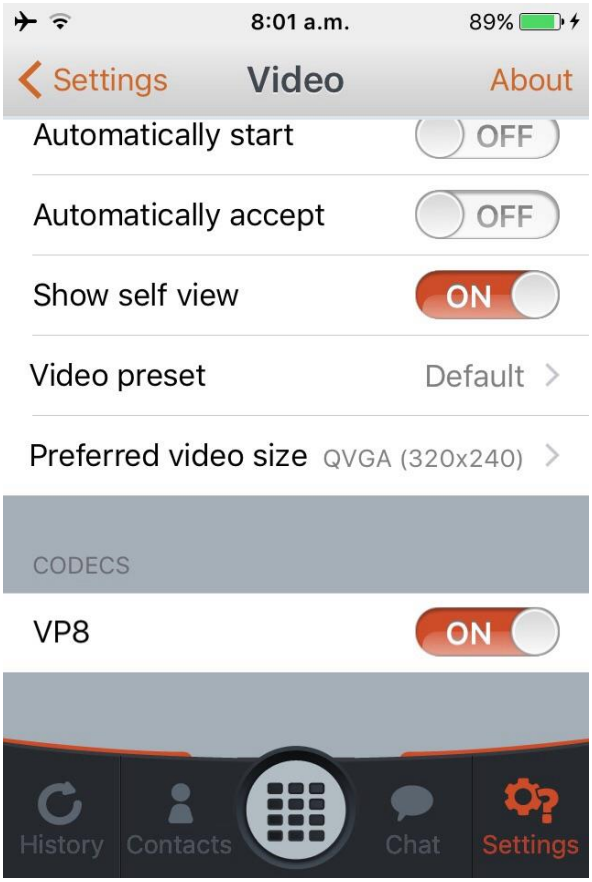


Figura 35 Configuración de video.

Como se puede apreciar en la figura 35 además de la selección del códec también está la selección de la calidad del video, acciones de envío automático de la imagen, mostrarse si mismo mediante cámara frontal, etc. Estas opciones son de carácter secundario para el funcionamiento de la aplicación pero están disponibles para su uso.

Para la configuración de la seguridad, se debe ir a network y dentro de network al apartado de encriptación. La figura 36 muestra lo anterior dicho.

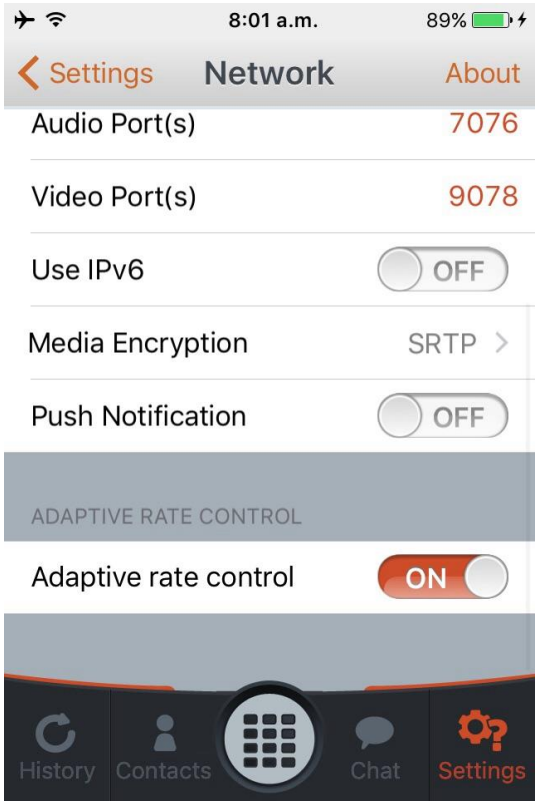


Figura 36 Configuración de Red.

Dentro de encriptación media se puede seleccionar el tipo de seguridad que se desea. La figura 37 muestra los tipos disponibles.

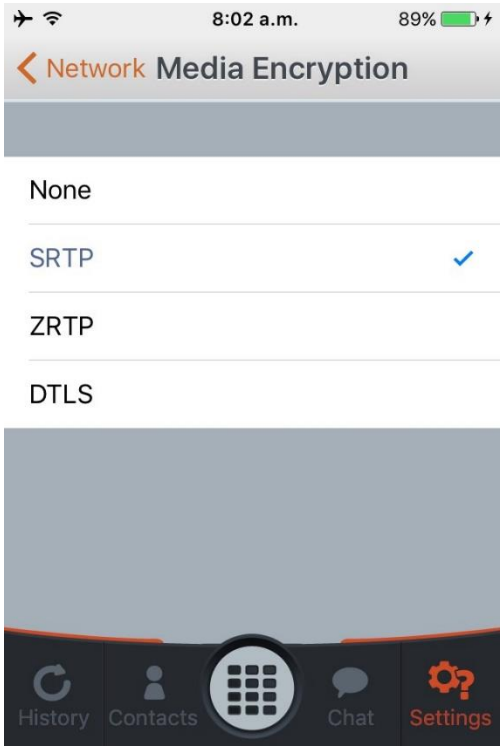


Figura 37 Tipos de seguridad de Linphone.

En el caso de esta tesis se ha seleccionado la de SRTP basándose en que es el tipo de seguridad que ha sido implementado en el servidor Asterisk.

Una vez que se han seleccionado las configuraciones necesarias para el cliente, este está listo para ser usado.

2.6. Configuración de Sistema de Seguridad anti ataques.

Para darle robustez al sistema se ha creado una configuración en Linux Centos para evitar el ataque de piratas cibernéticos al servidor diseñado. Para esto se ha instalado un software llamado Fail2Ban que bloquea los ataques tanto por SSH como ataques directos a los servicios de VoIP.

Pero antes de pasar a configurar dicho software se ponen diferentes configuraciones propias de Asterisk y que permiten evitar ataques menos significativos.

Configuraciones de Seguridad en Asterisk

- Para evitar dar información sobre la versión de Asterisk se modifica en la ruta etc/asterisk/sip.conf y se añaden las siguientes líneas:

  ```
  [general]
  useragent=Asterisk PBX
  ```

- Para evitar que usuarios anónimos usen el servidor, se crea un contexto default que cuelgue todas las llamadas de los usuarios que se registraron anónimamente. Para esto se tiene que ir al Sip.conf y agregar las líneas siguientes:

  ```
  [general]
  context=default
  ```

Luego se debe ir al extensión.conf en la ruta siguiente /etc/asterisk/extensions.conf y agregar las líneas que a continuación se ponen:

```
[default]
exten => _X.,1,Hangup(21)
exten => s,1,Hangup(21)
```

De esta manera, un atacante anónimo al conseguir pasar por alto la autenticación no podría realizar ninguna llamada.

- Limitar las llamadas simultaneas de cada extensión

Para esto se debe ir al sip.conf y agregar las líneas siguientes

[número de extensión]
call-limit=1

Por último se deben revisar los logs de las llamadas para ver si son coherente con los usuarios que se tienen. El archivo con esta información se encuentra en la ruta:

/var/log/asterisk/cdr-csv/Master.csv

Configuración de Fail2Ban

Para configurar el Fail2Ban lo primero que se hace es activar el log de seguridad de Asterisk. El archivo se encuentra en la ruta /etc/asterisk/logger.conf

Una vez que se ha accedido al mencionado archivo se debe modificar en la sección general, el parámetro dateformat y configúralo de la siguiente forma:

[general]

dateformat=%F %T

Luego en la sección logfiles debe quedar de la siguiente forma:

[logfile]
security => security

Una vez que se encuentra el log activado se puede instalar el softwarer Fail2Ban. Para esto se puede descargar de la siguiente dirección:

http://dl.fedoraproject.org/pub/epel/6/x86_64/epel-release-6-8.noarch.rpm

Una vez instalado se debe añadir un filtro dentro de la aplicación que es específico para el log de seguridad de Asterisk. Este filtro se puede localizar en la siguiente dirección: /etc/fail2ban/filter.d/asterisk.conf.

Al abrir el archivo se encontrarán líneas similares a las que se ponen ms abajo y al configurarlas deben quedar de la siguiente manera:

```
 # Fail2Ban configuration file
#
# $Revision: 250 $
#
 [INCLUDES]
 # Read common prefixes. If any customizations available -- read them from
# common.local
#before = common.conf
 [Definition]
 #_daemon = asterisk
 # Option:  failregex
# Notes.:  regex to match the password failures messages in the logfile. The
#          host must be matched by a group named "host". The tag "<HOST>" can
#          be used for standard IP/hostname matching and is only an alias for
#          (?:::f{4,6}:)?(?P<host>\S+)
# Values:  TEXT
#
failregex                                   =                                   SECURITY.*
SecurityEvent="FailedACL".*RemoteAddress=".+?/.+?/<HOST>/.+?".*
  SECURITY.*
SecurityEvent="InvalidAccountID".*RemoteAddress=".+?/.+?/<HOST>/.+?".*
  SECURITY.*
SecurityEvent="ChallengeResponseFailed".*RemoteAddress=".+?/.+?/<HOST>/.+
?".*
  SECURITY.*
SecurityEvent="InvalidPassword".*RemoteAddress=".+?/.+?/<HOST>/.+?".*
 # Option:  ignoreregex
# Notes.:  regex to ignore. If this regex matches, the line is ignored.
# Values:  TEXT
#
ignoreregex =
```

Las líneas anteriores son un ejemplo exacto que trae el propio programa para realizar las configuraciones que se explican.

Para continuar con la configuración del Fail2Ban se debe añadir además las reglas de restricciones en el archivo jail.conf. Tal archivo se encuentra ubicado en la siguiente dirección: /etc/fail2ban/jail.conf

```
[asterisk]
enabled  = true
filter   = asterisk
action   = iptables-allports[name=ASTERISK, protocol=all]
logpath  = /var/log/asterisk/security
maxretry = 5
bantime  = 86400
findtime = 86400
```

Las líneas anteriores habilitan en el Fail2Ban las opciones para Asterisk y en conjunto con el firewall iptables, deniegan el acceso a los usuarios que intenten como máximo 5 veces un registro sin lograr el éxito. Se ponen además otras opciones que indican el tiempo que los usuarios estarán bloqueados por el firewall.

Se puede conocer el estado del Fail2Ban lanzando la siguiente línea de comando desde la consola de Centos.

```
[root@asterisk158 diecpbx]# fail2ban-client status asterisk
```

La figura 38 muestra el estado del Fail2Ban de uno de los servidores con los que se ha trabajado.

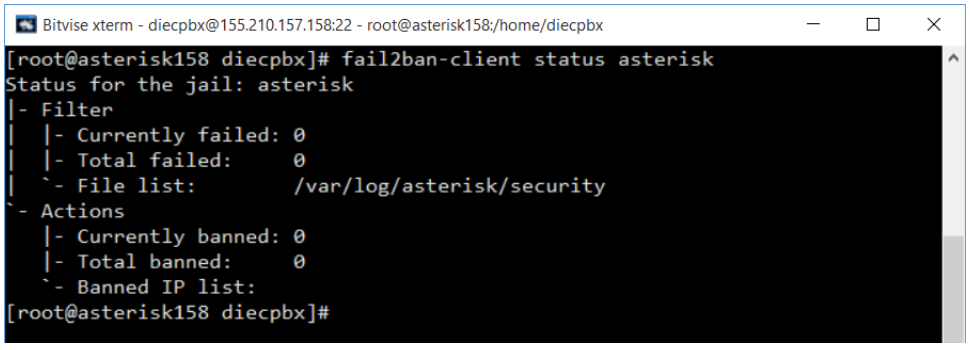


Figura 38 Estado de Fail2Ban

Para la protección contra ataques SSH, también existe una configuración en Fail2Ban. Al igual que con Asterisk, se debe crear una regla en el archivo jail.conf pero en esta ocasión destinada para SSH. La siguiente configuración es la indicada para lograrlo.

```
[ssh-iptables]
enabled  = true
filter   = sshd
action   = iptables[name=SSH, port=ssh, protocol=tcp]
           sendmail-whois[name=SSH, dest=root, sender=fail2ban@example.com]
logpath  = /var/log/secure
maxretry = 5
```

Tal como en el ejemplo de ataques contra Asterisk, en esta ocasión se restringen los accesos SSH cuando los atacantes hayan fallado 5 intentos al tratar de acceder. Para esto se crea una regla automática en el firewall iptables y se envía un email de notificación al administrador.

De esta manera el Fail2Ban queda configurado para estos dos servicios importantes. Cabe destacar que existen otros servicios que pudieran ser protegidos con el mismo software, sin embargo no son del interés de esta tesis.

2.7. Conclusiones

A lo largo de este capítulo se ha podido explicar la implementación de los servicios propuestos. Se ha realizado desde la instalación de Asterisk, hasta la configuración de los archivos principales y necesarios para el establecimiento, mantenimiento y finalización de una video llamada cifrada. Además se configuró un ejemplo de un cliente para poder servirse de Asterisk. Se debe destacar que Linphone, la aplicación cliente, es de código abierto lo que permite la implementación de aplicaciones clientes basadas en sus librerías. Esto significa que en un futuro se pudiera realizar una investigación sobre cómo implementar una aplicación cliente

para que el sistema diseñado quede completo. Finalmente se trataron aspectos de la seguridad del sistema como tal. De esta forma se explicó cómo se puede proteger al servidor de ataques mediante configuraciones sencillas propias de Asterisk y en combinación con la aplicación Fail2Ban.

Capítulo 3.

“Diseño de Herramienta de captura de tráfico”

3.0. Introducción.

A lo largo de este capítulo se explicará cómo a partir de dos softwares de análisis de tráfico en la red, se puede conformar una herramienta que permita la captura de información para poderla analizar y sacar conclusiones futuras en cuanto a los consumos de ancho de banda. Dichos softwares serán el Tcpdump y el Wireshark, donde este último será fundamental para en los capítulos de validación de pruebas, dejar en evidencia el empleo del SRTP como protocolo de cifrado y demostrar que es posible mediante el mismo tener una comunicación segura.

3.1. Herramienta de Captura de tráfico

Para la captura y análisis del tráfico de VoIP se ha implementado un método basado en dos softwares muy relacionados: Tcpdump y Wireshark. Cada uno es capaz por si solo de hacer todo que se necesita, sin embargo al unirlos en dicho método, se facilita y se simplifica mucho el trabajo a realizar.

3.1.1. Tcpdump

Tcpdump es una herramienta en línea de comandos cuya utilidad principal es analizar el tráfico que circula por la red. Permite al usuario capturar y mostrar a tiempo real los paquetes transmitidos y recibidos en la red a la cual el ordenador está conectado. Tcpdump funciona en la mayoría de los sistemas operativos UNIX: Linux, Solaris, BSD, Mac OS X, HP-UX y AIX entre otros. En esos sistemas, tcpdump hace uso de la biblioteca libpcap para capturar los paquetes que circulan por la red. Existe una adaptación de tcpdump para los sistemas Microsoft Windows que se llama WinDump y que hace uso de la biblioteca Winpcap.

El usuario puede aplicar varios filtros para que sea más depurada la salida. Un filtro es una expresión que va detrás de las opciones y que nos permite seleccionar los paquetes que estamos buscando. En ausencia de ésta, el tcpdump volcará todo el tráfico que vea el adaptador de red seleccionad. [29]

Dada una breve descripción del tcpdump se puede pasar a la explicación de cómo se realizó la programación de un script de consola en Centos para automatizar el proceso de captura de los datos.

Se creó un archivo al que se llamó captura en el que se escribieron las siguientes líneas:

```
hora=`date +"%T"`
/usr/sbin/tcpdump -i eth0 -w capturaPBX157"_"$hora -nn -s 80&
sleep $1
rm ./fichproc
ps aux | grep "tcpdump" | cut -b 10-14 | head -n3 > ./fichproc
fich="./fichproc"
read PROC<$fich
kill -15 $PROC
```

A grandes rasgos las líneas anteriores lo que hacen es definir una variable llamada *hora* donde se almacene la fecha y la hora en la que se realizó la captura. Luego se manda a ejecutar el tcpdump diciéndole que escuche por la interfaz eth0 y que escriba lo escuchado con el nombre capturaPBX157 seguido de la variable hora que contiene la información del momento en que el que se ha realizado la captura. Luego se lanza un comando sleep quien hace esperar la cantidad de segundos que se hayan configurado una vez que se ha lanzado la ejecución del script. Ya al final se hace un filtrado de los procesos que se encuentran en ejecución en el administrador de procesos de Linux y al localizar el proceso del tcpdump, se manda a cerrar o finalizar.

De esta manera quedó el método para capturar los paquetes. Es decir, cuando se quiere capturar una llamada de VoIP, se manda a ejecutar el script programado y automáticamente se crea un archivo con todos los paquetes que hayan pasado por la interfaz eth0. Pero no son todos los paquetes los que se quieren analizar, sino los paquetes que vienen con determinado protocolo, desde un ip específico, hacia un puerto determinado, etc. Es por tal causa que se complementó este método con el procesamiento de filtrado que puede hacer el Wireshark en un entorno visual.

3.1.2. Wireshark

Wireshark es una herramienta multiplataforma de análisis de red, producto de la evolución de Ethereal. Funciona igual que cualquier otro sniffer como Windump, TCPDump ó dsniff. Pero, al contrario de estos, lo hace mostrando los datos a través de un entorno gráfico de forma más amigable y entendible. [30]

Tiene diversas características que incluye lo siguiente:

- Inspección profunda de cientos de protocolos
- Captura en vivo y análisis offline
- Es multiplataforma. Se ejecuta en Windows, Linux, OS X, Solaris, FreeBSD, y NetBSD.
- Los datos de red capturados se pueden consultar a través de una interfaz gráfica de usuario
- Tiene capacidad de poderosos filtros de visualización
- Contiene un analizador de VoIP. [30]

Wireshark puede ser descargado desde el siguiente enlace:

https://www.wireshark.org/#download

Luego de la breve descripción del Wireshark se puede explicar cómo se trabajó con dicho software en beneficio de este proyecto de investigación. Una vez instalado el software y con las capturas que se desean analizar guardadas en un archivo Tcpdump, se puede importar la información a la aplicación. La figura 39 y 40 muestran dicho procedimiento.

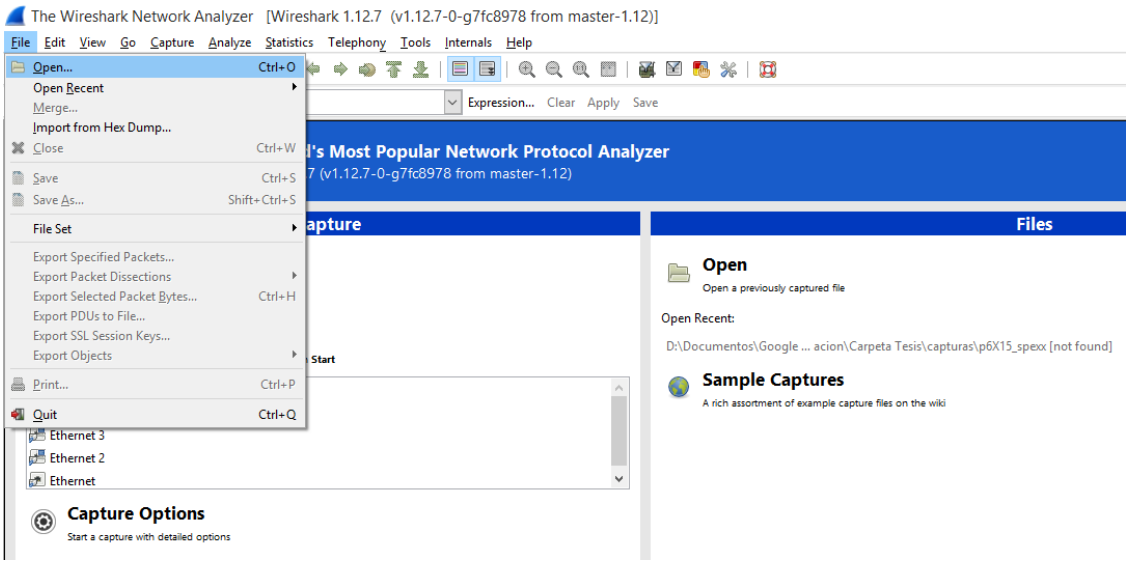


Figura 39 Abrir un archivo del Tcpdump al Wireshark.

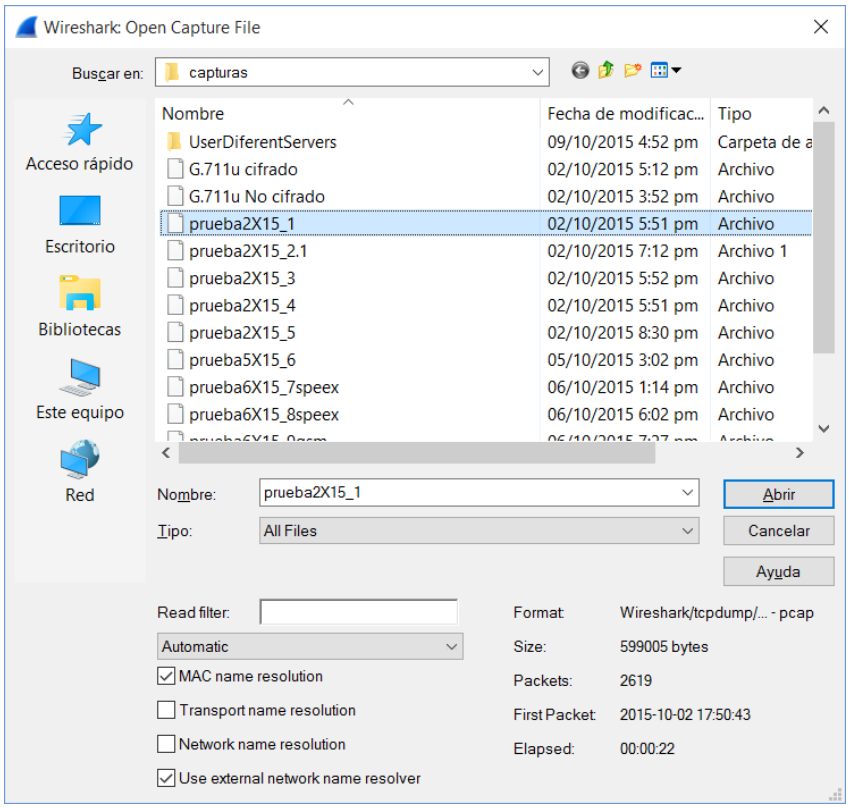


Figura 40 Selección del archivo que se desea importar.

Una vez que se ha importado la información, esta es mostrada en la consola del Wireshark de modo que en este punto se tiene acceso a todos los paquetes que fueron capturados por ejemplo en la interfaz eth0. La figura 41 muestra dicha consola de información.

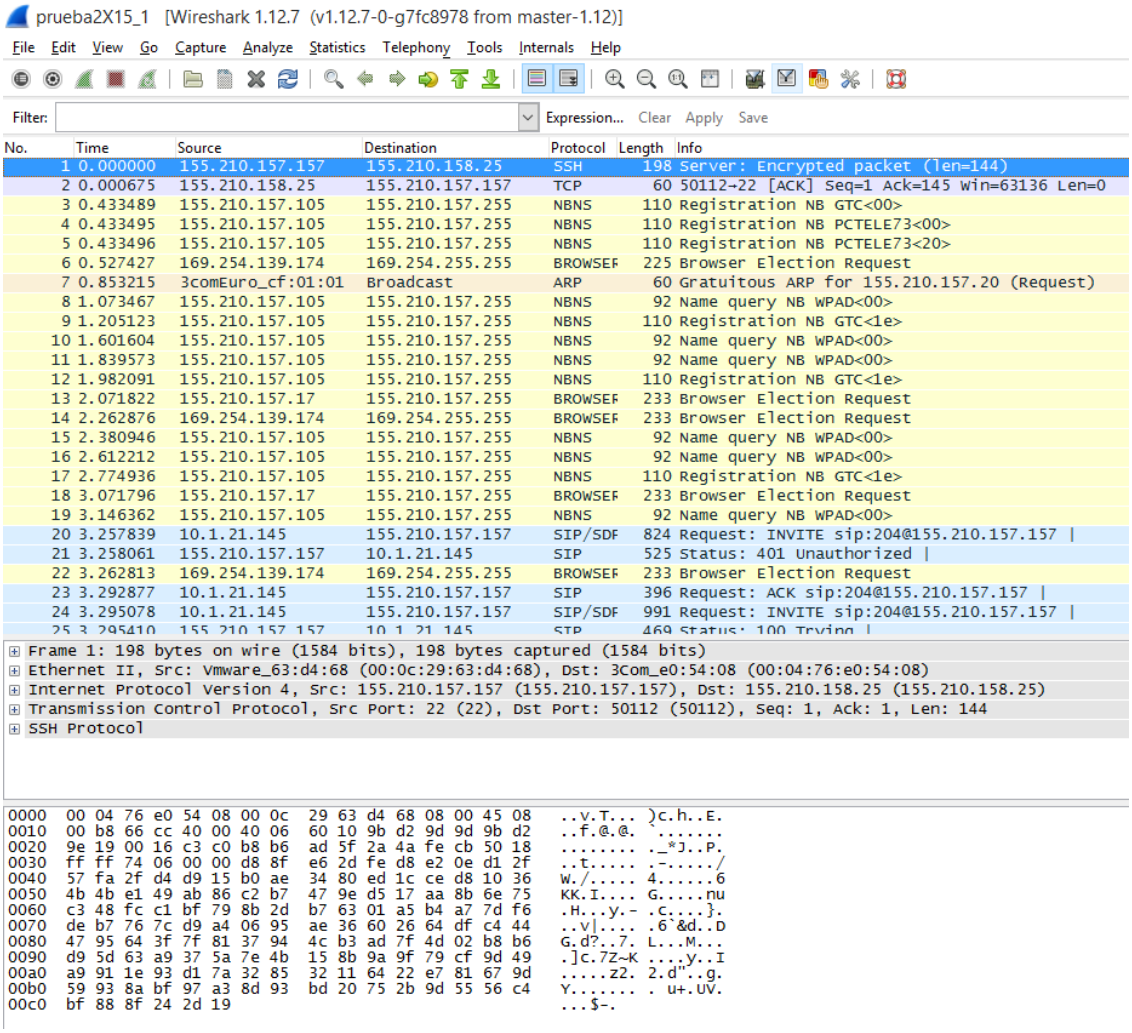


Figura 41 Visualización de la información capturada.

Ya que se tiene la información se pueden hacer varias cosas en dependencia de lo que se desee. Se puede desde mostrar la información de señalización de llamada de VoIP mediante un diagrama en flujo, hasta el cálculo del ancho de banda consumido por determinada cantidad de paquetes después de aplicar los filtros necesarios.

Para mostrar el flujo de señalización de llamadas basta con ir a las opciones de VoIP Call, incluidas en el menú superior llamado Telephony. La figura 42 muestra dicho acceso.

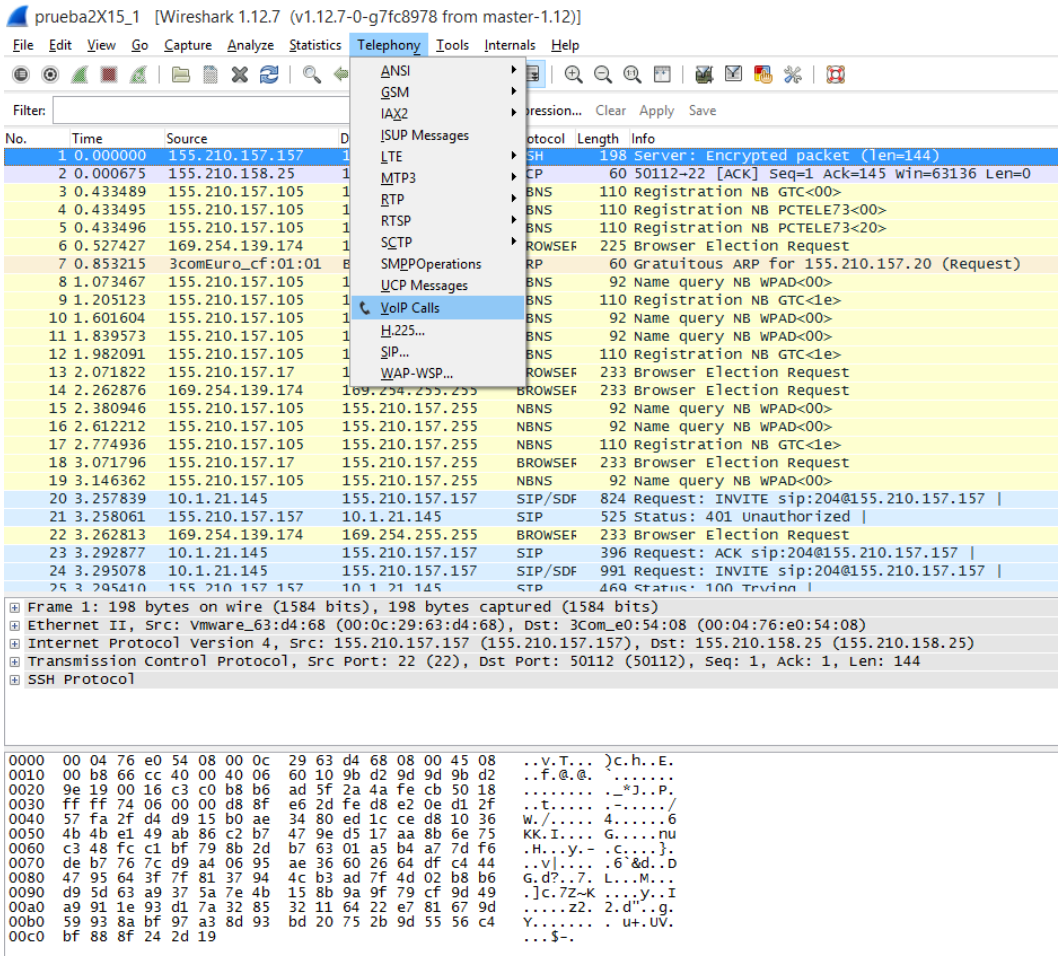


Figura 42 Acceso a opciones de VoIP.

Una vez que se selecciona el submenú VoIP calls la llamada de VoIP es mostrada en pantalla como lo evidencia la figura 43.

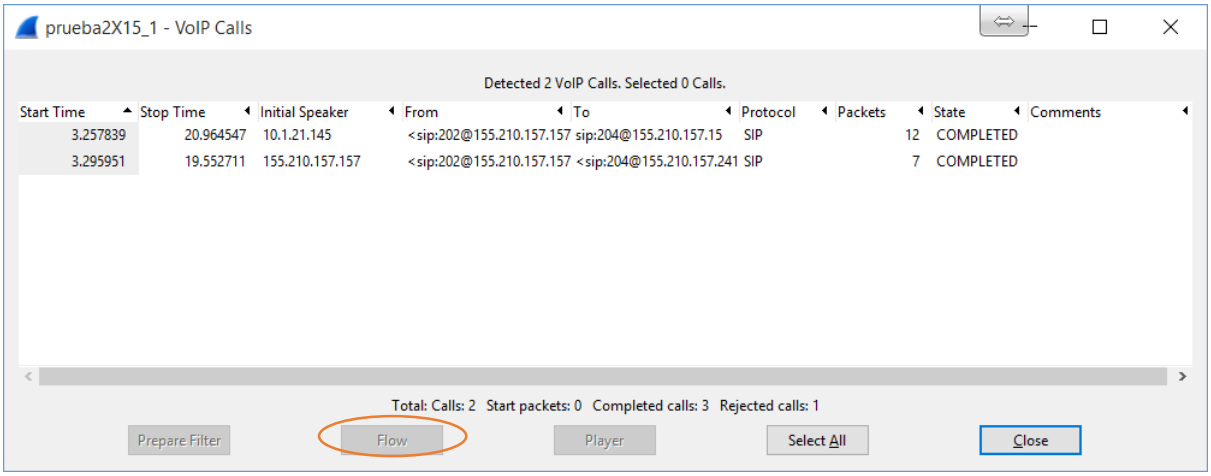


Figura 43 Detección de una llamada VoIP.

Seleccionando la llamada que se desea procesar y dando click en el botón Flow que muestra la figura 43 se obtiene el flujo de señalización SIP del que se hablaba anteriormente. La figura 44 muestra un ejemplo de mencionado flujo.

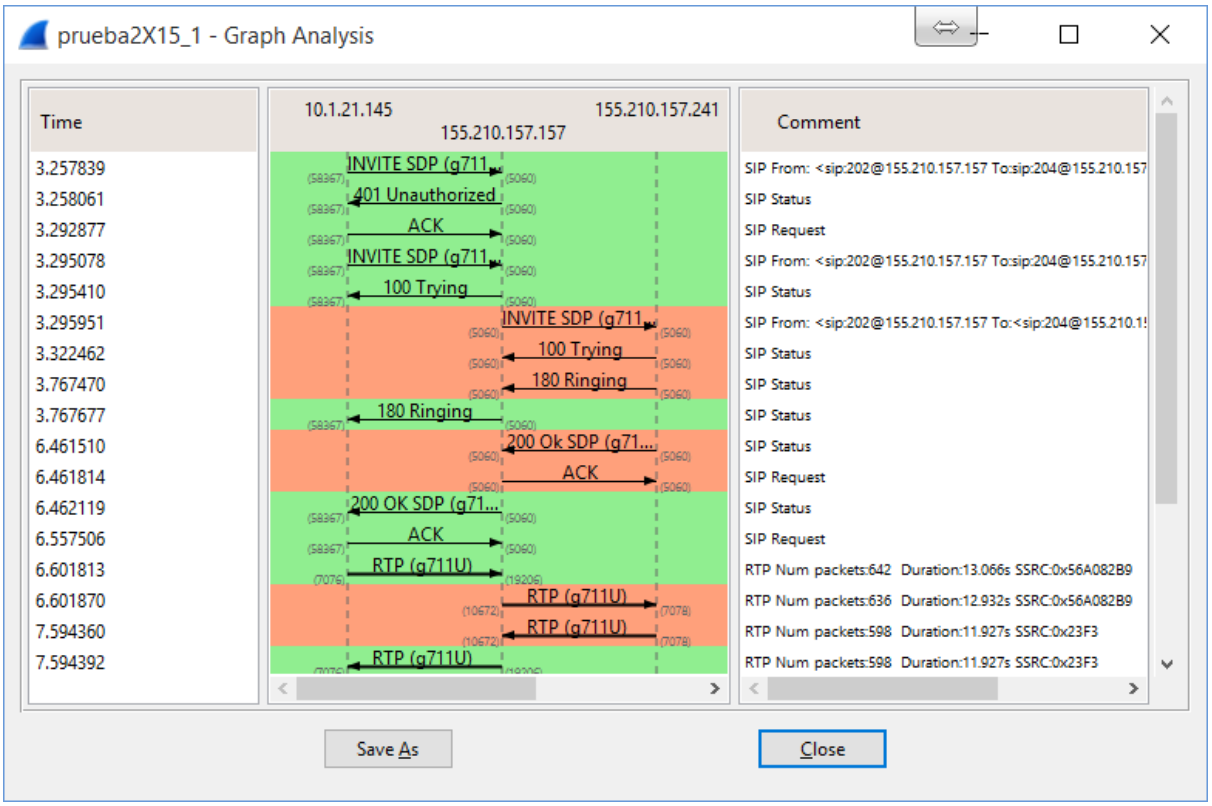


Figura 44 Ejemplo de flujo de una llamada de VoIP mediante Wireshark.

