PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

FACULTAD DE INGENIERÍA

MAESTRÍA EN REDES DE COMUNICACIÓN

TRABAJO PREVIO A LA OBTENCION DEL TÍTULO DE:

MASTER EN REDES DE COMUNICACIÓN

TEMA:

"ESTUDIO DE LAS CARACTERISTICAS DE NUEVAS ARQUITECTURAS WEB BASADAS EN WEBRTC ALOJADA EN LA NUBE Y FACTIBLE IMPLEMENTACION PARA APLICACIONES DE VOZ SOBRE IP (VoIP)."

AUTOR:

DANNYLL MICHELLC ZAMBRANO ZAMBRANO

DIRECTOR:

ING. GUSTAVO CHAFLA ALTAMIRANO, Ph. D

Quito – 2015

DEDICATORIA

Al ser Creador de la Vida, que permite visualizar la luz y vivir experiencias cada día.

A mi Familia; particularmente a mi esposa Yohanna, a mis hijos Dannyll y Dylan; para que vean su porvenir en la preparación y la Academia.

A mis Padres, fuente inagotable de apoyo y consejos.

AGRADECIMIENTO

A mi Familia; esposa e hijos, por ser apoyo constante y no permitirme NUNCA desmayar para alcanzar las metas trazadas.

Al Sr. Ing. Gustavo Chafla A. Ph. D, por sus valioso aporte a este trabajo y responder cada interrogante planteada en el desarrollo de la tesis.

A los docentes de la maestría en la PUCE y a todos quienes de una u otra forma estuvieron atentos y apoyando este trabajo de investigación.

Dannyll Michellc Zambrano Z.

INDICE

Resumen

El trabajo presenta un estudio de una arquitectura WEB para aplicaciones de red como es el caso de Voz sobre IP (VoIP), mediante servidores WebRTC en la nube, que ofrecen un servicio gratuito o de pago.

Mediante un navegador web, por ejemplo Google Chrome, Mozilla Firefox y Opera, que soportan WebRTC; se han realizados llamadas desde dos locaciones geográficas distantes y mediante las herramientas "webrtc-internals" y Wireshark se han tomado valores de Jitter, RTT y perdidas de paquetes para establecer la QoS en la transmisión de paquetes de VoIP.

Con los parámetros analizados se establece la factibilidad de tener comunicación de VoIP mediante WebRTC, valiéndose de servidores WebRTC en la Nube.

Palabras Claves: VoIP, WebRTC, Jitter, RTT, Perdida de Paquetes, Navegadores Web, QoS

Abstract

This research paper presents a study of a WEB architecture for network applications such as Voice over IP (VoIP), WebRTC using cloud servers, that they offer free or paid service.

Using a web browser, eg Google Chrome , Mozilla Firefox and Opera , which support WebRTC; have made calls from two distant geographic locations and through " WebRTC - internals " and Wireshark tools have been taken values Jitter, RTT , and packet loss for establishing QoS in the VoIP packet transmission.

With the parameters analyzed the feasibility of having WebRTC communication via VoIP, using WebRTC cloud servers is established.

Keywords: VoIP, WebRTC , Jitter, RTT , packet loss , Web Browsers , QoS

Introducción

El trabajo presenta un estudio de una arquitectura WEB para aplicaciones de red como es el caso de Voz sobre IP (VoIP), tomando como referencia los modelos de la Computación en la Nube, siendo una nueva tecnología que ha permitido desplegar un mercado de servicios por demanda, que ha adquirido gran popularidad, lo que ha llevado a los usuarios a utilizar soluciones sustentados en ella; realizando un análisis del software web que se prevé utilizar en base a lineamientos y directrices de diseños de redes que dan sustento a la infraestructura de las mismas.

WebRTC (Web Real Time Communications) es una herramienta que aparece para implementar comunicación en tiempo real a través de cualquier dispositivos electrónicos conectado a la WEB, de una manera sencilla, rápida y eficaz, acorde a la convergencia de redes y aplicaciones que se tiene hoy en día; siendo un proyecto de código abierto auspiciado por Google y basado en estándares definidos por el W3C (World Wide Web Consortium) y el IETF. En razón de esta tendencia y con el surgir de nuevas tecnologías que se les podría denominar emergentes, es necesario también definir nuevas arquitecturas WEB, para aplicaciones nativas de escritorio que migran a aplicaciones WEB.

Este trabajo se enfoca en el desarrollo de una metodología de aplicaciones en un ambiente WEB para permitir al Usuario una nueva oportunidad de acercarse a las comunicaciones modernas, ya que en esencia WebRTC permite la comunicación mediante los navegadores, facilitando la interacción de múltiples formas y dentro de un mismo contexto.

Este perfil de trabajo de tesis para la maestría de Redes de Comunicaciones presenta una justificación y planteamiento del problema enfocado en documentos académicos desarrollados que guardan relación con el tema, se establece el alcance y los objetivos de la investigación sustentados en una metodología de investigación y planteamiento de una hipótesis; en la parte final se plantea un temario tentativo, el cronograma y la bibliografía que dan forma y sustento a la presentación del trabajo de investigación.

Justificación

Desde principios del año 2000 las organizaciones y proveedores han sabido aprovechar los beneficios de la aplicación de Voz sobre IP (VoIP) permitiendo que los usuarios se comuniquen, siempre y cuando exista conectividad a Internet, desde diversas locaciones en todo el globo terrestre, prestando un servicio a usuarios y clientes prácticamente sin límites, de esta manera la aplicación de Voz sobre IP (VoIP) hoy en día con las soluciones en la Nube y el desarrollo de la Web RTC, puede desplegar todo su potencial.

Para conectarse al Internet se ha necesitado de un navegador, los cuales constantemente reciben mejoras en forma de actualizaciones y/o nuevas versiones, con lo que se mejora el rendimiento y amplía la gama de opciones para los usuarios, de ahí que en la actualidad se plantea la comunicación multimedia (audio y video) independientemente de complementos externos denominados plugins, basados en nuevos estándares Web que proveen al navegador de aquellas características como por ejemplo HTML 5, CCS3, etc.

Uno de los grandes retos para la web actual, es permitir la comunicación humana a través de voz y video en tiempo real (RTC - Real time Communication); históricamente éste tipo de mecanismos y tecnologías han estado bajo la tutela de sectores corporativos, haciendo uso de infraestructuras tecnológicas complejas y en algunos casos con requerimientos de licencias de audio y video para ser puestas en funcionamiento en sitio [1]

La tecnología de WebSockets y el desarrollo de WebRTC han posibilitado que aplicaciones conocidas de comunicación se puedan dar en tiempo real; siendo WebRTC un proyecto de código abierto desarrollado por Google y a partir de los estándares definidos por el W3C (World Wide Web Consortium) y el IETF, que se basa en permitir a los desarrolladores web la capacidad de desarrollar aplicaciones multimedia (como por ejemplo, vídeo o chat), sobre navegadores, con capacidad de comunicación en tiempo real y sin necesidad de descarga de ningún tipo de plugins o aplicación adicional [2]. Esto ha permitido que variables dependientes como la seguridad o tipo de códecs, para la comunicaciones de aplicaciones multimedia, hayan

disminuido y presenten menos inconvenientes; de ahí que este estándar soportado por algunos navegadores (Chrome, Chrome Canary, Firefox, Mozilla, Opera...) [2] permitan de una forma sencilla y fácil la conexión de equipos terminales y/o finales como son móviles, tablets y portátiles.

Actualmente, las aplicaciones que dan servicio mediante comunicaciones en tiempo real están compuestas por tres elementos principales: un framework (dispositivo/sistema operativo del usuario); una interfaz de usuario; y una media engine, responsable de la transmisión/recepción de los archivos utilizados para dicha comunicación, en otras palabras la media engine se encarga de las funciones que permiten entregar audio y vídeo con la calidad esperada [3]

WebRTC realiza el papel de media engine, basándose de una serie de estándares y APIs, para así obtener una nueva solución a las comunicaciones en tiempo real basada en la utilización directa de aplicaciones sobre los navegadores [3]

Adicionalmente, se debe considerar que WebRTC está ligado a la existencia de HTML5, la cual permite el manejo de Websockets, o expresado de otra manera permite que una vez si se produce una variante en el servidor, esa variante se manifieste y se implemente en el cliente, sin que se realice la petición cliente-servidor [4].

En razón de lo manifestado en párrafo anteriores, se justifica la importancia de este estudio debido al rápido crecimiento de los alojamientos en la nube y la dependencia de una gran mayoría de usuarios de aplicaciones y servicios desde la web, que gracias a esta tecnología WebRTC, la cual únicamente es un medio para alcanzar un objetivo, combina la solución en la Nube junto con la Voz sobre IP (VoIP) lo que desde un punto de vista económico, permite a una organización, empresa o usuario en general el ahorro de costos, aunado a la parte tecnológica que permite de una manera sencilla, rápida y eficaz, mediante un único navegador que funciona sin necesidad de tener que instalar plugins ni infraestructuras físicas, a parte del propio dispositivo (PC, Tablet, Móvil); la posibilidad de comunicación en tiempo real.

Planteamiento del Problema

Ante la exigencia de comunicación en tiempo real de una manera rápida, eficaz y sencilla aunado al vertiginoso desarrollo de aplicaciones Web, se plantea un nuevo escenario muy alentador para la comunicación mediante WebRTC.

WebRTC es una tecnología introducido por Google para implementar en la Web capacidades de comunicación en base a aplicaciones de Voz sobre IP (VoIP), además de garantizar que el sistema logre enviar datos (texto, imágenes, captura de pantalla, etc.) y lograr brindar al "usuario de la web" de un interfaz de comunicaciones sin que sea necesario la instalación de software externo, plugins y/o softphones [5]; por lo tanto Web RTC proporciona la comunicación entre el usuario final o en otras palabras el usuario que ejecuta el navegador y el servidor donde se encuentra alojado el servicio web, una forma de transmitir información que permita emprender una comunicación Cliente-Servidor, además que facilite la conexión con otros clientes y o usuarios que también estén accediendo al mismo servicio web. [6]

Es necesario dejar en claro que el objetivo de WebRTC no es ser un "softphone web", la concepción propia abarca más allá que aquello, más bien concibe a la Voz sobre IP (VoIP) desde el punto de vista donde la web tenga un papel preponderante y fundamental en relación a lo que representa la "VoIP pura" y no ligada coyunturalmente a la Telefonía IP [7], lo que supondría un cambio en la arquitectura de red ya que podría suceder que muchas empresas que actualmente viven de las llamadas telefónicas, tengan que migrar y ofrecer nuevos servicios.[8]

La principal ventaja para los operadores con WebRTC es que se abren sus infraestructuras de VoIP y SIP a una nueva red de acceso: la Web 2.0. Los medios en WebRTC van de navegador a navegador directamente, porque se reduce la latencia y se emplea más eficientemente el ancho de banda [9]. Sin embargo, la señalización va a través del servidor Web, lo cual facilita además la interoperabilidad con sistemas fuera de Internet. Para la interconexión de estos dos mundos, serán necesarios "gateways", que permitan la interoperabilidad, seguridad y control

de sesiones de los navegadores a los dispositivos SIP, tanto a nivel de señalización como de medios [10].

Tomando en consideración lo mencionado, el problema radica en que no se cuenta con datos e información mediante un estudio que presenten las características de una arquitectura web sustentada en WebRTC, concebida no como un softphone sino que permita conocer la viabilidad de implementar aplicaciones en tiempo real como por ejemplo Voz sobre IP (VoIP) mediante aplicaciones WEB y represente una opción para el usuario final.

OBJETIVOS

Objetivo General:

Investigar las características de las arquitecturas Web modernas basadas en WebRTC y alojamiento en la Nube, para en base al estudio establecer la viabilidad de implementar comunicaciones en tiempo real específicamente Voz sobre IP (VoIP).

Objetivos Específicos:

1. Analizar el estado del arte referente a computación en la nube, respecto a la arquitectura y almacenamiento en la actualidad.
2. Establecer las características y reconocer las ventajas que representa la aplicación de Voz sobre IP (VoIP), priorizando en los estudios sobre alojamiento en la nube.
3. Investigar el estado actual de la nueva tecnología WebRTC, sus características y analizar una estructura de red funcional que soporte la misma.
4. Investigar y concebir una posible arquitectura Web que utilice tecnología WebRTC, de manera que permita la implementación de aplicaciones Web con capacidad para aplicaciones como Voz sobre IP (VoIP).

CAPÍTULO 1

Introducción

En el Capítulo 1 se contextualizan los elementos sobre los que se fundamenta la Computación en la Nube, haciendo énfasis en los modelos de servicios que se tienen y los modelos de despliegue en una arquitectura que presenta beneficios y riesgos, donde se deben implementar seguridades para su funcionamiento.

1. Estado del arte arquitectura web y la Computación en la nube

1.1. Definición de Computación en la Nube

El concepto de Computación en la Nube, de la traducción de los términos en inglés "Cloud Computing", tiene una concepción tecnológica y es cada vez más común que las organizaciones y/o empresas la conciban como una tecnología para la solución de muchos problemas en el orden de infraestructura tecnológica y reducción de costos.

La Computación en la Nube es la evolución de un conjunto de tecnologías que afectan al enfoque de las organizaciones y empresas en la construcción de sus infraestructuras de TI (*IT*: *Information Technology)*. Al igual que ha sucedido con la evolución de la Web, con la Web 2.0 y la Web Semántica, la computación en nube no incorpora nuevas tecnologías. Se han unido tecnologías potentes e innovadoras, para construir este nuevo modelo y arquitectura de la Web. [11]

Siendo en Internet un fundamento esencial, la Nube no es solamente el Internet, va más allá, ya que se puede utilizar tecnología en el instante que se requiera. Se puede concebir que la computación en la Nube proporciona un servicio de software y/o hardware (infraestructura), como por ejemplo podría ser una aplicación de escritorio ejecutada luego de ser descargada o el acceso a una aplicación empresarial o institucional para realizar consultas en una base de datos.