Para el cálculo del ancho de banda basta con filtrar la información que se tiene e ir al menú Statistic y luego al submenú Summary, dónde se obtendrá la información deseada. Si por ejemplo se desea calcular el ancho de banda consumido en un sentido de la comunicación se filtra poniendo rtp para que muestre solo los paquetes rtp dentro de todos los paquetes de información, seguido se pone el nexo and para agregar otra regla de filtrado y se coloca el filtro ip.src seguido por dos signos = que indicarán la asignación del IP de la fuente de los datos.

El filtro quedaría estructurado de la siguiente forma:

rtp and ip.src==xxx.xxx.xxx.xxx

Si se desea además ser más específico y filtrar la información dejando solo la información que va destinada a un puerto con el objetivo de tener nada más paquetes o de audio o de video, se le agrega a la sentencia anterior el filtro udp.port seguido de dos signos = y el número de puerto. Quedaría de la siguiente manera.

rtp and ip.src==xxx.xxx.xxx.xxx and udp.port==xxxx

La figura 45 muestra un ejemplo de la aplicación del filtro anterior descrito.

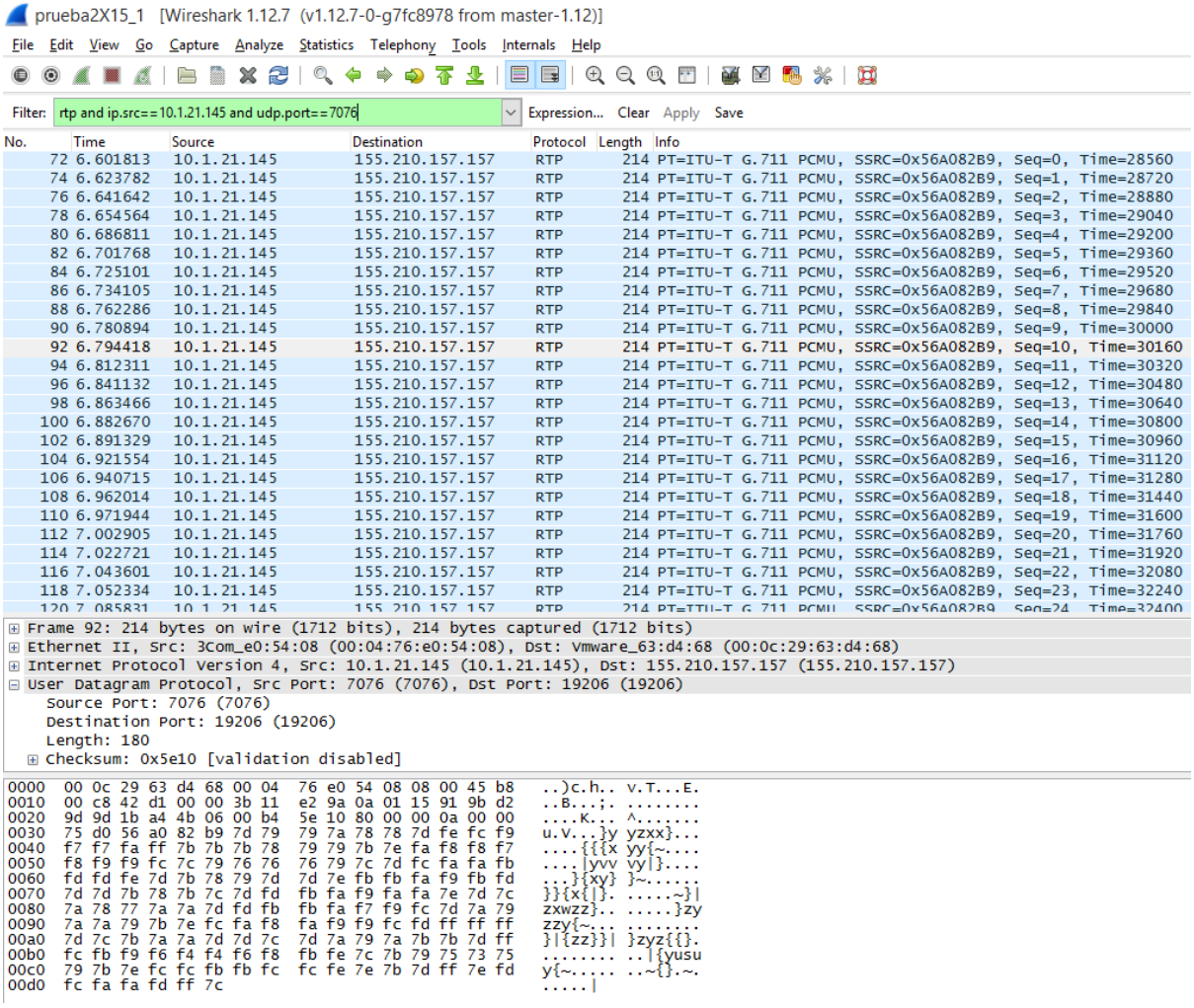


Figura 45 Ejemplo de filtro de información de multimedia.

Como se había explicado se puede saber el ancho de banda que consumen los paquetes una vez que se ha filtrado los paquetes accediendo a los menús convenientes. La figura 46 muestra la información al respecto.

Se puede ver claramente el ancho de banda consumido por los paquetes de audio. La señalización en rojo en la figura 46 muestra que los paquetes filtrados consumen un ancho de 0.084Mbps o lo que es lo mismo 84kbps.

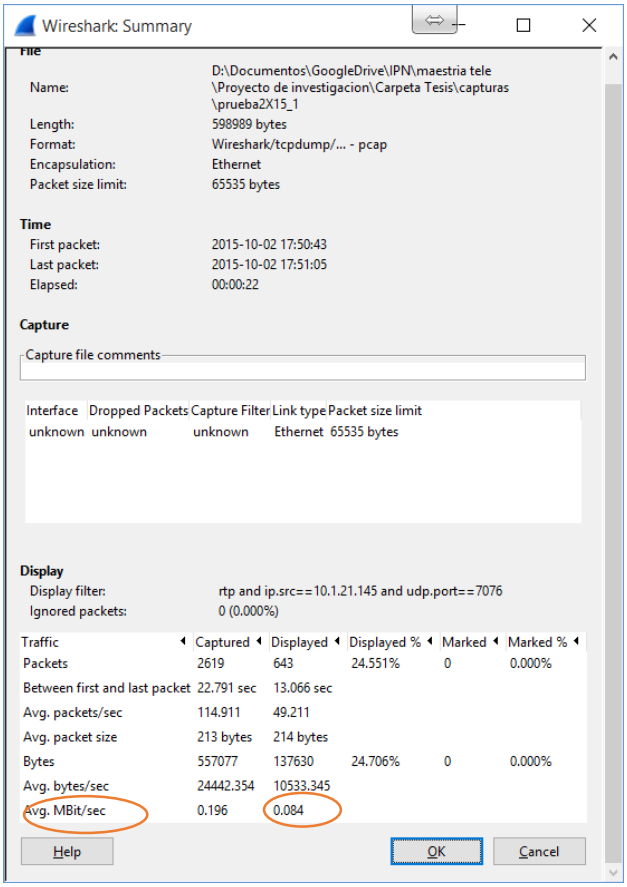


Figura 46 Cálculo del ancho de banda de los paquetes filtrados.

Además con el Wireshark se puede visualizar el consumo del ancho de banda en una gráfica. Para esto se puede acceder desde el menú Statistics y luego accediendo al menú IO Graph tal como muestra la figura 47.

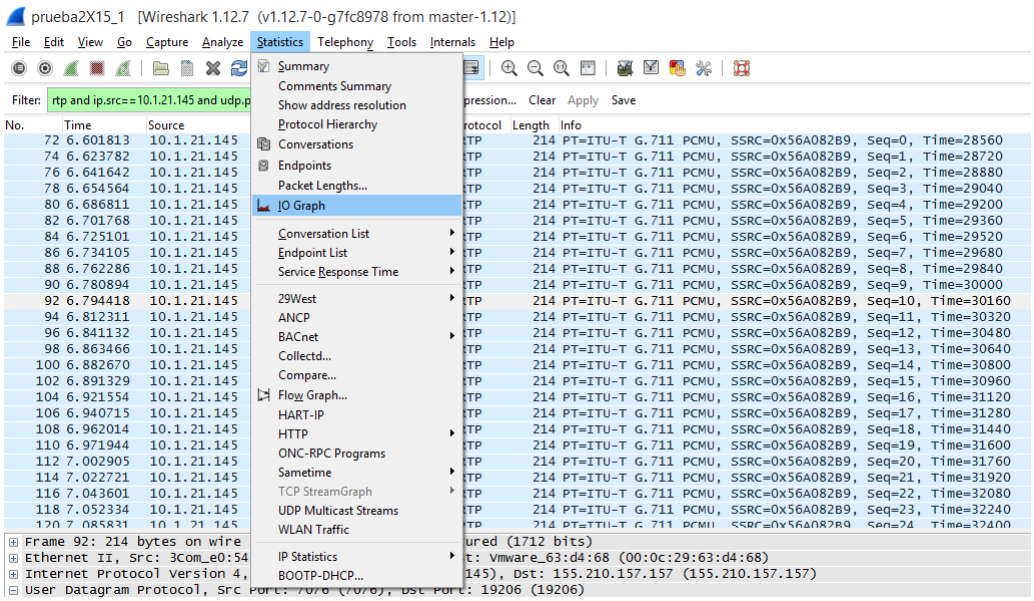


Figura 47 Acceso a gráficos de anchos de banda.

Al acceder a la gráfica de ancho de banda, automáticamente se muestra todo el tráfico capturado. Sin embargo se puede graficar en colores diferentes determinada cantidad de paquetes según lo que se quiera. La figura 48 muestra un ejemplo de donde se muestra todo el tráfico capturado con respecto al tráfico después de aplicar el filtro del ejemplo descrito en líneas arriba.

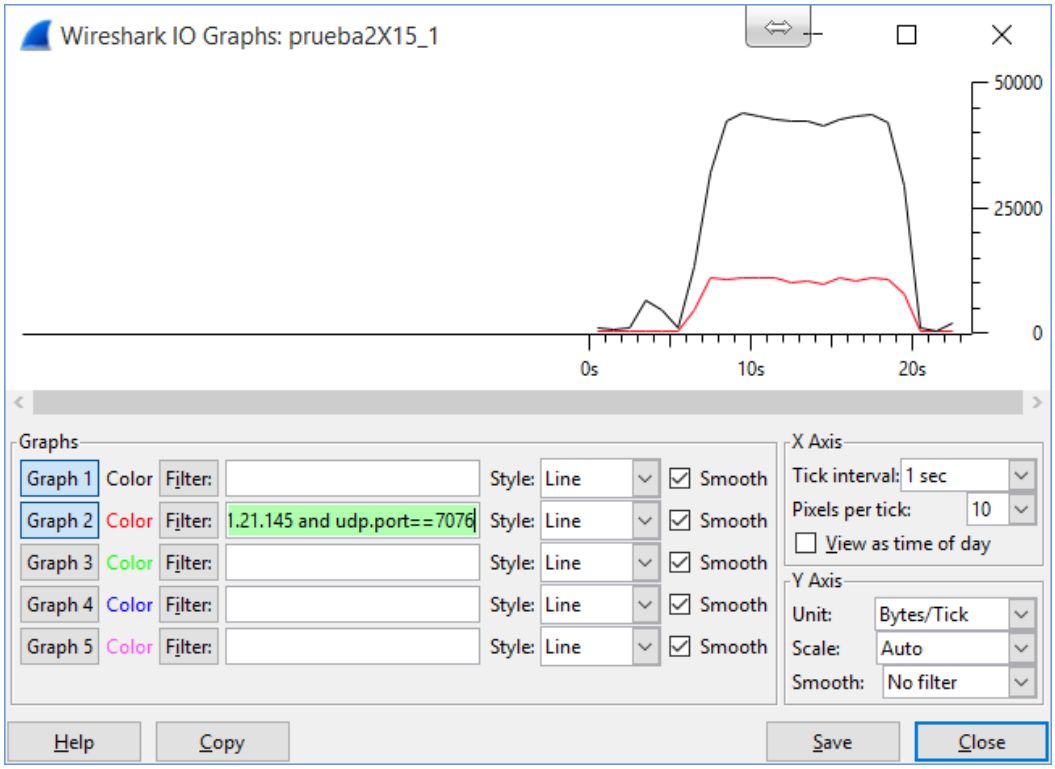


Figura 48 Gráfico de ancho de banda con filtro aplicado.

3.2. Conclusiones

Se pudo demostrar que mediante la unificación de los softwares Tcpdump y Wireshark se puede conformar una herramienta muy útil para realizar mediciones de consumos de anchos de bandas. Son muchas las aplicaciones que tienen estos dos softwares para el análisis de tráfico. Estos métodos serán usados en capítulos posteriores para la validación, análisis y obtención de resultados de esta tesis.

Capítulo 4.

“Diseño del Banco de pruebas”

4.0. Introducción

En el siguiente capítulo se diseñará un banco de pruebas para probar la funcionalidad del Sistema de VoIP propuesto. Mediante el Wireshark, software explicado en el capítulo anterior, se harán capturas que dejarán evidenciado la presencia del protocolo SIP y SRTP. Dichas capturas serán el medio para validar que la información está protegida mientas viaja por la red. Además la información recopilada en dicho banco de pruebas podrá ser usada para un análisis de consumo de ancho de banda en el posterior capítulo.

4.1. Diseño de un escenario real de VoIP y de un escenario de laboratorio para realizar de pruebas.

A modo general y elaborando un esquema donde se imite la realidad, se ha creado un escenario donde convergen distintos tipos de redes, como red FasthEthernet, WiFi y distintas redes móviles como 3G, 4G y LTE. La idea es hacer que a partir de equipos terminales registrados en un servidor de VoIP Asterisk, se puedan realizar llamadas entre los mismos. Además se puede tener más de un servidor Asterisk donde usuarios de un servidor puedan comunicarse con usuarios de otro servidor independientemente de donde se encuentren localizados. La Figura 49, muestra lo anterior descrito.

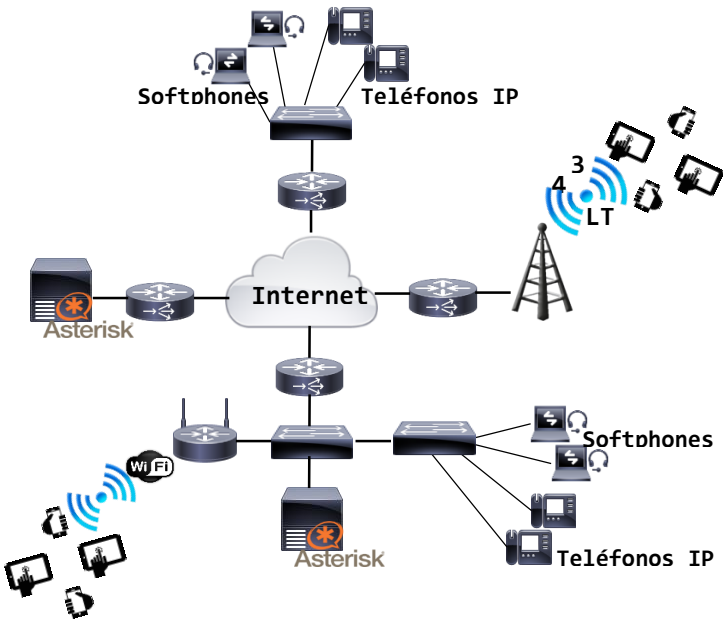


Figura 49 Esquema general de Sistema de VoIP con acceso desde diferentes redes.

A partir de la Figura 49 se quiere elaborar un ambiente de laboratorio de modo que se puedan realizar pruebas controladas y obtener información para el análisis posterior. Es por eso que se ha decidido conformar otro esquema a partir de elementos fundamentales que serán descritos a continuación.

La Figura 50 muestra los elementos implicados y el diseño del escenario de laboratorio que se utilizará para realizar pruebas. A modo general, el escenario está conformado por una PC Física que tiene dos interfaces: FastEthernet 1 y FastEthernet 2 que estarán conectados a un Hub. Dentro de la PC Física se encuentran corriendo dos máquinas virtuales: PBX157 y PBX158 que cada una contiene tres interfaces virtuales: ETH0, ETH1 y ETH2. Las interfaces de cada máquina virtual fueron configuradas de tal forma que siempre se tenga que salir al exterior entiéndase al hub para comunicarse entre ellas. Los ips de cada una interfaces fueron distribuidos de la siguiente forma: Ver Tabla 6 y Tabla 7

PBX157

Tabla 3 Interfaces PBX157 y sus IPs

ETH 0	158.210.157.157
ETH 1	192.168.2.157
ETH 2	192.168.7.157

PBX158

Tabla 4 Interfaces de la PBX158 y sus IPs

ETH 0	158.210.157.158
ETH 1	192.168.2.158
ETH 2	192.168.7.158

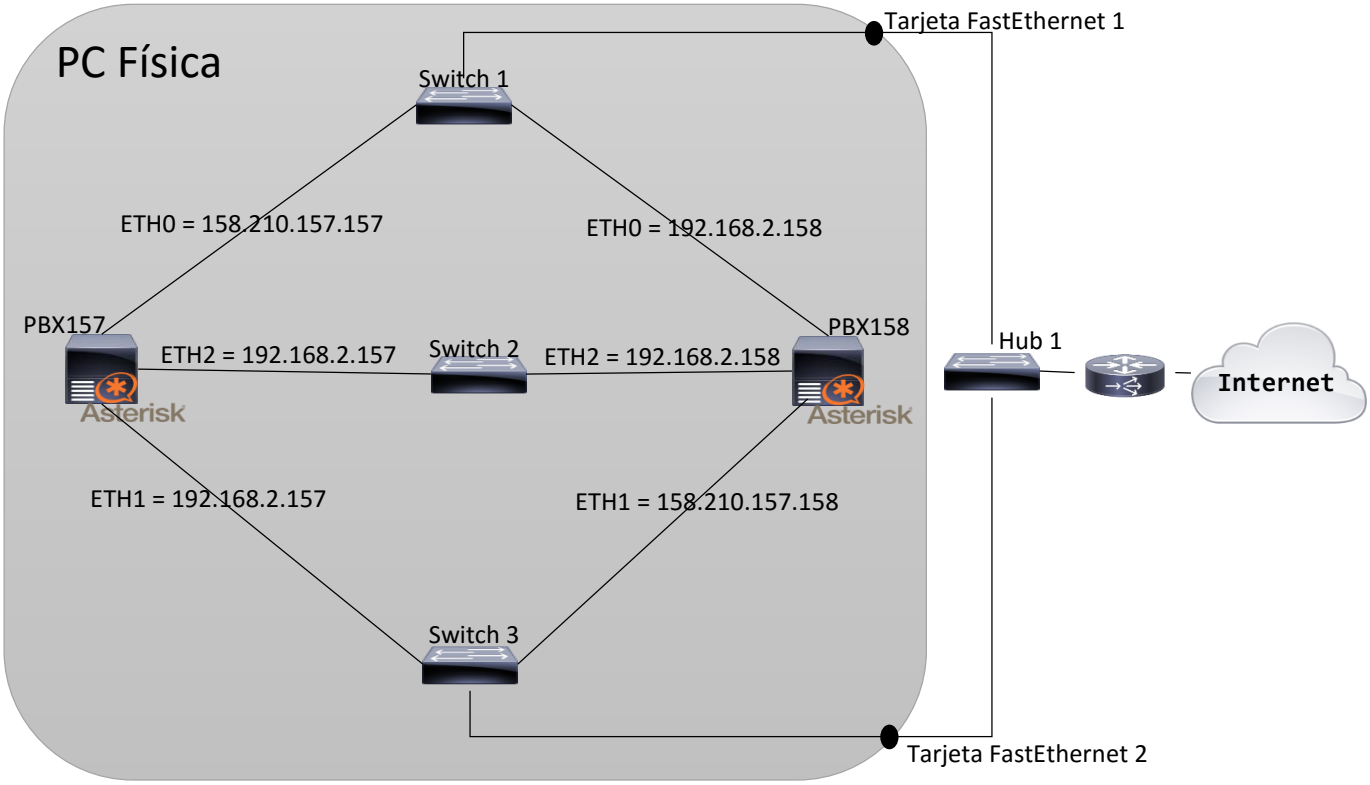


Figura 50 Diagrama de Laboratorio

La figura 50 muestra un esquema de laboratorio que simula la figura 49.

El hecho de que se hayan configurado los IPs de esta forma tiene el propósito de simular el esquema mostrado en la Figura 49, donde se ve que por ejemplo, los servidores Asterisk se encuentran cada uno en redes distintas y con la nube de internet en medio.

A continuación se muestran la Figura 51 y Figura 52 que son las consolas de cada uno de los servidores PBX157 y PBX158 con sus interfaces y los IPs configurados.

```
[root@asterisk157 diecpbx]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0C:29:63:D4:68
          inet addr:155.210.157.157  Bcast:155.210.157.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe63:d468/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:455994 errors:0 dropped:0 overruns:0 frame:0
          TX packets:87650 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:43856531 (41.8 MiB)  TX bytes:11797201 (11.2 MiB)

eth1      Link encap:Ethernet  HWaddr 00:0C:29:63:D4:72
          inet addr:192.168.2.157  Bcast:192.168.2.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe63:d472/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:361735 errors:0 dropped:0 overruns:0 frame:0
          TX packets:40 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:30352041 (28.9 MiB)  TX bytes:2664 (2.6 KiB)

eth2      Link encap:Ethernet  HWaddr 00:0C:29:63:D4:7C
          inet addr:192.168.7.157  Bcast:192.168.7.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe63:d47c/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:116 errors:0 dropped:0 overruns:0 frame:0
          TX packets:26 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:7036 (6.8 KiB)  TX bytes:1488 (1.4 KiB)
```

Figura 51 Interfaces de PBX157

```
[root@asterisk158 diecpbx]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0C:29:44:9B:93
          inet addr:192.168.2.158  Bcast:192.168.2.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe44:9b93/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:359835 errors:0 dropped:0 overruns:0 frame:0
          TX packets:20 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:30196477 (28.7 MiB)  TX bytes:1376 (1.3 KiB)

eth1      Link encap:Ethernet  HWaddr 00:0C:29:44:9B:9D
          inet addr:155.210.157.158  Bcast:155.210.157.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe44:9b9d/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:457268 errors:0 dropped:0 overruns:0 frame:0
          TX packets:115720 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:45133997 (43.0 MiB)  TX bytes:32856069 (31.3 MiB)

eth2      Link encap:Ethernet  HWaddr 00:0C:29:44:9B:A7
          inet addr:192.168.7.158  Bcast:192.168.7.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe44:9ba7/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:116 errors:0 dropped:0 overruns:0 frame:0
          TX packets:17 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:7036 (6.8 KiB)  TX bytes:1110 (1.0 KiB)
```

Figura 52 Interfaces de PBX158

Los IPs 155.210.157.157 y 155.210.157.158 correspondientes a las interfaces ETH0 y ETH1 respectivamente, son IPs públicos. De esta manera se podrán ser alcanzados desde cualquier punto del internet.

4.2. Pruebas de funcionalidad de servicios con distintos códecs de audio y el códec de video Vp8.

Las siguientes pruebas tienen el objetivo de capturar el tráfico de llamadas realizadas para hacer un análisis posterior. Además se quiere evidenciar la utilización de distintos códecs contenidos en la plataforma implementada que demuestran lo versátil y adaptativo que puede ser el sistema en relación a la codificación de audios de llamadas. Se harán pruebas de audio con códecs de audio como: g711u, speex, gsm, opus y g722. En el caso de las pruebas de video se harán solo con un códec, el Vp8 debido a la tendencia actual de emplearlo para los servicios de multimedias y por la inestabilidad presentada por el sistema al utilizar códecs más antiguos como son el H263 y H264. Sin embargo se recomendará un estudio posterior, la posibilidad de poderlos incluir en el stack de códecs de video soportados por el sistema de video llamadas. Todas las pruebas desde la 7 hasta la 12, son realizadas en el mismo entorno y que se muestra en la siguiente Figura 53.

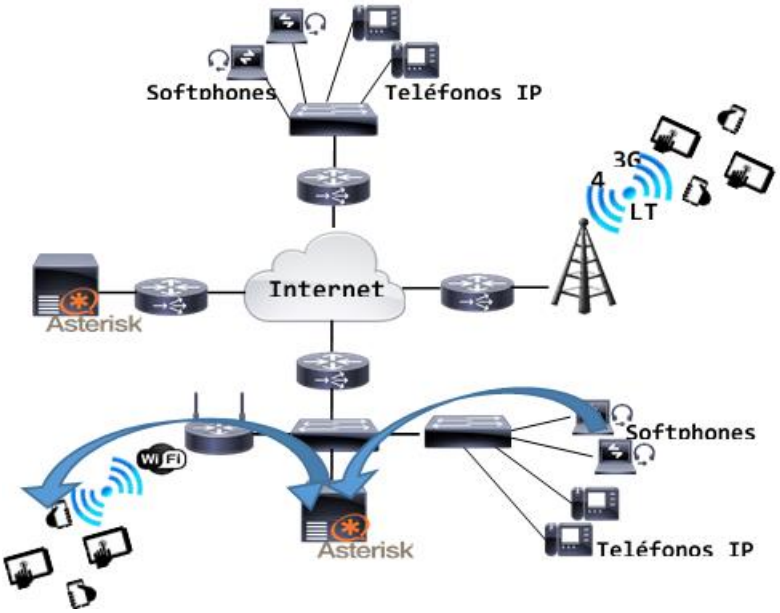


Figura 53 Entorno de llamada mediante una misma PBX.

4.2.1. Prueba 1. Realización de una llamada cifrada utilizando códec speex y Vp8.

Para esta realización y las posteriores no se exhibirán las configuraciones empleadas para el éxito de las pruebas debido a que son idénticas a las configuraciones explicadas con anterioridad a lo largo del capítulo. Sin embargo se hará alusión a la utilización de los distintos códecs según la prueba presente.

La figura 54 representa la identificación de una llamada de VoIP por el software WireShark una vez que se han capturados los paquetes durante una conversación entre dos clientes.

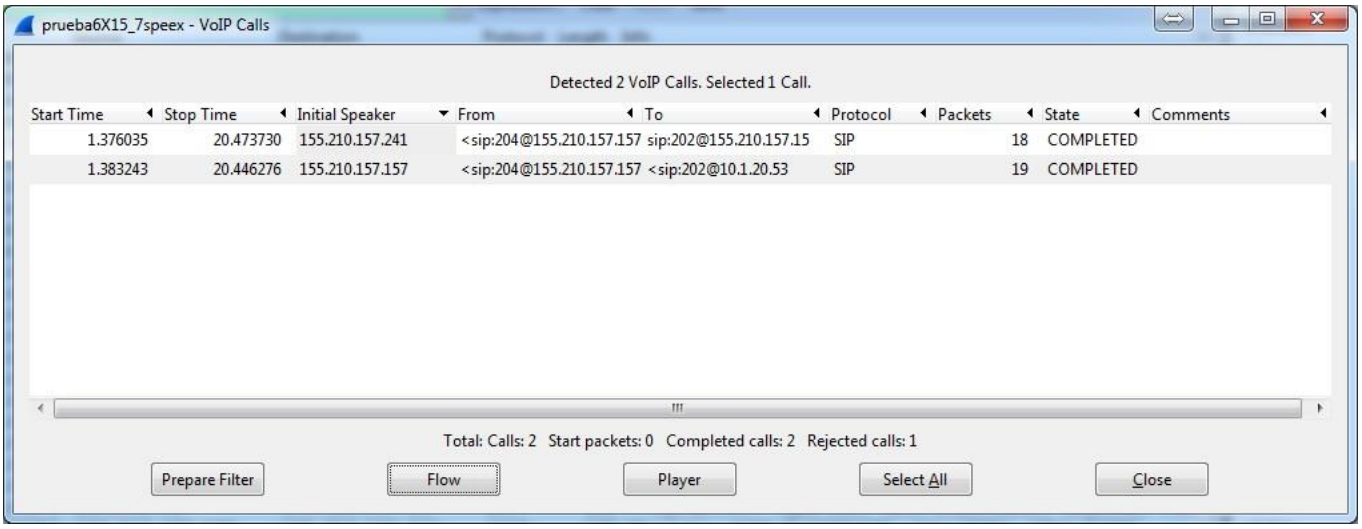


Figura 54 Detección de una llamada de VoIP por el software WireShark.

En la figura se puede ver una llamada realizada desde la extensión 204 a las 202 pertenecientes ambas a la PBX157.

En la figura 55 se puede ver el flujo SIP de señalización y dentro de las tramas SRTP el códec speex. También se pueden ver dentro de SRTP tramas que tienen información de video con el códec Vp8.

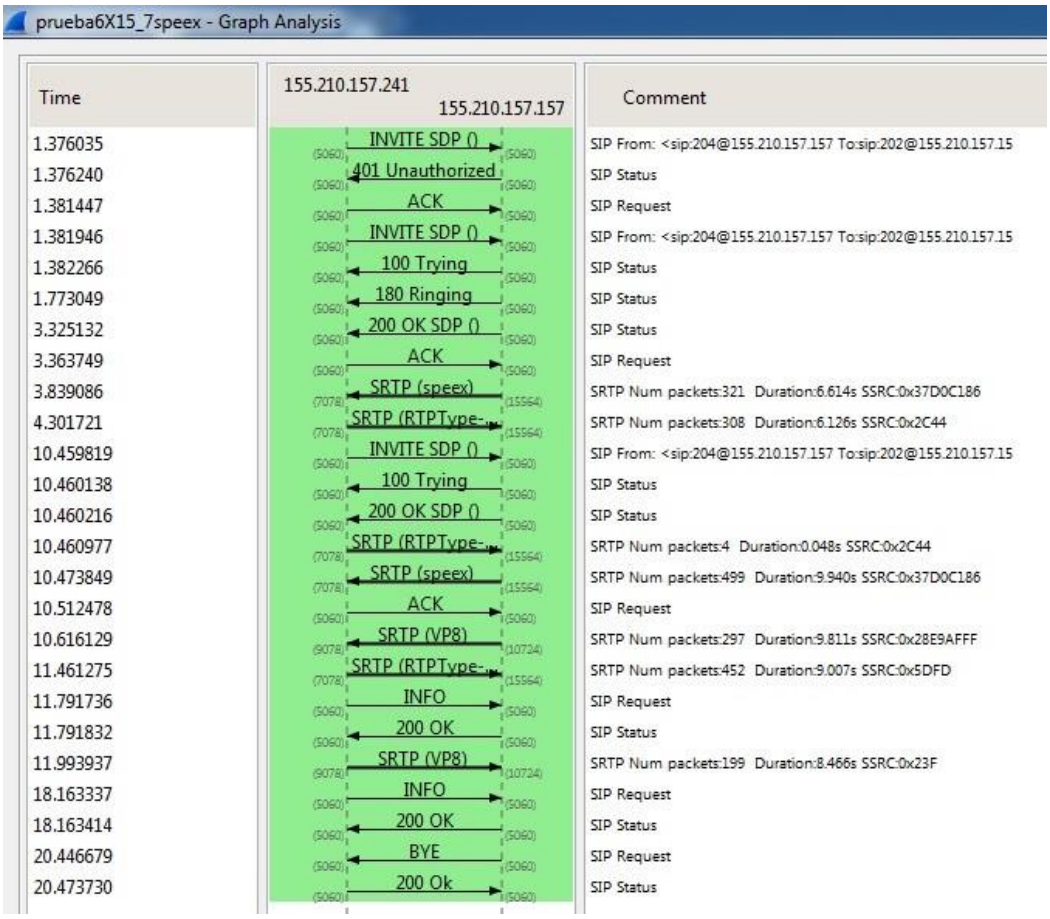


Figura 55 Diagrama de flujo SIP con códec speex y Vp8 cifrado.

El flujo representado en la anterior figura va desde el cliente con IP 155.210.1357.241 al servidor 155.210.157.157.

La próxima Figura 56 es el flujo SIP que va hacia el otro cliente que se encuentra en el IP 10.1.20.53. En dicha figura también se puede apreciar las tramas SRTP con los códecs de video Vp8 y de audio speex.