Existen organismos de carácter internacional que propugnan la estandarización de Tecnologías de la Información, y en lo referente a la Computación en la Nube el National Institute of Standards and Technology (NIST) y su Information Technology Laboratory, definen la computación en nube (cloud computing) como: *"Un modelo que permite el acceso bajo demanda a través de la Red a un conjunto compartido de recursos de computación configurables (redes, servidores, almacenamiento, aplicaciones y servicios) que se pueden aprovisionar rápidamente con el mínimo esfuerzo de gestión o interacción del proveedor del servicio"* [12]

Se puede decir que la Computación en la Nube es un modelo de TI que engloba las tecnologías Web, valiéndose de las existentes, y que tiene como meta o fin optimizar y proporcionar el uso de los recursos en la red esencialmente el Internet, los cuales pueden ser ofrecidos bajo distintos esquemas de acuerdo a la necesidad y requerimientos de los usuario finales.

1.2. Características de la computación en la nube

Para una comprender y conceptualizar de mejor manera la computación en la nube, se deben conocer las características principales que la diferencian de los modelos existentes en TI; y las mismas se muestran en a Figura 1.1. [12], [13]

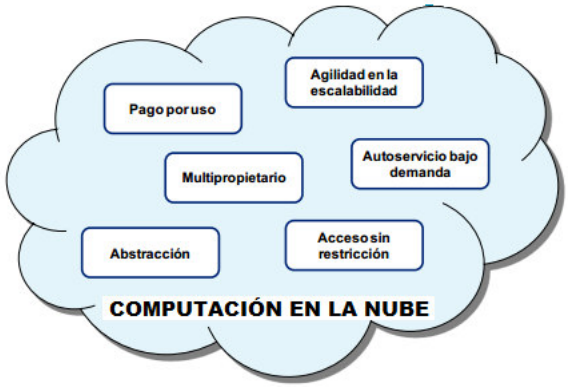


Figura 1.1: Características de la Computación en la Nube [12]

1.2.1. Pago por uso

Una de las características principales de la Computación en la Nube es la implementación de la facturación sustentada en el consumo, en el cual el pago del usuario final depende del servicio contratado.

1.2.2. Abstracción

Por medio de la cual se tiene la capacidad de aislar los recursos informáticos contratados al proveedor de servicios de la nube, de los equipos informáticos del cliente. Esto se da mediante

la virtualización, con lo que la organización usuaria no requiere de personal dedicado al mantenimiento de la infraestructura, actualización de sistemas, pruebas y demás tareas asociadas que quedan del lado del servicio contratado.

1.2.3. Agilidad en la escalabilidad

Se puede aumentar o disminuir las funcionalidades ofrecidas al cliente, en función de sus necesidades puntuales sin necesidad de nuevos contratos ni penalizaciones. De la misma manera, el coste del servicio asociado se modifica también en función de las necesidades puntuales de uso de la solución. Esta característica, relacionada con el pago por uso, evita los riesgos inherentes de un posible mal dimensionamiento inicial en el consumo o en la necesidad de recursos.

1.2.4. Multiusuario

Esta característica permite a varios usuarios compartir los medios y recursos informáticos, permitiendo la optimización de su uso.

1.2.5. Autoservicio bajo demanda

Se permite al usuario acceder de manera flexible a las capacidades de computación en la nube de forma automática a medida que las vaya requiriendo, sin necesidad de una interacción humana con su proveedor o proveedores de servicios de computación en la nube.

1.2.6. Acceso sin restricciones

Posibilita a los usuarios acceder a los servicios contratados de computación en la nube en cualquier lugar, en cualquier momento y con cualquier dispositivo que disponga de conexión a Internet. El acceso a los servicios de computación en la nube se realiza a través de la red, mediante cualquier dispositivo de usuario final.

1.3 Modelos de Servicios

Se debe tener en claro que los modelos de servicios se refieren a los servicios específicos a los que se puede acceder en una arquitectura dentro de la computación en la nube, con esta premisa se tienen tres modelos de servicios.[12]

1.3.1 Infraestructura como servicio- IaaS (Infrastructure as a Service)

El usuario final contrata la infraestructura tecnológica con recursos como capacidad de procesamiento, de almacenamiento o comunicaciones, que son controladas por el proveedor; siendo el usuario quien pueda utilizar para ejecutar cualquier software; desde sistemas operativos hasta aplicaciones. [12]

Las mismas que pueden ser utilizadas como un recurso temporal o ser utilizado por un lapso de tiempo mayor, los costes estarán determinados por variables como: el consumo de recursos y la duración del uso de la infraestructura. En la figura 1.2 se muestra un esquema de las IaaS en la nube.

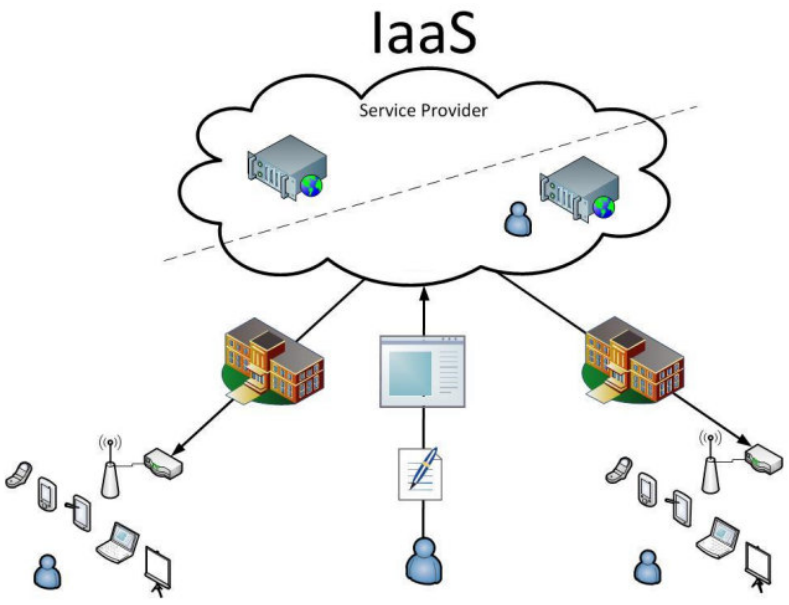


Figura 1.2: Esquema de las IaaS en la nube [12]

Uno de los servicios que toma importancia concerniente a la infraestructura es el Almacenamiento en la Nube.

Pertenece al nivel de infraestructura como servicio, la manera de implementación es a través del Service Oriented Architecture (SOA), y la ubicación tiene diferentes variaciones, el almacenamiento en la nube puede localizarse en: [14]

- Un Datacenter público,
- Un Datacenter privado, o
- Separado del almacenamiento primario.

La manera de acceso puede ser de dos formas: Directamente como bloques o archivos; o indirectamente a través de aplicaciones que están ubicadas en el mismo lugar del almacenamiento.

Las tecnologías que proporcionan métodos de almacenamiento son:

- Storage Area Network (SAN): son switches de alta velocidad que permiten que múltiples computadoras tengan acceso compartido a varios dispositivos de almacenamiento;
- Network-Attached Storage (NAS): vienen como aplicaciones NAS o Gateways NAS, son servidores de archivos virtuales que tienen soporte a protocolos como NFS (siglas de NFS), siendo un dispositivo que directamente accede a la red y que tiene capacidades de compartir archivos. [15]

Los protocolos utilizados para el Almacenamiento en la nube son:

- REST: Representation State Transfer [14], es un protocolo que define las operaciones en recursos y en formatos de datos. Basado en principios o reglas de arquitectura de red, los estados y la funcionalidad de la aplicación se representan mediante recursos, utiliza el protocolo HTTP (definir siglas) para transferencia de información.
- SOAP: Simple Object Access Protocol [15], es un protocolo basado en XML para aplicaciones que envían o reciben mensajes en internet, siendo una recomendación de

la W3C. SOAP fue diseñado para ser simple, extensible e independiente de cualquier plataforma o modelo de programación; utiliza HTTP como protocolo de transferencia, aunque puede ser utilizado también en RPC (Remote Procedure Call).

1.3.2 Plataforma como servicio - PaaS (Platform as a Service)

El proveedor ofrece las herramientas de programación y la plataforma de desarrollo, el usuario por su parte podrá montar y ejecutar sus propias aplicaciones, ya sean desarrolladas o adquiridas; conservando el control de la aplicación pero no de la totalidad de la plataforma tecnológica. [12]

En la figura 1.3 se muestra un esquema de las PaaS en la nube.

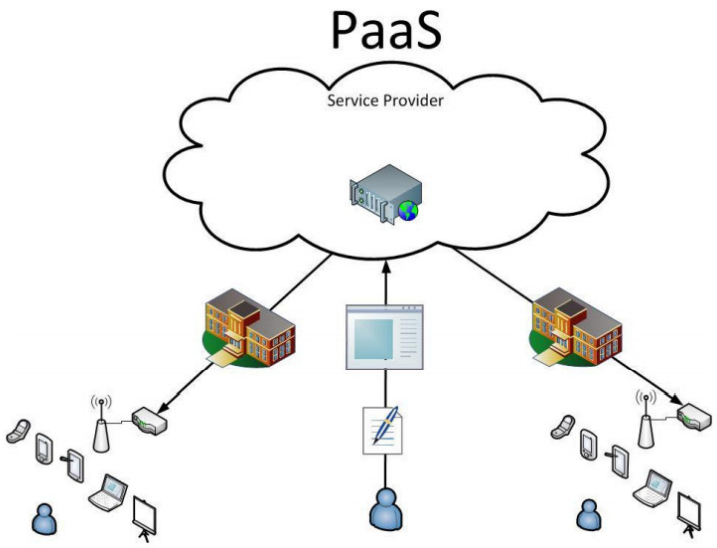


Figura 1.3: Esquema de las PaaS en la nube [12]

1.3.3. Software como servicio – SaaS (Software as a Service)

Al usuario se le ofrece la capacidad de que las aplicaciones suministradas se desenvuelvan en una infraestructura de la nube, siendo las aplicaciones accesibles a través de un navegador web, como en el correo electrónico Web. Posiblemente, este es el ejemplo más representativo, por lo extendido, de este modelo de servicio. El usuario carece de cualquier control sobre la

infraestructura o sobre las propias aplicaciones, excepción hecha de las posibles configuraciones de usuario o personalizaciones que se le permitan. [12]

En la figura 1.4 se muestra un esquema de las SaaS en la nube.

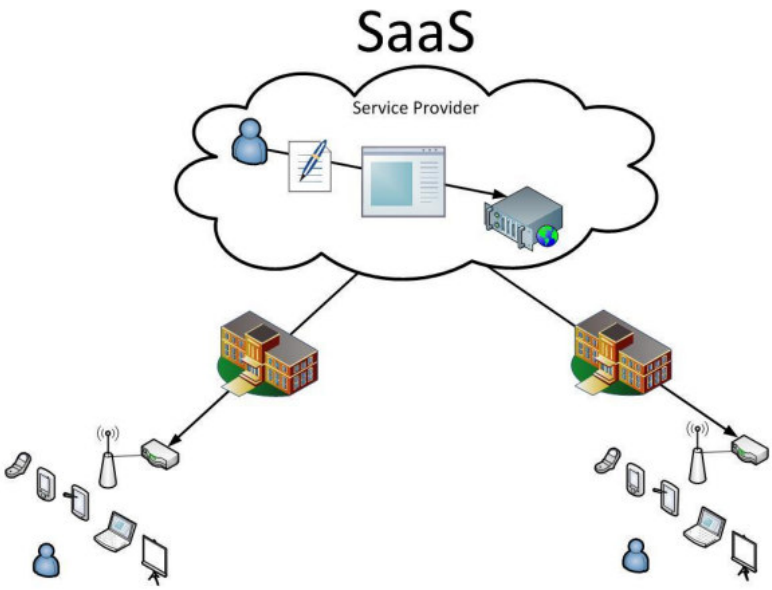


Figura 1.4: Esquema de las SaaS en la nube [12]

1.4. Modelos de Despliegue

La computación en la nube, se puede implementar de diversas maneras; dependiendo de factores como los requisitos de seguridad, las habilidades de TI, etc.; en función del tipo de acceso a la nube, la industria de TI ha señalado cuatro modelos de despliegue de la computación en la nube [12]:

1. Nube Publica
2. Nube privada
3. Nube Hibrida
4. Nube Comunitaria

1.4.1. Nubes Públicas

Es una nube en la cual los proveedores de servicios ofrecen a un usuario en Internet su infraestructura (esto es, su software o hardware) por medio de servicios, de forma gratuita o mediante el abono de cierta cantidad de dinero relacionada con el volumen de información o tiempo de uso de los mismos.[12]

Las aplicaciones e información se almacenan en servidores externos y el servicio se ofrece a través de Internet. El uso de nubes públicas permite ampliar fácilmente los recursos necesitados, ya que éstas suelen tener más tamaño que las nubes privadas, normalmente implantadas en una única organización.

1.4.2. Nubes Privadas

La tendencia actual de las grandes empresas y organizaciones es implementar una nube interna, denominada nube privada, la cual puede ser implementada y gestionada por la propia empresa u organización o por un proveedor externo; habitualmente, el usuario es también propietario de la infraestructura de nube privada, y tiene control total de las aplicaciones desplegadas en ella, posibilitando un alto grado de control sobre el desarrollo de aplicaciones, confiabilidad y seguridad. [12]

Los principales inconvenientes de este modelo son los analizados para el paradigma tradicional, por ejemplo los relativos a la ampliación de los sistemas informáticos. Esto obliga a adquirir nuevos sistemas antes de hacer uso de ellos, contrariamente a lo ofrecido por las nubes públicas, donde ampliar los recursos se reduce a contratarlos con el proveedor de servicios.

1.4.3. Nubes Hibridas

Es una combinación de los modelos de nubes públicas y de nubes privadas, de manera que se aprovecha la ventaja de localización física de la información gestionada por las nubes privadas

con la facilidad de ampliación de recursos de las nubes públicas; se requiere tener cuidado cuando se van a decidir los recursos que se van a localizar en la parte de la nube publica y en la nube privada. [12]

Actualmente este tipo de nubes está teniendo buena aceptación en las empresas y organizaciones, por lo que se están desarrollando software de gestión de nube que permita controlar la nube privada e incorporar al mismo tiempo recursos y servicios de proveedores públicos de la nube.[16]

1.4.4. Nubes Comunitarias

Ha sido organizada para servir a una función o propósito común. Es preciso compartir objetivos comunes (misión, políticas, seguridad). Puede ser administrada bien por las organizaciones constituyentes, o por un tercero. Este modelo es el definido por el NIST, aunque la mayoría de organizaciones, proveedores y usuarios de la nube aceptan los tres modelos de despliegue: pública, privada e híbrida. [12]

1.5. Arquitectura

La arquitectura es el conjunto de capas que se encuentran acopladas entre sí para brindar la funcionalidad del sistema, en este caso la arquitectura de Computación en la Nube es similar a la arquitectura de red, desde un nivel físico hasta un nivel de aplicación. Esto debido a que Computación en la Nube utiliza protocolos similares a los se usan en Internet como medio de comunicación, ya sea basado en web o no basado en web. En la figura 1.5 se visualiza una arquitectura genérica para Computación en la Nube, que consta de las siguientes capas: [17]

- Recursos físicos: incluyen elementos como servidores, almacenamiento y red.
- Virtualización: incluye infraestructura virtual como un servicio.
- Infraestructura: incluye software de plataforma como servicio.
- Plataforma: incluye componentes de aplicación como servicio.

- Aplicación: incluye servicios basados en web y software como servicio.

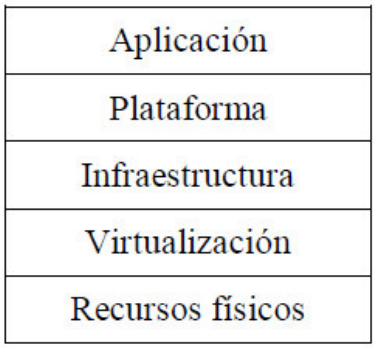


Figura 1.5: Arquitectura genérica de Nube de Computación [17]

1.6. Beneficios y Riesgos

1.6.1. Beneficios

La computación en la nube presenta los siguientes beneficios: [18]

- Ahorro: la optimización y el menor gasto en licencias como en la administración del servicio y en los equipos necesarios. Si se cuenta con una infraestructura 100% basada en la nube, en teoría solo serán necesarios los terminales.
- Implementación rápida: mediante la infraestructura de la nube, es posible comenzar a trabajar al instante; las aplicaciones en la nube estarán disponibles para el usuario, inclusive con un alto nivel de personalización.
- Actualizaciones automáticas: sin afectar los recursos de TI, el usuario podrá actualizar una aplicación.
- Portabilidad de información: si bien es cierto en un principio los proveedores en la nube dirigían sus servicios a los usuarios corporativos, con el paso del tiempo los usuarios particulares han comenzado a usar este concepto manera masiva y casi sin darse cuenta con el uso de servicios para teléfonos móviles (smartphones particularmente), tablets, etc.

1.6.2. Riesgos

Con el modelo de la computación en la nube, se presentan riesgos los cuales se sintetizan a continuación: [19]

- Privacidad de los datos: el riesgo es mayor cuando los datos e información de diversa índole está alojada en la nube.
- Seguridad: es primordial manejar con seguridad la información ante cualquier amenaza externa e interna.
- Licencias de software: se debe tomar en cuenta que exista compatibilidad entre el software con licencia y el software en la nube.
- Interoperabilidad: debe de estar garantizada la interoperabilidad entre todos los servicios.
- SLA (Services Level Agreement): se debe establecer el cumplimiento de acuerdos a nivel de servicio (SLA) antes de confiar a un tercero (proveedor) las aplicaciones de una empresa u organización.
- Escalabilidad a largo plazo: a medida que crezca el número de usuarios y se comparta la infraestructura de la nube, la sobrecarga en los servidores de los proveedores aumentara; pudiendo producir degradaciones en el servicio o jitter altos.
- Aplicaciones: las aplicaciones del modelo computación en la nube deben estar diseñadas de modo que se puedan dividir entre múltiples servidores.

CAPÍTULO 2

Introducción

En el capítulo 2 se describen los conceptos que hacen referencia a la aplicación de Voz sobre IP (VoIP), los protocolos que operan en lo concerniente al transporte, señalización y enrutamiento del tráfico de esta aplicación, particularmente en un entorno Web y especificando parámetros como jitter, retardo y perdida de paquetes que permiten una Calidad de Servicio (QoS) en la comunicación.

2. Voz sobre IP (VoIP), estado actual

2.1. Definición

La VoIP (Voz sobre IP) es exactamente lo que su nombre indica, el envío de voz a través de una red basada en el protocolo IP (Protocolo Internet). Esto es completamente diferente de la telefónica pública mediante conmutación de circuitos, en la que se asigna recursos a cada individuo al llamar. [20]

En las redes IP por conmutación de paquetes, cada uno de estos es enviado de manera independiente, tiene su propio encabezado IP, y se enviará por separado por parte de los Routers (nodos).

Una topología de la arquitectura de Red para la aplicación de VoIP se muestra en la figura 2.1.

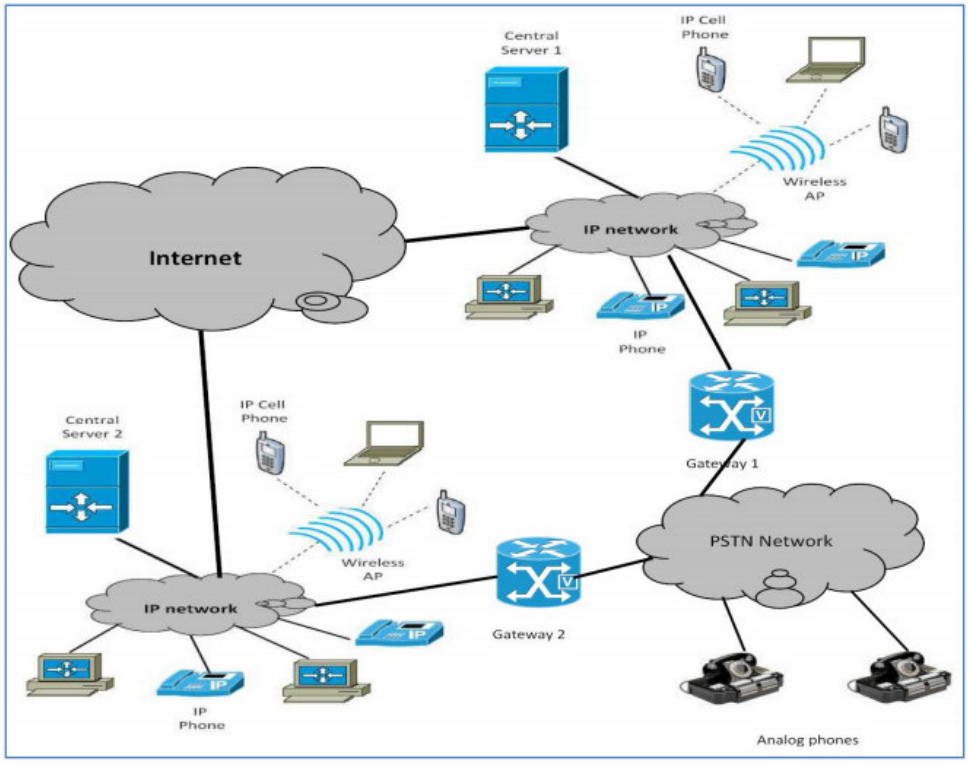


Figura 2.1: Esquema de Red para aplicación de VoIP [20]

2.2. Protocolos de VoIP en la Nube

En una llamada mediante VoIP se tienen protocolos los cuales permiten el tráfico de voz en una red WAN, específicamente en la Nube. Existen varios protocolos de VoIP, pero en forma general se tienen: protocolos señalización, protocolos de transporte y protocolos de enrutamiento.

Los protocolos de enrutamiento hacen posible que los paquetes de VoIP en la Nube, toman sus rutas en una red WAN (Internet) a través de los Routers.

Los protocolos de señalización manejan las funciones derivadas de la arquitectura del sistema telefónico, y los protocolos de transporte llevar los paquetes de voz generada en el códec.

Los terminales IP utilizan el protocolo de señalización para registrarse con el servidor de llamadas, configurar y culminar las llamadas, además los protocolos de señalización se utilizan también para funciones tales como servicios de directorio y en las pantallas. Una vez que la llamada que se ha establecido, los paquetes de datos de voz se envían directamente entre los teléfonos mediante encapsulación por medio de un protocolo de transporte (p.e. el protocolo RTP). [20]

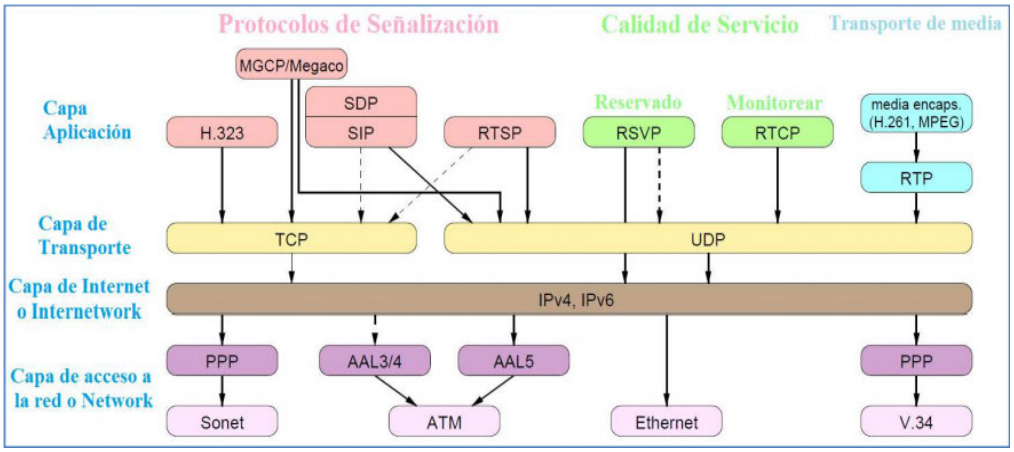


Figura 2.2: Protocolos en VoIP [20]

2.2. 1. Protocolos de Señalización

A pesar de que la arquitectura VoIP es completamente diferente a la utilizada por telefonía tradicional, se tiene la necesidad básica de señalización. De alguna manera, los teléfonos con un identificador y/o números deben ser comunicados, y las rutas para el tráfico de paquetes tienen que estar configuradas; estas funciones son manejadas por el protocolo de señalización.[20]

2.2.1.1. H. 323

La Unión Internacional de Telecomunicaciones – Telecom (UIT-T) con la recomendación H.323, estableció el inicio de un estándar de videoconferencia, que incluye voz, vídeo y datos. Dado que aborda, entre otras cosas, control de llamadas y la gestión de punto a punto y multipunto, administración del sistema de puerta de enlace de tráfico multimedia, los parámetros de ancho de banda y participación de los usuarios, a la vez se convirtió en el estándar de facto para la telefonía por Internet y VoIP. Hoy en día, con el surgimiento del protocolo SIP, se lo utiliza menos.

El objetivo de H. 323 es definir los terminales y otras entidades que prestan servicios de comunicaciones multimedia en redes basadas en conmutación de paquetes que no proporcionan una calidad de servicio garantizada

El estándar original data de 1996, hasta la última versión 7 que está vigente desde el 2009.

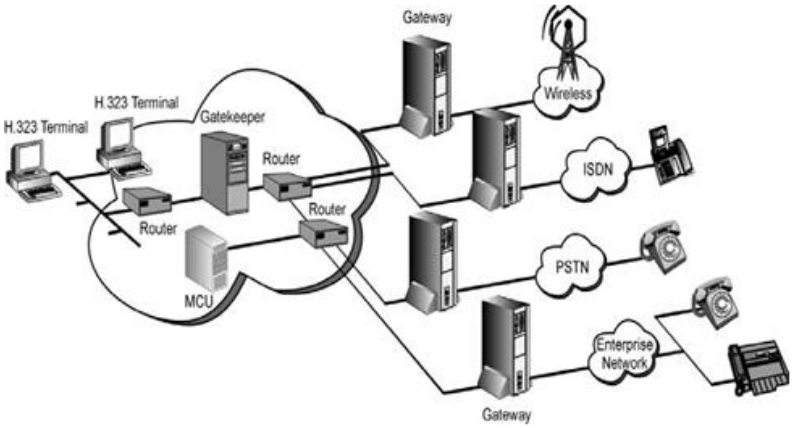


Figura 2.3: Componentes del protocolo H.323 [20]

Hay varios componentes que conforman un sistema H. 323, a continuación se detallan algunos de ellos.

- Terminales: Es el dispositivo final, hardware o software, que provee comunicación full dúplex con otro dispositivo final, transmitiendo información de control, transmisión de audio, códec (video), y la interfaz de red. Todos los terminales soportan comunicación de voz y video, la comunicación de datos es opcional.
- Gateway: Controla la comunicación y la traducción de protocolos entre el terminal H.323 en una red de conmutación de paquetes hacia una red de Conmutación de Circuitos, como la Red Telefónica Pública Conmutada (PSTN). Además se encarga de la Configuración de llamada, terminación y posiblemente traducción entre códecs a través de H. 225.
- Gatekeeper: Este es un componente opcional que puede manejar servicios de control de llamadas, tales como solicitudes de ubicación y de la confirmación, la resolución de nombres, admisión y control de ancho de banda, a través de H. 225.
- Multipoint controller: El controlador permite las conferencias entre tres o más terminales H. 323. También se encarga del intercambio de tráfico multimedia entre los nodos, a través de H. 245.

Para el control y la señalización H.323 hace uso de los siguientes protocolos:

- H.225: permite la comunicación ente un terminal-gatekeeper y un gatekeeper. Se le denomina también RAS (registro, admisión y estado); contiene mensajes de registro y anulación en el gatekeeper, los mensajes de admisión de llamadas, llamada final, etc.
- Q.931 / H.225: Para señalización de llamada entre los extremos en H.323 se utiliza el protocolo Q. 931. Sin embargo, Q.931 no tiene ciertos campos de datos que se necesitan para la comunicación de VoIP (por ejemplo direcciones IP y puertos de

escucha), para resolver este problema, los mensajes Q. 931 se embeben en mensajes H.225 que llevan la información completa.

- H.245: se utiliza para negociar los parámetros de audio (y video) entre los terminales. La negociación incluye códecs, direcciones IP y puerto.