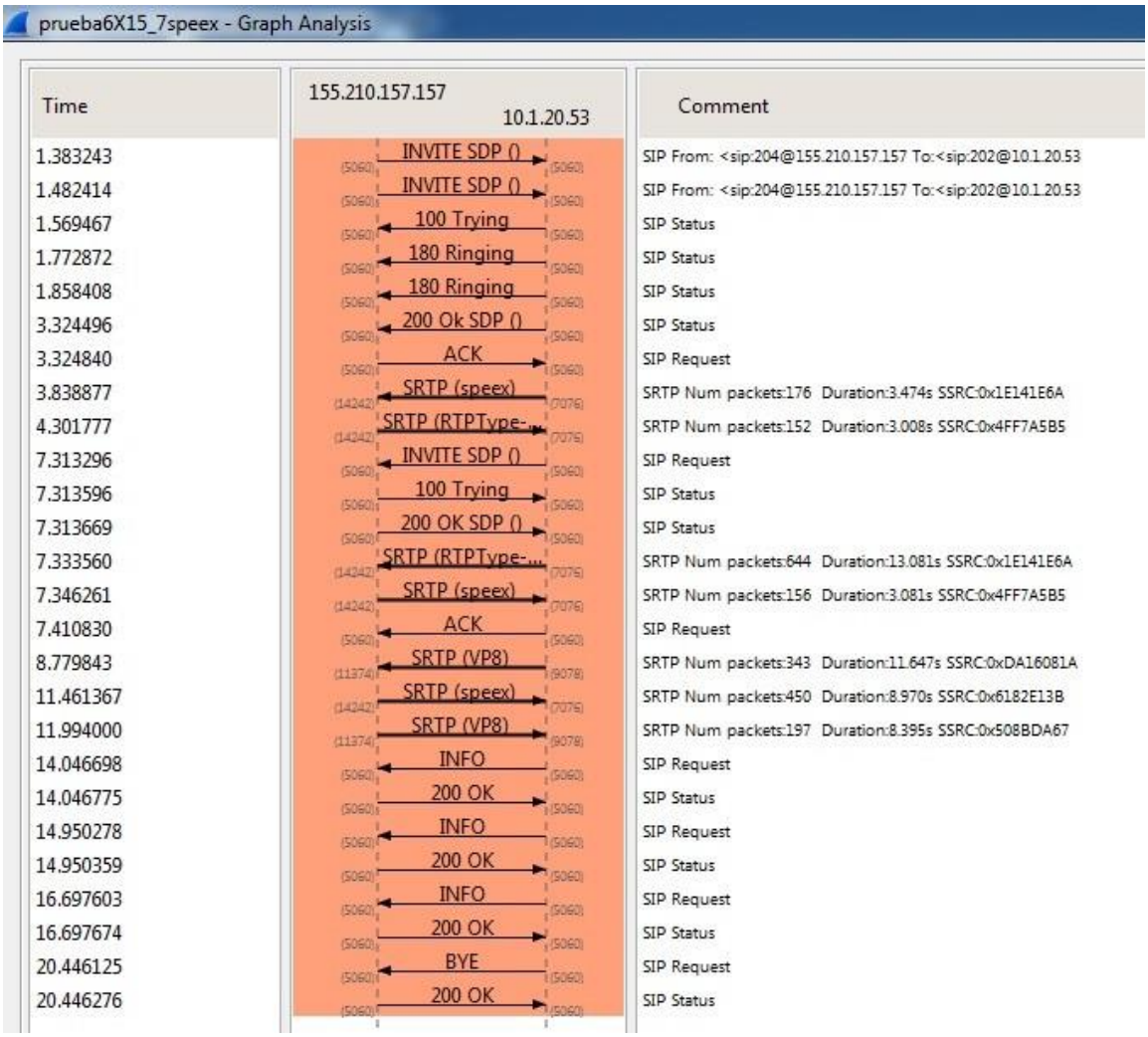


Figura 56 Flujo SIP de una llamada cifrada con códec speex y vp8.

4.2.2. Prueba 2 Realización de una llamada no cifrada utilizando códec speex y Vp8.

La siguiente prueba es idéntica a la anterior lo que sin usar cifrado. El único objetivo de esta prueba es demostrar que el consumo de ancho de banda con un códec específico, no varía al tener cifrado, o sea, el consumo del ancho de banda es el mismo. Sin embargo esos resultados serán tratados en el posterior capítulo, por lo que

sólo se presentará la prueba como parte del banco que se está conformando en este capítulo.

Como se muestra en la figura 57, la llamada como en casos anteriores ha sido detectada por el Wirehsark, una vez realizado este procedimiento, se puede sacar los flujos de señalización SIP para ver las tramas con los códecs configurados.

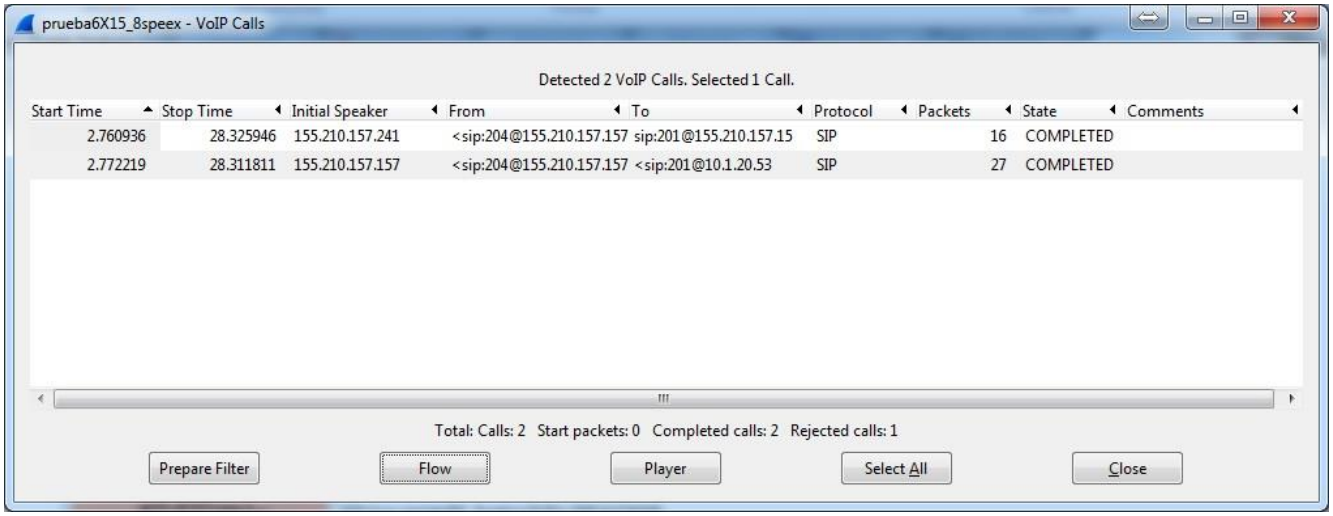


Figura 57 Detección de la llamada VoIP.

La siguiente Figura 58 muestra el flujo SIP entre el cliente con IP 155.210.157.241 y el servidor PBX157. Como se puede apreciar hay inexistencia de paquetes SRTP por lo que la comunicación no es cifrada.

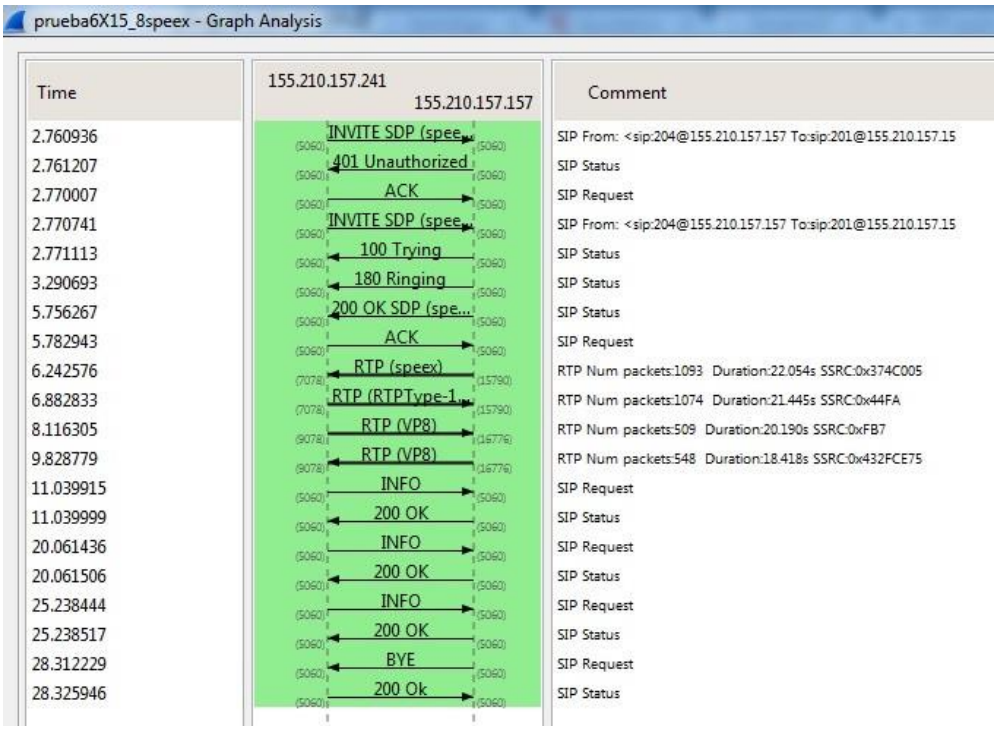


Figura 58 Flujo SIP de una llamada no cifrada con códec Vp8 y speex.

La siguiente Figura 59 es el flujo SIP pero desde el otro extremo del cliente o sea desde el IP 10.1.20.53 al servidor PBX157.

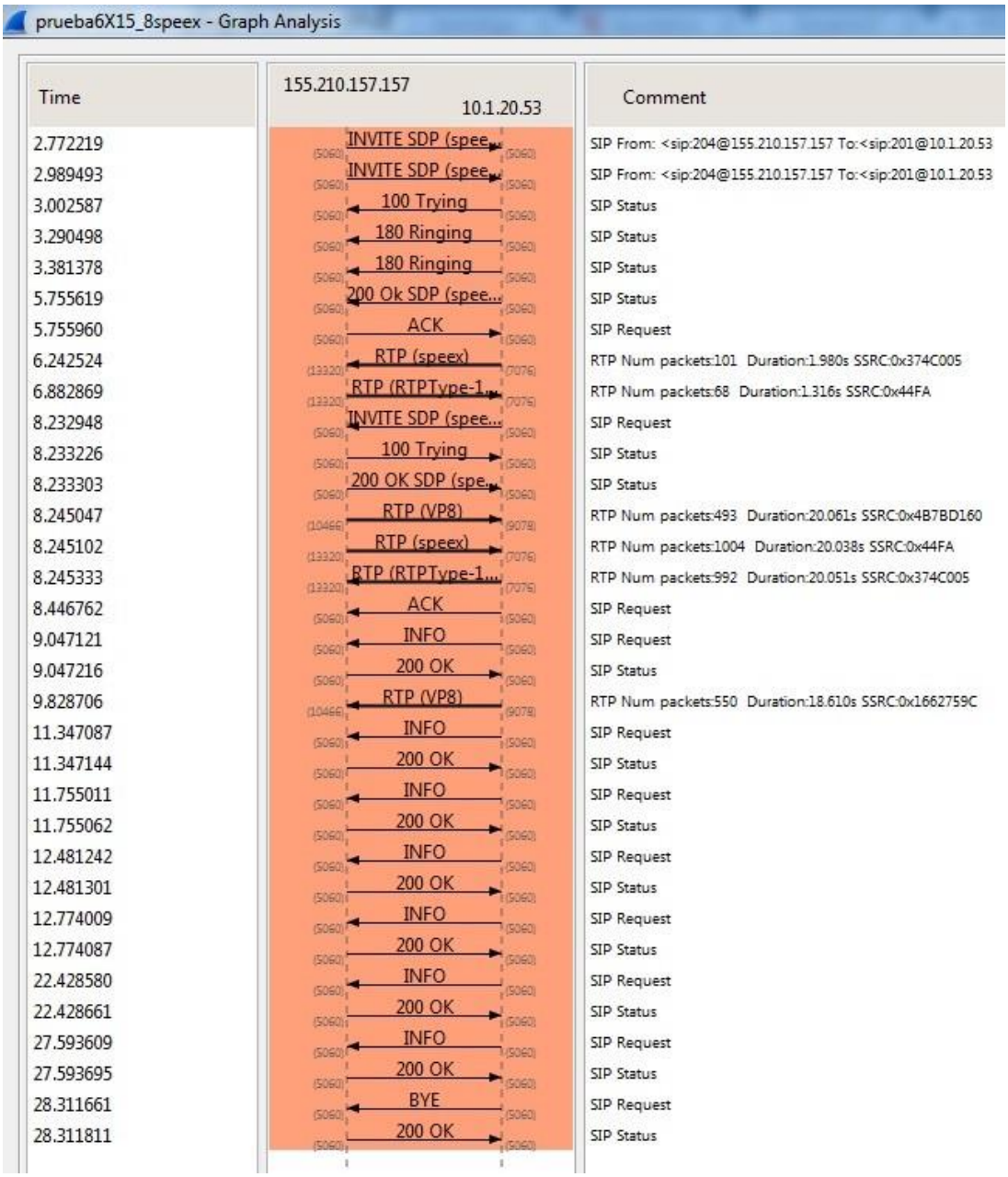


Figura 59 Flujo SIP de una llamada no cifrada con speex y vp8.

4.2.3. Prueba 3 Realización de una llamada cifrada utilizando códec GSM y Vp8.

En la presente prueba se ha configurado el servidor y los clientes para que soporten el códec GSM y el Vp8 que es el mismo que se ha manejado en todas las pruebas anteriores.

La siguiente figura 60 muestra la detección de la llamada por el software Wireshark. Se puede apreciar una llamada completada que va desde la extensión 204 a la 202 y los IPs de los clientes son 10.1.20.53 y 155.210.157.15. Ambas extensiones pertenecen al servidor PBX157

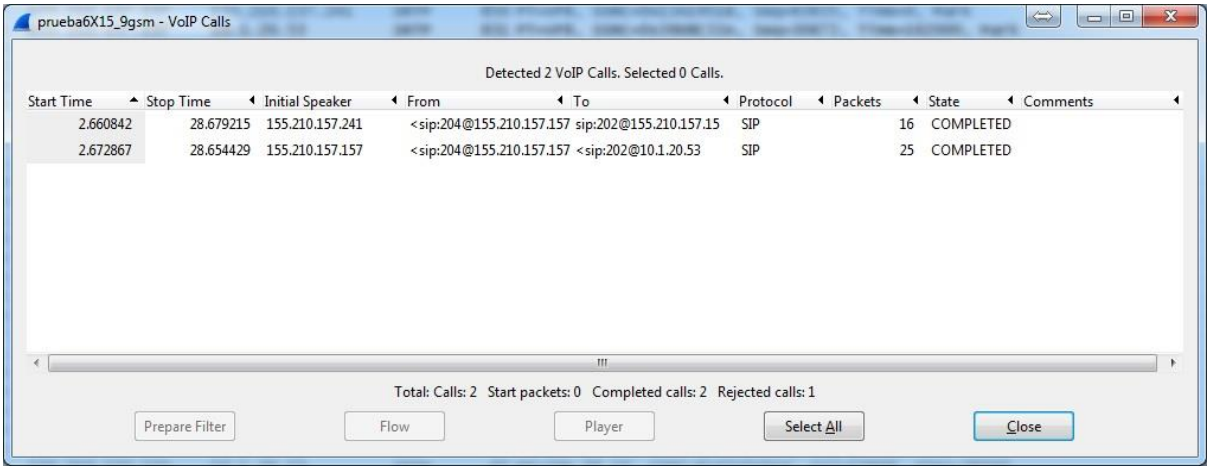


Figura 60 Detección de la llamada de VoIp por el Wireshark.

Una vez detectada la llamada se realizó la captura del flujo de señalización SIP donde se puede apreciar los códecs involucrados en esta prueba. La siguientes figuras 61 y 62 muestran dicha información**.**

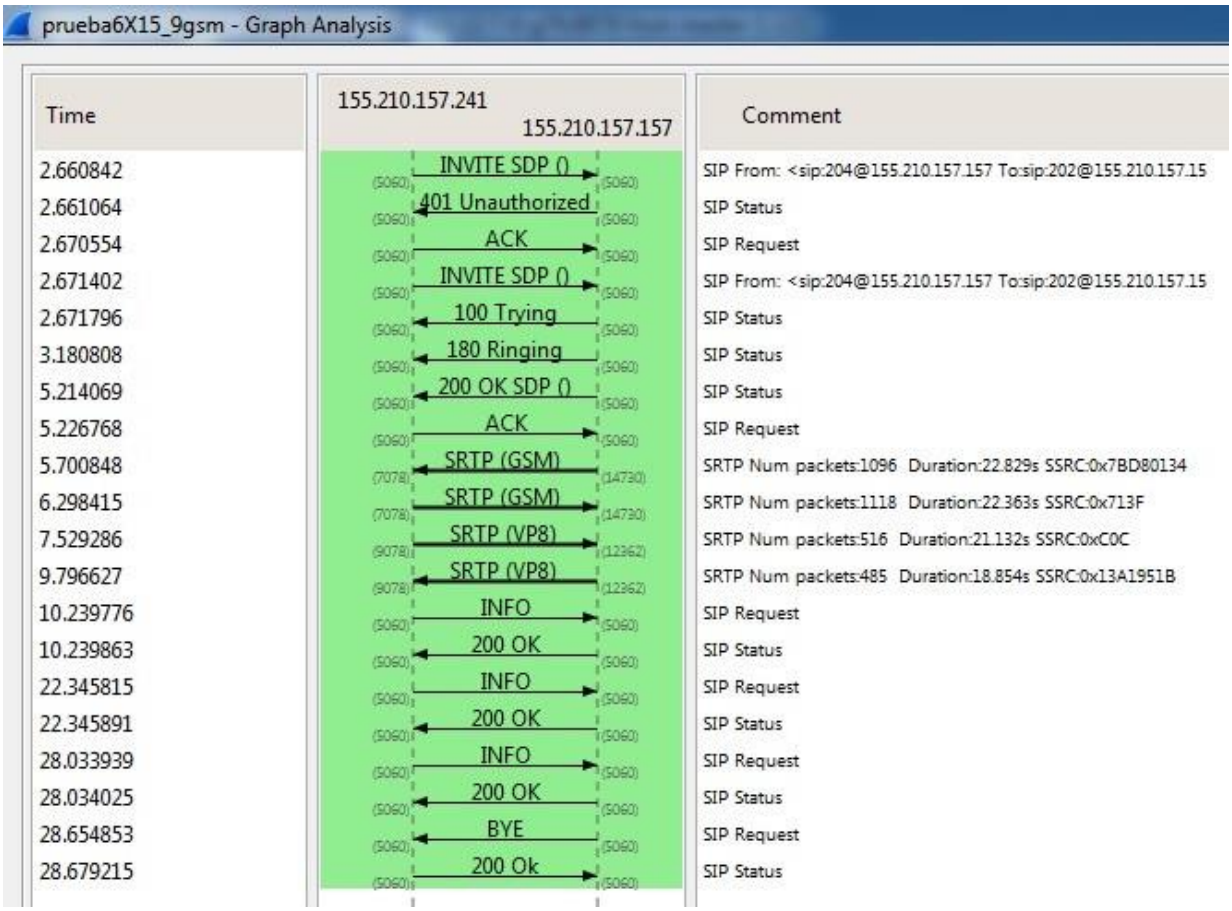


Figura 61 Flujo SIP de una llamada cifrada con GSM y Vp8.

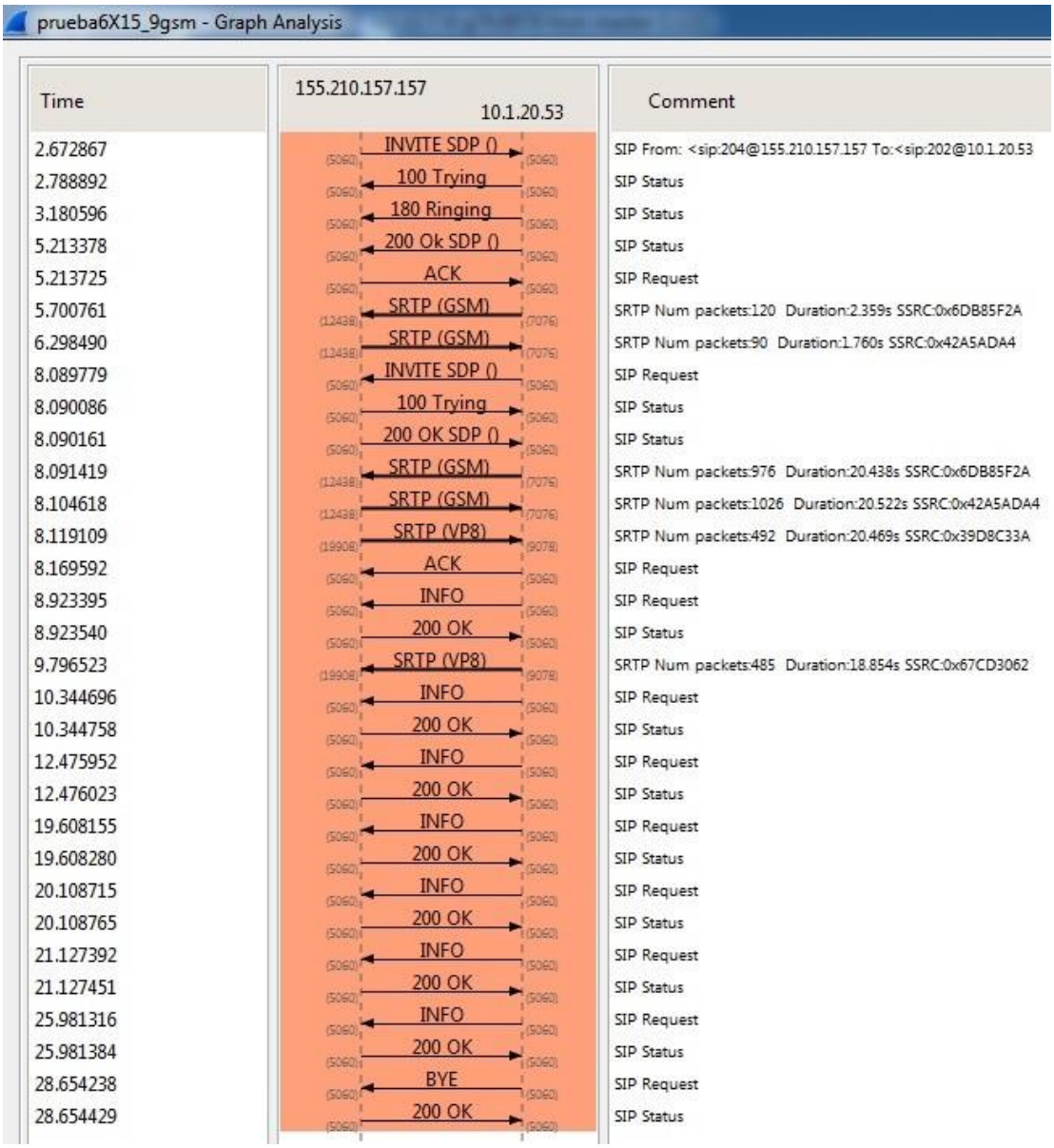


Figura 62 Flujo SIP de una llamada cifrada con GSM y Vp8.

4.2.4. Prueba 4 Realización de una llamada cifrada utilizando códec Opus y Vp8.

La siguiente prueba se trata de la realización de una llamada de video y de voz con los códecs Vp8 y opus.

La figura 63 representa la detección de la llamada mediante el software WireShark, donde se puede ver como desde el IP 155.210.157.241 desde la extensión 204 se realiza una llamada al IP 10.1.20.53 a la extensión 202 mediante el servidor PBX157.

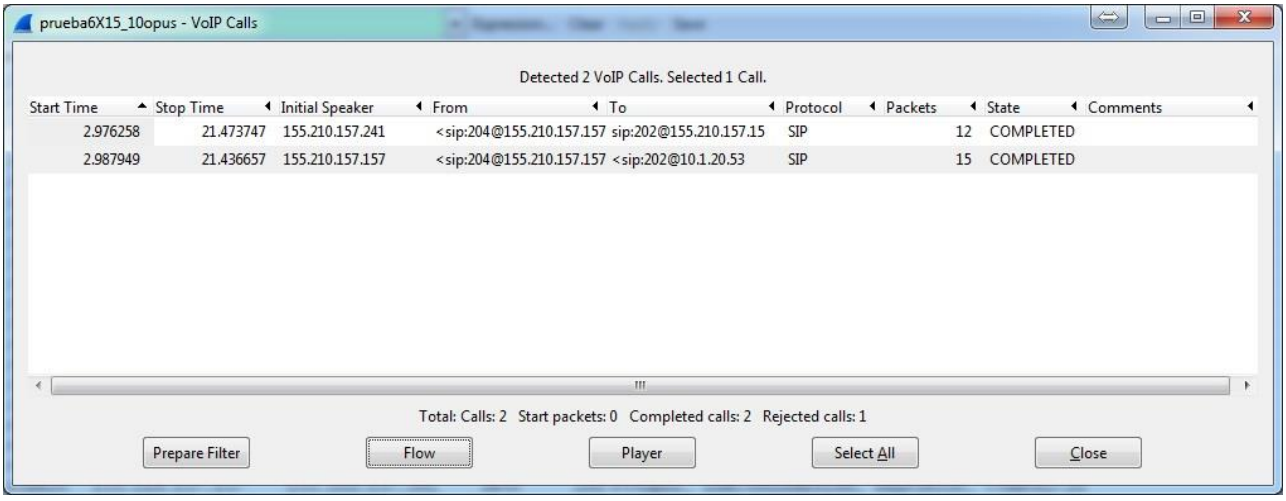


Figura 63 Detección de una llamada VoIP mediante Wireshark.

La próximas figuras 64 y 65 muestran el flujo de señalización donde se ven claramente los códecs empleados así como el protocolo SRTP de cifrado de los datos.

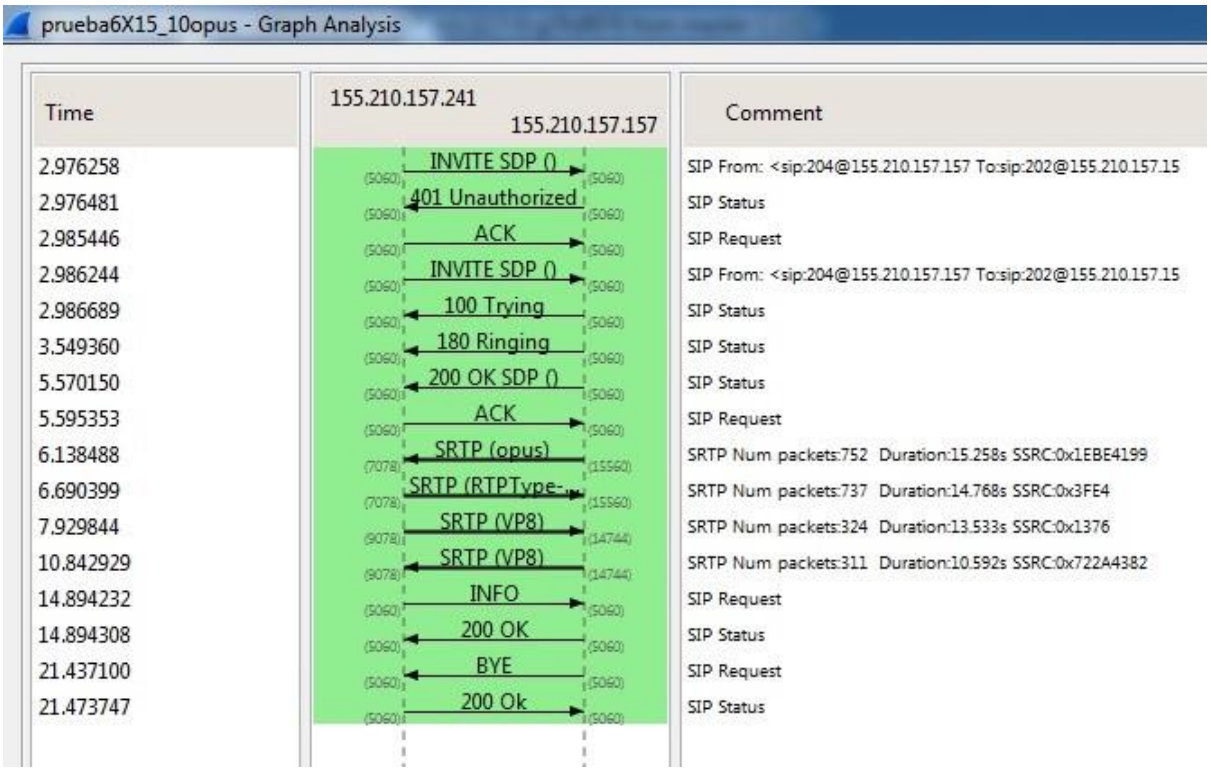


Figura 64 Flujo de señalización de una llamada cifrada con opus y Vp8.

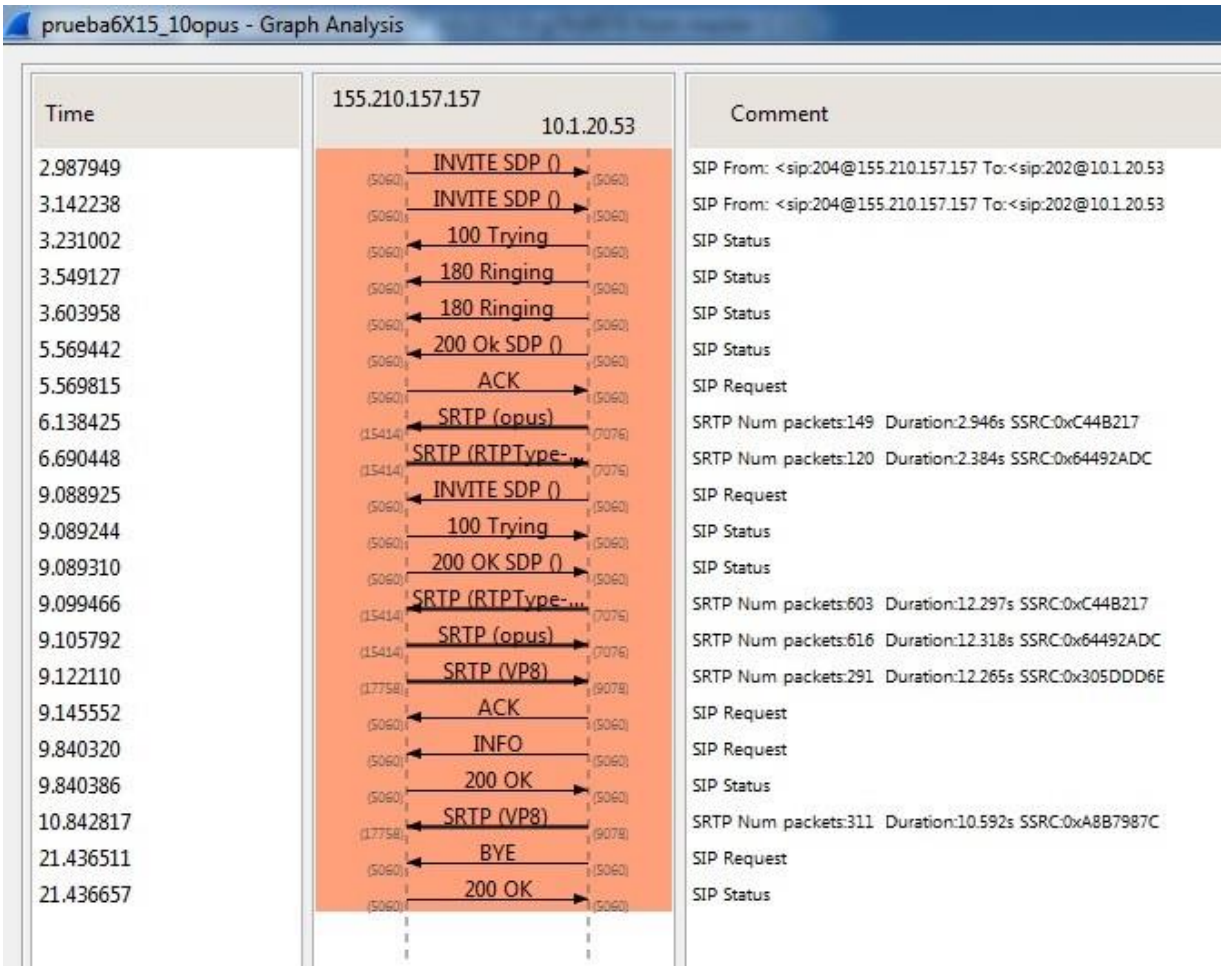


Figura 65 Flujo de señalización de una llamada cifrada con opus y Vp8.

4.2.5. Prueba 5 Realización de una llamada cifrada utilizando códec g.722 y Vp8

La próxima realización que se muestra fue hecha con los códecs g.722 y Vp8.