2.2.1.2. Protocolo de Inicio de Sesión (SIP)

Los primeros trabajos sobre SIP empezaron en 1999 con el RFC 2543. SIP opera en la capa de aplicación a los efectos de iniciar las sesiones de usuario para las transmisiones multimedia, tales como voz, video, chat, juegos y realidad virtual. Estas sesiones pueden ser unicast o multicast y puede funcionar con o sin un servidor de llamadas o de la puerta de enlace.

Al ser compatible con la mayoría de códecs, SIP es utilizado para señalización en las nuevas aplicaciones. A menudo se dice que el SIP es mucho más fácil de leer y utilizar que otros protocolos de señalización, esto puede ser debido a que SIP es muy similar en su estructura a la del Protocolo de Transferencia de Hipertexto (HTTP). Vamos a ver que hay una cierta cantidad de verdad que este sentimiento. [20]

El protocolo SIP no maneja todo lo referente a señalización, y en las implementaciones se suele utilizar otro protocolo llamado Protocolo de Descripción de Sesión (SDP), para negociar los parámetros de la conexión multimedia.

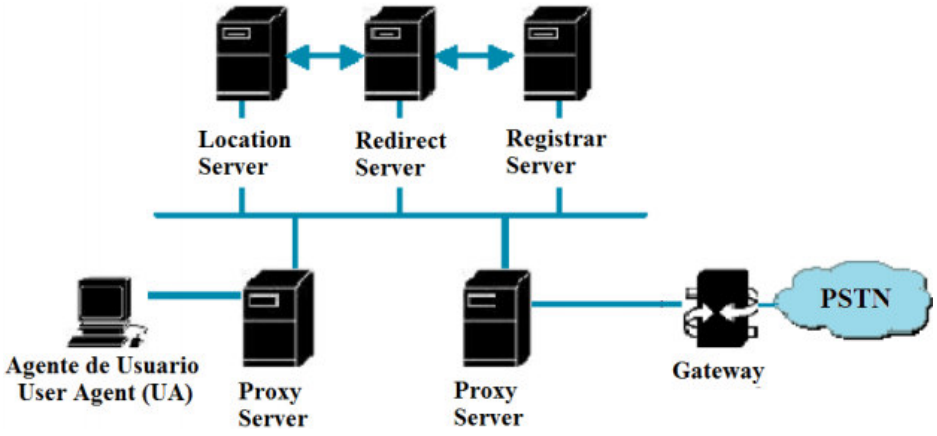


Figura 2.4: Componentes del protocolo SIP [20]

2.2.1.2.1 Componentes

Los componentes de SIP son:

- Agentes de Usuario (UA): es la parte lógica que inicia o responde a las transacciones SIP. La UA puede ser un cliente o servidor
    - Agentes de Usuario Clientes (UAC): Solicita las llamadas y acepta las respuestas. Por lo general, es el teléfono SIP que inicia la llamada.
    - Agentes de usuario Servidores (UAS): Acepta solicitudes y envía las respuestas a las llamadas.
- Servidores Proxy: Componente intermedio que reenvía las peticiones de un UAC a un UA u otro proxy. Esto se realiza principalmente no solo para el encaminamiento de paquetes sino también para el cumplimiento de las políticas, tales como la autenticación.
- Servidores de registro: Los UAS que aceptan registrar los mensajes y actualizan la ubicación.
- Servidores de Redirección: Envían solicitudes de UAC para un conjunto alternativo de identificadores de recursos uniformes.

2.2.1.2.2 Direccionamiento

El protocolo SIP puede comenzar conversaciones teniendo como identificador una dirección IP o el nombre de usuario con el fin de permitir al UAC comunicarse con otro usuario o recurso de la red.

El direccionamiento con el protocolo SIP es estándar y similar al correo electrónico, teniendo una de las siguientes formas (el puerto es opcional, y si no se especifica, se usa 5060 por default):

sip:user@domain:port

sip:user@host:port

sip:phone number@domain

En la figura 2.5, se observa una de las formas de direccionamiento SIP, y se utiliza el puerto 5060.

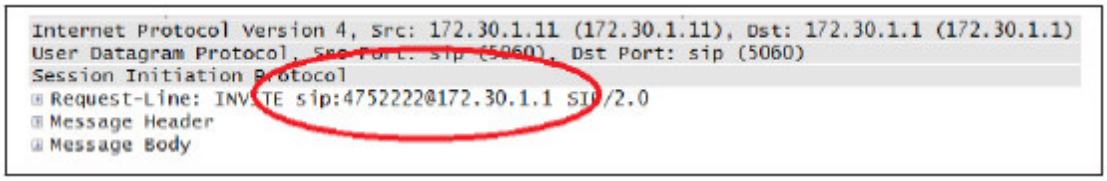


Figura 2.5: Direccionamiento SIP [20]

2.2.1.2.3 Protocolo de Descripción de Sesión (SDP)

El Protocolo de Descripción de Sesión se define en el RFC 4566. El objetivo de SDP es establecer un protocolo de propósitos generales que permita describir el contenido de los medios que se van a transferir.

Dentro de la información importante en el campo SDP, se tiene la sesión, duración, tipo de medio y la información relevante de media stream por ejemplo el número de puerto, el tipo de códec, etc.

2.2.1.3. Protocolo de Intercambio entre Asterisk (IAX)

El protocolo IAX (Inter-Asterisk eXchange protocol), se publicó en la RFC-5456, se basó en sus inicios como un protocolo para servidores de VoIP que trabajaban con Asterisk, en la actualidad se lo utiliza también para comunicación entre equipos finales o terminales.

A diferencia de H. 323 y SIP, el protocolo IAX envía la señalización en los mismos paquetes de comunicación, lo que da funcionalidad al protocolo, permitiendo disminuir el número de conexiones establecidas de manera simultáneas, siendo esto una ventaja en ambientes que los firewalls y NATs pueden ocasionar inconvenientes.

El protocolo utilizado en la actualidad es el IAX2, el cual no es un estándar por lo que en muchos de los dispositivos de diferentes marcas que se encuentran en el mercado no se lo puede implementar.

Una de las desventajas de IAX2 con respecto a SIP, es que la señalización y datos al viajar en el mismo paquete necesariamente todo el tráfico pasa por el servidor de IAX2, lo cual aumenta el ancho de banda y el procesamiento del servidor cuando existe un número alto de llamadas simultaneas.

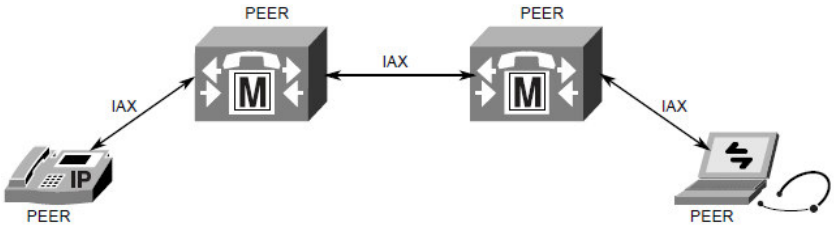


Figura 2.6 Modelo de llamada IAX [20]

2.2.1.4. Protocolo MGCP (Media Gateway Control Protocol)

El protocolo MGCP tiene su origen en el SGCP (de Cisco y Bellcore), es un protocolo de control de dispositivos, donde un gateway esclavo (MG, *Media Gateway*) es controlado por un maestro (MGC, *Media Gateway Controller*) o CAs (*Call Agents*).

MGCP soporta un control de señalización de llamada escalable, integrando el control de QoS (*Quality of Service*, Calidad de Servicio) en el gateway. Su compatibilidad con normas de IETF y con H.323 lo hace ideal para aplicaciones de multimedia sobre redes IP.

Este protocolo presenta una arquitectura de control de llamada donde la inteligencia está fuera de las gateways y es manejada por elementos de control de llamada externos, conocidos como Agentes de Llamada.

El protocolo MGCP presupone que estos elementos del control de llamada, o Agentes de Llamada, se sincronizan entre sí para enviar órdenes coherentes y respuesta a las gateways.

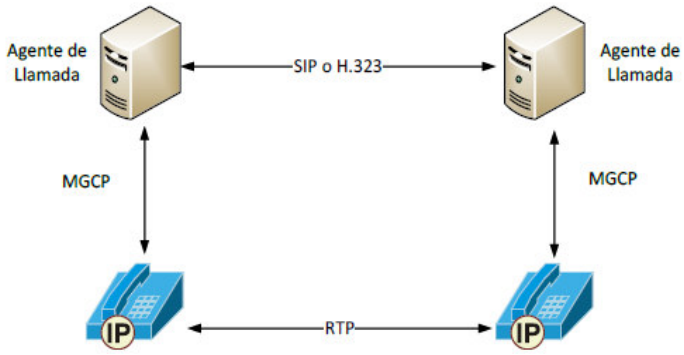


Figura 2.7: Señalización en el protocolo MGCP [20]

2.2.2. Protocolos de Transporte

Un protocolo de transporte cumple en forma general con la función de trasladar los paquetes de VoIP desde el origen (transmisor) hasta el destino (receptor) de manera correcta, es decir que cumplan con los parámetros de calidad de servicio adecuados.

El protocolo de transporte empleado para VoIP es el RTP junto con el protocolo de control RTCP proporciona servicios de control y otras funcionalidades. Además se tiene una variante llamada SRTP (Secure RTP) usada para aportar características de cifrado al canal RTP.

2.2.2.1 Real-time Transport Protocol (RTP)

El protocolo RTP se utiliza para transmitir los datos en tiempo real, generalmente paquetes de voz y/o vídeo. Una vez que la sesión de medios se ha establecido, los paquetes RTP comienzan a fluir entre los extremos en ambas direcciones. Los paquetes de cada uno de los orígenes son aunados en el receptor mediante un número que los identifica.

Fue publicado por primera vez como estándar en 1996 como RFC 1889, y actualizado posteriormente. Éste ofrece entrega de datos multicast para aplicaciones de streaming, videoconferencia, etc., siempre que la red proporcione los servicios.

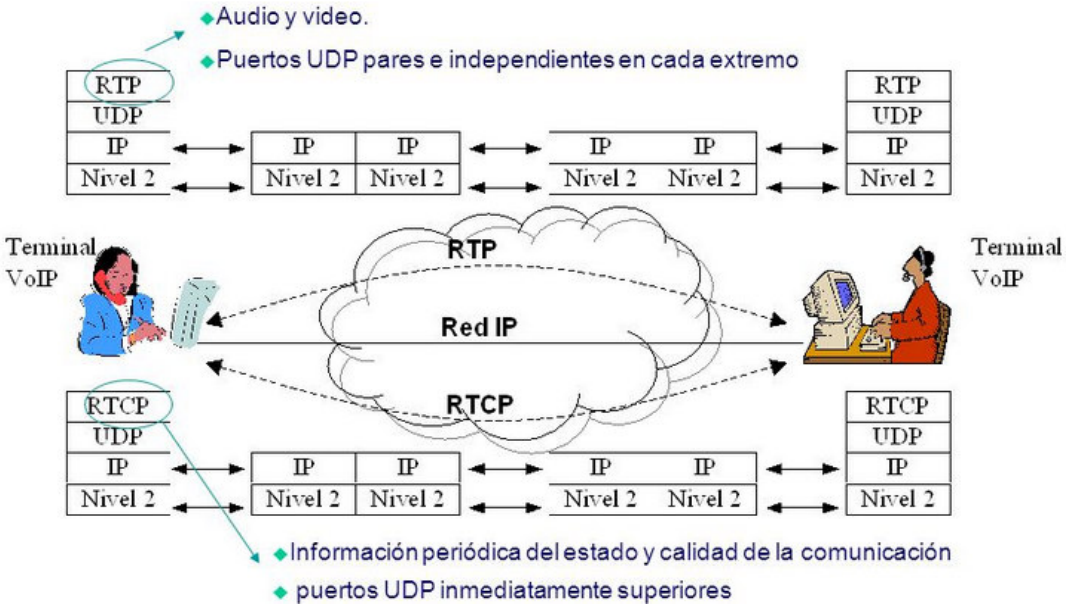


Figura 2.8: Protocolos de Tiempo Real sobre IP

Es importante destacar en este caso que RTP no ofrece garantías sobre la calidad del servicio ni sobre el retraso de la entrega de datos, por lo que necesita el apoyo de capas de nivel inferior. Debido a que en la transmisión de datos en tiempo real es esencial que los parámetros de latencia, perdida de paquetes y jitter estén dentro de valores consideraros normales para que la calidad de la llamada sea buena, el protocolo RTP para control utiliza el protocolo RTCP.

En la figura 8 se observa un paquete RTP real capturado con el software Wireshark. Los dos primeros campos son muy pequeñas e incluyen información sobre el contenido del paquete. Además se tienen los campos de PT o tipo de carga, el número de secuencia (11639), timestamp (998248329), y el SSRC, identificador o fuente de sincronización.

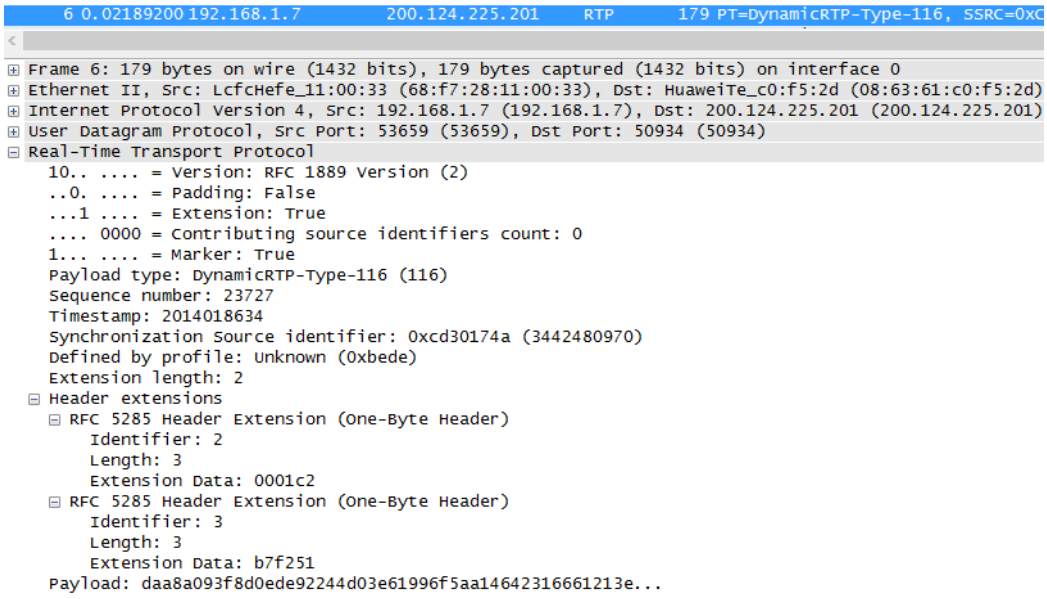


Figura 2.9: Captura paquete RTP (Autor de la Tesis 2015)

2.2.2.2 RTP Control Protocol (RTCP)

El protocolo RTCP se encarga de monitorizar la calidad del servicio y de proporcionar información acerca de los participantes en una sesión de intercambio de datos. El protocolo, definido en el RFC 3550, no está diseñado para soportar todas las necesidades de comunicación de una aplicación, sólo las más básicas.

La principal función de RTCP es proporcionar una retroalimentación útil para mantener una calidad de distribución adecuada: los receptores de una sesión emplean RTCP para informar al emisor sobre la calidad de su recepción, incluyendo el número de paquetes perdidos, *jitter* (la variación en la latencia) y RTT (Round Trip Time, tiempo empleado por un paquete en realizar todo el circuito: llegar al receptor y volver de nuevo al emisor).

Los paquetes RTCP se envían de modo que el tráfico en la red no aumente linealmente con el número de agentes participantes en la sesión, ajustando el intervalo de envío de acuerdo al tráfico. Para ello, RTCP se encarga de transmitir periódicamente paquetes de control a todos los participantes de una sesión.

La aplicación de VoIP se la considera en muchos casos como una aplicación simple, pero en realidad es una aplicación crítica a la cual se le debe asignar recursos en la red pues maneja datos en tiempo real; de ahí que el protocolo RTCP ofrece información valiosa de los recursos asignados. En la figura 2.10 se muestra la captura de una llamada en la cual se mezclan paquetes RTCP y RTP.

Time	Source	Destination	Protocol	Length	Info
274 1.65889500	192.168.1.7	200.124.225.201	RTP	141	PT=DynamicRTP-Type-111, SSRC=0xD8AFAE22, Seq=4381, Time=3426034497
278 1.67867400	192.168.1.7	200.124.225.201	RTP	146	PT=DynamicRTP-Type-111, SSRC=0xD8AFAE22, Seq=4382, Time=3426035457
281 1.68806000	192.168.1.7	200.124.225.201	RTCP	112	Sender Report
285 1.69319200	192.168.1.7	200.124.225.201	RTP	136	PT=DynamicRTP-Type-116, SSRC=0xCD30174A, Seq=23752, Time=2014169024
286 1.69960000	192.168.1.7	200.124.225.201	RTP	138	PT=DynamicRTP-Type-111, SSRC=0xD8AFAE22, Seq=4383, Time=3426036417

Figura 2.10: paquete RTP y RTCP en una llamada (Autor de la Tesis 2015)

En una llamada, como se muestra en la figura 2.11, lo que se desea es que los emisores proporcionen información acerca del flujo RTP, y los receptores proporcionan a la vez información para el remitente, mediante mensajes de informes de emisor y receptor; siendo el objetivo principal mantener la calidad de la llamada; teniendo en cuenta que RTCP se encapsula en mensajes del protocolo UDP.

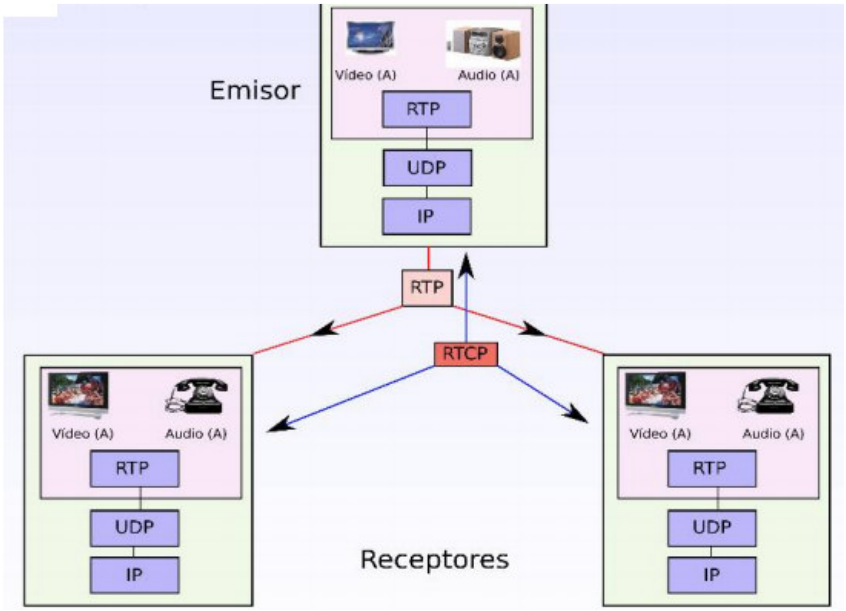


Figura 2.11: Flujo de información entre emisor y receptores, con protocolo RTCP

2.2.2.3 Secure Real-time Transport Protocol (SRTP)

El protocolo SRTP, es una extensión del protocolo RTP que agrega funciones de seguridad, como autenticación de mensaje, confidencialidad y protección de respuesta, mayormente pensadas para comunicaciones mediante la aplicación VoIP.

El protocolo SRTP fue concebido y elaborado por expertos de las empresas Cisco y Ericsson, se publicó en marzo de 2004 por el Internet Engineering Task Force (IETF) en el RFC 3711. SRTP utiliza el cifrado y la autenticación para reducir al mínimo el riesgo de ataques como denegación de servicio (DoS).

Este protocolo puede lograr un alto rendimiento en diversos entornos de comunicaciones que incluyen tantos dispositivos inalámbricos y con cables.

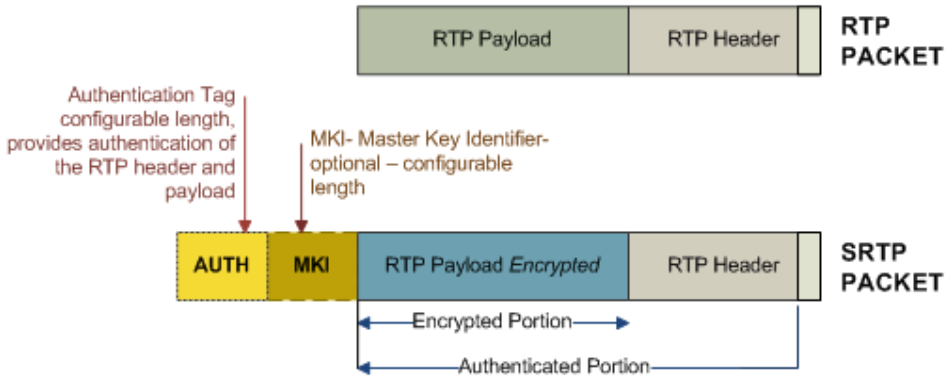


Figura 2.12: Estructura del paquete SRTP

2.2.3. Protocolos de Enrutamiento

Los datos de la aplicación de VoIP en la Nube, toman sus rutas en Internet según la información de las tablas de enrutamiento que el nodo (router) determine como óptima.

Esta tabla puede ser estática o dinámica, en la practica la tabla dinámica es la de uso común, ya que se actualiza dinámicamente en razón de los cambios que se presentan en la topología de red.

Frente al enrutamiento estático, el enrutamiento dinámico ofrece nuevas posibilidades, se adapta mejor a nuevas circunstancias pero requiere una mayor complejidad en los sistemas y en la gestión de estos.

Un protocolo de enrutamiento es un software complejo que se ejecuta de manera simultánea en un conjunto de routers, con el objetivo de completar y actualizar su tabla de enrutamiento con los mejores caminos para intercambiar información con otras redes. Así, podríamos resumir que un protocolo de enrutamiento tiene como objetivos los siguientes:

- Descubrir redes lejanas con las que intercambiar información
- Mantener la información de enrutamiento actualizada de manera fiable
- Elegir el mejor camino posible en cada momento hacia las redes de destino
- Encontrar un nuevas rutas para sustituir a aquellas que dejen de estar disponibles en los términos necesarios.

En la figura 2.13, se muestra la clasificación de los protocolo de enrutamiento dinámico.

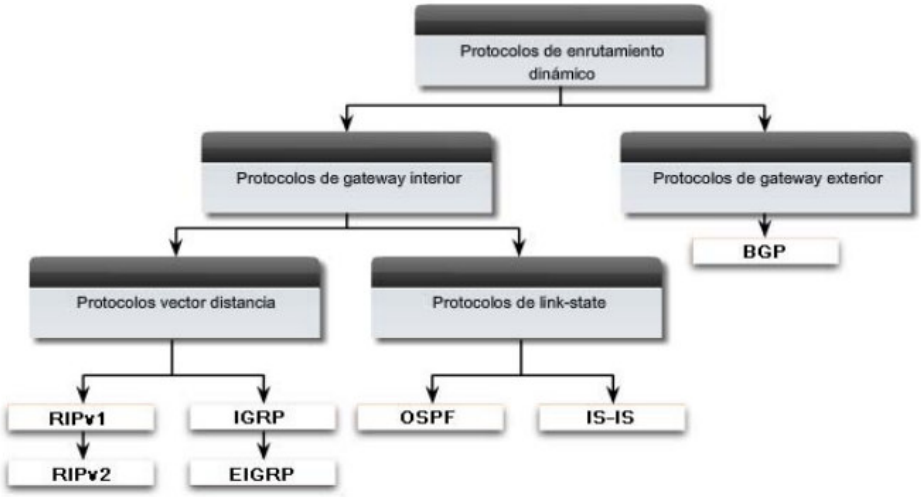


Figura 2.13: Clasificación de los protocolo de enrutamiento dinámico [21]

En la tabla 2.1 se puede observar las características de los protocolo de enrutamiento dinámico más comunes

	RIPv2	EIGRP	OSPF	IS-IS	BGP
Interior o Exterior	Interior	Interior	Interior	Interior	Exterior
Tipo de Protocolo	Vector Distancia	Vector Distancia	Estado de Enlace	Vector Distancia	Vector Camino
Métrica	Numero de Saltos	Numero de Saltos	Costo	Costo	Multiples Atributos
Limite Numero de Saltos	15	224	Ninguno	Ninguno	-
Tamaño de la red	Pequeño	Grande	Grande	Grande	Muy Grande
Tiempo de convergencia	Lento	Muy rápido	Rápido	Rápido	Muy lento
Algoritmo	Bellman-Ford	Dual	Dijkstra	Dijkstra	Algoritmo Mejor Ruta
Direccionamiento sin clase	Si	Si	Si	Si	Si
Requiere diseño jerárquico	No	No	Si	Si	No
Distancia administrativa	120	5/90/170	110	115	20/200
Actualizaciones	30 seg	Solo cuando ocurren cambios	Solo cuando ocurren cambios	Solo cuando ocurren cambios	Solo cuando ocurren cambios

Tabla 2.1: Características protocolo de enrutamiento dinámico [22]

En la figura 2.14 se aprecia un ambiente típico en la nube, a sabiendas que el camino por el cual circula el paquete no se lo puede controlar y no depende de los usuarios de la comunicación sino más bien del ISP (Proveedor de Servicios Internet) ya que tienen configurados los protocolos de enrutamiento en sus nodos, y de manera específica de la información que en un momento determinado tengan la tablas de enrutamiento.

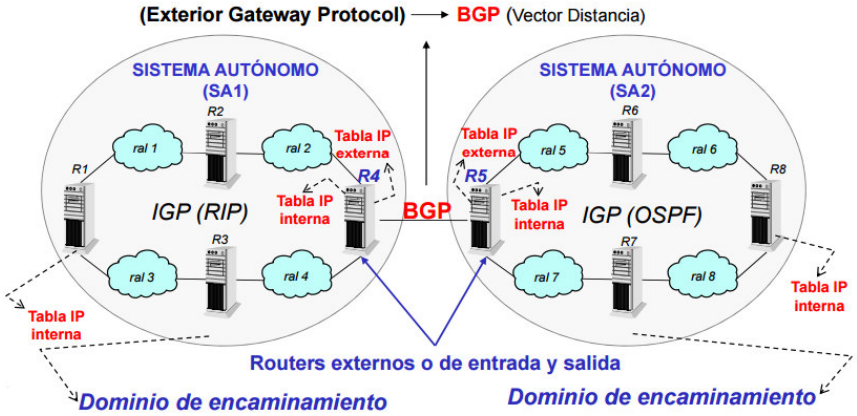


Figura 2.14: Protocolos externos de enrutamiento especifico en el ambiente de internet [23]

De lo anteriormente mencionado se tiene en esta topología los siguientes protocolos de enrutamiento: BGP (del inglés Border Gateway Protocol), OSPF (del inglés Open Shortest Path First) y RIP (del inglés Routing Information Protocol), que son los protocolos que encaminan el tráfico de paquetes de VoIP en la Nube.

2.3. VoIP en un entorno WEB

En los últimos años se ha observado un aumento de la migración de las aplicaciones de escritorio a la Web, e incluso con la reciente aparición de las aplicaciones móviles, también en este segmento algunas empresas deciden crear aplicaciones Web, que permite que la misma aplicación se ejecute en cualquier dispositivo y con cualquier sistema operativo.

Las aplicaciones Web ya no son sólo las páginas con contenido estático y ahora son capaces de hacer las transacciones típicas de las aplicaciones de escritorio, y se denominan Rich Internet Application (RIA).

Hoy se diseñan las páginas web con muchas más funcionalidades, con contenido cargado en el fondo y con mucha más programación en la aplicación cliente.

Para crear aplicaciones Web que permiten realizar llamadas de VoIP, es necesario que las aplicaciones pueden acceder a algunas funciones, que normalmente no ofrecen los navegadores sin tener que recurrir a algunos plugins que facilitan estos permisos. Los requerimientos mínimos son el acceso a los micrófonos, cámaras y un medio de transmisión de los datos obtenidos de estos dispositivos a un servidor o directamente al destinatario.