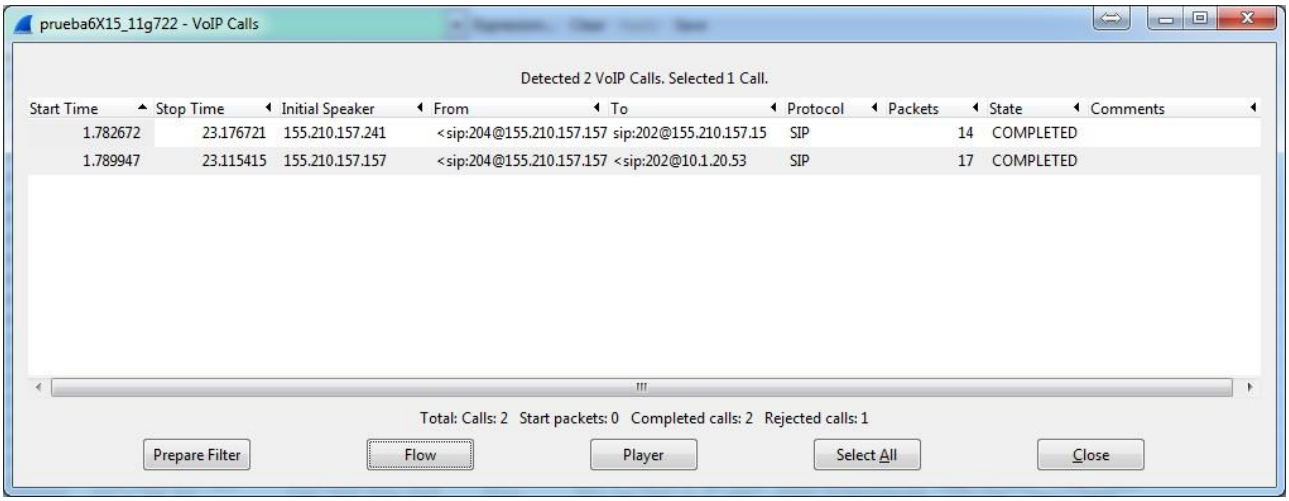


Figura 66 Detección de una llamada de VoIP mediante el Wireshark.

Como se puede apreciar en la figura 66, se detecta una llamada que es realizada desde la extensión 204 ubicada en el IP 155.210.157.241 hacia la extensión 202 que se encuentra en el IP 10.1.20.53 mediante la PBX157.

La siguientes figuras 67 y 68 muestran entonces el flujo de señalización SIP donde puede ser evidenciado los códecs empleados así como el protocolo de cifrado SRTP con el que se ha trabajado.

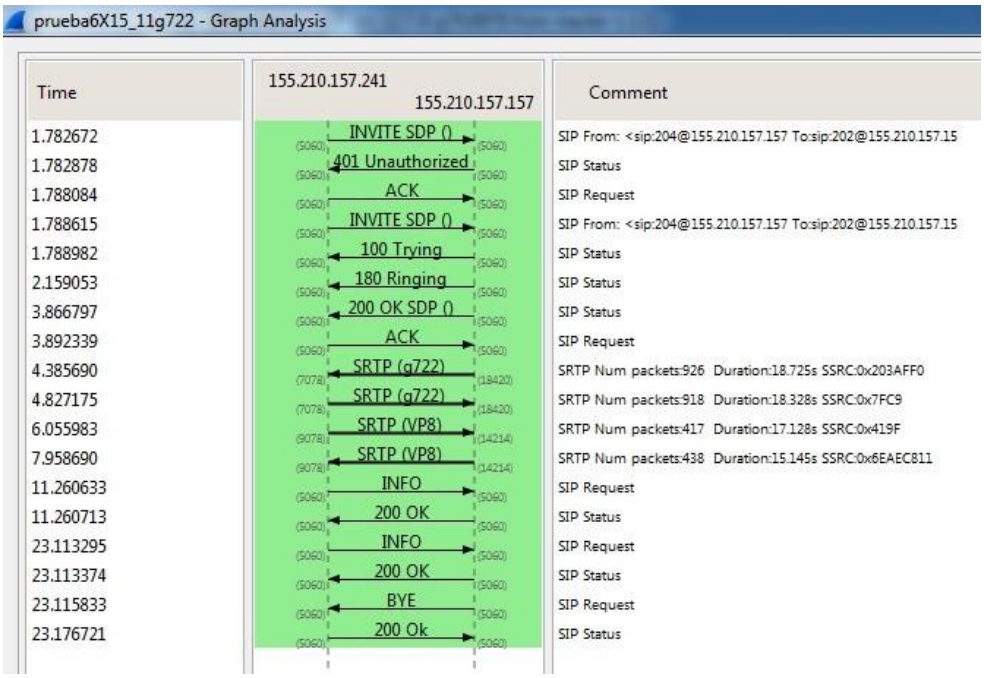


Figura 67 Flujo de señalización SIP de llamada cifrada con g722 y Vp8.

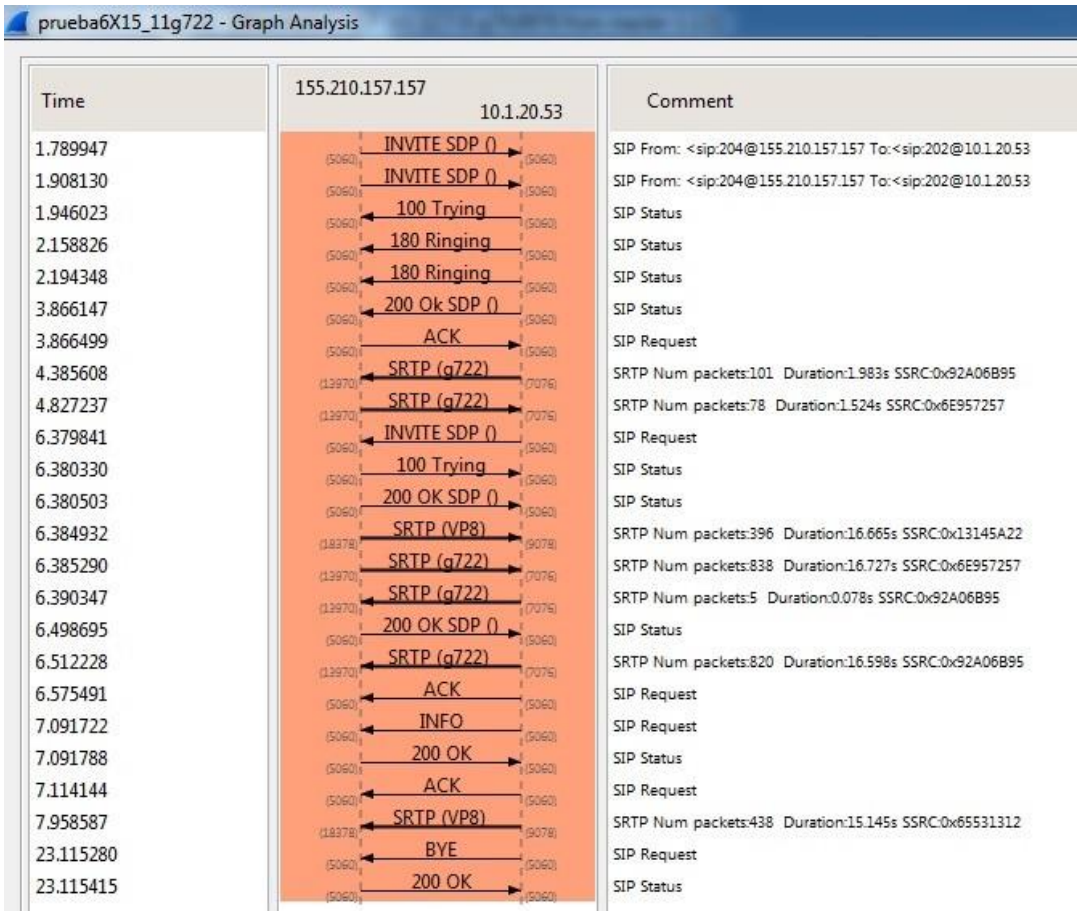


Figura 68 Flujo de señalización SIP de una llamada cifrada con g722 y Vp8.

4.2.6. Prueba 6 Realización de una llamada cifrada utilizando códec g.711 (uLaw) y Vp8.

La figura 69 muestra la detección de la llamada realizada desde el IP 10.1.23.183 con extensión 202 al IP 155.210.157.241 a la extensión 204 mediante la PBX157.

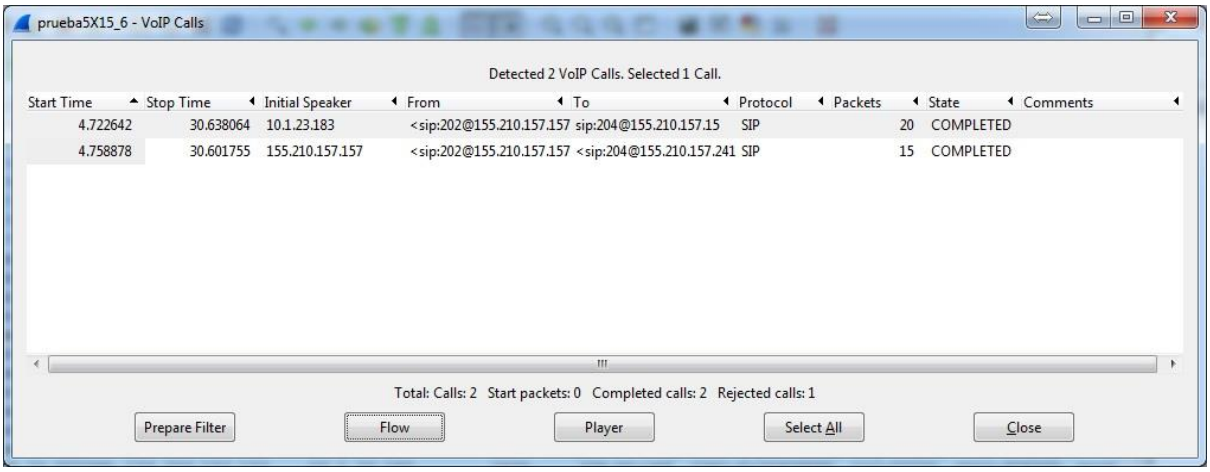


Figura 69 Detección de una llamada VoIP mediante Wireshark.

Las siguientes figuras 70 y 71 muestran el flujo de señalización SIP que evidencian la presencia del códec g711u y Vp8 dentro del protocolo de transporte seguro SRTP.

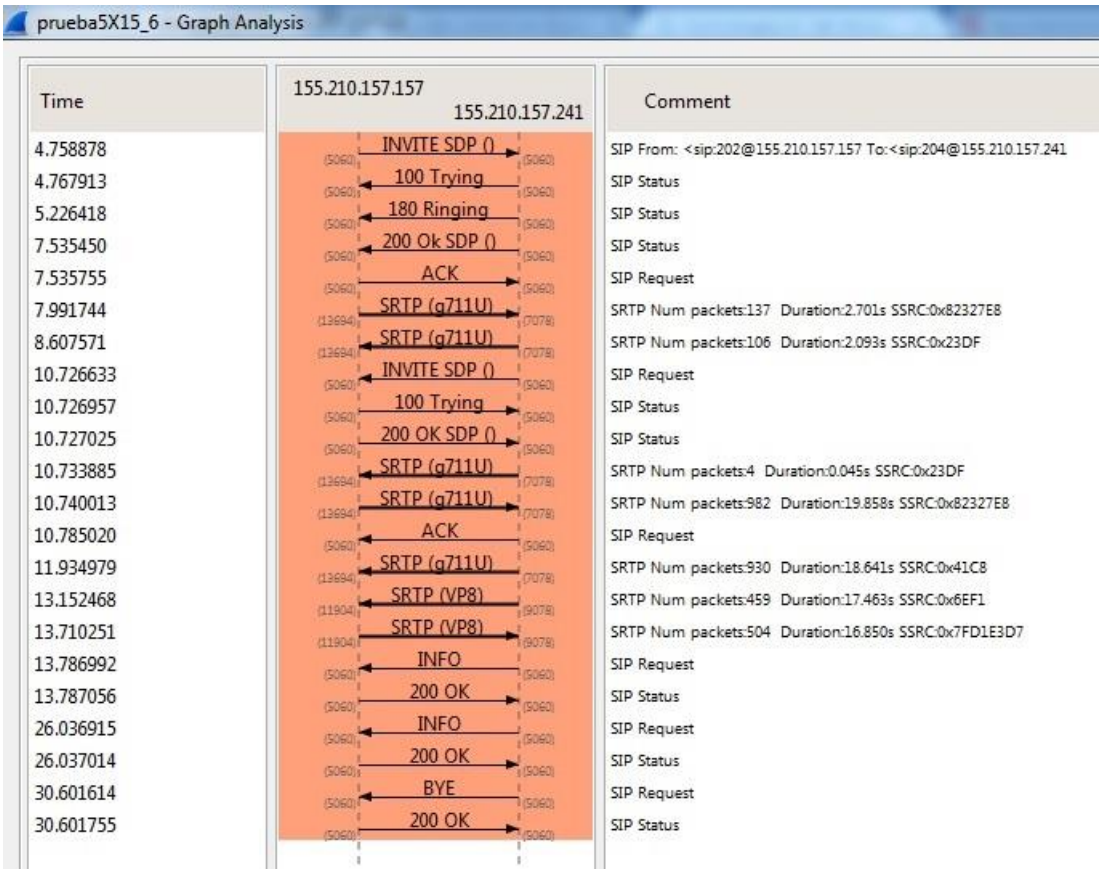


Figura 70 Flujo de señalización SIP con g711u y Vp8.

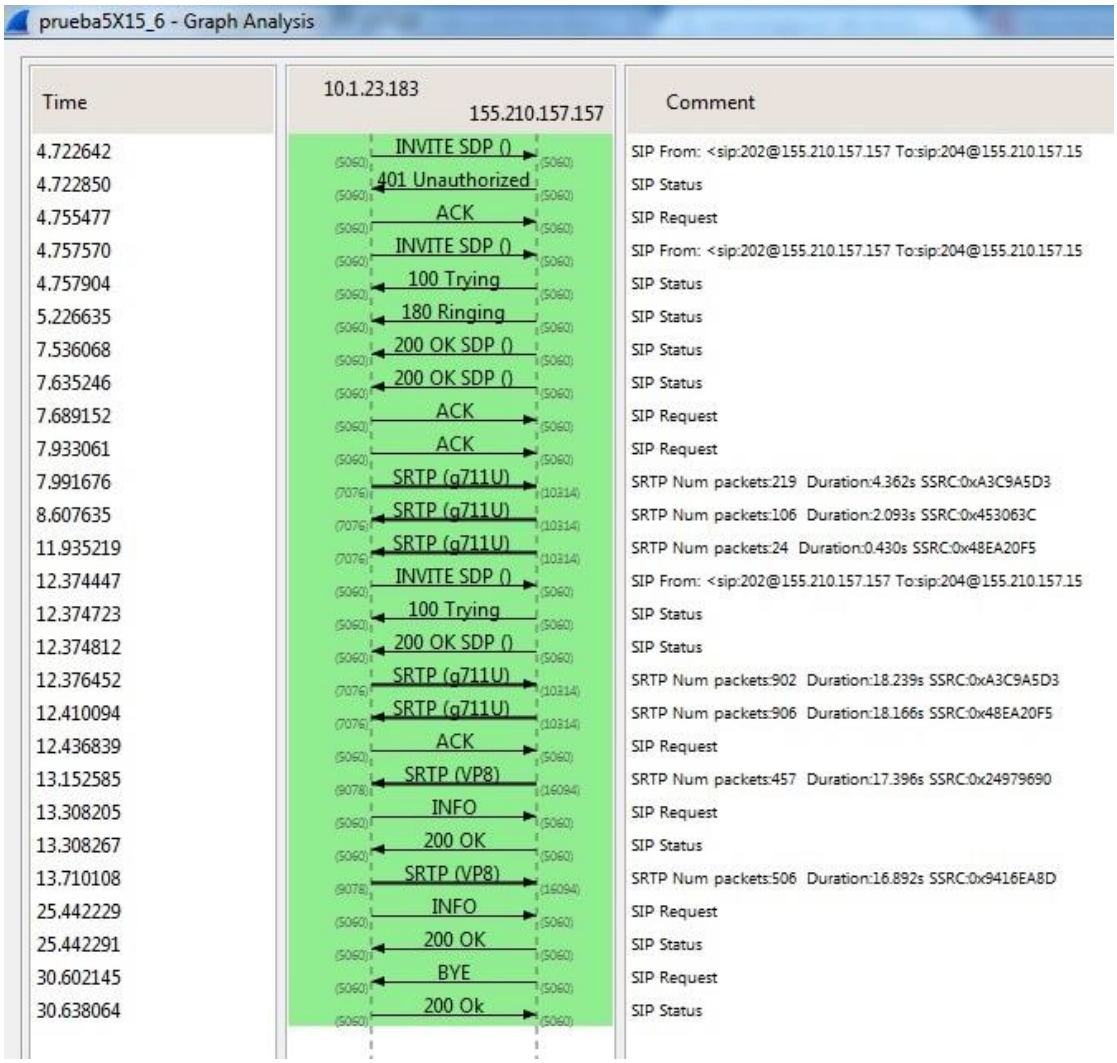


Figura 71 Flujo de señalización SIP con g711u y Vp8.

4.2.7. Conclusiones

Se han presentado diferentes pruebas, en diferentes configuraciones y distintos códecs lo que demuestran la funcionalidad del sistema. No se ha querido tratar en el cuerpo de los epígrafes el análisis de los consumos de anchos de bandas para que sea tratado en el capítulo final de esta tesis. Como se ha podido percibir, es posible la

realización de un sistema que permita las realizaciones de llamadas de video y de voz cifrados y ha quedado demostrado a lo largo del capítulo.

Capítulo 5.

“Resultados y Conclusiones”

Capítulo 5. Resultados y Conclusiones

5.0. Introducción

En el siguiente capítulo quedarán expuestos los resultados de las pruebas realizadas en el capítulo 4. Se entrará en detalles de los anchos de bandas consumidos por los distintos códecs de audio que fueron probados. De tal modo se podrá visualizar mediante gráficas cuál es el más indicado en cuanto al menor consumo. Finalmente se llegará a conclusiones que demostrarán el cumplimiento de las tareas y los objetivos propuestos.

5.1. Prueba 1. Realización de una llamada cifrada utilizando códec Speex y Vp8.

La prueba 1 como se había indicado en el capítulo anterior es una realización donde los códecs involucrados fueron Speex y Vp8. Dicha prueba fue efectuada entre dos extensiones que pertenecen a una misma PBX. La figura 72 muestra un ejemplo que visualiza en qué lugar específico de la red se realizó la llamada.

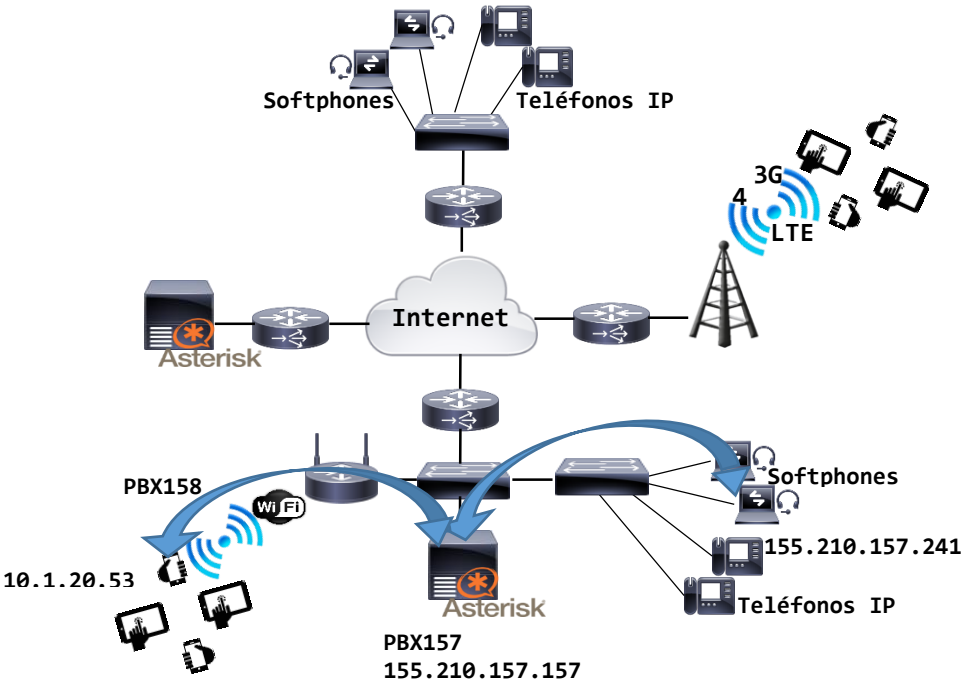


Figura 72 Escenario de llamada de la prueba 1.

Basado en la herramienta de análisis y captura explicada en el capítulo 3 y con la conformación de diferentes filtros de información, se pudo capturar el tráfico de audio y de video tanto de subida como de bajada entre los clientes involucrados.

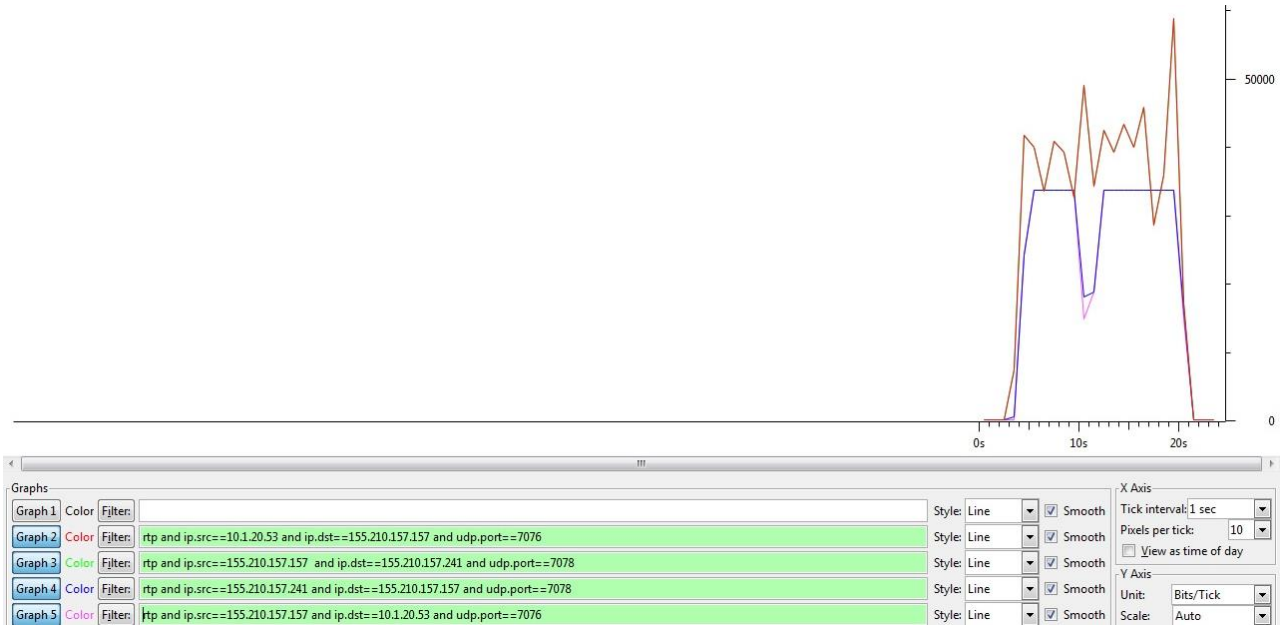


Figura 73 Gráficas de tráfico de audio cifrado en una llamada con códec speex.

Como se puede ver en la figura 73 se han puesto 4 filtros:

1. Rtp and ip.src==10.1.20.53 and ip.dst==155.210.157.157 and udp.port==7076
2. Rtp and ip.src==155.210.157.157 and ip.dst==155.210.157.241 and udp.port==7078
3. Rtp and ip.src==155.210.157.241 and ip.dst==155.210.157.157 and udp.port==7078
4. Rtp and ip.src==155.210.157.157 and ip.dst==10.1.20.53 and udp.port==7076.

En la Figura 73 sólo se pueden ver dos gráficas, una de color azul y una de color rojo, sin embargo en el mismo gráfico se encuentran cuatro representaciones: una gráfica azul, una roja, una verde y una magenta. La verde y la magenta no son

visibles y esto es debido a que coinciden con el ancho de banda de la gráfica azul y rojo respectivamente.

El filtro 1 grafica el tráfico que tiene como puerto destino el 7076. Dicho tráfico es de audio y corresponde a los paquetes que han sido enviados desde el cliente con dirección 10.1.20.53 y hacia el servidor con dirección 155.210.157.157.

El filtro 2 grafica el tráfico que tiene como puerto destino el 7078. Dicho tráfico es de audio y corresponde a los paquetes que han sido enviados desde el servidor con dirección 155.210.157.157 y hacia el cliente con dirección 155.210.157.241.

El filtro 3 grafica el tráfico que tiene como puerto destino el 7078. Dicho tráfico es de audio y corresponde a los paquetes que han sido enviados desde el cliente con dirección 155.210.157.241 y hacia el servidor con dirección 155.210.157.157.

El filtro 4 grafica el tráfico que tiene como puerto destino el 7076. Dicho tráfico es de audio y corresponde a los paquetes que han sido enviados desde el servidor con dirección 155.210.157.157 y hacia el cliente con dirección 10.1.20.53.

Si se aplican los filtros del 1 al 4 a todo el tráfico capturado se puede calcular con exactitud el ancho de banda consumido por speex en los cuatro flujos de información. Entiéndase los dos flujos de subidas y los dos de bajadas correspondientes a la comunicación. Destacar que el ancho de banda mostrado en la figura 96 tiene en cuenta 14 octetos de la cabecera de Ethernet, por lo que no es el ancho de bando que se desea calcular, sino un ancho de banda dado por el Wirshark que se tomará como referencia de comparación. Para calcular el ancho de banda se tienen que restar a cada uno de los paquetes los 14 octetos de la cabecera. Se debe hacer notar que se debe estar claro en cuanto a que nivel se desea calcular el ancho de banda, ya que este será mayor o menor en dependencia de las cabeceras agregadas por niveles. En todas las pruebas siguientes el ancho

de banda será calculado a nivel de IP, quitando las cabeceras de la capa física. En particular la prueba 1 tiene un desnivel de información durante la transmisión de los datos. Si se observa la figura 73 se puede apreciar que cercano a los 10 segundos existe 1 segundo en los cuales los clientes no transmitieron datos. Por lo que esto afectará también el resultado debido a que se está teniendo en cuenta 1 segundo en el cual no hay transmisión alguna. Se recomienda tener en cuenta todos estos detalles para tener un resultado lo más acertado posible.

La figuras 74 muestra dicha información correspondiente al filtro 1.

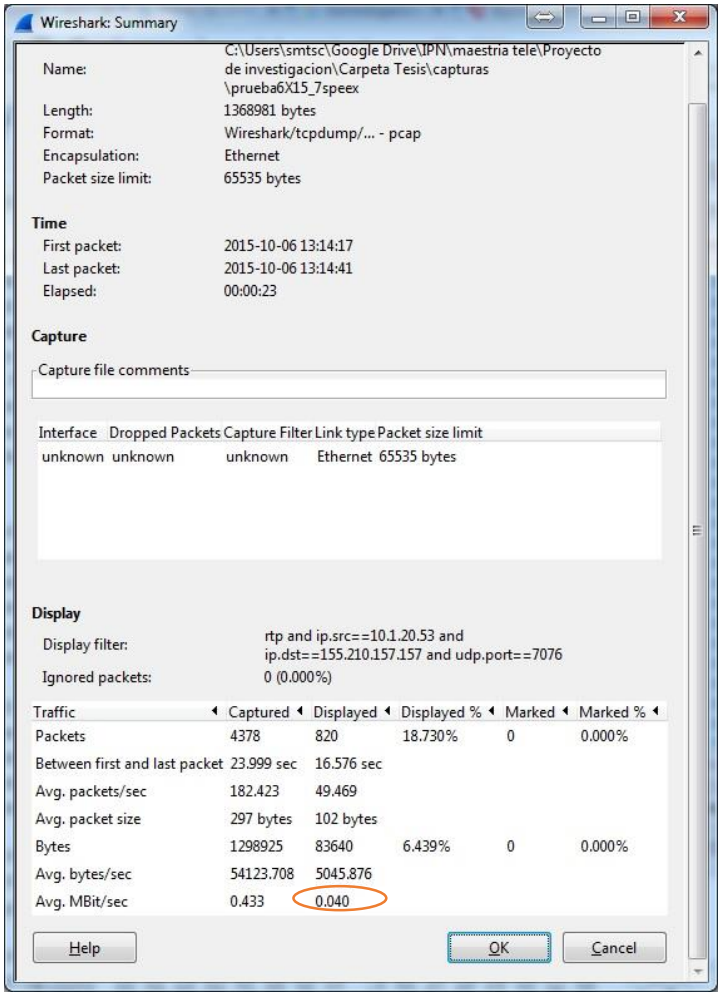


Figura 74 Tráfico capturado con filtro 1.

De la propia figura 74 se puede sacar cierta información que será útil a la hora de hacer el cálculo. Por ejemplo se puede ver que el promedio de envios de paquetes es aproximadamente 50 paquetes por segundo y que el tamaño promedio por paquetes es de 102 bytes. Que la cantidad de paquetes filtrados son 820 y que el tiempo aproximado de los paquetes filtrados es de 16.5 s. Con estos datos se puede calcular de la siguiente manera el ancho de banda correspondiente al filtro 1.

Se llamará:

T al tamaño de promedio de los paquetes en Bytes
P a la cantidad de paquetes filtrados
t al tiempo que ocupan los paquetes en segundo

AB1=[(T-14)*P*8bit]/t

Para el filtro 1 se tiene que:

AB1=(102 bytes-14 bytes)*820*8bit/16.5 s= 34986.666 bit/s=34.9 kbps.

Ha de notarse que teniendo en cuenta las cabeceras restadas, el ancho de banda del filtro 1 sería de:

AB1=0.040 Mbps o lo que es lo mismo 40 kbps

Se hace notar la diferencia entre ambos resultados.

Para el caso de los restantes filtros, se ha realizado un procedmiento exactamente igual, por lo que no serán mostradas todas las figuras. Sin embargo será calculado el ancho de banda de la misma forma.

Para el filtro 2 se tiene que:

T= 102 bytes
P= 820
t=16.5
AB2=(102 bytes-14 bytes)*820*8bit/16.5 s= 34986.666 bit/s=34.9 kbps.

Teniendo en cuenta las cabeceras restadas, el ancho de banda del filtro 2 sería de:

AB2=0.040 Mbps o lo que es lo mismo 40kbps.

Se hace notar la diferencia.

Para el filtro 3 se tiene que:

T= 84 bytes
P= 766
t=17
AB3=(84 bytes-14 bytes)*766*8bit /17 s= 2472.244 bit/s=24.72 kbps.

Teniendo en cuenta las cabeceras restadas, el ancho de banda del filtro 2 sería de:

AB3=0.030 Mbps o lo que es lo mismo 30 kbps

Se hace notar la diferencia.

Para el filtro 4 se tiene que:

T= 84 bytes
P= 758
t=16.13
AB4=(84 bytes-14 bytes)*758*8bit /16.13 s= 26041.71 bit/s=26.041 kbps.

Teniendo en cuenta las cabeceras restadas, el ancho de banda del filtro 2 sería de:

AB4=0.032 Mbps o lo que es lo mismo 32 kbps

Se hace notar la diferencia.

Agrupando los anchos de bandas calculados al aplicar los 4 filtros de la prueba se obtiene la siguiente información:

Tabla 5 Anchos de banda de speex

AB Filtro 1	34.9 Kbps
AB Filtro 2	34.9 Kbps
AB Filtro 3	24.7 Kbps
AB Filtro 4	26.0 Kbps

Como en un sentido la taza de transmisión es diferente a la otra, no se puede generalizar haciendo un promedio de anchos de bandas para speex. A de notarse que ambos clientes utilizados se encontraban en diferentes redes, diferentes dispositivos, con sistemas operativos diferentes, por lo que pudiera ser la causa de las diferencias notables en un sentido y en otro. No obstante se ha tomado un tiempo de prueba lo suficientemente grande para hacer los cálculos, por lo que los valores se encuentran en el rango teórico, confirmando que los resultados obtenidos son fiables. Esto es aplicable para el resto de las pruebas que se analizarán a lo largo del capítulo.

Haciendo un análisis muy similar al anterior se puede calcular el ancho de banda promedio para el codec de video Vp8. La figura 75 es muy similar a la figura 95

anteriormente mostrada, con la diferencia en que en esta se muestra una gráfica de 4 filtros de la captura de los paquetes correspondientes al códec Vp8.

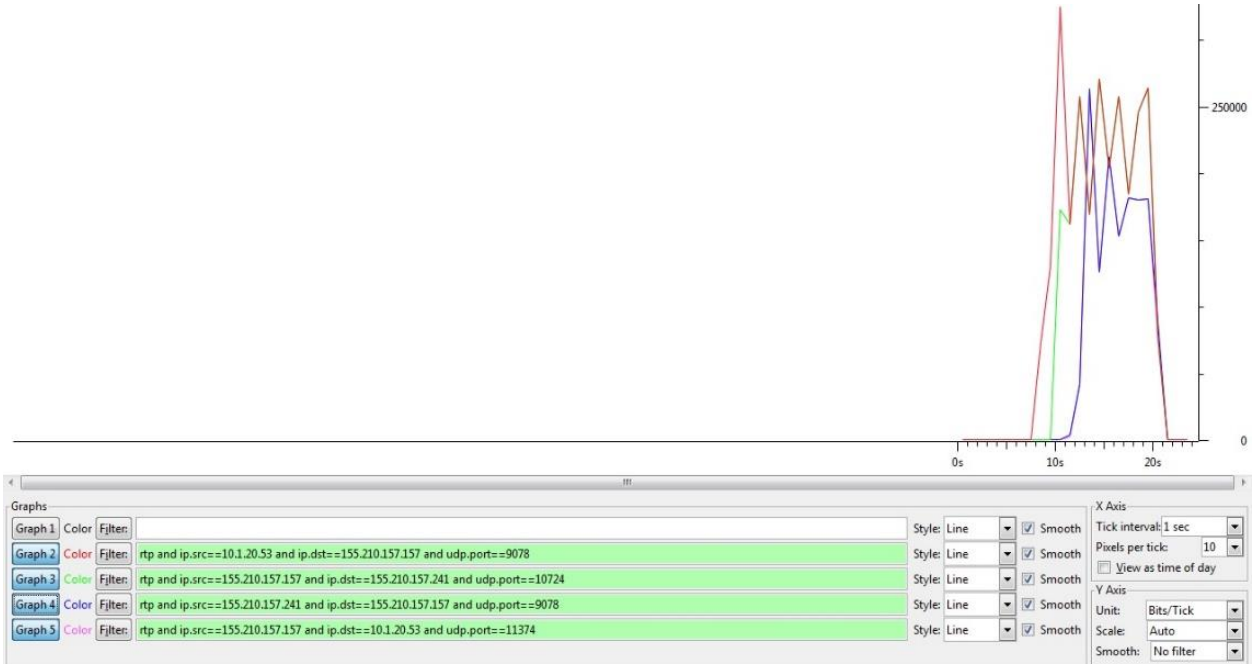


Figura 75 Gráficas de tráfico de video en una llamada con códec Vp8.