2.4 Parámetros de Calidad de Servicio para el tráfico de VoIP

La VoIP es implementada con más frecuencia a través de redes IP donde convergen los paquetes de voz, vídeo y datos; cuando los recursos de la red están congestionadas pueden afectar seriamente a la calidad del tráfico de la aplicación de VoIP provocando una mala experiencia de usuario para los suscriptores. [24] [25]

Por lo tanto, es muy importante en una red de una empresa y/o entidad implementar políticas QoS para el tráfico VoIP. Esto puede ayudar a garantizar una buena calidad de voz cuando los recursos de la red están congestionados.

Hay una serie de factores que pueden afectar la calidad del tráfico de VoIP tal como la percibe el usuario final. Algunos de los factores comunes incluyen retardo, jitter y pérdida de paquetes. Estos factores pueden ser los indicadores claves del estado en general de la red de voz.

Los problemas de la calidad del tráfico en VoIP vienen derivados de dos factores principalmente:

- Internet (incluida la Nube) es un sistema basado en conmutación de paquetes y por tanto la información no viaja siempre por el mismo camino. Esto produce efectos como la pérdida de paquetes o el jitter.
- Las comunicaciones VoIP son en tiempo real lo que produce que efectos como el eco, la pérdida de paquetes y el retardo o latencia sean muy molestos y perjudiciales y deban ser evitados.

En la tabla 2.2 se pueden observar lo valores referenciales de retardo, jitter y pérdida de paquetes en el tráfico de paquetes de VoIP

	Buena	Aceptable	Pobre
Retardo (Round Trip)	0ms–150ms	150ms–300ms	> 300ms
Jitter	0ms–20ms	20ms–50ms	> 50ms
Pérdida de Paquetes	0%–0.5%	0.5%–1.5%	> 1.5%

Tabla 2.2: Valor referencial de retardo, jitter y pérdida de paquetes en el tráfico de VoIP [41]

2.4.1. Retardo

El retardo es el tiempo que tarda el tráfico de VoIP para llegar de un extremo a otro, por lo general, se denomina el tiempo de extremo a extremo. El retardo puede ser valorado en una vía o un retardo de ida y vuelta. La recomendación ITU G. 114 establece que el retardo aceptable para la voz es 150 ms.

Cualquier retardo mayor a 150 ms puede causar una degradación en calidad de voz y una mala experiencia para el usuario ya que el oído humano es capaz de detectar latencias de unos 250 ms, y hay casos de 200 ms en el caso de personas bastante sensibles. Si se supera ese umbral la comunicación se vuelve molesta.

Puede ocurrir que a los paquetes le tome un largo período en alcanzar su destino, debido a que pueden permanecer en largas colas o tomen una ruta menos directa para prevenir la congestión de la red. En algunos casos, los retardos excesivos pueden inutilizar aplicaciones tales como VoIP.

No hay una solución que se pueda implementar de manera sencilla, muchas veces depende de los equipos por los que pasan los paquetes, es decir, de la red misma, y no se dispone del control de la red WAN (Nube).

Si el problema de retardo está en la red LAN propia (Intranet) se puede aumentar el ancho de banda, la velocidad del enlace o priorizar esos paquetes dentro de la red.

2.4.2. Jitter

El Jitter es la variación en el tiempo de llegada de un paquete de un extremo a otro. Si la demora de las transmisiones varía demasiado en una llamada VoIP, la calidad de la llamada es muy deficiente.

El jitter es causado por la congestión de red, perdida de sincronización o por las diferentes rutas seguidas por los paquetes para llegar al destino. La solución más ampliamente adoptada es la utilización del jitter buffer.

El jitter buffer consiste básicamente en asignar una pequeña cola o almacén para ir recibiendo los paquetes y sirviéndolos con un pequeño retraso, si algún paquete no está en el buffer (se perdió o no ha llegado todavía) cuando sea necesario se descarta.

Normalmente en los terminales IP (hardware y software) se pueden modificar los buffers, pero hay que tomar en cuenta lo siguiente un aumento del buffer implica menos perdida de paquetes pero más retraso, y una disminución implica menos retardo pero más pérdida de paquetes.

2.4.3. Pérdida de paquetes

Este valor viene dado por el número de paquetes perdidos en la ruta de datos al llevar el tráfico de VoIP de un extremo a otro. Un 3 % de perdida de paquetes es generalmente considerado como el máximo límite tolerable para una buena calidad de voz. Las redes de VoIP se deben diseñar para valores menores a 1,5 % de pérdida de paquetes con el fin de garantizar una buena calidad de voz.

Las comunicaciones en tiempo real están basadas en el protocolo UDP. Este protocolo no está orientado a conexión y si se produce una pérdida de paquetes no se reenvían; otra cusa de perdida de paquetes también se produce por descartes de paquetes que no llegan a tiempo al receptor.

Sin embargo la voz es bastante predictiva y si se pierden paquetes aislados se puede recomponer la voz de una manera bastante óptima. El problema es mayor cuando se producen pérdidas de paquetes en ráfagas.

CAPÍTULO 3

Introducción

En el capítulo 3 se realiza el análisis teórico de WebRTC, desde sus componentes y arquitectura así como también los protocolos de señalización que intervienen en el proceso de la comunicación, además se analiza la compatibilidad de esta tecnología con los distintos navegadores.

3. Arquitectura y soluciones Web, mediante WEBRTC

3.1 Definiciones

3.1.1 HTML 5 (Hyper Text Markup Language)

HTML5 (del inglés Hyper Text Markup Language) es un lenguaje de marcado o lenguaje de marcas, usado para estructurar y presentar el contenido para la web. [26]

Es uno de los aspectos fundamentales para el funcionamiento de los sitios, pero no es el primero. Es de hecho la quinta revisión del estándar que fue creado en 1990. A fines del 2012, la W3C (World Wide Web) la recomendó para transformarse en el estándar a ser usado en el desarrollo de proyectos nuevos.

Por así decirlo, qué el HTML5 está relacionado también con la entrada en decadencia del viejo estándar HTML 4. Con HTML5, los navegadores como Firefox, Chrome, Explorer, Safari y más pueden saber cómo mostrar una determinada página web, saber dónde están los elementos, dónde poner las imágenes, dónde ubicar el texto. En este sentido, el HTML5 no se diferencia demasiado de su predecesor; la diferencia principal, sin embargo, es el nivel de sofisticación del código que se puede construir usando HTML5.

En términos de lenguaje de marcado, el HTML5 introduce algunos elementos que hacen que se actualice la página al mismo tiempo que se ejecuta. Así, muchas de las novedades están relacionadas con la forma de construir los sitios web que se tiene en la actualidad.

Una de las más importantes novedades está relacionada con la inserción de multimedia en los sitios web, que ahora contarán con etiquetas HTML especiales para poder ser incluidos. Por otro lado, algunos aspectos de diseño también son incluidos en el lenguaje, así como también algunos detalles de navegación algunas líneas.

Con el uso de HTML5, se puede reducir la dependencia de los plug-ins que se tienen que instalar para poder ver una determinada web. Caso emblemático, el de Adobe Flash, que se ve claramente perjudicado por la instauración de este estándar. Por otro lado, fue un avance

importante para dispositivos que de forma nativa no soportaban Flash, y que no soportaban tampoco plug-ins necesarios para hacerlo.

Aunque HTML5 no ha sido finalizado, casi todas sus etiquetas pueden ser utilizadas con seguridad en las páginas Web de hoy.

3.1.2 CSS3 (Cascading Style Sheets)

CSS (en inglés Cascading Style Sheets) u hojas de estilo en cascada es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML. [27]

El W3C (World Wide Web Consortium) es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para los agentes de usuario o navegadores.

CSS se creó para separar el contenido de la forma, a la vez que permite a los diseñadores mantener un control mucho más preciso sobre la apariencia de las páginas.

CSS sirve para definir la estética de un sitio web en un documento externo y eso mismo permite que modificando ese documento (la hoja CSS) se logre cambiar la estética entera de un sitio web, en otras palabras, el mismo sitio web puede variar totalmente de estética cambiando solo la CSS, sin tocar para nada los documentos HTML u otros que lo componen; con CSS3 se suman muchos nuevos efectos que harán de la que la parte visual de nuestra página sea más agradable y llamativa, se podrá tener sombras, transformaciones de figuras, creación sencilla de bordes y efectos 3D.

Además esta nueva especificación viene con interesantes novedades que permitirán hacer webs más elaboradas y dinámicas, con mayor separación entre estilos y contenidos. Dará soporte a muchas necesidades de las webs actuales, sin tener que recurrir a trucos de diseñadores o lenguajes de programación.

3.1.3 API (Aplication Programming Interface)

Una API (del inglés Application Programming Interface) es un conjunto de reglas (código) y especificaciones que las aplicaciones pueden seguir para comunicarse entre ellas: sirviendo de interfaz entre programas diferentes de la misma manera en que la interfaz de usuario facilita la interacción humano-software. [28]

Las API pueden servir para comunicarse con el sistema operativo (WinAPI), con bases de datos o con protocolos de comunicaciones. Las API son valiosas, ante todo, porque permiten hacer uso de funciones ya existentes en otro software o de la infraestructura ya existente en otras plataformas; reutilizando así código que se sabe que está probado y que funciona correctamente.

En el caso de herramientas propietarias (es decir, que no sean de código abierto), son un modo de hacer saber a los programadores de otras aplicaciones cómo incorporar una funcionalidad concreta sin por ello tener que proporcionar información acerca de cómo se realiza internamente el proceso.

Como ejemplo de API se tiene que los desarrolladores de un programa cualquiera para Windows que se conecte a Internet no necesitan incluir en su código las funciones necesarias para reconocer la tarjeta de red, ya que basta una 'llamada' a la API correspondiente del sistema operativo.

Otro ejemplo es que los webmasters pueden incluir en sus webs de forma automática productos actualizados de Amazon o eBay, permitiendo iniciar el proceso de compra desde su web; al igual que los botones de "+1" de los blogs son llamadas a la API de Google.

3.1.4 SCRIPT

Los scripts son un conjunto de instrucciones generalmente almacenadas en un archivo de texto que deben ser interpretados línea a línea en tiempo real para su ejecución. [29]

Los scripts en el ámbito Web son utilizados al navegar por internet, más específicamente, se ejecutan en los navegadores como Internet Explorer, Google Chrome o Firefox, estos scripts son llamados "Scripts del lado del cliente", dado que son ejecutados en la computadora de quien visita el sitio web; en contraposición están los "Scripts del lado del servidor", que se ejecutan en el servidor del sitio web (el usuario solo ve el resultado del script).

Los scripts del lado del cliente suelen ser escritos en el lenguaje JavaScript o también en VBScript (sólo para Internet Explorer y Chrome)

Estos scripts le agregan muchas funcionalidades a los sitios web como validación de datos de un formulario, juegos, botones interactivos, cargar información de forma asíncrona, etc. Es muy común que los usuarios se quejen porque los navegadores dan "error en script". Usualmente el error desaparece si el usuario emplea otro navegador web o elimina todo el caché e historial del navegador afectado.

3.1.5 JAVA SCRIPT

Javascript es un lenguaje de programación que se puede utilizar para construir sitios Web y para hacerlos más interactivos. [30]

Aunque comparte muchas de las características y de las estructuras del lenguaje Java, fue desarrollado independientemente. El lenguaje Javascript puede interactuar con el código HTML, permitiendo a los programadores web utilizar contenido dinámico.

El lenguaje Javascript es opensource, por lo cualquier persona puede utilizarlo sin comprar una licencia.

3.2. Web Real-Time Communication (WEBRTC)

3.2.1 Definición

WebRTC (del inglés Web Real-Time Communication) es un conjunto de estándares, protocolos y APIs de JavaScript, los cuales combinados permiten una comunicación peer-to-

peer (igual a igual) de información de audio, video y datos por medio de los navegadores (browser). [31]

En lugar de depender de plug-ins de terceros o algún software propietario, WebRTC convierte la comunicación en tiempo real en una característica estándar que cualquier aplicación web puede aprovechar a través de una sencilla API de JavaScript.

Con muchos requerimientos y de alta calidad, las aplicaciones de RTC (Real-Time Communication) como audio y video-teleconferencia, y además del intercambio peer-to-peer de datos; requieren que el navegador provea una gran cantidad de nuevas funciones como por ejemplo capacidades de audio y de procesamiento de vídeo, nuevas APIs de aplicaciones, y soporte a los nuevos protocolos de red.

En la práctica, el navegador abstrae la mayor parte de estos requerimientos complejos en la funcionalidad de tres APIs principales:

- MediaStream: permite a un navegador web acceder a la cámara y el micrófono.
- PeerConnection: establece las llamadas de audio y vídeo
- DataChannel: permiten a los navegadores a compartir datos a través de peer-to-peer

3.2.2. Las normas y el Desarrollo de WebRTC

La habilitación de la comunicación en tiempo real mediante un navegador web es una propuesta ambiciosa, y posiblemente, una de las adiciones más significativas en la plataforma web desde sus inicios.

WebRTC rompe con el modelo tradicional de cliente - servidor en el ámbito de comunicaciones, lo que se traduce en una reingeniería completa de la capa de red en el navegador web, y también trae una pila de medios completamente nuevo, que se requiere para permitir eficiente, procesamiento en tiempo real de audio y vídeo. [31]

Como resultado de esto, la arquitectura WebRTC consta de más de una docena de normas diferentes, que abarca tanto las aplicaciones de navegador y los APIs, así como muchos protocolos diferentes y formatos de datos necesarios para que funcione:

- El grupo de trabajo Web en Tiempo Real Comunicaciones (WebRTC) W3C es responsable de definir las API del navegador.
- El grupo de trabajo de la IETF denominado Comunicación en tiempo real en la Web los navegadores (RTCWEB), es responsable de definir los protocolos , formatos de datos , seguridad y todos los demás aspectos necesarios para permitir la comunicación de igual a igual en el navegador.

El propósito principal de WebRTC es permitir la comunicación en tiempo real entre los navegadores, está diseñada de tal manera que se puede integrar con los sistemas de comunicación existentes, por ejemplo la aplicación de VOIP, varios clientes SIP, e incluso la red telefónica pública conmutada (PSTN).

Los estándares y normas de WebRTC no definen los requisitos de interoperabilidad específicos o las API, pero lo intentan utilizar los mismos conceptos y protocolos de las aplicaciones existentes cuando sea posible.