Como se puede ver en la figura 75 se han puesto 4 filtros:

1. Rtp and ip.src==10.1.20.53 and ip.dst==155.210.157.157 and udp.port==9078
2. Rtp and ip.src==155.210.157.157 and ip.dst==155.210.157.241 and udp.port==10724
3. Rtp and ip.src==155.210.157.241 and ip.dst==155.210.157.157 and udp.port==9078
4. Rtp and ip.src==155.210.157.157 and ip.dst==10.1.20.53 and udp.port==11374.

El filtro 1 grafica el tráfico que tiene como puerto destino el 9078. Dicho tráfico es de video y corresponde a los paquetes que han sido enviados desde el cliente con dirección 10.1.20.53 y hacia el servidor con dirección 155.210.157.157.

El filtro 2 grafica el tráfico que tiene como puerto destino el 10724. Dicho tráfico es de video y corresponde a los paquetes que han sido enviados desde el servidor con dirección 155.210.157.157 y hacia el cliente con dirección 155.210.157.241.

El filtro 3 grafica el tráfico que tiene como puerto destino el 9078. Dicho tráfico es de video y corresponde a los paquetes que han sido enviados desde el cliente con dirección 155.210.157.241 y hacia el servidor con dirección 155.210.157.157.

El filtro 4 grafica el tráfico que tiene como puerto destino el 11374. Dicho tráfico es de audio y corresponde a los paquetes que han sido enviados desde el servidor con dirección 155.210.157.157 y hacia el cliente con dirección 10.1.20.53.

A diferencia de la figura 95 en la figura 97 se ven bien definidos los 4 filtros, esto es debido a que las diferencias en taza de transmisión en un sentido y en otro. Cuando se analiza ancho de banda de video y en especial con Vp8, se pueden tener estas diferencias debido a que Vp8 es un códec adaptativo por lo que su ancho de banda varía en dependencia de los recursos de la red. Es por esto que es tan difícil poder decir con exactitud que ancho de banda consume. Sin embargo se ha realizado el mismo procedimiento que el caso anterior y se han obtenido los siguientes resultados:

Para el filtro 1 se tiene que:

AB1=(955 bytes-14 bytes)*343*8bit/11.64= 222595 bit/s=222.5 kbps.

Ha de notarse que teniendo en cuenta las cabeceras restadas, el ancho de banda del filtro 1 sería de:

AB1=0.225 Mbps o lo que es lo mismo 225 Kbps.

Se hace notar la diferencia aunque no es mucha entre ambos resultados.

Para el filtro 2 se tiene que:

T= 953 bytes
P= 297
t=9.81
AB2=(953 bytes-14 bytes)*297*8bit/9.51 s= 234601 bit/s=234.6 kbps.

Teniendo en cuenta las cabeceras restadas, el ancho de banda del filtro 2 sería de:

AB2=0.231 Mbps o lo que es lo mismo 231kbps.

Se hace notar la diferencia.

Para el filtro 3 se tiene que:

T= 890 bytes
P= 201
t=9.0
AB3=(890 bytes-14 bytes)*201*8bit /9 s= 156512 bit/s=156.512 kbps.

Teniendo en cuenta las cabeceras restadas, el ancho de banda del filtro 3 sería de:

AB3= 0.159 Mbps o lo que es lo mismo 159 kbps

Se hace notar la diferencia.

Para el filtro 4 se tiene que:

T= 899 bytes
P= 197
t=8.3
AB4=(899 bytes-14 bytes)*197*8bit /8.3 s= 168043 bit/s=168.043 kbps.

Teniendo en cuenta las cabeceras restadas, el ancho de banda del filtro 4 sería de:

AB4=0.169 Mbps o lo que es lo mismo 169 kbps

Se hace notar la diferencia.

Agrupando los anchos de bandas calculados al aplicar los 4 filtros de la prueba se obtiene la siguiente información en la Tabla 9

Tabla 6 Anchos de bandas de Vp8

AB Filtro 1	222.5 Kbps
AB Filtro 2	234.6 Kbps
AB Filtro 3	156.2 Kbps
AB Filtro 4	168.0 Kbps

5.2. Prueba 2 Realización de una llamada no cifrada utilizando códec speex y Vp8.

La prueba 2 es muy similar a la 1 con la diferencia en el cifrado. En la prueba 1 se ha desactivado la seguridad y el único objetivo de esto es poder comparar los anchos de bandas una vez que se ha cifrado con respecto a cuando no se hace. La única diferencia notable entre una prueba cifrada y no cifrada son 10 octetos que se agregan por el cifrado, por lo que no es mucha la diferencia. No obstante quedará demostrado a lo largo de todos los resultados mostrados. Se han mantenido los mismos códecs para tener una base para comparar. La prueba fue efectuada entre dos extensiones que pertenecen a una misma PBX. La figura 76 lo pone en evidencia.

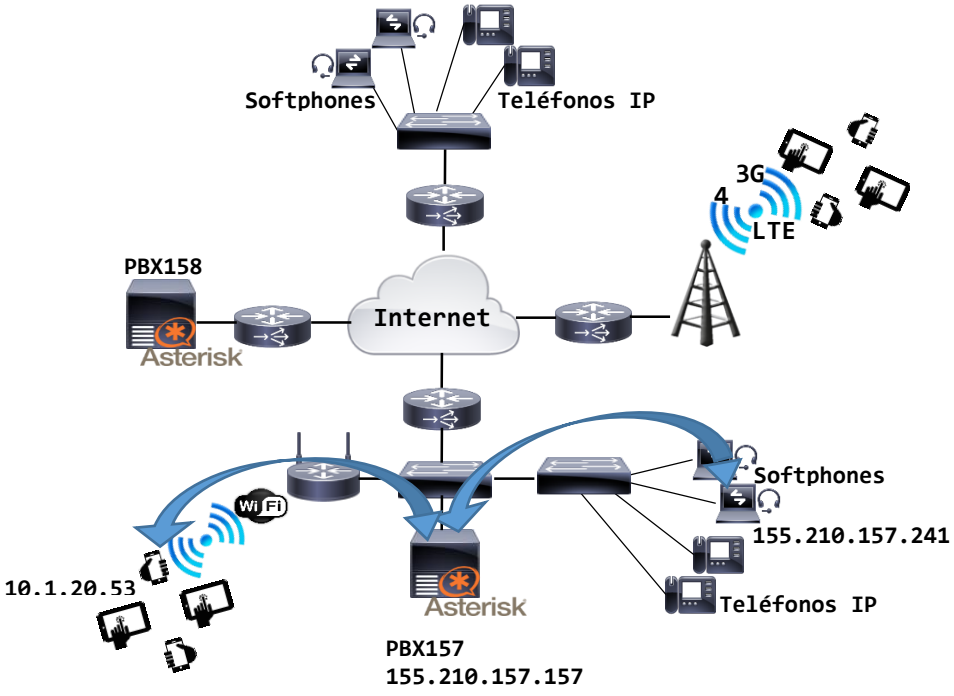


Figura 76 Escenario de llamada de la prueba 8.

Basado en la herramienta de análisis y captura explicada en el capítulo 3 y con la conformación de diferentes filtros de información, se pudo capturar el tráfico de audio y de video tanto de subida como de bajada entre los clientes involucrados.

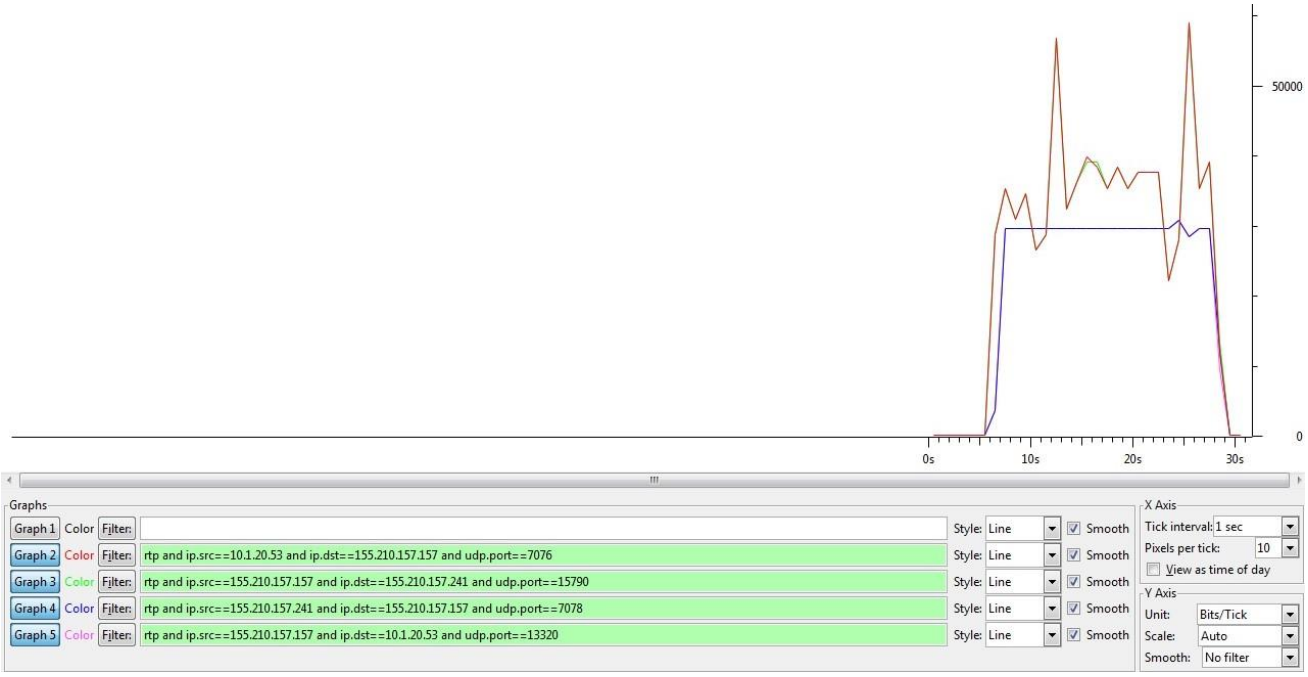


Figura 77 Gráficas de tráfico de audio no cifrado en una llamada con códec speex.

Como se puede ver en la figura 77 se han puesto 4 filtros:

1. Rtp and ip.src==10.1.20.53 and ip.dst==155.210.157.157 and udp.port==7076
2. Rtp and ip.src==155.210.157.157 and ip.dst==155.210.157.241 and udp.port==15790
3. Rtp and ip.src==155.210.157.241 and ip.dst==155.210.157.157 and udp.port==7078

4. Rtp and ip.src==155.210.157.157 and ip.dst==10.1.20.53 and udp.port==13320.

El filtro 1 grafica el tráfico que tiene como puerto destino el 7076. Dicho tráfico es de audio y corresponde a los paquetes que han sido enviados desde el cliente con dirección 10.1.20.53 y hacia el servidor con dirección 155.210.157.157.

El filtro 2 grafica el tráfico que tiene como puerto destino el 15790. Dicho tráfico es de audio y corresponde a los paquetes que han sido enviados desde el servidor con dirección 155.210.157.157 y hacia el cliente con dirección 155.210.157.241.

El filtro 3 grafica el tráfico que tiene como puerto destino el 7078. Dicho tráfico es de audio y corresponde a los paquetes que han sido enviados desde el cliente con dirección 155.210.157.241 y hacia el servidor con dirección 155.210.157.157.

El filtro 4 grafica el tráfico que tiene como puerto destino el 13320. Dicho tráfico es de audio y corresponde a los paquetes que han sido enviados desde el servidor con dirección 155.210.157.157 y hacia el cliente con dirección 10.1.20.53.

Tal cual pasó en la prueba 1, en la prueba 2 no se pueden distinguir los cuatro gráficos. La causa es la misma que se explicó con anterioridad.

Si se aplican los filtros del 1 al 4 a todo el tráfico capturado se puede calcular con exactitud el ancho de banda consumido por speex en los cuatro flujos de información. Entiéndase los dos flujos de subidas y los dos de bajadas correspondientes a la comunicación. Los resultados arrojados son los siguientes:

Para el filtro 1 se tiene que:

AB1=(92 bytes-14 bytes)*1093*8bit/22.05= 30931 bit/s= 30.93 kbps.

Ha de notarse que teniendo en cuenta las cabeceras restadas, el ancho de banda del filtro 1 sería de:

AB1=0.036 Mbps o lo que es lo mismo 36 Kbps.

Se hace notar la diferencia aunque no es mucha entre ambos resultados.

Se puede notar además como cuando no es cifrado el tamaño de los paquetes de ser 102 bytes, donde claramente se puede apreciar una diferencia de 10 bytes debido al cifrado.

Para el filtro 2 se tiene que:

T= 92 bytes
P= 1095
t=22.08s
AB2=(92 bytes-14 bytes)*1095*8bit/22.08 s= 30945 bit/s=30.94 kbps.

Teniendo en cuenta las cabeceras restadas, el ancho de banda del filtro 2 sería de:

AB2=0.036 Mbps o lo que es lo mismo 36 kbps.

Se hace notar la diferencia.

Para el filtro 3 se tiene que:

T= 74 bytes

P= 1076
t=21.445 s
AB3=(74 bytes-14 bytes)*1076*8bit /22.445 s= 23010 bit/s=23.01 kbps.

Teniendo en cuenta las cabeceras restadas, el ancho de banda del filtro 3 sería de:

AB3= 0.30 Mbps o lo que es lo mismo 30 kbps

Se hace notar la diferencia.

Además se ve claramente la diferencia de 10 octetos por el cifrado, o sea, en esta prueba 2 los paquetes tienen un promedio de tamaño de 74 bytes a diferencia de la prueba 1 que son 84 bytes.

Para el filtro 4 se tiene que:

T= 74 bytes
P= 1072
t=21.4
AB4=(74 bytes-14 bytes)*1072*8bit /21.4 s= 24044 bit/s=24.04 kbps.

Teniendo en cuenta las cabeceras restadas, el ancho de banda del filtro 4 sería de:

AB4=0.030 Mbps o lo que es lo mismo 30 kbps

Se hace notar la diferencia.

Agrupando los anchos de bandas calculados al aplicar los 4 filtros de la prueba se obtiene la siguiente información en la Tabla 10:

Tabla 7 Anchos de bandas de speex no cifrado

AB Filtro 1	30.9 Kbps
AB Filtro 2	30.9 Kbps
AB Filtro 3	23.0 Kbps
AB Filtro 4	24.0 Kbps

Haciendo un análisis similar al anterior se puede calcular el ancho de banda promedio para el codec de video Vp8. La figura 78 muestra una gráfica de 4 filtros de la captura de los paquetes correspondientes al códec Vp8.

Figura 78 Gráficas de tráfico de video en una llamada con códec Vp8.

Como se puede ver en la figura 78 se han puesto 4 filtros:

1. Rtp and ip.src==10.1.20.53 and ip.dst==155.210.157.157 and udp.port==9078
2. Rtp and ip.src==155.210.157.157 and ip.dst==155.210.157.241 and udp.port==16776
3. Rtp and ip.src==155.210.157.241 and ip.dst==155.210.157.157 and udp.port==9078
4. Rtp and ip.src==155.210.157.157 and ip.dst==10.1.20.53 and udp.port==10466.

El filtro 1 grafica el tráfico que tiene como puerto destino el 9078. Dicho tráfico es de video y corresponde a los paquetes que han sido enviados desde el cliente con dirección 10.1.20.53 y hacia el servidor con dirección 155.210.157.157.

El filtro 2 grafica el tráfico que tiene como puerto destino el 16776. Dicho tráfico es de video y corresponde a los paquetes que han sido enviados desde el servidor con dirección 155.210.157.157 y hacia el cliente con dirección 155.210.157.241.

El filtro 3 grafica el tráfico que tiene como puerto destino el 9078. Dicho tráfico es de video y corresponde a los paquetes que han sido enviados desde el cliente con dirección 155.210.157.241 y hacia el servidor con dirección 155.210.157.157.

El filtro 4 grafica el tráfico que tiene como puerto destino el 10466. Dicho tráfico es de audio y corresponde a los paquetes que han sido enviados desde el servidor con dirección 155.210.157.157 y hacia el cliente con dirección 10.1.20.53.

Como se puede ver la única diferencia entre los filtros correspondientes a los paquetes de audio y los correspondientes a los de video son el puerto destino. Por lo que se puede identificar el tipo de paquetes según esta información.

En el caso de esta prueba las tazas de transmisión son similares por lo que ocurre que los gráficos se solapan y es difícil ver las diferencias entre los cuatro filtros.

Si se aplican los filtros del 1 al 4 a todo el tráfico capturado se puede calcular con exactitud el ancho de banda consumido por Vp8 en los cuatro flujos de información. Entiéndase los dos flujos de subidas y los dos de bajadas correspondientes a la comunicación. Los resultados arrojados son los siguientes:

Para el filtro 1 se tiene que:

AB1=(953 bytes-14 bytes)*550*8bit/18.6 s= 222129 bit/s=222.1 kbps.

Ha de notarse que teniendo en cuenta las cabeceras restadas, el ancho de banda del filtro 1 sería de:

AB1=0.225 Mbps o lo que es lo mismo 225 Kbps.

Se hace notar la diferencia aunque no es mucha entre ambos resultados.

En este caso no se puede definir concretamente cuantos octetos se agregan al cifrar la información o no debido a que el tamaño de los paquetes en video es variable. Por lo que no se puede sacar una relación como se hizo en audio.

Para el filtro 2 se tiene que:

T= 953 bytes
P= 548
t=18.4
AB2=(953 bytes-14 bytes)*548*8bit/18.4 s= 2237 bit/s=223.7 kbps.

Teniendo en cuenta las cabeceras restadas, el ancho de banda del filtro 2 sería de:

AB2=0.227 Mbps o lo que es lo mismo 227 kbps.

Se hace notar la diferencia.

Para el filtro 3 se tiene que:

T= 897 bytes
P= 509
t=20.190
AB3=(897 bytes-14 bytes)*509*8bit /20.2 s= 17799 bit/s=177.99 kbps.

Teniendo en cuenta las cabeceras restadas, el ancho de banda del filtro 3 sería de:

AB3= 0.181 Mbps o lo que es lo mismo 181 kbps

Se hace notar la diferencia.

Para el filtro 4 se tiene que:

T= 889 bytes
P= 493
t=20 s
AB4=(899 bytes-14 bytes)*493*8bit /20 s= 17452 bit/s=174.5 kbps.

Teniendo en cuenta las cabeceras restadas, el ancho de banda del filtro 4 sería de:

AB4=0.175 Mbps o lo que es lo mismo 175 kbps

Se hace notar la diferencia.

Agrupando los anchos de bandas calculados al aplicar los 4 filtros de la prueba se obtiene la siguiente información:

Tabla 8 Anchos de bandas de Vp8 cifrado

AB Filtro 1	222.1 Kbps
AB Filtro 2	223.2 Kbps
AB Filtro 3	177.9 Kbps
AB Filtro 4	174.5 Kbps

5.3. Prueba 3 Realización de una llamada cifrada utilizando códec GSM y Vp8

La prueba 3 como se había indicado en el capítulo anterior es una realización donde los códecs involucrados fueron GSM y Vp8. Dicha prueba fue efectuada entre dos extensiones que pertenecen a una misma PBX. La figura 79 muestra un ejemplo que visualiza en qué lugar específico de la red se realizó la llamada.

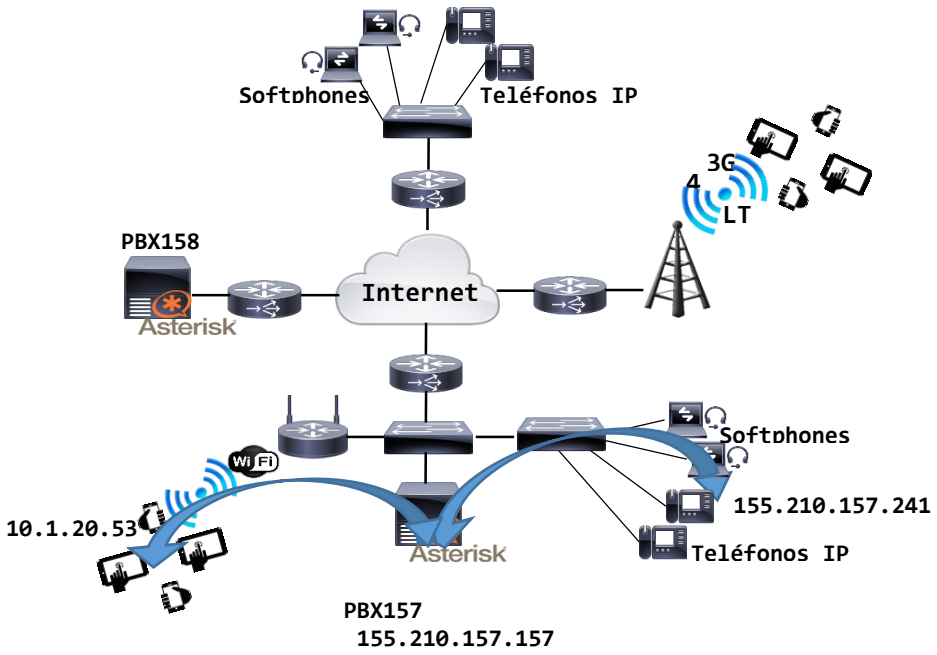


Figura 79 Escenario de llamada de la prueba 3.

Basado en la herramienta de análisis y captura explicada en el capítulo 3 y con la conformación de diferentes filtros de información, se pudo capturar el tráfico de audio y de video tanto de subida como de bajada entre los clientes involucrados.

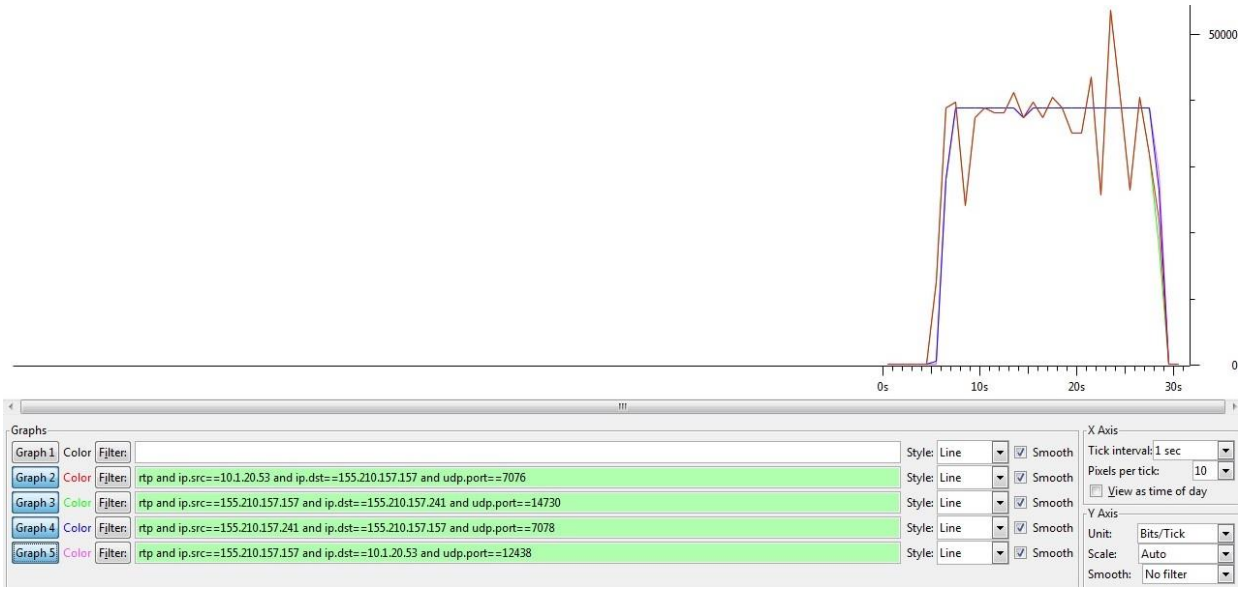


Figura 80 Gráficas de tráfico de audio cifrado en una llamada con códec GSM.

Como se puede ver en la Figura 80 se han puesto 4 filtros:

1. Rtp and ip.src==10.1.20.53 and ip.dst==155.210.157.157 and udp.port==7076
2. Rtp and ip.src==155.210.157.157 and ip.dst==155.210.157.241 and udp.port==14730
3. Rtp and ip.src==155.210.157.241 and ip.dst==155.210.157.157 and udp.port==7078
4. Rtp and ip.src==155.210.157.157 and ip.dst==10.1.20.53 and udp.port==12438.

El filtro 1 grafica el tráfico que tiene como puerto destino el 7076. Dicho tráfico es de audio y corresponde a los paquetes que han sido enviados desde el cliente con dirección 10.1.20.53 y hacia el servidor con dirección 155.210.157.157.

El filtro 2 grafica el tráfico que tiene como puerto destino el 14730. Dicho tráfico es de audio y corresponde a los paquetes que han sido enviados desde el servidor con dirección 155.210.157.157 y hacia el cliente con dirección 155.210.157.241.

El filtro 3 grafica el tráfico que tiene como puerto destino el 7078. Dicho tráfico es de audio y corresponde a los paquetes que han sido enviados desde el cliente con dirección 155.210.157.241 y hacia el servidor con dirección 155.210.157.157.

El filtro 4 grafica el tráfico que tiene como puerto destino el 12438. Dicho tráfico es de audio y corresponde a los paquetes que han sido enviados desde el servidor con dirección 155.210.157.157 y hacia el cliente con dirección 10.1.20.53.

Tal como en pruebas anteriores en esta tampoco se pueden diferenciar por colores los 4 filtros. Por lo que la gráfica azul es similar y casi idéntica a la verde y la roja a la magenta.

Si se aplican los filtros del 1 al 4 a todo el tráfico capturado se puede calcular con exactitud el ancho de banda consumido por GSM en los cuatro flujos de información. Entiéndase los dos flujos de subidas y los dos de bajadas correspondientes a la comunicación.

Los resultados obtenidos son los siguientes:

Para el filtro 1 se tiene que:

AB1=(97 bytes-14 bytes)*1100*8bit/22.9 s= 31895 bit/s=31.8 kbps.

Ha de notarse que teniendo en cuenta las cabeceras restadas, el ancho de banda del filtro 1 sería de:

AB1=0.037 Mbps o lo que es lo mismo 37 kbps

Se hace notar la diferencia entre ambos resultados.

Para el caso de los restantes filtros, se ha realizado un procedmiento exactamente igual, por lo que no serán mostradas todas las figuras. Sin embargo será calculado el ancho de banda de la misma forma.

Para el filtro 2 se tiene que:

T= 97 bytes
P= 1096
t=22.8 s
AB1=(97 bytes-14 bytes)*1096*8bit/22.8 s= 31918.666 bit/s=31.91 kbps.

Teniendo en cuenta las cabeceras restadas, el ancho de banda del filtro 2 sería de:

AB2=0.037 Mbps o lo que es lo mismo 37 kbps.

Se hace notar la diferencia.

Para el filtro 3 se tiene que:

T= 97 bytes
P= 1119
t= 23.42 s
AB1=(97 bytes-14 bytes)*1119*8bit /23.42 s= 31725 bit/s=31.7 kbps.

Teniendo en cuenta las cabeceras restadas, el ancho de banda del filtro 2 sería de:

AB3=0.037 Mbps o lo que es lo mismo 37 kbps

Se hace notar la diferencia.

Para el filtro 4 se tiene que:

T= 97 bytes
P= 1116
t=22.3 s
AB1=(97 bytes-14 bytes)*1116*8bit /22.3 s= 33229 bit/s=33.2 kbps.

Teniendo en cuenta las cabeceras restadas, el ancho de banda del filtro 2 sería de:

AB4=0.039 Mbps o lo que es lo mismo 39 kbps

Se hace notar la diferencia.

Agrupando los anchos de bandas calculados al aplicar los 4 filtros de la prueba se obtiene la siguiente información. Ver Tabla 12

Tabla 9 Anchos de bandas de GSM

AB Filtro 1	31.8 Kbps
AB Filtro 2	31.9 Kbps
AB Filtro 3	31.7 Kbps
AB Filtro 4	33.2 Kbps

Haciendo un análisis similar al anterior se puede calcular el ancho de banda promedio para el codec de video Vp8. La figura 81 muestra una gráfica de 4 filtros de la captura de los paquetes correspondientes al códec Vp8.

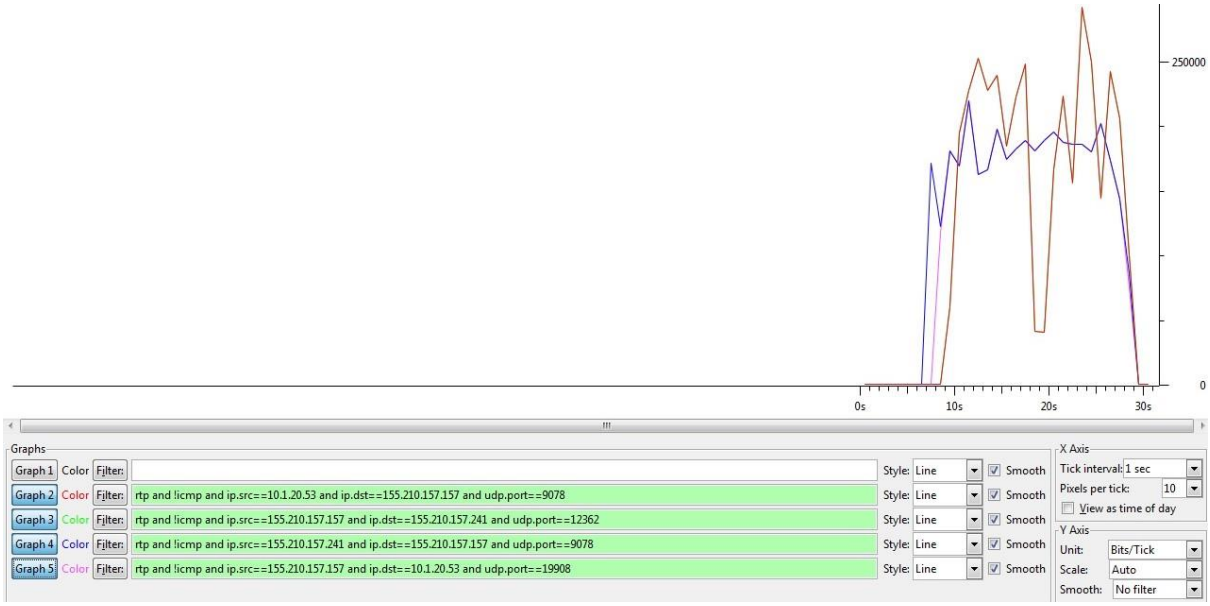


Figura 81 Gráficas de tráfico de video en una llamada con códec Vp8.

Como se puede ver en la figura 81 se han puesto 4 filtros:

1. Rtp and !icmp and ip.src==10.1.20.53 and ip.dst==155.210.157.157 and udp.port==9078
2. Rtp and !icmp and ip.src==155.210.157.157 and ip.dst==155.210.157.241 and udp.port==12362
3. Rtp and !icmp and ip.src==155.210.157.241 and ip.dst==155.210.157.157 and udp.port==9078
4. Rtp and !icmp and ip.src==155.210.157.157 and ip.dst==10.1.20.53 and udp.port==19908.

El filtro 1 grafica el tráfico que tiene como puerto destino el 9078. Dicho tráfico es de video y corresponde a los paquetes que han sido enviados desde el cliente con dirección 10.1.20.53 y hacia el servidor con dirección 155.210.157.157.

El filtro 2 grafica el tráfico que tiene como puerto destino el 12362. Dicho tráfico es de video y corresponde a los paquetes que han sido enviados desde el servidor con dirección 155.210.157.157 y hacia el cliente con dirección 155.210.157.241.

El filtro 3 grafica el tráfico que tiene como puerto destino el 9078. Dicho tráfico es de video y corresponde a los paquetes que han sido enviados desde el cliente con dirección 155.210.157.241 y hacia el servidor con dirección 155.210.157.157.

El filtro 4 grafica el tráfico que tiene como puerto destino el 19908. Dicho tráfico es de audio y corresponde a los paquetes que han sido enviados desde el servidor con dirección 155.210.157.157 y hacia el cliente con dirección 10.1.20.53.

En este caso tampoco se puede divisar la diferencia entre los 4 filtros, por lo que haciendo la explicación la gráfica de color rojo idéntica a la magenta y la de color verde a la azul.

Si se aplican los filtros del 1 al 4 a todo el tráfico capturado se puede calcular con exactitud el ancho de banda consumido por Vp8 en los cuatro flujos de información. Entiéndase los dos flujos de subidas y los dos de bajadas correspondientes a la comunicación. Los resultados obtenidos son los siguientes:

Para el filtro 1 se tiene que:

T= 958 bytes
P= 485
t=18.8 s

AB1=(958 bytes-14 bytes)*485*8bit/18.8 s= 194825 bit/s=194.8 kbps.

Se hace notar la diferencia entre ambos resultados.

Para el caso de los restantes filtros, se ha realizado un procedmiento exactamente igual, por lo que no serán mostradas todas las figuras. Sin embargo será calculado el ancho de banda de la misma forma.

Para el filtro 2 se tiene que:

T= 958 bytes
P= 485
t=18.8 s
AB2=(958 bytes-14 bytes)*485*8bit/18.8 s= 194825 bit/s=194.8 kbps.

Teniendo en cuenta las cabeceras restadas, el ancho de banda del filtro 2 sería de:

AB2=0.197 Mbps o lo que es lo mismo 197 kbps.

Se hace notar la diferencia.

Para el filtro 3 se tiene que:

T= 927 bytes
P= 518
t=22.3 s
AB3=(927 bytes-14 bytes)*518*8bit /22.3 s= 169662 bit/s=169.6 kbps.

Teniendo en cuenta las cabeceras restadas, el ancho de banda del filtro 2 sería de:

AB3=0.172 Mbps o lo que es lo mismo 172 kbps

Se hace notar la diferencia.

Para el filtro 4 se tiene que:

T= 928 bytes
P= 492
t=20.4 s
AB1=(928 bytes-14 bytes)*492*8bit /20.4 s= 176348 bit/s=176.34 kbps.

Teniendo en cuenta las cabeceras restadas, el ancho de banda del filtro 2 sería de:

AB4=0.179 Mbps o lo que es lo mismo 179 kbps

Se hace notar la diferencia.

Agrupando los anchos de bandas calculados al aplicar los 4 filtros de la prueba se obtiene la siguiente información. Ver Tabla 13

Tabla 10 Anchos de banda de Vp8

AB Filtro 1	194.8 Kbps
AB Filtro 2	194.8 Kbps
AB Filtro 3	169.6 Kbps
AB Filtro 4	176.3 Kbps

5.4. Prueba 4 Realización de una llamada cifrada utilizando códec Opus y Vp8.

La prueba 4 como se había indicado en el capítulo anterior es una realización donde los códecs involucrados fueron opus y Vp8. Dicha prueba fue efectuada entre dos extensiones que pertenecen a una misma PBX. La figura 82 muestra un ejemplo que visualiza en qué lugar específico de la red se realizó la llamada.

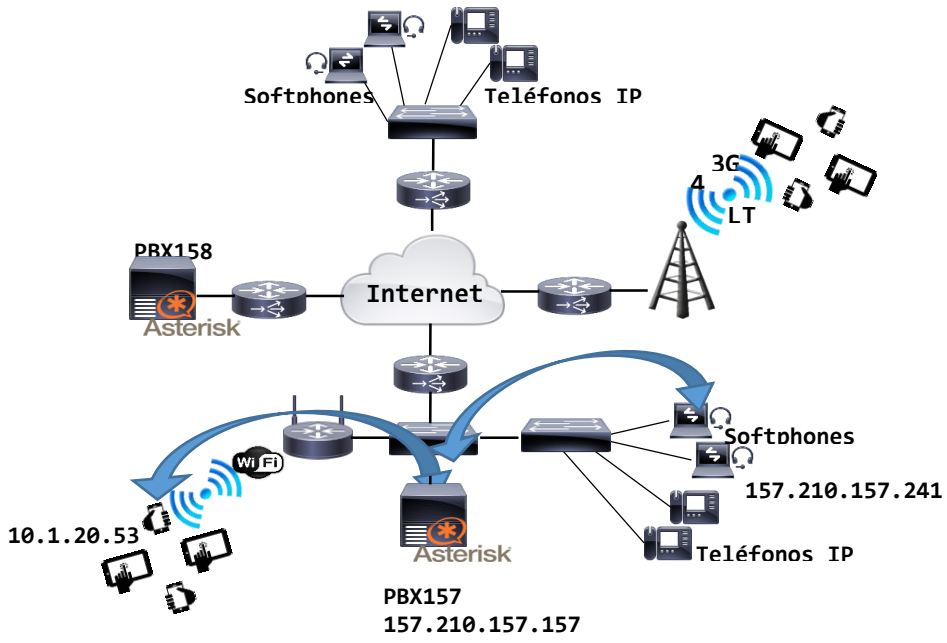


Figura 82 Escenario de llamada de la prueba 4.

Basado en la herramienta de análisis y captura explicada en el capítulo 3 y con la conformación de diferentes filtros de información, se pudo capturar el tráfico de audio y de video tanto de subida como de bajada entre los clientes involucrados.

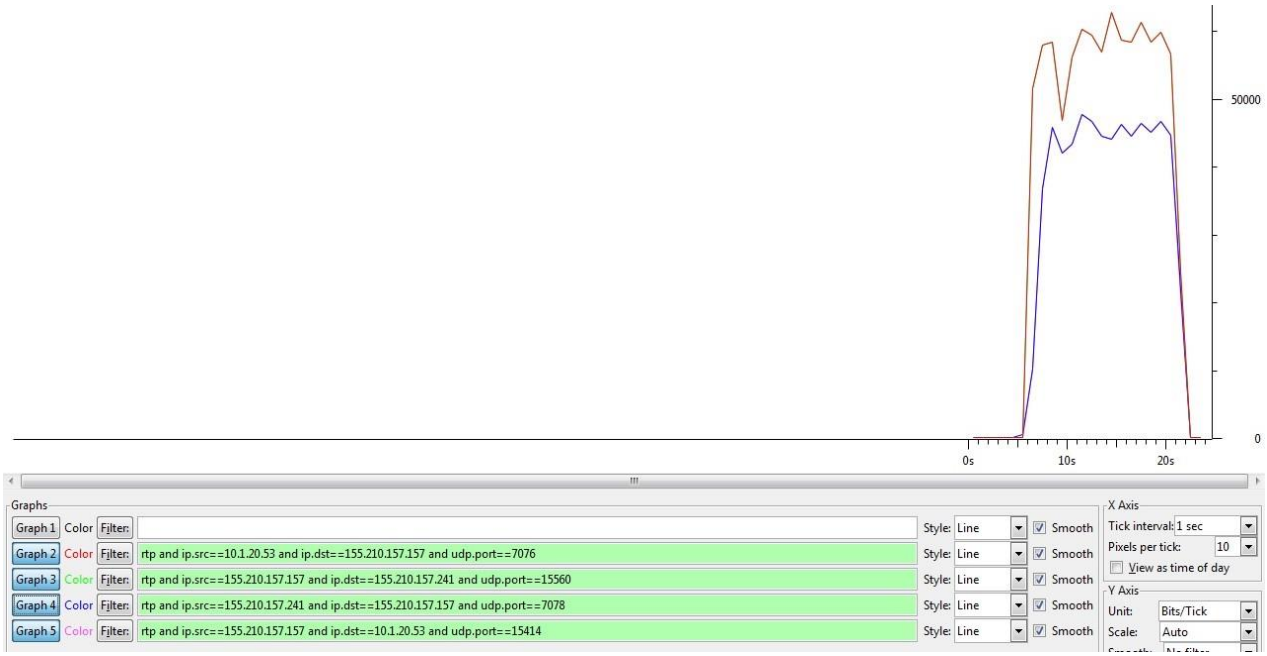


Figura 83 Gráficas de tráfico de audio cifrado en una llamada con códec opus.

Como se puede ver en la figura 83 se han puesto 4 filtros:

1. Rtp and ip.src==10.1.20.53 and ip.dst==155.210.157.157 and udp.port==7076
2. Rtp and ip.src==155.210.157.157 and ip.dst==155.210.157.241 and udp.port==15560
3. Rtp and ip.src==155.210.157.241 and ip.dst==155.210.157.157 and udp.port==7078
4. Rtp and ip.src==155.210.157.157 and ip.dst==10.1.20.53 and udp.port==15414.

El filtro 1 grafica el tráfico que tiene como puerto destino el 7076. Dicho tráfico es de audio y corresponde a los paquetes que han sido enviados desde el cliente con dirección 10.1.20.53 y hacia el servidor con dirección 155.210.157.157.

El filtro 2 grafica el tráfico que tiene como puerto destino el 15560. Dicho tráfico es de audio y corresponde a los paquetes que han sido enviados desde el servidor con dirección 155.210.157.157 y hacia el cliente con dirección 155.210.157.241.

El filtro 3 grafica el tráfico que tiene como puerto destino el 7078. Dicho tráfico es de audio y corresponde a los paquetes que han sido enviados desde el cliente con dirección 155.210.157.241 y hacia el servidor con dirección 155.210.157.157.

El filtro 4 grafica el tráfico que tiene como puerto destino el 15414. Dicho tráfico es de audio y corresponde a los paquetes que han sido enviados desde el servidor con dirección 155.210.157.157 y hacia el cliente con dirección 10.1.20.53.

El filtro azul es similar al verde y el rojo al magenta por lo que tampoco se puede ver la diferencia entre ellos.

Si se aplican los filtros del 1 al 4 a todo el tráfico capturado se puede calcular con exactitud el ancho de banda consumido por opus en los cuatro flujos de información. Entiéndase los dos flujos de subidas y los dos de bajadas correspondientes a la comunicación. Los resultados obtenidos son los siguientes:

Para el filtro 1 se tiene que:

T= 147 bytes
P= 752
t= 15.25 s
AB1=(147 bytes-14 bytes)*752*8bit/15.25 s= 52467 bit/s=52.4 kbps.

Teniendo en cuenta las cabeceras restadas, el ancho de banda del filtro 2 sería de:

AB1=0.058 Mbps o lo que es lo mismo 58 kbps.

Se hace notar la diferencia entre ambos resultados.

Para el filtro 2 se tiene que:

T= 147 bytes
P= 752
t= 15.25 s
AB1=(147 bytes-14 bytes)*752*8bit/15.25 s= 52467 bit/s=52.4 kbps.

Teniendo en cuenta las cabeceras restadas, el ancho de banda del filtro 2 sería de:

AB1=0.058 Mbps o lo que es lo mismo 58 kbps.

Se hace notar la diferencia entre ambos resultados.

Para el filtro 3 se tiene que:

T= 111 bytes
P= 738
t=15.8 s
AB3=(111 bytes-14 bytes)*738*8bit /15.8 s= 36246 bit/s=36.2 kbps.

Teniendo en cuenta las cabeceras restadas, el ancho de banda del filtro 2 sería de:

AB3=0.041 Mbps o lo que es lo mismo 41 kbps

Se hace notar la diferencia.

Para el filtro 4 se tiene que:

T= 111 bytes
P= 736
t=14.7 s
AB1=(111 bytes-14 bytes)*736*8bit /14.7 s= 38852 bit/s=38.8 kbps.

Teniendo en cuenta las cabeceras restadas, el ancho de banda del filtro 2 sería de:

AB4=0.044 Mbps o lo que es lo mismo 44 kbps

Se hace notar la diferencia.

Agrupando los anchos de bandas calculados al aplicar los 4 filtros de la prueba se obtiene la siguiente información. Ver tabla 14

Tabla 11 Anchos de banda de opus

AB Filtro 1	52.4 Kbps
AB Filtro 2	52.4 Kbps
AB Filtro 3	36.2 Kbps
AB Filtro 4	38.8 Kbps

Haciendo un análisis similar al anterior se puede calcular el ancho de banda promedio para el codec de video Vp8. La figura 84 muestra una gráfica de 4 filtros de la captura de los paquetes correspondientes al códec Vp8.

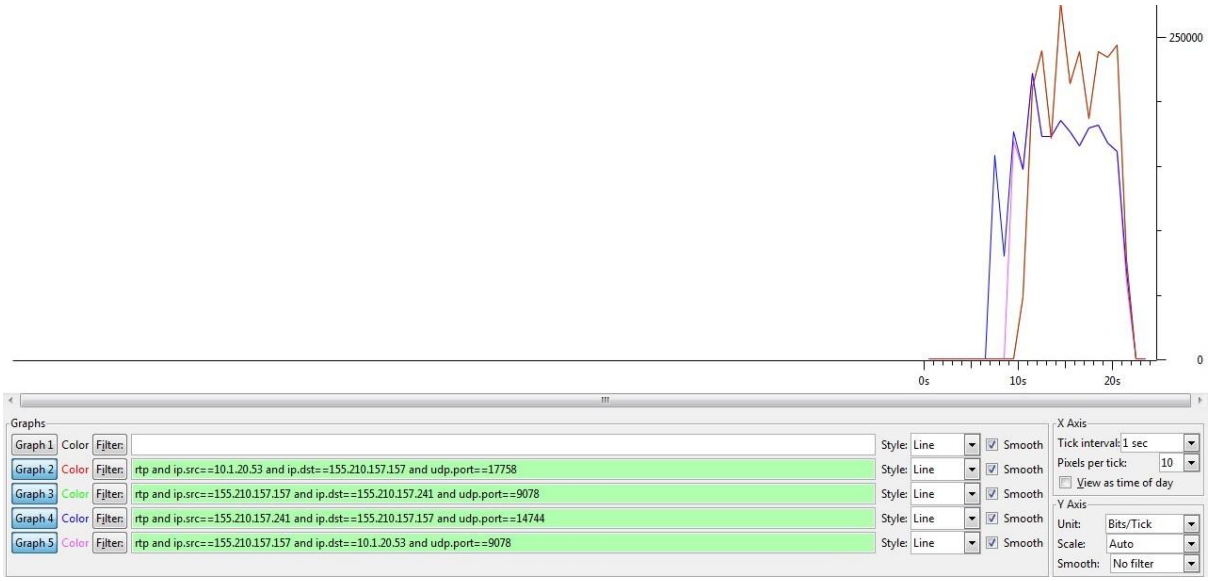


Figura 84 Gráficas de tráfico de video en una llamada con códec Vp8.

Como se puede ver en la figura 84 se han puesto 4 filtros:

1. Rtp and ip.src==10.1.20.53 and ip.dst==155.210.157.157 and udp.port==17758
2. Rtp and ip.src==155.210.157.157 and ip.dst==155.210.157.241 and udp.port==9078
3. Rtp and ip.src==155.210.157.241 and ip.dst==155.210.157.157 and udp.port==14744
4. Rtp and ip.src==155.210.157.157 and ip.dst==10.1.20.53 and udp.port==9078.

El filtro 1 grafica el tráfico que tiene como puerto destino el 17758. Dicho tráfico es de video y corresponde a los paquetes que han sido enviados desde el cliente con dirección 10.1.20.53 y hacia el servidor con dirección 155.210.157.157.

El filtro 2 grafica el tráfico que tiene como puerto destino el 9078. Dicho tráfico es de video y corresponde a los paquetes que han sido enviados desde el servidor con dirección 155.210.157.157 y hacia el cliente con dirección 155.210.157.241.

El filtro 3 grafica el tráfico que tiene como puerto destino el 14744. Dicho tráfico es de video y corresponde a los paquetes que han sido enviados desde el cliente con dirección 155.210.157.241 y hacia el servidor con dirección 155.210.157.157.

El filtro 4 grafica el tráfico que tiene como puerto destino el 9078. Dicho tráfico es de audio y corresponde a los paquetes que han sido enviados desde el servidor con dirección 155.210.157.157 y hacia el cliente con dirección 10.1.20.53.

Al igual que en casos anteriores, las gráficas de tráficos de subida y de bajada son muy similares por lo que sólo se pueden apreciar dos colores en la figura, sin embargo están representados los resultados de los 4 filtros. La gráfica azul es igual a la verde y la roja igual a la magenta.

Si se aplican los filtros del 1 al 4 a todo el tráfico capturado se puede calcular con exactitud el ancho de banda consumido por Vp8 en los cuatro flujos de información. Entiéndase los dos flujos de subidas y los dos de bajadas correspondientes a la comunicación. Los resultados obtenidos son los siguientes:

Para el filtro 1 se tiene que:

T= 956 bytes
P= 311
t= 10.5 s
AB1=(956 bytes-14 bytes)*311*8bit/10.5 s= 223209 bit/s=223.2 kbps.

Teniendo en cuenta las cabeceras restadas, el ancho de banda del filtro 2 sería de:

AB1=0.224 Mbps o lo que es lo mismo 224 kbps.

Se hace notar la diferencia entre ambos resultados.

Para el filtro 2 se tiene que:

T= 956 bytes
P= 311
t= 10.5 s
AB1=(956 bytes-14 bytes)*311*8bit/10.5 s= 223209 bit/s=223.2 kbps.

Teniendo en cuenta las cabeceras restadas, el ancho de banda del filtro 2 sería de:

AB1=0.224 Mbps o lo que es lo mismo 224 kbps.

Se hace notar la diferencia entre ambos resultados.

Para el filtro 3 se tiene que:

T= 925 bytes

P= 326

t=14.7 s

AB3=(925 bytes-14 bytes)*326*8bit /14.7 s= 16162 bit/s=161.6 kbps.

Teniendo en cuenta las cabeceras restadas, el ancho de banda del filtro 2 sería de:

AB3=0.163 Mbps o lo que es lo mismo 163 kbps

Se hace notar la diferencia.

Para el filtro 4 se tiene que:

T= 927 bytes
P= 291
t=12.2 s
AB4=(927 bytes-14 bytes)*291*8bit /12.2 s= 174218 bit/s=174.2 kbps.

Teniendo en cuenta las cabeceras restadas, el ancho de banda del filtro 2 sería de:

AB4=0.176 Mbps o lo que es lo mismo 176 kbps

Se hace notar la diferencia.

Agrupando los anchos de bandas calculados al aplicar los 4 filtros de la prueba se obtiene la siguiente información. Ver tabla 15

Tabla 12 Anchos de banda de Vp8

AB Filtro 1	223.2 Kbps
AB Filtro 2	223.2 Kbps
AB Filtro 3	161.6 Kbps
AB Filtro 4	174.2 Kbps

5.5. Prueba 5 Realización de una llamada cifrada utilizando códec g.722 y Vp8

La prueba 5 es una realización donde los códecs involucrados fueron g.722 y Vp8. Dicha prueba fue efectuada entre dos extensiones que pertenecen a una misma PBX. La figura 86 muestra un ejemplo que visualiza en qué lugar específico de la red se realizó la llamada.

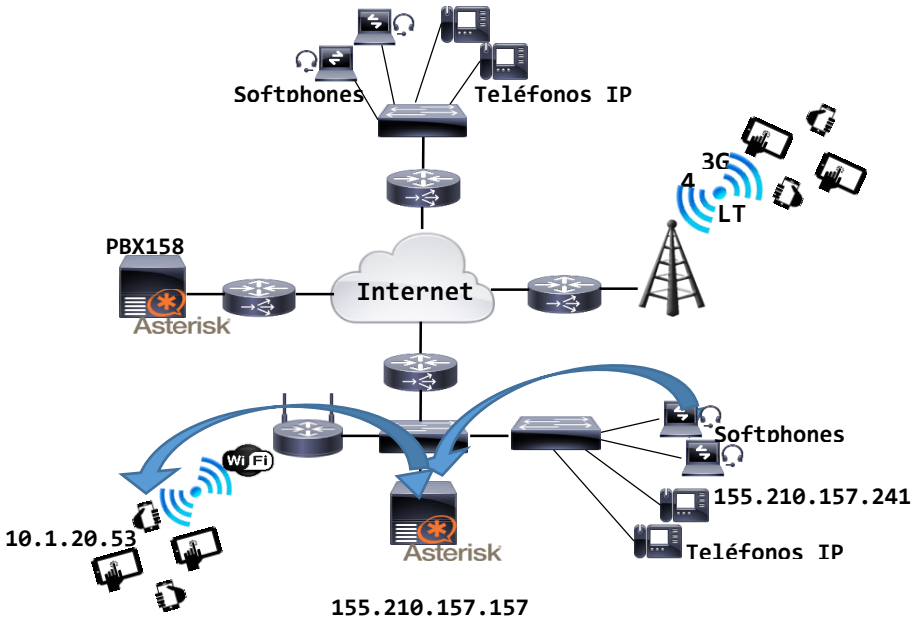


Figura 85 Escenario de llamada de la prueba 5

Basado en la herramienta de análisis y captura explicada en el capítulo 3 y con la conformación de diferentes filtros de información, se pudo capturar el tráfico de audio y de video tanto de subida como de bajada entre los clientes involucrados.

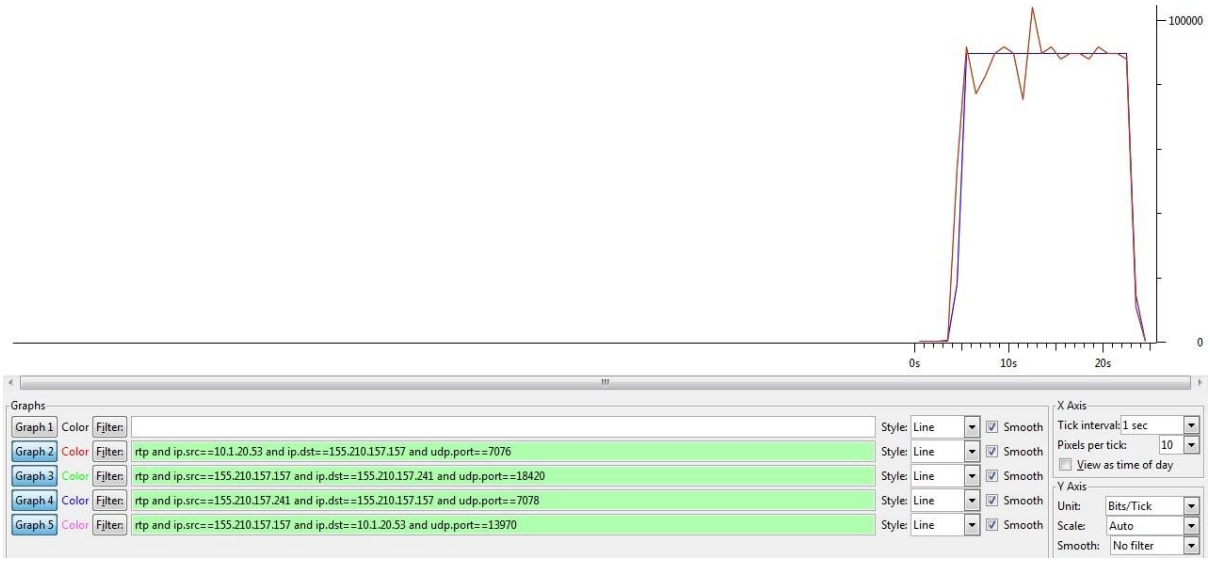


Figura 86 Gráficas de tráfico de audio cifrado en una llamada con códec g.722.

Como se puede ver en la figura 86 se han puesto 4 filtros:

5. Rtp and ip.src==10.1.20.53 and ip.dst==155.210.157.157 and udp.port==7076
6. Rtp and ip.src==155.210.157.157 and ip.dst==155.210.157.241 and udp.port==18420
7. Rtp and ip.src==155.210.157.241 and ip.dst==155.210.157.157 and udp.port==7078
8. Rtp and ip.src==155.210.157.157 and ip.dst==10.1.20.53 and udp.port==13970.

El filtro 1 grafica el tráfico que tiene como puerto destino el 7076. Dicho tráfico es de audio y corresponde a los paquetes que han sido enviados desde el cliente con dirección 10.1.20.53 y hacia el servidor con dirección 155.210.157.157.

El filtro 2 grafica el tráfico que tiene como puerto destino el 18420. Dicho tráfico es de audio y corresponde a los paquetes que han sido enviados desde el servidor con dirección 155.210.157.157 y hacia el cliente con dirección 155.210.157.241.

El filtro 3 grafica el tráfico que tiene como puerto destino el 7078. Dicho tráfico es de audio y corresponde a los paquetes que han sido enviados desde el cliente con dirección 155.210.157.241 y hacia el servidor con dirección 155.210.157.157.

El filtro 4 grafica el tráfico que tiene como puerto destino el 13970. Dicho tráfico es de audio y corresponde a los paquetes que han sido enviados desde el servidor con dirección 155.210.157.157 y hacia el cliente con dirección 10.1.20.53.

Tal cual como pruebas anteriores están representadas en la figura 11 los 4 filtros, no es perceptible los colores magenta y verde debido a que son similares a rojo y azul respectivamente.

Si se aplican los filtros del 1 al 4 a todo el tráfico capturado se puede calcular con exactitud el ancho de banda consumido por opus en los cuatro flujos de información. Entiéndase los dos flujos de subidas y los dos de bajadas correspondientes a la comunicación. Los resultados obtenidos son:

Para el filtro 1 se tiene que:

T= 224 bytes
P= 926
t= 18.7 s
AB1=(224 bytes-14 bytes)*926*8bit/18.7 s= 83191 bit/s=83.1 kbps.

Teniendo en cuenta las cabeceras restadas, el ancho de banda del filtro 2 sería de:

AB1=0.089 Mbps o lo que es lo mismo 89 kbps.

Se hace notar la diferencia entre ambos resultados.

Para el filtro 2 se tiene que:

T= 224 bytes
P= 926
t= 18.7 s
AB2=(224 bytes-14 bytes)*926*8bit/18.7 s= 83191 bit/s=83.1 kbps.

Teniendo en cuenta las cabeceras restadas, el ancho de banda del filtro 2 sería de:

AB2=0.089 Mbps o lo que es lo mismo 89 kbps.

Se hace notar la diferencia entre ambos resultados.

Para el filtro 3 se tiene que:

T= 224 bytes
P= 919
t=19.2 s
AB3=(224 bytes-14 bytes)*919*8bit /19.2 s= 80412 bit/s=80.4 kbps.

Teniendo en cuenta las cabeceras restadas, el ancho de banda del filtro 2 sería de:

AB3=0.086 Mbps o lo que es lo mismo 86 kbps

Se hace notar la diferencia.

Para el filtro 4 se tiene que:

T= 224 bytes
P= 916
t=18.2 s
AB4=(224 bytes-14 bytes)*916*8bit /18.2 s= 84553 bit/s=84.5 kbps.

Teniendo en cuenta las cabeceras restadas, el ancho de banda del filtro 2 sería de:

AB4=0.90 Mbps o lo que es lo mismo 90 kbps

Se hace notar la diferencia.

Agrupando los anchos de bandas calculados al aplicar los 4 filtros de la prueba se obtiene la siguiente información. Ver tabla 16

Tabla 13 Anchos de banda de g722

AB Filtro 1	83.1 Kbps
AB Filtro 2	83.1 Kbps
AB Filtro 3	80.4 Kbps
AB Filtro 4	84.5 Kbps

Haciendo un análisis similar al anterior se puede calcular el ancho de banda promedio para el codec de video Vp8. La figura 87 muestra una gráfica de 4 filtros de la captura de los paquetes correspondientes al códec Vp8.

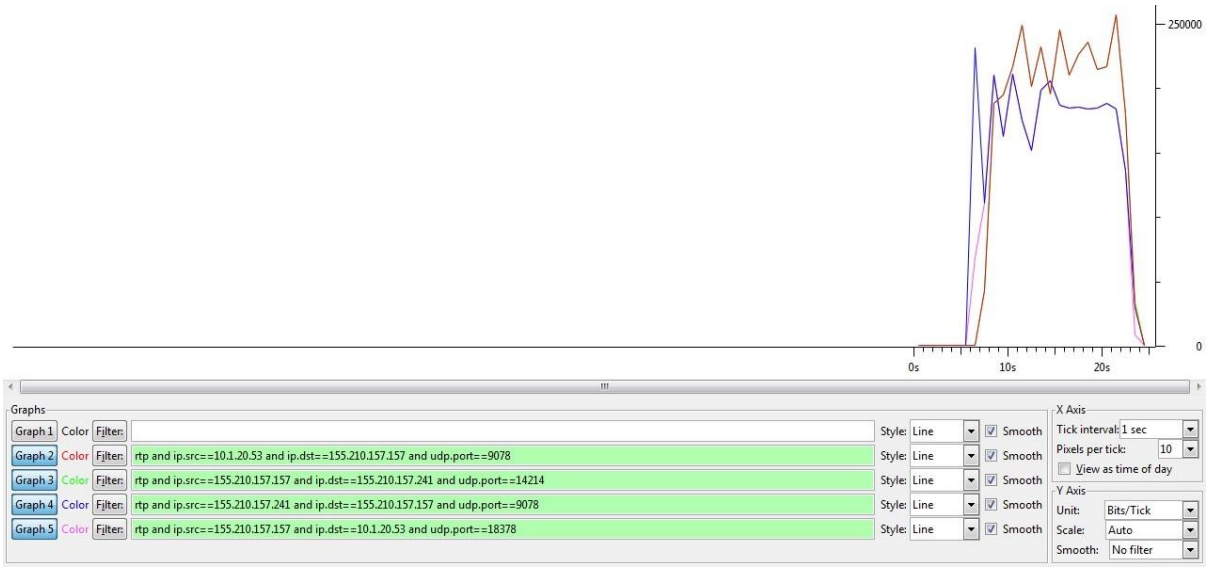


Figura 87 Gráficas de tráfico de video en una llamada con códec Vp8.

Como se puede ver en la figura 87 se han puesto 4 filtros:

1. Rtp and ip.src==10.1.20.53 and ip.dst==155.210.157.157 and udp.port==9078
2. Rtp and ip.src==155.210.157.157 and ip.dst==155.210.157.241 and udp.port==14214
3. Rtp and ip.src==155.210.157.241 and ip.dst==155.210.157.157 and udp.port==9078
4. Rtp and ip.src==155.210.157.157 and ip.dst==10.1.20.53 and udp.port==18378.

El filtro 1 grafica el tráfico que tiene como puerto destino el 9078. Dicho tráfico es de video y corresponde a los paquetes que han sido enviados desde el cliente con dirección 10.1.20.53 y hacia el servidor con dirección 155.210.157.157.

El filtro 2 grafica el tráfico que tiene como puerto destino el 14214. Dicho tráfico es de video y corresponde a los paquetes que han sido enviados desde el servidor con dirección 155.210.157.157 y hacia el cliente con dirección 155.210.157.241.

El filtro 3 grafica el tráfico que tiene como puerto destino el 9078. Dicho tráfico es de video y corresponde a los paquetes que han sido enviados desde el cliente con dirección 155.210.157.241 y hacia el servidor con dirección 155.210.157.157.

El filtro 4 grafica el tráfico que tiene como puerto destino el 18378. Dicho tráfico es de audio y corresponde a los paquetes que han sido enviados desde el servidor con dirección 155.210.157.157 y hacia el cliente con dirección 10.1.20.53.

Tal cual la explicación anterior, las 4 gráficas están representadas en la figura 109. El resto no puede ser mostrado por las mismas causas.

Si se aplican los filtros del 1 al 4 a todo el tráfico capturado se puede calcular con exactitud el ancho de banda consumido por Vp8 en los cuatro flujos de información. Entiéndase los dos flujos de subidas y los dos de bajadas correspondientes a la comunicación. Los resultados obtenidos son:

Para el filtro 1 se tiene que:

T= 951 bytes
P= 438
t= 15.1 s
AB1=(951 bytes-14 bytes)*438*8bit/15.1 s= 217433 bit/s=217,4 kbps.

Teniendo en cuenta las cabeceras restadas, el ancho de banda del filtro 1 sería de:

AB1=0.220 Mbps o lo que es lo mismo 220 kbps.

Se hace notar la diferencia entre ambos resultados.

Para el filtro 2 se tiene que:

T= 950 bytes
P= 439
t= 15.2 s
AB2=(950 bytes-14 bytes)*439*8bit/15.2 s= 216265 bit/s=216.2 kbps.

Teniendo en cuenta las cabeceras restadas, el ancho de banda del filtro 2 sería de:

AB2=0.219 Mbps o lo que es lo mismo 219 kbps.

Se hace notar la diferencia entre ambos resultados.

Para el filtro 3 se tiene que:

T= 927 bytes
P= 420
t=18.3 s
AB3=(927 bytes-14 bytes)*420*8bit /18.3 s= 167632 bit/s=167.6 kbps.

Teniendo en cuenta las cabeceras restadas, el ancho de banda del filtro 3 sería de:

AB3=0.170 Mbps o lo que es lo mismo 170 kbps

Se hace notar la diferencia.

Para el filtro 4 se tiene que:

T= 925 bytes
P= 396
t=16.6 s
AB4=(925 bytes-14 bytes)*396*8bit /16.6 s= 173858 bit/s=173.8 kbps.

Teniendo en cuenta las cabeceras restadas, el ancho de banda del filtro 4 sería de:

AB4=0.176 Mbps o lo que es lo mismo 176 kbps

Se hace notar la diferencia.

Agrupando los anchos de bandas calculados al aplicar los 4 filtros de la prueba se obtiene la siguiente información ver tabla 17

Tabla 14 Anchos de banda de Vp8

AB Filtro 1	217.4 Kbps
AB Filtro 2	216.2 Kbps
AB Filtro 3	167.6 Kbps
AB Filtro 4	173.8 Kbps

5.6. Prueba 6 Realización de una llamada cifrada utilizando códec g.711 (uLaw) y Vp8

La prueba 6 es una realización donde los códecs involucrados fueron g.711u y Vp8. Dicha prueba fue efectuada entre dos extensiones que pertenecen a una misma

PBX. La figura 89 muestra un ejemplo que visualiza en qué lugar específico de la red se realizó la llamada.

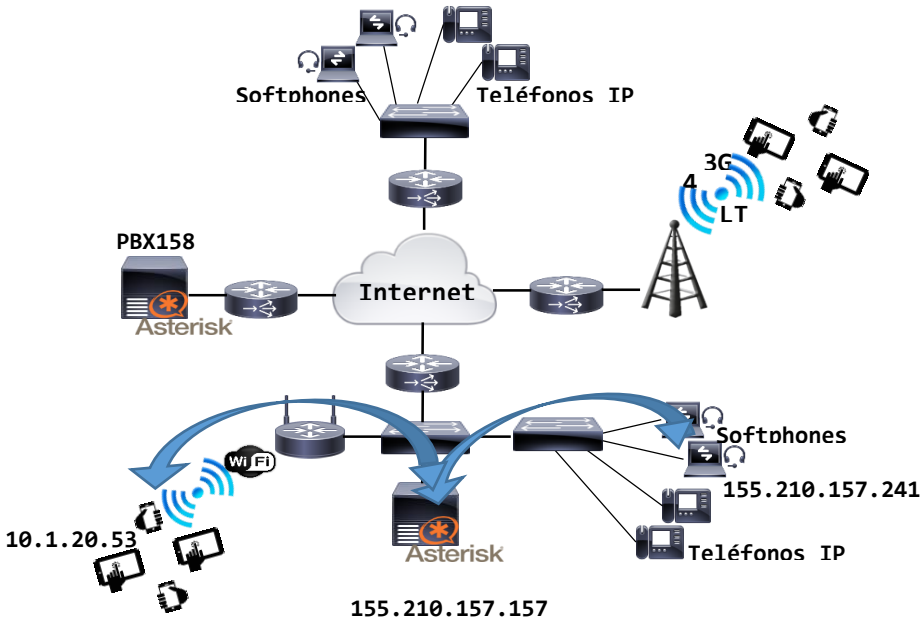


Figura 88 Escenario de llamada de la prueba 6.