Con WebRTC no sólo se trata de llevar la comunicación en tiempo real por medio de un navegador, sino también busca mostrar las capacidades de la Web para las telecomunicaciones en todo el mundo; no en vano se trata de un desarrollo significativo y muchos proveedores de telecomunicaciones y empresas están siguiendo muy de cerca el desarrollo y establecimiento de WebRTC, y lo ven como mucho más que un simple API de navegador.

3.3. Compatibilidad de navegadores de forma nativa con WEBRTC

Aunque el objetivo de WEBRTC es ser transparente para el usuario, esto no significa que todos los navegadores tengan las mismas características; diferentes navegadores tendrán más desarrollo en alguna característica específica de los componentes de WEBRTC. [32]

Hay varios sitios web que pueden decir si el navegador soporta una tecnología específica, uno de ellos es "http://caniuse.com/rtcpeerconnection", que indica los navegadores que soportan WebRTC.

La comprobación de la aplicación de VoIP por medio de WEBRTC, en Smartphone esta fuera de los alcance de este trabajo de tesis, por lo cual no se tomó en cuenta los sistemas operativos iOS y Android.

Figura 3.1: Características de WEBRTC y navegadores que lo soportan [32]

3.3. 1. Compatibilidad con Chrome, Firefox y Opera

Los navegadores Chrome, Firefox y Opera soportan WEBRTC, y debería de funcionar en cualquier sistema operativo ya sea Windows, Linux y Mac.

Los proveedores de navegadores, como Chrome y Firefox, también han estado trabajando juntos para solucionar los temas de la interoperabilidad, para que todos puedan comunicarse entre sí con facilidad.

Las versiones de los navegadores que soportan WebRTC es:

- Google Chrome versión 23 y superior
- Mozilla Firefox versión 22 y superior
- Opera versión 18 y superior

3.3. 2. Compatibilidad con Apple

Apple ha hecho pocos esfuerzos para habilitar WebRTC en el navegador Safari o en iOS, hay rumores de apoyo a WEBRTC, pero no hay fecha oficial de soporte. Una solución que se ha utilizado para nativos híbrido basado en web o aplicaciones iOS es incorporar el código WebRTC directamente en sus aplicaciones y cargar una aplicación WebRTC en una página de WebView.

3.3. 3. Compatibilidad con Internet Explorer

Microsoft no ha anunciado planes para activar WebRTC en Internet Explorer. Han propuesto una solución alternativa para permitir comunicación de audio y vídeo en el navegador, esta alternativa fue rechazada en favor de WebRTC. Desde entonces, Microsoft ha sido un socio silencioso en el desarrollo de la esta tecnología.

No se ha tomado en cuenta el navegador Edge de Microsoft para el desarrollo de este trabajo de tesis.

3.4. Análisis de la arquitectura WEBRTC

WebRTC ofrece a los desarrolladores de aplicaciones web la capacidad de escribir aplicaciones multimedia de mayor nivel, en tiempo real en la web, sin necesidad de plugins, descarga o instalar algún software adicional. Su propósito es ayudar a construir una sólida plataforma de "Comunicación en Tiempo Real" (RTC) que funciona a través de múltiples navegadores web, en múltiples plataformas.

Es importante aclarar que WebRTC es el nombre utilizado por el organismo de estandarización W3C (World Wide Web Consortium) y RTCWeb es el nombre utilizado por el grupo de trabajo "The Real-Time Communications on the Web" del IETF (Internet Engineering Task Force), pero ambos buscan el desarrollo de WebRTC. [33] [34]

El W3C se encarga de estandarizar la tecnología desde la perspectiva de los navegadores y tecnologías Web (HTML5, CSS, etc.) y de definir APIs estándar para que los desarrolles Web utilicen WebRTC en sus aplicaciones.

El IEFT se ha centrado en los protocolos y herramientas que la tecnología utilizará a nivel de transporte, como SRTP, STUN/ICE/TURN, y códec.

Aunque se ha avanzado mucho, el proceso de estandarización de WebRTC está llevando más tiempo del previsto, debido sobre todo a ciertos elementos claves en la comunicación en tiempo real: método de establecimiento y finalización de llamadas en conferencias con muchos participantes, códec de vídeo, seguridad, QoS, etc.

En la Tabla 3.1 se muestra una comparación muy general, entre los tradicionales sistemas IP de comunicaciones en tiempo real y WebRTC. WebRTC ignora la parte de señalización, permitiendo a los fabricantes de sistemas VoIP emplear WebRTC independientemente del protocolo de señalización empleado para establecer la llamada. Las opciones más populares para la señalización son: JSON (JavaScript Object Notation) sobre WebSockets y SIP (Session Initiation Protocol) sobre WebSockets.

Los componentes dentro de WebRTC, como se muestra en la Figura 3.2, se tiene los codecs, motores de medios y capa de transporte. Esto facilita el uso de comunicaciones en tiempo real por parte de los desarrolladores de aplicaciones, sin necesidad de ser expertos en estas tecnologías.

Característica	VoIP	WebRTC
Señalización	SIP (principalmente) y H.323	Sin definir
Medios	RTP/RTCP	RTP/RTCP
Codecs de voz	Serie G.7xxx (principalmente)	G.711 y Opus
Codecs de vídeo	H.263, H.264	VP8
Seguridad de los medios	SRTP/TLS/IPsec	SRTP

Tabla 3.1: Comparativa entre sistemas tradicionales de VoIP y WebRTC [33]

La arquitectura global de WebRTC se ve de la siguiente manera (figura 3.2)

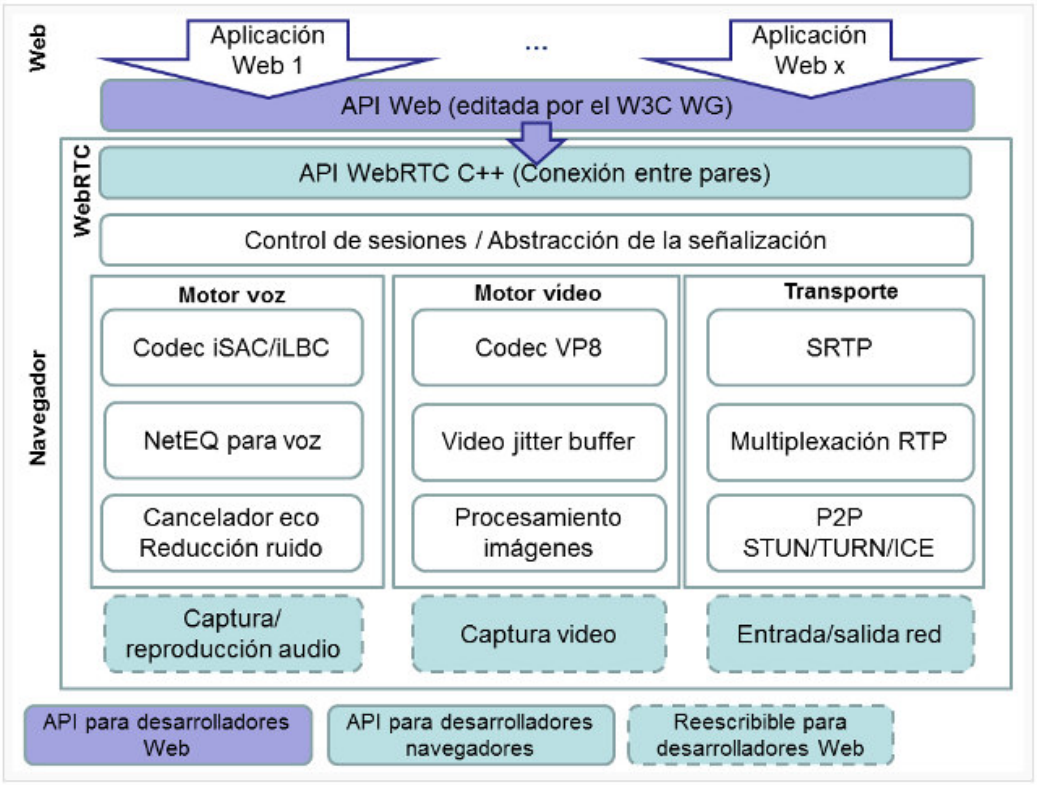


Figura 3.2: Arquitectura de WebRTC [33]

3.4.1. Motores de Audio y Video

La necesidad de contar con un navegador que sea capaz de proveer una videoconferencia de alta calidad, requiere que el navegador pueda acceder al hardware del sistema para capturar tanto audio como video y no depender de plug-ins o controladores personalizados, sólo con una simple API que sea consistente. [31]

Sin embargo, los flujos de audio y vídeo en el origen requieren que cada flujo deba ser procesado para mejorar la calidad; se necesita que el sincronismo y la tasa de bits de salida se ajuste al ancho de banda del canal de transmisión y la latencia entre los clientes.

En el extremo receptor el proceso se invierte, el cliente debe decodificar los flujos en tiempo real, debe ser capaz de adaptarse al jitter de la red y a los retardos de latencia.

De ahí que la captura y procesamiento de audio y video es un problema complejo; sin embargo, WebRTC trae motores de audio y vídeo con todas las funciones en el navegador como se muestra en la Figura 3.3, que se encargan de todo el procesamiento de señales.

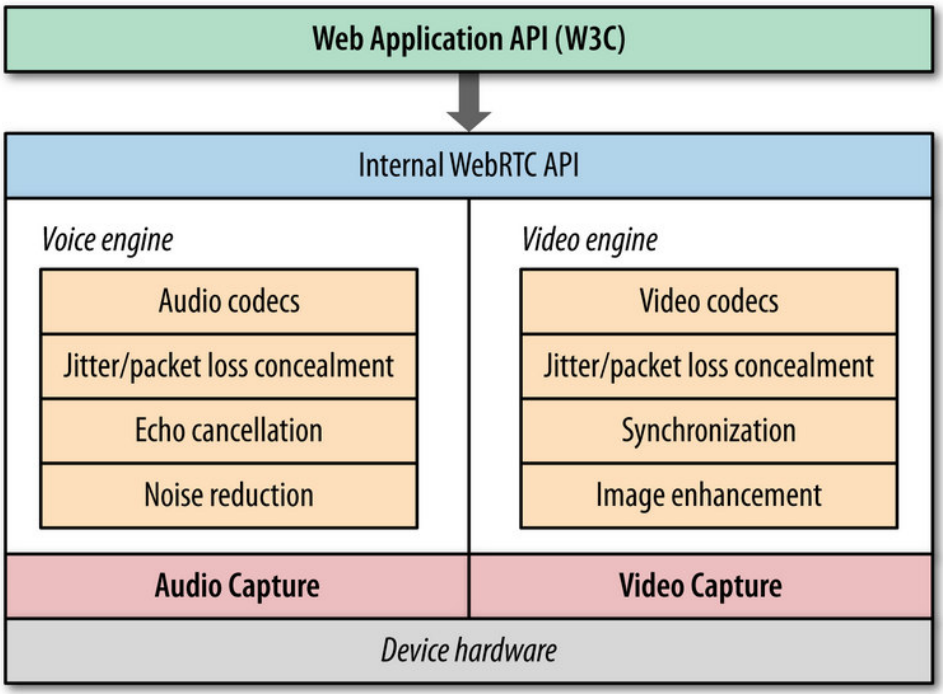


Figura 3.3: Motores de Audio y video en WebRTC [31]

El flujo de audio transmitido se procesa para reducir el ruido y cancelar el eco, luego se codifica de forma automática con uno de los códecs de banda estrecha optimizado o un códecs de audio de banda ancha; por último, para los errores se utiliza un algoritmo de ocultamiento para minimizar los efectos negativos de jitter y pérdida de paquetes en la red. El motor de vídeo realiza un procesamiento similar mediante la selección de los ajustes de compresión, utilización de códecs, etc., optimizando la calidad de la imagen.

Todo el procesamiento lo realiza directamente el navegador, además realiza de manera dinámica los ajustes si los parámetros cambian con el flujo de audio y video debido a las condiciones de la red. Una vez que todo este trabajo se realiza, la aplicación web recibe un flujo de medios optimizado.

Cuando el navegador solicita audio y vídeo, se debe prestar especial atención al tamaño y la calidad de los flujos; si el hardware es capaz de capturar los flujos de alta calidad, también el procesamiento del CPU y ancho de banda deben ser capaces de trabajar a ese ritmo.

Las implementaciones actuales de WebRTC actuales utilizan los códecs Opus y VP8:

- El códec Opus se utiliza para audio y soporta codificación con tasas de bits constante y variable y requieren de 6 a 510 Kbit/s de ancho de banda. El códec se puede cambiar sin problemas y se adapta al ancho de banda variable.
- El códec VP8 utilizado para la codificación de vídeo también requiere de 100-2.000 + Kbit / s de ancho de banda, y la tasa de bits depende de la calidad de los streams.

Para la realización de este trabajo de tesis el códec de audio para VoIP dependerá del servidor WebRTC, en su mayoría utilizan OPUS.

3.5. APIs de WebRTC

Las APIs WebRTC de W3C permiten a una aplicación JavaScript aprovechar las capacidades nuevas del navegador para aplicaciones en tiempo real; el navegador en este caso proporciona

la funcionalidad necesaria para la comunicación de audio, video y datos, como se muestra en la Figura 3.4. [8] [31]

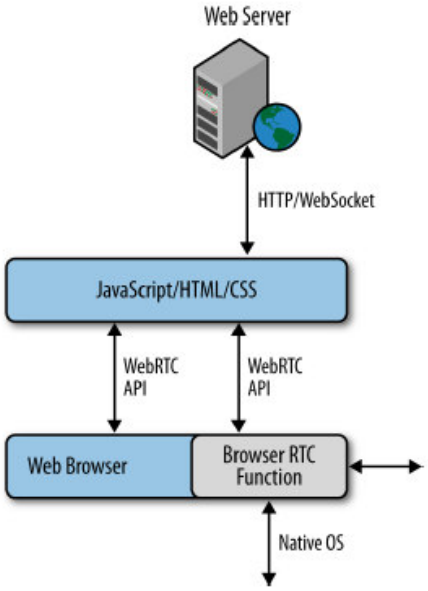


Figura 3.4: Comunicación en Tiempo Real en el Navegador [8]

El cifrado es una característica obligatoria de WebRTC, se aplica a todos los componentes incluidos los mecanismos de señalización; todos los flujos de medios enviados a través de WebRTC se codifican mediante protocolos de encriptación estandarizados y bien conocidos. El protocolo de cifrado que se utiliza depende del tipo de medio o canal, asi los flujos de datos son encriptados usando Datagramas de Transport Layer Security (DTLS) y los flujos de medios son encriptados usando Secure Real Transport Protocolo (SRTP). [35]

Las APIs están diseñadas en torno a tres conceptos principales: MediaStream, PeerConnection y DataChannel

En el apartado 3.2.1, se establecieron las tres APIs que tiene WebRTC, para realizar este trabajo de tesis solo se analizara las que son necesarias para la transmisión de VOIP.