Basado en la herramienta de análisis y captura explicada en el capítulo 3 y con la conformación de diferentes filtros de información, se pudo capturar el tráfico de audio y de video tanto de subida como de bajada entre los clientes involucrados.

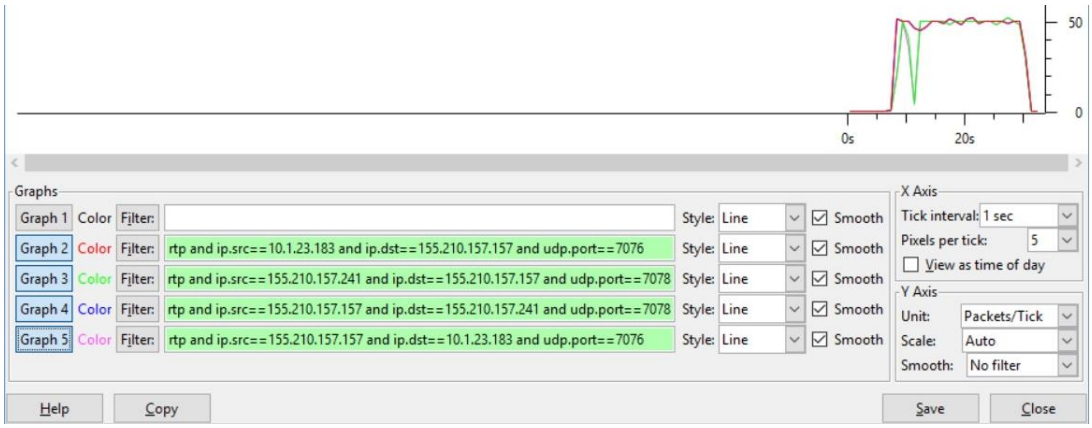


Figura 89 Gráficas de tráfico de audio cifrado en una llamada con códec g.711u.

Como se puede ver en la figura 89 se han puesto 4 filtros:

1. Rtp and ip.src==10.1.20.183 and ip.dst==155.210.157.157 and udp.port==7076
2. Rtp and ip.src==155.210.157.241 and ip.dst==155.210.157.157 and udp.port==7078
3. Rtp and ip.src==155.210.157.157 and ip.dst==155.210.157.241 and udp.port==7078
4. Rtp and ip.src==155.210.157.157 and ip.dst==10.1.20.183 and udp.port==7076.

El filtro 1 grafica el tráfico que tiene como puerto destino el 7076. Dicho tráfico es de audio y corresponde a los paquetes que han sido enviados desde el cliente con dirección 10.1.20.183 y hacia el servidor con dirección 155.210.157.157.

El filtro 2 grafica el tráfico que tiene como puerto destino el 7078. Dicho tráfico es de audio y corresponde a los paquetes que han sido enviados desde el servidor con dirección 155.210.157.157 y hacia el cliente con dirección 155.210.157.241.

El filtro 3 grafica el tráfico que tiene como puerto destino el 7078. Dicho tráfico es de audio y corresponde a los paquetes que han sido enviados desde el cliente con dirección 155.210.157.241 y hacia el servidor con dirección 155.210.157.157.

El filtro 4 grafica el tráfico que tiene como puerto destino el 7076. Dicho tráfico es de audio y corresponde a los paquetes que han sido enviados desde el servidor con dirección 155.210.157.157 y hacia el cliente con dirección 10.1.20.53.

Si se aplican los filtros del 1 al 4 a todo el tráfico capturado se puede calcular con exactitud el ancho de banda consumido por opus en los cuatro flujos de información. Entiéndase los dos flujos de subidas y los dos de bajadas correspondientes a la comunicación.

Para el filtro 1 se tiene que:

T= 224 bytes
P= 1121
t= 22.6 s
AB1=(224 bytes-14 bytes)*1121*8bit/22.6 s= 83330 bit/s=83.3 kbps.
Teniendo en cuenta las cabeceras restadas, el ancho de banda del filtro 2 sería de:

AB1=0.089 Mbps o lo que es lo mismo 89 kbps.

Se hace notar la diferencia entre ambos resultados.

Para el filtro 2 se tiene que:

T= 224 bytes
P= 1119
t= 22.6 s
AB2=(224 bytes-14 bytes)*1119*8bit/22.6 s= 83182 bit/s=83.1 kbps.

Teniendo en cuenta las cabeceras restadas, el ancho de banda del filtro 2 sería de:

AB2=0.089 Mbps o lo que es lo mismo 89 kbps.

Se hace notar la diferencia entre ambos resultados.

Para el filtro 3 se tiene que:

T= 224 bytes
P= 1042
t=23.0 s
AB3=(224 bytes-14 bytes)*1042*8bit /23.0 s= 76111 bit/s=76.1 kbps.

Teniendo en cuenta las cabeceras restadas, el ancho de banda del filtro 3 sería de:

AB3=0.081 Mbps o lo que es lo mismo 81 kbps

Se hace notar la diferencia.

Para el filtro 4 se tiene que:

T= 224 bytes
P= 1036
t=21.9 s
AB4=(224 bytes-14 bytes)*1036*8bit /21.9 s= 79473 bit/s=79.4 kbps.

Teniendo en cuenta las cabeceras restadas, el ancho de banda del filtro 4 sería de:

AB4=0.85 Mbps o lo que es lo mismo 85 kbps

Se hace notar la diferencia.

Agrupando los anchos de bandas calculados al aplicar los 4 filtros de la prueba se obtiene la siguiente información:

Tabla 15 Anchos de banda de G711 u

AB Filtro 1	83.3 Kbps
AB Filtro 2	83.1 Kbps
AB Filtro 3	76.1 Kbps
AB Filtro 4	79.4 Kbps

Haciendo un análisis similar al anterior se puede calcular el ancho de banda promedio para el codec de video Vp8. La figura 90 muestra una gráfica de 4 filtros de la captura de los paquetes correspondientes al códec Vp8.

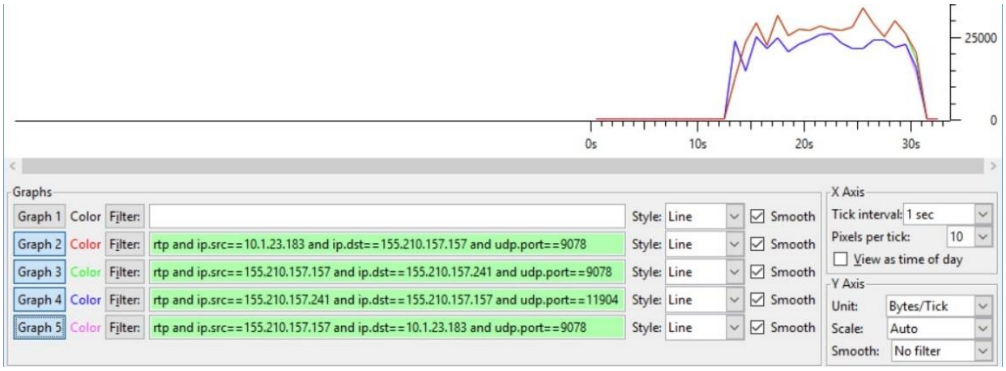


Figura 90 Gráficas de tráfico de video en una llamada con códec Vp8.

Como se puede ver en la figura 90 se han puesto 4 filtros:

1. Rtp and ip.src==10.1.23.183 and ip.dst==155.210.157.157 and udp.port==9078
2. Rtp and ip.src==155.210.157.157 and ip.dst==155.210.157.241 and udp.port==9078
3. Rtp and ip.src==155.210.157.241 and ip.dst==155.210.157.157 and udp.port==11904
4. Rtp and ip.src==155.210.157.157 and ip.dst==10.1.23.183 and udp.port==9078.

El filtro 1 grafica el tráfico que tiene como puerto destino el 9078. Dicho tráfico es de video y corresponde a los paquetes que han sido enviados desde el cliente con dirección 10.1.23.183 y hacia el servidor con dirección 155.210.157.157.

El filtro 2 grafica el tráfico que tiene como puerto destino el 9078. Dicho tráfico es de video y corresponde a los paquetes que han sido enviados desde el servidor con dirección 155.210.157.157 y hacia el cliente con dirección 155.210.157.241.

El filtro 3 grafica el tráfico que tiene como puerto destino el 11904. Dicho tráfico es de video y corresponde a los paquetes que han sido enviados desde el cliente con dirección 155.210.157.241 y hacia el servidor con dirección 155.210.157.157.

El filtro 4 grafica el tráfico que tiene como puerto destino el 9078. Dicho tráfico es de audio y corresponde a los paquetes que han sido enviados desde el servidor con dirección 155.210.157.157 y hacia el cliente con dirección 10.1.20.53.

Si se aplican los filtros del 1 al 4 a todo el tráfico capturado se puede calcular con exactitud el ancho de banda consumido por Vp8 en los cuatro flujos de información. Entiéndase los dos flujos de subidas y los dos de bajadas correspondientes a la comunicación.

Para el filtro 1 se tiene que:

T= 934 bytes
P= 506
t= 16.8 s
AB1=(934 bytes-14 bytes)*506*8bit/16.8 s= 221676 bit/s=221.6 kbps.

Teniendo en cuenta las cabeceras restadas, el ancho de banda del filtro 2 sería de:

AB1=0.224 Mbps o lo que es lo mismo 224 kbps.

Se hace notar la diferencia entre ambos resultados.

Para el filtro 2 se tiene que:

T= 934 bytes
P= 504
t= 16.8 s
AB2=(934 bytes-14 bytes)*504*8bit/16.8 s= 220800 bit/s=220.8 kbps.

Teniendo en cuenta las cabeceras restadas, el ancho de banda del filtro 2 sería de:

AB2=0.223 Mbps o lo que es lo mismo 223 kbps.

Se hace notar la diferencia entre ambos resultados.

Para el filtro 3 se tiene que:

T= 874 bytes
P= 461

t=18.7 s
AB3=(874 bytes-14 bytes)*461*8bit /18.7 s= 169608 bit/s=169.6 kbps.

Teniendo en cuenta las cabeceras restadas, el ancho de banda del filtro 2 sería de:

AB3=0.172 Mbps o lo que es lo mismo 172 kbps

Se hace notar la diferencia.

Para el filtro 4 se tiene que:

T= 878 bytes
P= 457
t=17.3 s
AB4=(878 bytes-14 bytes)*457*8bit /17.3 s= 182588 bit/s=182.5 kbps.

Teniendo en cuenta las cabeceras restadas, el ancho de banda del filtro 4 sería de:

AB4=0.184 Mbps o lo que es lo mismo 184 kbps

Se hace notar la diferencia.

Agrupando los anchos de bandas calculados al aplicar los 4 filtros de la prueba se obtiene la siguiente información:

Tabla 16 Anchos de banda de Vp8

AB Filtro 1	221.6 Kbps
AB Filtro 2	220.8 Kbps
AB Filtro 3	169.6 Kbps
AB Filtro 4	182.2 Kbps

5.7. Conlusiones

Recopilando la informacion de todas las pruebas realizadas se puede llegar a las siguientes conclusiones:

1. En cuanto al códec Vp8, tras analizar todas las pruebas, se deduce que consume un ancho de banda que se encuentra entre 156 kbps y 223 kbps. Las variaciones son debido a que las pruebas fueron realizadas desde distintas redes para simular la realidad por lo que en redes como FastEthernet consumirá un poco más de ancho de banda comparado con redes como Wifi. Esto se debe al carácter adaptativo del códec.
2. En cuanto a los códecs de audio más consumidor de ancho de banda, el G.722 es el primero en la lista, con ancho de banda que va de 80.4 kbps y 84.5 kbps. Cabe destacar otro códec con un consumo muy similar al de G.722 y es el G.711u con un ancho de banda que va de 76.1 kbps y 83.3 kbps.
3. En cuanto al codec de audio con menor consumo se tiene el speex, conun ancho de banada que va de 24.7 kbps a 34.9 kbps cuando esta cifrado y de 23 kbps a 30.9 kbps cuando no está cifrado. Con un consumo muy similar se tiene el codec GSM, con un consumo de 31.7 kbps y 33.2 kbps.
4. En cuanto al consumo de ancho de banda teniendo en cuenta el cifrado y el no cifrado se llega a la conclusion de que no existe una variacion significativa

en el envío de paquetes como para decir que al cifrar los datos se consume mucho más. La diferencia radica que se agregan 10 octetos más cuando se esta cifrado que cuando no se está. Por lo que el procedimiento de ponerle seguridad a las comunicaciones de VoIP no representa un gasto de recursos significativos.

5. Se cumplieron los objetivos enfocados a la seguridad de los servicios implementados, debido a que se demostró mediante las capturas realizadas la presencia del protocolo de cifrado SRTP.
6. En cuanto a la trasmisión de video llamdas seguras se pudo demostrar la presencia del codec Vp8 y de diverers códecs de audios embebidos en el protocolo SRTP.
7. Se cumplió con ofrecer al Sistema un método de seguridad antiataques cibernéticos que evitarían robo de información así como mal uso de la tecnología implementada.
8. Finalmente se demuestra que es posible implementar un sistema de video llamadas seguras utilizando una PBX-Asterisk.

.Referencias

1. Rehman, U.U. *Security Analysis of VoIP Architecture for Identifying SIP Vulnerabilities* School of Electrical Engineering and Computer Science National University of Sciences and Technology Islamabad, Pakistan 2014 [cited; Available from: http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=7021022&url=http%3A%2F%2Fieeexplore.ieee.org%2Fiel7%2F7011185%2F7020997%2F07021022.pdf%3Farnumber%3D7021022

2. Deters, R.K.L.a.R. *Intrusion Prevention in Asterisk-based Telephony System*. 2014 [cited; Available from: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6924302&newsearch=true&queryText=Intrusion%20Prevention%20in%20Asterisk-based%20Telephony%20System

3. Mohammad Hasanzadeh, H.H., Habibolah Asghari. *Plaintext transmission over Session Initiation Protocol*. 2014 [cited; Available from: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=7000781&queryText=Plaintext%20transmission%20over%20Session%20Initiation%20Protocol&newsearch=true

4. Sun, L. *Guide to Voice and Video over IP*. 2013 [cited; Available from: http://www.springer.com/us/book/9781447149040.

5. Gupta, P. *GSM and PSTN Gateway for Asterisk EPBX*. 2013 [cited; Available from: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6616225&newsearch=true&queryText=GSM%20and%20PSTN%20Gateway%20for%20Asterisk%20EPBX

6. Gupta, P. *SIP Server Security with TLS: Relative Performance Evaluation*. 2012 [cited; Available from: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6412062&newsearch=true&queryText=SIP%20Server%20Security%20with%20TLS:%20Relative%20Performance%20Evaluation

7. Oceguera, T. F. (2008). La VoIP en PBXs, su aplicación en Cuba., in Telecomunicaciones y Electrónica Telecomunicaciones. La Habana, Instituto Superior Politécnico José Antonio Echeverría.

8. R.G, C. (2008). Propuesta de un sistema que sirva como interfaz de conexión de un teléfono analógico a una computadora para tener VoIP. La Habana, Cuba, Instituto Superior Politécnico José Antonio Echeverría.
9. EscuderoPascual, A. (2010). "VoIP para el desarrollo. Una guía para crear una infraestructura de voz en regiones en desarrollo." International Development Research Center.
10. Store, S. C. (2015). "TELEFONO IP CISCO ". from http://www.computostore.com.mx/equipos-telefonia-ip/
11. Actual, M. (2015). "Teléfonos IP." from http://www.mercadoactual.es/telefonia/pda/telefonos-voip/.
12. Anónimo. (2008). "Equalcom."
13. español, C. d. u. d. A. e. (2015). "Asterisk-ES." from http://comunidad.asterisk-es.org/index.php?title=Introduccion_a_Asterisk.
14. Solutec. (2013). "Soluciones en Tecnología de la Información." from http://www.solutecperu.com/spsac/que-es-asterisk.
15. Handley H., y.o., *Request for Comments 2543 SIP: Session Initiation Protocol.* . 1999
16. Mesa, Y.V., *Conferencia de VoIP*. 2009, Cujae
17. John G. Van Bosse, y.o., *Signaling in Telecommunication Networks*. 2007, New Jersey: John Wiley & Sons, Inc.
18. Rosenberg J., y.o., *SCTP as a Transport for SIP.* 2001
19. Camarillo, G., *SIP Demystified*. 2002, New York McGraw-Hill Companies
20. Manuel Moreno Martín, y.o., *Una primera aproximación al protocolo SIP*.
21. Boucadair, M., *Inter-Asterisk Exchange (IAX): Deployment Scenarios in SIP-Enabled Networks*. John Wiley & Sons Ltd. ed. 2009, Francia: John Wiley & Sons, Ltd
22. M. Spencer, y.o., *Request for Comments: 5456 IAX: Inter-Asterisk eXchange Version 2,* 2010

23. Rodríguez, D. Protocolo RTP y Protocolo SIP. DEPARTAMENTO DE CIENCIAS EXACTAS Y NATURALES. Licenciatura en Ciencias de la Computación, UNIVERSIDAD DE SONORA. M.
24. Anónimo.(2013) "3Cu Electrónica." from https://sites.google.com/site/3cuelectronica/home/voip/rtp.
25. M. Baugher, D. M., Cisco Systems, Inc, M. Naslund, E. Carrara, K. Norrman, Ericsson Research (2013). "The Secure Real-time Transport Protocol (SRTP) RFC 3711."
26. brachmann, S. "Las ventajas de Centos."
27. Wikypedia (2015). “Vp8.” From https://es.wikipedia.org/wiki/VP8
28. Menéndez, J. J. (2007). "Tráfico RTP directo entre dispositivos con NAT (y parche)."
29. Galache, A. D. (2013). "VoIP Asterisk: Parametro Qualify."
30. Communications, B. (2015). "Linphone overview." from http://www.linphone.org/technical-corner/linphone/overview.
31. Wikypedia. (2015). "Tcpdump." from https://es.wikipedia.org/wiki/Tcpdump.
32. Combs, G. (2015). "Wireshark." from https://www.wireshark.org/.

.Bibliografía

- Oceguera, T. F. (2008). La VoIP en PBXs, su aplicación en Cuba., in Telecomunicaciones y Electrónica Telecomunicaciones. La Habana, Instituto Superior Politécnico José Antonio Echeverría.
- R.G, C. (2008). Propuesta de un sistema que sirva como interfaz de conexión de un teléfono analógico a una computadora para tener VoIP. La Habana, Cuba, Instituto Superior Politécnico José Antonio Echeverría.
- EscuderoPascual, A. (2010). "VoIP para el desarrollo. Una guía para crear una infraestructura de voz en regiones en desarrollo." International Development Research Center.
- Store, S. C. (2015). "TELEFONO IP CISCO ". from http://www.computostore.com.mx/equipos-telefonia-ip/
- Actual, M. (2015). "Teléfonos IP." from http://www.mercadoactual.es/telefonia/pda/telefonos-voip/.
- Anónimo. (2008). "Equalcom."
- español, C. d. u. d. A. e. (2015). "Asterisk-ES." from http://comunidad.asterisk-es.org/index.php?title=Introduccion_a_Asterisk.
- Solutec. (2013). "Soluciones en Tecnología de la Información." from http://www.solutecperu.com/spsac/que-es-asterisk.
- Handley H., y.o., *Request for Comments 2543 SIP: Session Initiation Protocol.* . 1999
- Mesa, Y.V., *Conferencia de VoIP*. 2009, Cujae
- John G. Van Bosse, y.o., *Signaling in Telecommunication Networks*. 2007, New Jersey: John Wiley & Sons, Inc.
- Rosenberg J., y.o., *SCTP as a Transport for SIP.* 2001
- Camarillo, G., *SIP Demystified*. 2002, New York McGraw-Hill Companies
- Manuel Moreno Martín, y.o., *Una primera aproximación al protocolo SIP*.

- Boucadair, M., *Inter-Asterisk Exchange (IAX): Deployment Scenarios in SIP-Enabled Networks*. John Wiley & Sons Ltd. ed. 2009, Francia: John Wiley & Sons, Ltd
- M. Spencer, y.o., *Request for Comments: 5456 IAX: Inter-Asterisk eXchange Version 2,* 2010
- Rodríguez, D. Protocolo RTP y Protocolo SIP. DEPARTAMENTO DE CIENCIAS EXACTAS Y NATURALES. Licenciatura en Ciencias de la Computación, UNIVERSIDAD DE SONORA. M.
- Anónimo.(2013) "3Cu Electrónica." from https://sites.google.com/site/3cuelectronica/home/voip/rtp.
- M. Baugher, D. M., Cisco Systems, Inc, M. Naslund, E. Carrara, K. Norrman, Ericsson Research (2013). "The Secure Real-time Transport Protocol (SRTP) RFC 3711."
- brachmann, S. "Las ventajas de Centos."
- Wikypedia (2015). “Vp8.” From https://es.wikipedia.org/wiki/VP8
- Menéndez, J. J. (2007). "Tráfico RTP directo entre dispositivos con NAT (y parche)."
- Galache, A. D. (2013). "VoIP Asterisk: Parametro Qualify."
- Communications, B. (2015). "Linphone overview." from http://www.linphone.org/technical-corner/linphone/overview.
- Wikypedia. (2015). "Tcpdump." from https://es.wikipedia.org/wiki/Tcpdump.
- Combs, G. (2015). "Wireshark." from https://www.wireshark.org/.
- Oliva, J. *Linux , Asterisk y Opensource*. 2010 [cited; Available from: http://jroliva.wordpress.com
- Castañeda, R., *Protocolos para voz IP*.
- Anónimo. *Voip Supply*. 2014 form http://www.voipsupply.com.
- Barberá,C.F. (2013) Análisis de prestaciones de un sistema de videoconferencia comercial., in Telecomunicaciones. Zaragoza, Dpto. Ingeniería Electrónica y Comunicaciones

- Royo, J.M. (2010) Análisis de un sistema CAC para telefonía IP. In Telecomunicaciones. Zaragoza. Dpto. Ingeniería Electrónica y Comunicaciones

. Anexos

Anexo A

A-1 Métodos SIP [38]

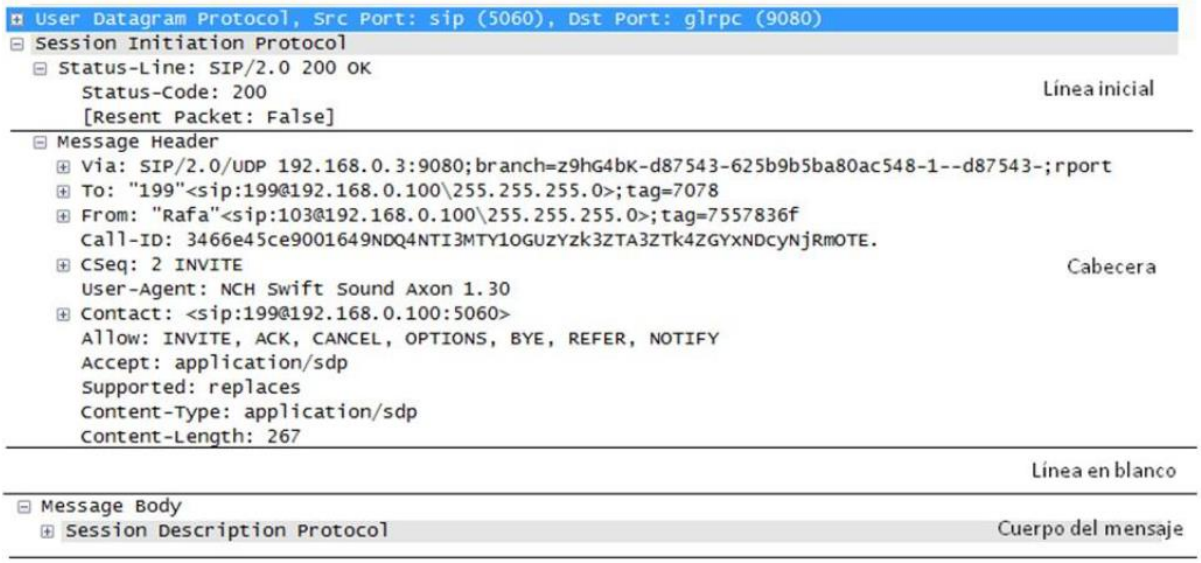
Método	Función
INVITE	Inicio de sesión.
ACK	Reconocimiento de INVITE
BYE	Terminación de sesión.
CANCEL	Cancelación de INVITE
REGISTER	Registro de URL
OPTIONS	Preguntar por opciones y capacidades
INFO	Trasporte de información en llamada
PRACK	Reconocimiento provisional
COMET	Notificación de precondición
REFER	Trasferencia a otra URL
SUSCRIBE	Requerir notificación de evento
UNSUSCRIBE	Cancelar notificación de evento
NOTIFY	Notificación de evento
MESSAGE	Mensaje instantáneo

A-2 Códigos SIP

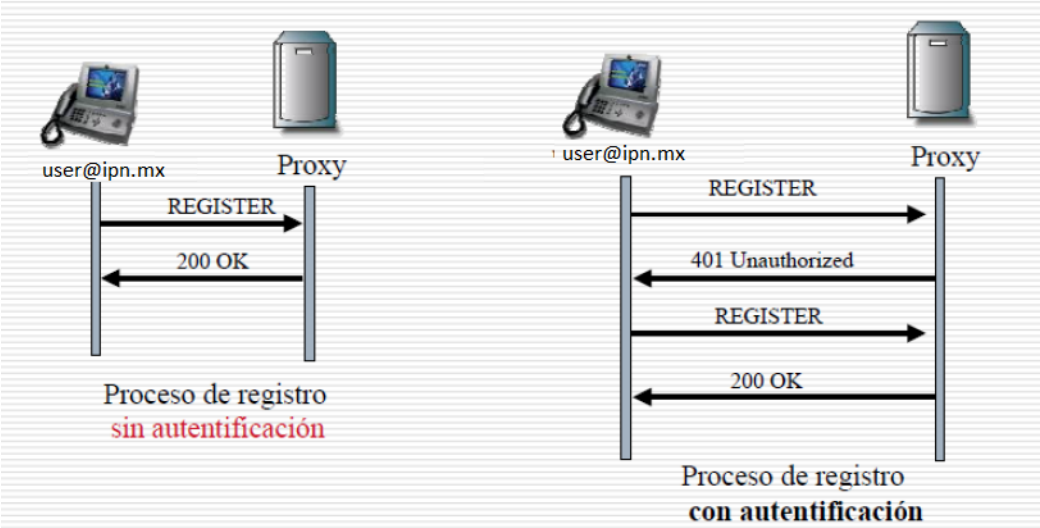
Código	Función
1xx	Información provisional, requerimiento en progreso pero no terminado
2xx	Completo: requerimiento completado satisfactoriamente
3xx	Redirección: petición debería redireccionarse
4xx	Error en de cliente (error en la petición)
5xx	Error de servidor
6xx	Falla Global

A-3 Paquete petición SIP

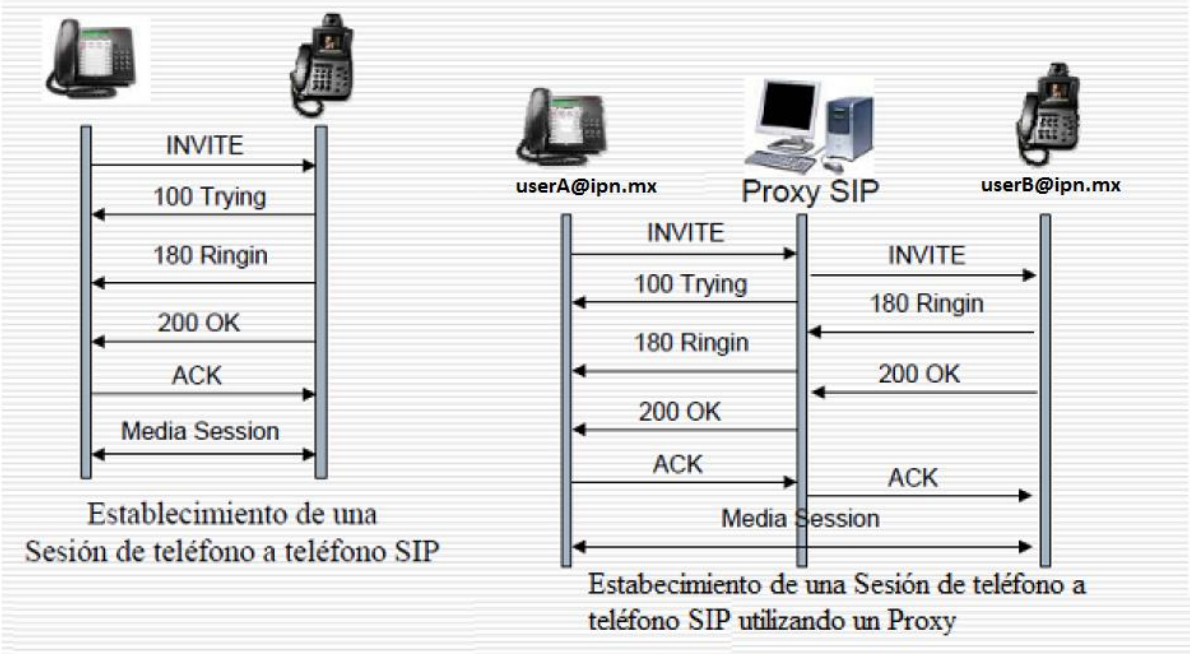
A-4 Paquete Respuesta SIP



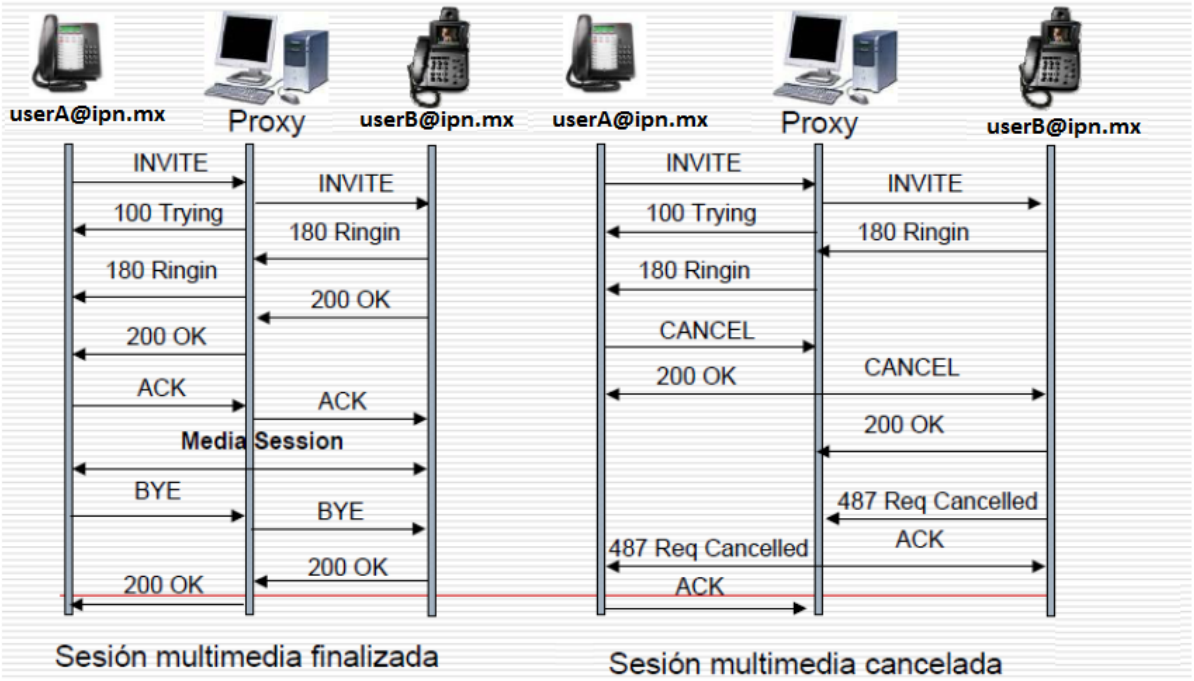
A-5 Proceso de Registro



A-6 Proceso de establecimiento de la sesión [38].



A-7 Proceso de finalización y de cancelación SIP[38]



Anexo B

B-1. Realización de una llamada sin cifrado entre dos extensiones de una misma PBX-Asterisk.

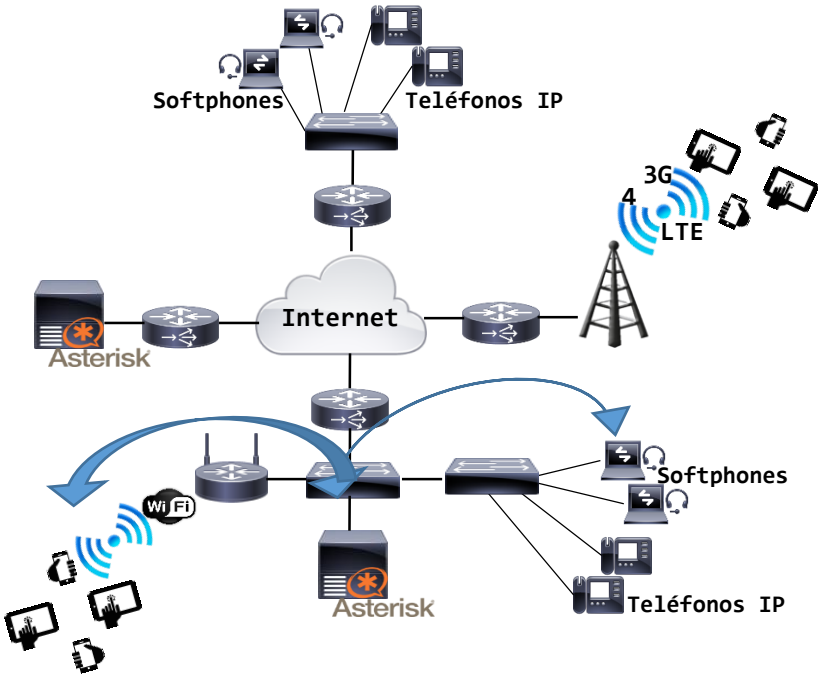


Figura 91 Entorno de una llamada mediante una misma PBX

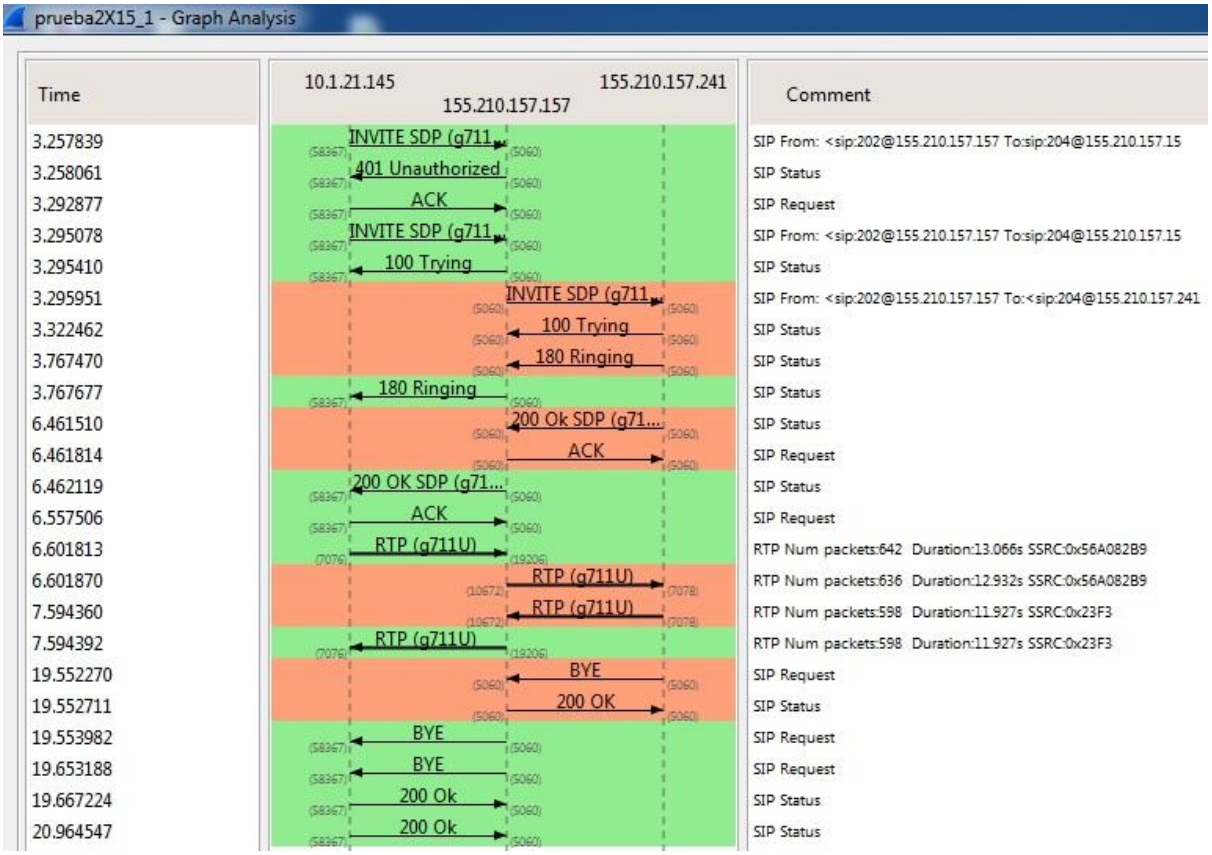


Figura 92 Diagrama de flujo de una llamada sin cifrar.

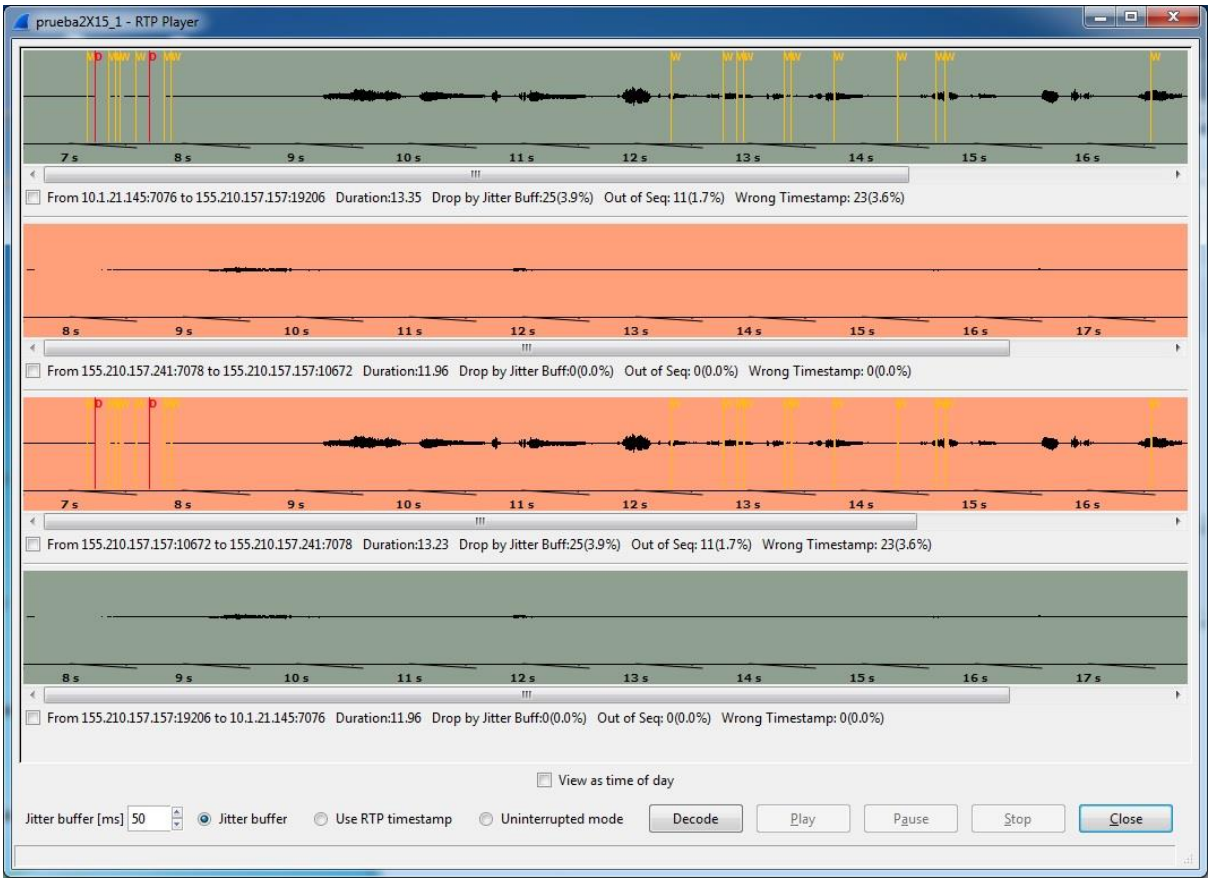


Figura 93 Captura de paquetes de audio de una llamada sin cifrar.

B-2. Realización de una llamada cifrada entre dos extensiones de una misma PBX-Asterisk.

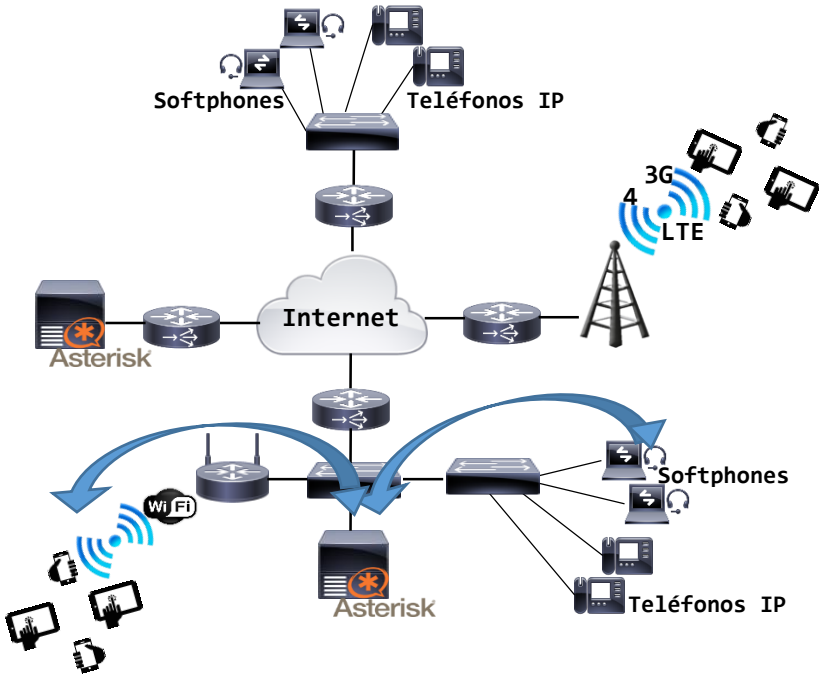


Figura 94 Entorno de una llamada mediante una misma PBX.

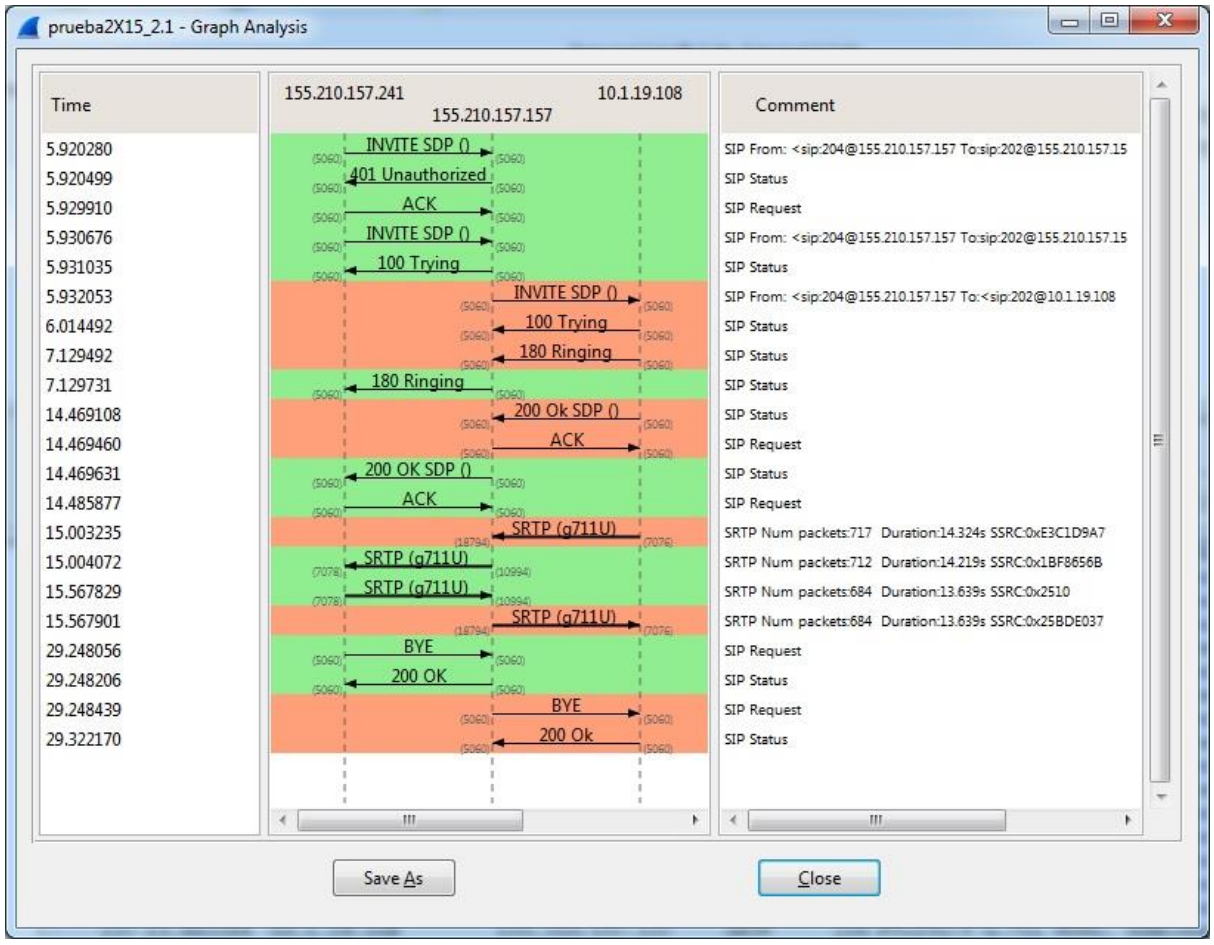


Figura 95 Diagrama de Flujo de una llamada cifrada

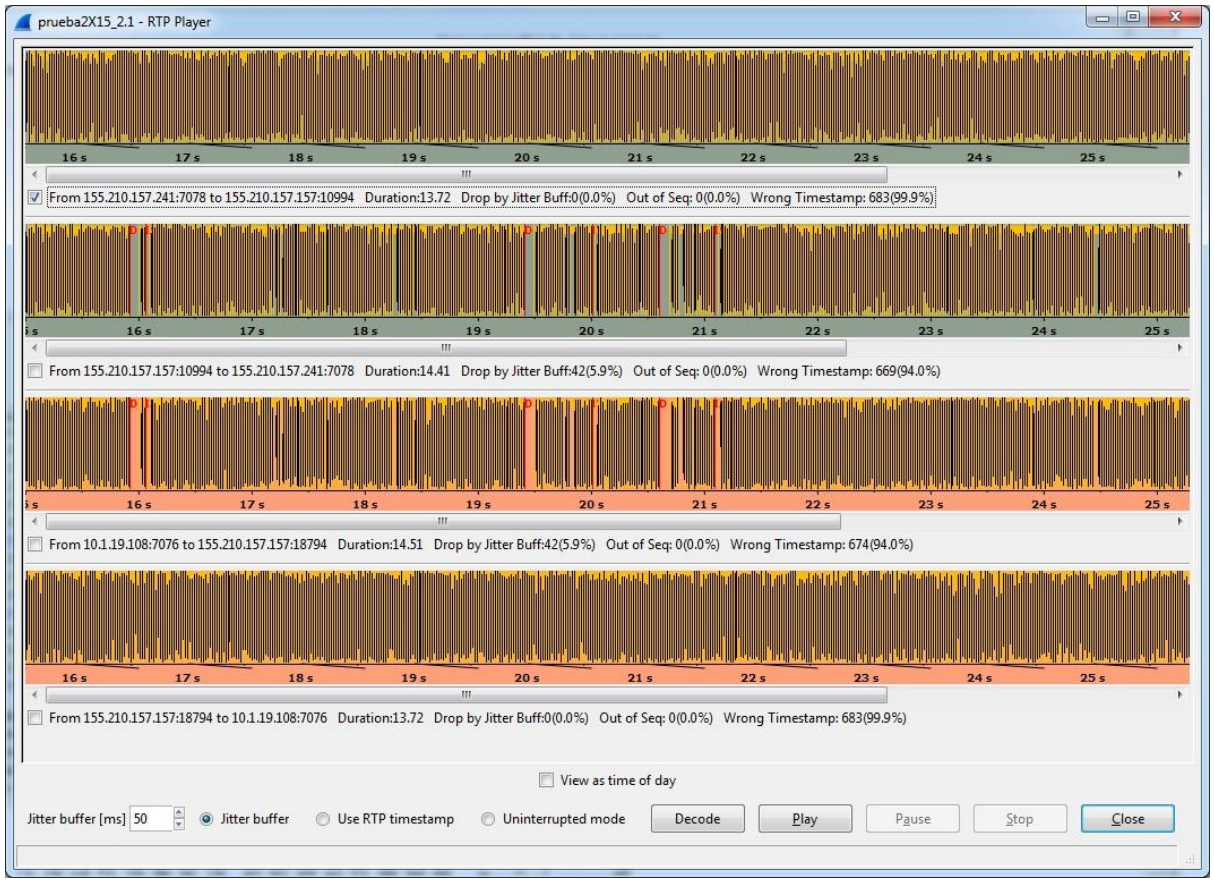


Figura 96 Captura de paquetes de audio de una llamada cifrada

B-5. Realización de una llamada de video sin cifrado entre dos extensiones una misma PBX-Asterisk.

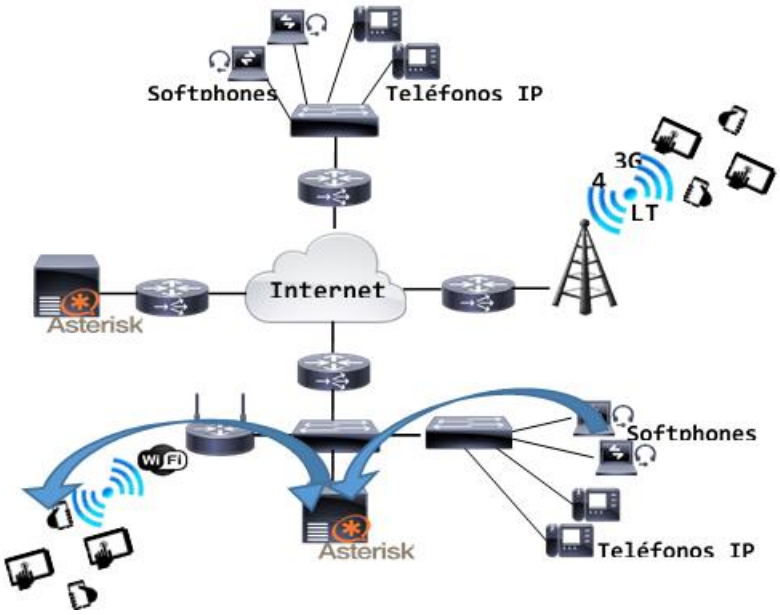


Figura 97 Entorno de llamada mediante una misma PBX.

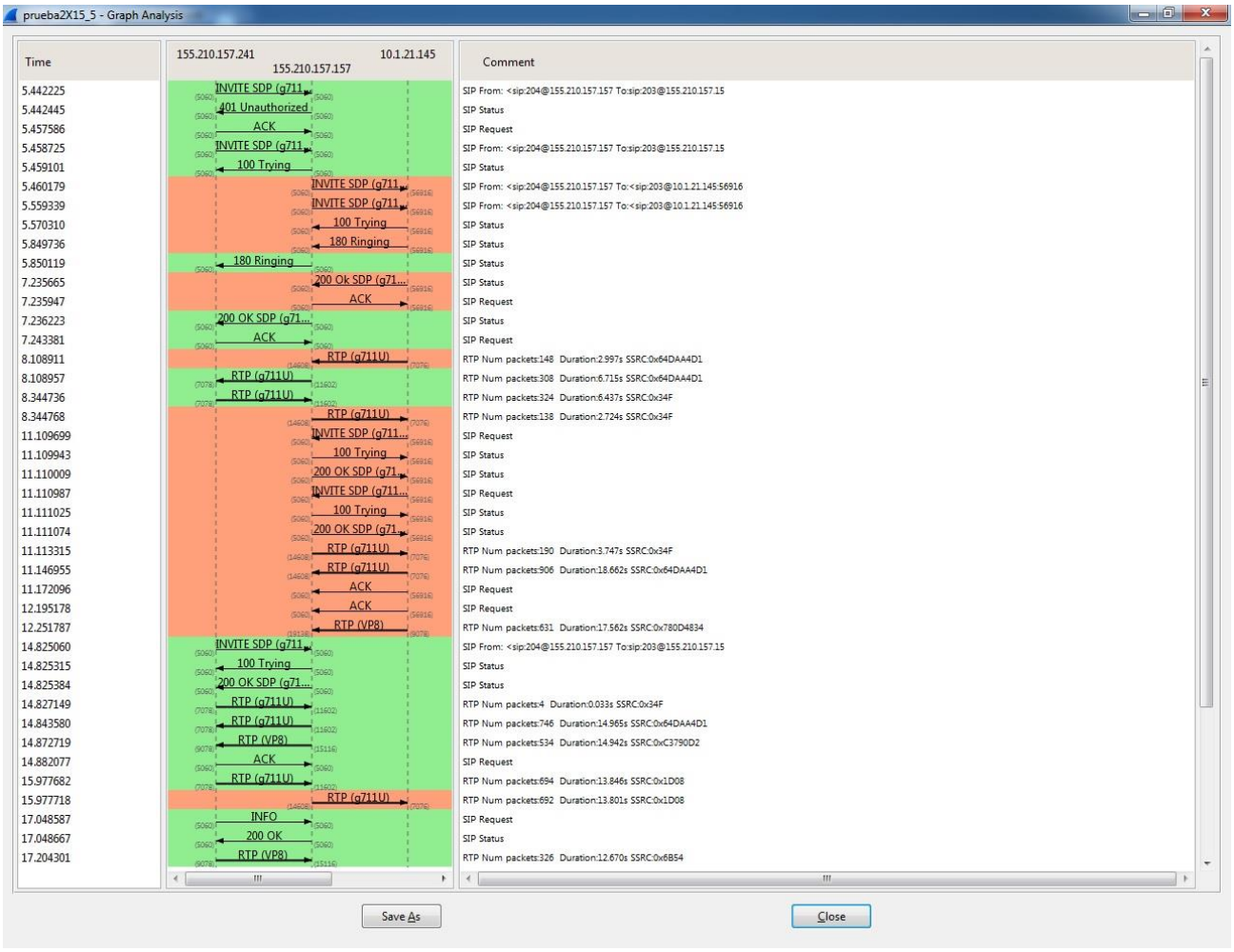


Figura 98 Diagrama del Flujo de audio y de video entre dos clientes.

B-6. Realización de una llamada de video cifrada entre dos extensiones una misma PBX-Asterisk.

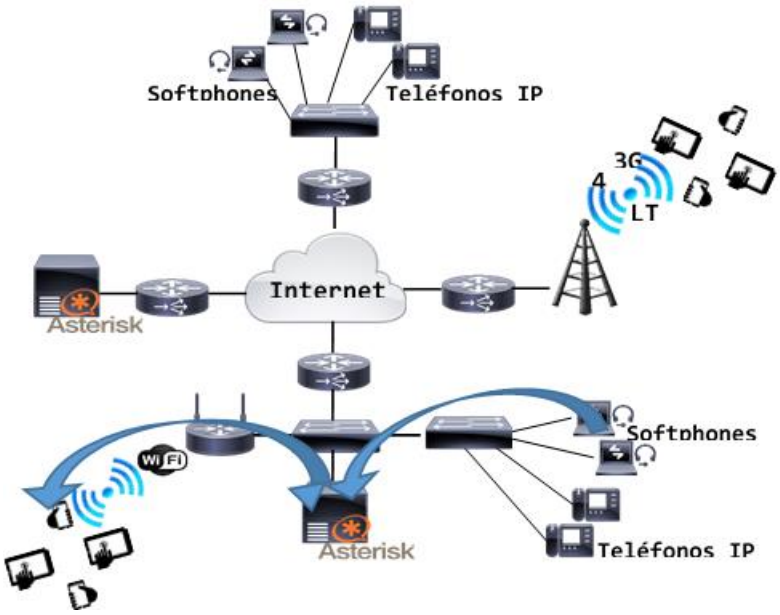


Figura 99 Entorno de llamada mediante una misma PBX.

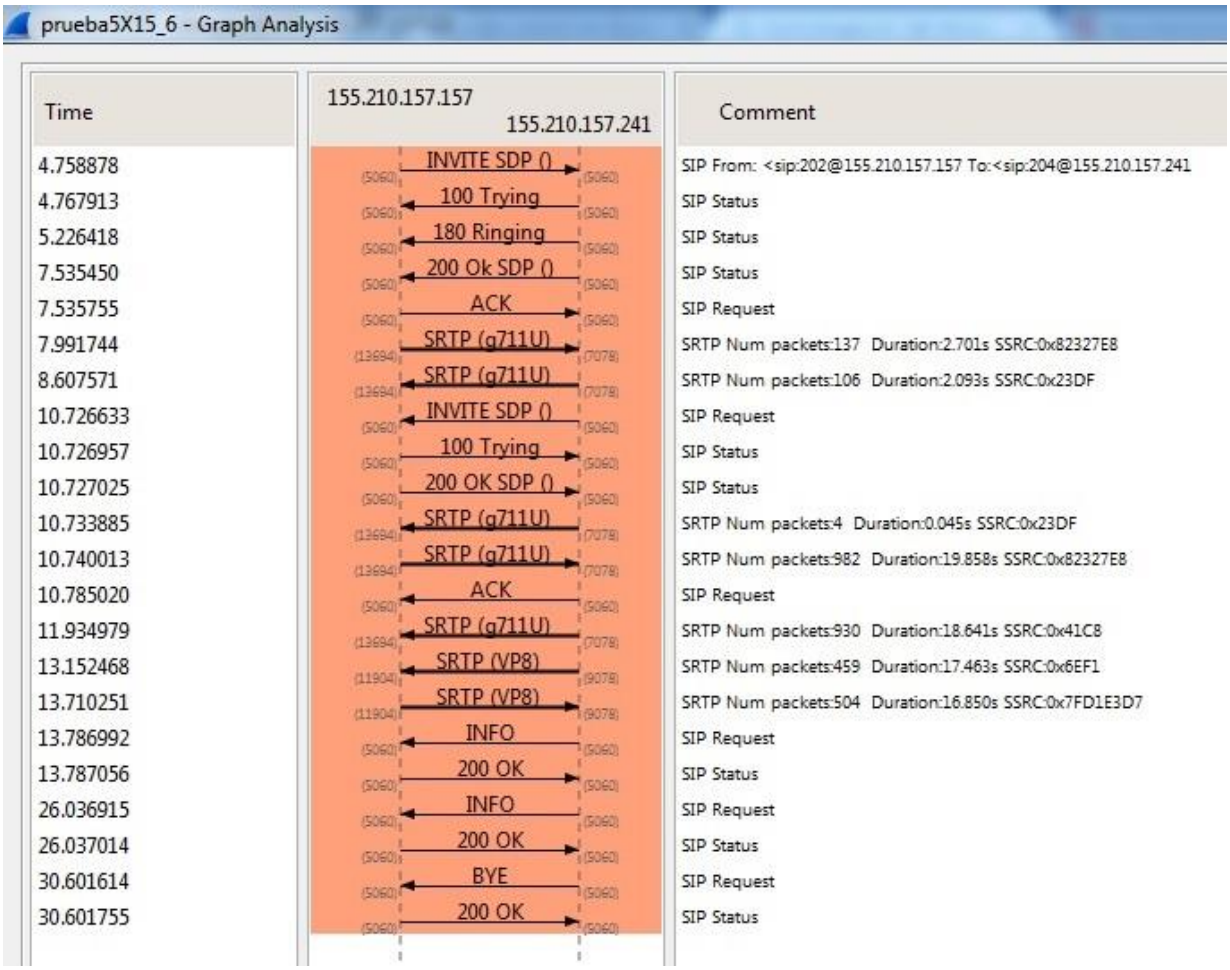


Figura 100 Diagrama de flujo SIP de llamada de video cifrada.

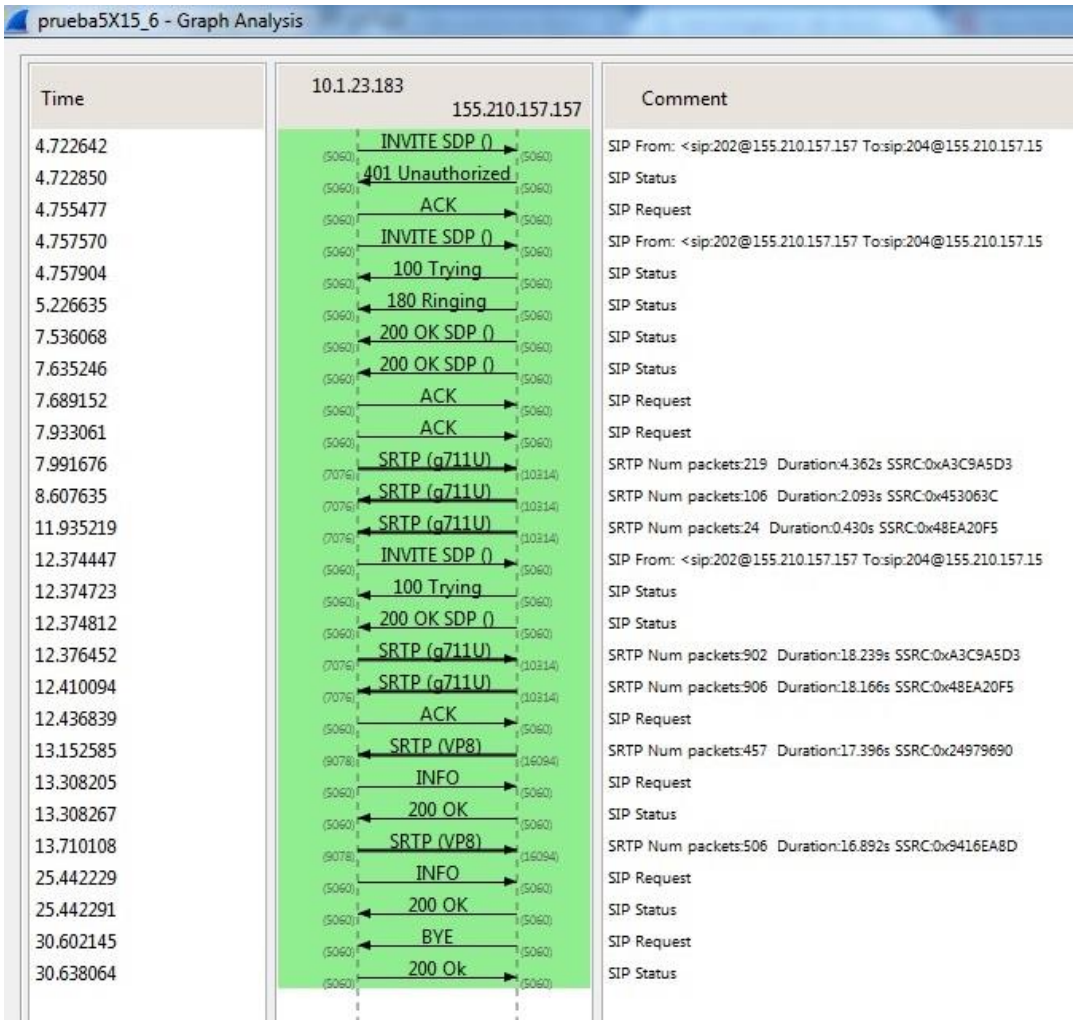


Figura 101 Diagrama de flujo SIP de llamada de video cifrada

Anexo C

C-1. Tabla Resumen de anchos de banda por códec.

Speex SRTP	Kbps	Vp8	Kbps	Opus	Kbps	Vp8	Kbps
AB Filtro 1	34.9	AB Filtro 1	222.5	AB Filtro 1	52.4	AB Filtro 1	223.2
AB Filtro 2	34.9	AB Filtro 2	234.6	AB Filtro 2	52.4	AB Filtro 2	223.2
AB Filtro 3	24.7	AB Filtro 3	156.2	AB Filtro 3	36.2	AB Filtro 3	161.6

AB Filtro 4	26.0	AB Filtro 4	168.0	AB Filtro 4	38.8	AB Filtro 4	174.2
Speex RTP	**Kbps**	**Vp8**	**Kbps**	**g722**	**Kbps**	**Vp8**	**Kbps**
AB Filtro 1	30.9	AB Filtro 1	222.1	AB Filtro 1	83.1	AB Filtro 1	217.4
AB Filtro 2	30.9	AB Filtro 2	223.2	AB Filtro 2	83.1	AB Filtro 2	216.2
AB Filtro 3	23.0	AB Filtro 3	177.9	AB Filtro 3	80.4	AB Filtro 3	167.6
AB Filtro 4	24.0	AB Filtro 4	174.5	AB Filtro 4	84.5	AB Filtro 4	173.8
GSM	**Kbps**	**Vp8**	**Kbps**	**g711u**	**Kbps**	**Vp8**	**Kbps**
AB Filtro 1	31.8	AB Filtro 1	194.8	AB Filtro 1	83.3	AB Filtro 1	221.6
AB Filtro 2	31.9	AB Filtro 2	194.8	AB Filtro 2	83.1	AB Filtro 2	220.8
AB Filtro 3	31.7	AB Filtro 3	169.6	AB Filtro 3	76.1	AB Filtro 3	169.6
AB Filtro 4	33.2	AB Filtro 4	176.3	AB Filtro 4	79.4	AB Filtro 4	182.2
Speex Iax2	**Kbps**	**GSM Iax2**	**Kbps**				
AB Filtro 1	28.1	AB Filtro 1	26.0				
AB Filtro 2	28.5	AB Filtro 2	26.3				
AB Filtro 3	23.8	AB Filtro 3	33.4				
AB Filtro 4	26.3	AB Filtro 4	33.4				

C-2. Relación de anchos de banda por códec.