3.5.1. MediaStream

La API MediaStream es una representación abstracta de un flujo real de datos de audio, video o la combinación de ambas; sirve por ejemplo para capturar la pantalla del escritorio de un par remoto.

Para crear y utilizar un flujo de datos local, la aplicación web debe solicitar el acceso del usuario a través de la función getUserMedia (); la aplicación especificara el tipo de medios de comunicación de audio o vídeo a la que se requiere el acceso.

El selector de dispositivos en la interfaz del navegador sirve como mecanismo para permitir o denegar el acceso. Una vez que la aplicación se ejecuta, puede revocar su propio acceso llamando a la función stop () en el LocalMediaStream.

3.5.2 PeerConnection

La API PeerConnection permite que dos usuarios se comuniquen directamente, de un navegador a otro; esto representa una asociación con el par remoto, que suele ser otra instancia de la misma aplicación JavaScript que se ejecuta en el navegador del otro extremo.

La comunicación se coordina mediante un canal de señalización que proporciona un código script de comandos en la página a través del servidor web; por ejemplo, usando XMLHttpRequest o WebSocket. Una vez que se establece una conexión entre pares, los flujos del stream se pueden enviar directamente en el navegador remoto.

La API PeerConnection utiliza el protocolo ICE (del inglés Interactive Connectivity Establishment) junto con los servidores STUN y TURN para que los flujos de datos encapsulados en UDP puedan a travesar NAT transversales y firewalls.

El protocolo ICE permite a los navegadores descubrir y conocer la suficiente información acerca de la topología de la red para encontrar la mejor ruta de comunicación disponible. El uso de ICE también proporciona una medida de seguridad, ya que evita páginas y aplicaciones web no confiables envíen datos a los hosts que no están esperando recibir los mismos.

El proceso de conexión se la realiza a través de la una negociación donde interviene el protocolo SDP, y se lo realiza de la siguiente manera:

- Se ejecuta la funcion “createOffer()” en el navegador, para lo cual se crea un objeto llamado RTCSessionDescription, al que se denomina “oferta”.
- Se ejecuta la función “setLocalDescription()”en el navegador, con el objeto RTCSessionDescription para completar los datos de información.
- Se remite el paquete SDP generado en esta sesión al otro extremo donde se encuentra el cliente, esto se lo realiza por el canal de señalización.
- En el otro extremo el navegador crea una respuesta, mediante la función “createAnswer()”, en el objeto RTCSessionDescription.
- Al finaliza el proceso de negociación, la aplicación llama a la función “setRemoteDescription()”, y se ejecuta en el navegador la configuración pactada y con esto se podrá empezar a comunicar.

En la figura 3.5 se observa el proceso de negociación descrito.

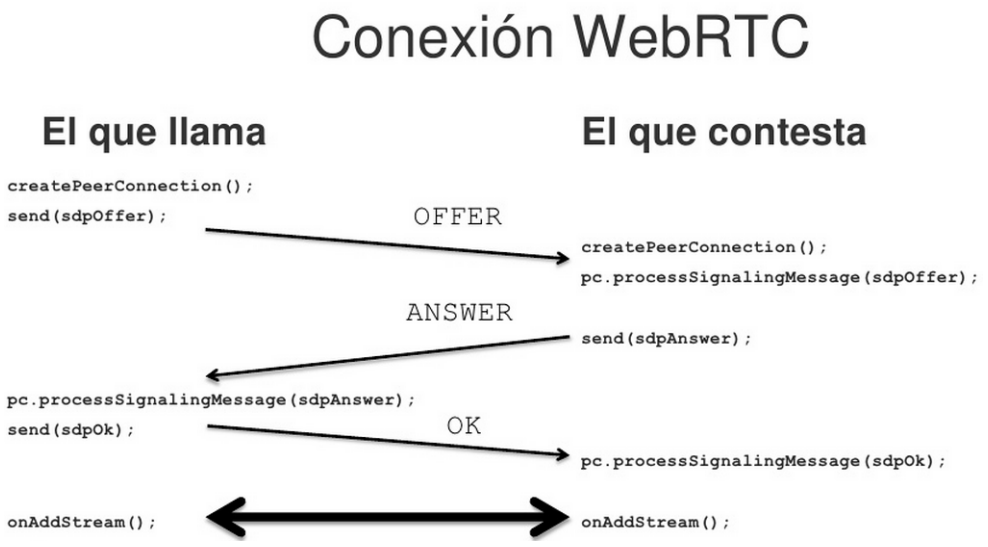


Figura 3.5: Proceso de negociación PeerConnection [31]

En el Apéndice D se han tomado las capturas para la comunicación de VOIP mediante WebRTC, con la herramienta Wireshark.

3.5.2.1. STUN (Session Traversal Utilities for Network Address Translators)

El servidor STUN es la primera alternativa para tener conexión entre dos peer (entidades iguales). Identificar a cada usuario en la Internet, y se lo utilizan otros protocolos para realizar conexiones entre peer.

El proceso se inicia cuando un cliente realiza una solicitud a un servidor habilitado con el protocolo STUN, este servidor identifica la dirección IP del cliente que realiza la petición y la devuelve la nueva dirección IP, con esto el cliente podrá identificarse con la IP dada.

Para que se pueda hacer uso del protocolo STUN en el cliente, se requiere que el servicio este habilitado en un servidor STUN para conectarse al mismo; en la actualidad Firefox y Chrome, provee servidores con este servicio.

3.5.2.2 TURN (Traversal Using Relay for Network Address Translators)

El servidor TURN es útil cuando hay elementos detrás de un NAT simétrico o Firewall que quieren estar en el lado de recepción de una conexión con otro elemento exterior. TURN no permite que los usuarios tengan servidores en puertos determinados si están tras un NAT; y si soporta la conexión de un usuario que esté tras un NAT con un peer único. Dicho de otro modo, su papel es proporcionar las mismas funciones de seguridad que dan los NAT simétricos, pero volteando (to turn) las tablas, para que el elemento en el tramo local esté en la parte de recepción, en vez de estar en la de transmisión, de una conexión pedida por el cliente externo. [36]

3.5.2.3 ICE (Interactive Connectivity Establishment)

El protocolo ICE es un mecanismo que permite a los peer establecer una conexión. En la práctica, los clientes por lo general no tienen una conexión directa a la Internet; están conectados a través de los dispositivos de red denominados routers, y tienen asignadas direcciones IP privadas, y como seguridad utilizan NAT, firewalls de red, y otros dispositivos. Con el fin de establecer una conexión igual a igual, por definición, los peer deben ser capaces de enrutar paquetes entre sí, ya que se encuentran en redes privadas distintas. Lo que se necesita es un camino de enrutamiento en la red pública entre los peer, para esto WebRTC provee soporte para estos casos, y lo gestiona de la siguiente manera: [31]

- Cada objeto de conexión PeerConnection contiene un " agente ICE
- Cada Agente ICE es responsable de reunir la IP local y el puerto (se denomina candidato)
- El Agente ICE es responsable de realizar comprobaciones de conectividad entre pares.
- El Agente ICE es responsable de enviar mensajes de actividad de conexión.

Una vez que una sesión local o remota se establece, el agente local ICE comienza automáticamente el proceso de descubrir todo el posible candidato IP y puerto para los peers locales, lo realiza de la siguiente manera:

- El Agente ICE realiza la consulta al sistema operativo para obtener las direcciones IP locales.
- Si se ha configurado el STUN, el agente ICE consulta a este servidor externo para conocer la IP publica y el puerto de los peers.
- Si se ha configurado el TURN, el agente ICE añade el servidor TURN como último recurso para conocer el candidato (dirección IP y puertos). Si la conexión “peer to peer” falla, los datos se transmiten a través de un intermediario especificado previamente.

Cada vez que se descubre un nuevo candidato (IP, puerto), el agente registra al candidato automáticamente con el objeto RTCPeerConnection y notifica a la aplicación a través de una función de devolución de llamada (onicecandidate).

Una vez que la recolección de ICE se ha completado, la misma llamada es devuelta para notificar a la aplicación.

3.6. Señalización en WebRTC

WebRTC permite la comunicación de igual a igual, pero cada aplicación WebRTC también necesitará un servidor de señalización para negociar y establecer la conexión.

En WebRTC la señalización es un concepto abstracto y no está definida; debido a esto la señalización es un tema tan complejo y difícil de resolver, los fabricantes de WebRTC ofrecen muchos recurso pero ningún protocolo tiene consenso como para que sea elevado a estándar y presentar la mejor opción de señalización al cliente.

Las opciones más populares para la señalización son: JSON (JavaScript Object Notation) sobre WebSockets y SIP (Session Initiation Protocol) sobre WebSockets. [34].

En la figura 3.6 se muestra algunas opciones de señalización en WebRTC.

Signaling Options in WebRTC

Technique	Advantages	Server Side Requirements
WebSocket	No signaling infrastructure required	WebSocket server with proprietary protocol implementation
XHR	No signaling infrastructure required	Web server with proprietary protocol implementation
SIP over WebSocket	Easy path to SIP interoperability (and PSTN connectivity)	SIP Registrar/Proxy server with support of SIP over WebSocket
XMPP/Jingle	Easy interoperability with XMPP clients	XMPP server with support of XMPP WebSocket transport
Data Channel	Low latency signaling and improved privacy	WebSocket or web server with proprietary implementation to establish the Data Channel

Figura 3.6: Opciones de Señalización en WebRTC [37]

No hay una opción única y estandarizada para utilizar un servidor de señalización, la elección depende de los requisitos de la aplicación. Sin embargo, se debe examinar las opciones comerciales y de código abierto disponibles, y por supuesto, prestar mucha atención a la señalización de transporte subyacente, ya que puede tener un impacto significativo tanto en la latencia del canal de señalización y la sobrecarga en la comunicación entre el cliente y el servidor.

3.6.1. WebSockets

WebSocket es una tecnología que permite un canal de comunicación bidireccional y full-duplex sobre un único socket TCP, orientado a mensajes de texto y datos entre cliente y servidor y está concebida para ser implementada en navegadores y servidores web, pero puede utilizarse por cualquier aplicación cliente/servidor.

La API de WebSocket está siendo normalizada por el W3C, mientras que el protocolo WebSocket ya fue normalizado por la IETF como el RFC 6455. Debido a que las conexiones TCP comunes sobre puertos diferentes al 80 son habitualmente bloqueadas por los administradores de redes, el uso de esta tecnología proporcionaría una solución a este tipo de limitaciones proveyendo una funcionalidad similar a la apertura de varias conexiones en distintos puertos, pero multiplexando diferentes servicios WebSocket sobre un único puerto TCP (a costa de una pequeña sobrecarga del protocolo).

La complejidad tras el Navegador Web se abstrae sobre una sencilla API que ofrece una serie de servicios adicionales:

- Negociación de la conexión y la aplicación de políticas en el origen de la comunicación.
- Interoperabilidad con la infraestructura HTTP existente.
- Comunicación orientada a mensajes y una eficiente encapsulación del mensaje en tramas

- Un sub protocolo de negociación y la extensibilidad

La URL de recursos de WebSocket utiliza su propio esquema personalizado, y puede ser de dos formas:

1.- ws para la comunicación de texto sin formato (por ejemplo, ws: //example.com/socket), y

2.- wss cuando se requiere un canal cifrado, utilizando los protocolos TCP y TLS.

3.7. Arquitectura según los componentes

WebRTC hace referencia por definición a aplicaciones peer-to-peer o igual a igual, no se debe entender que la comunicación en forma tácita es navegador a navegador; el análisis de la topología física enfoca este paradigma de la comunicación de igual a igual entre los navegadores.

Una de las topologías WebRTC más general se aprecia en la Figura 3.7, esta apoya en la llamada mediante SIP y tiene forma de un Trapecio. [8]

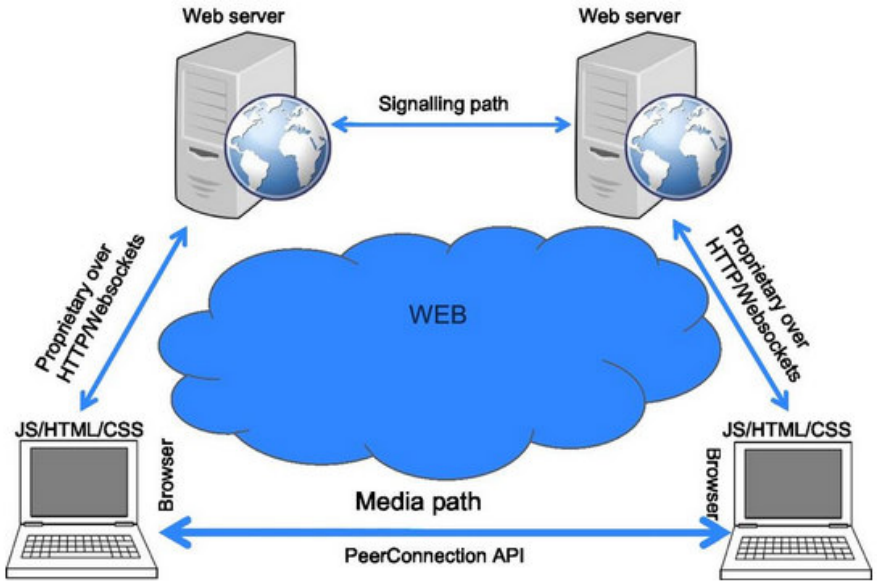


Figura 3.7: Trapecio WebRTC [8]

El escenario WebRTC más común es probable que sea aquel en el que ambos navegadores están ejecutando la misma aplicación web, desde la misma página web. En este caso la topología tiene forma de triángulo, como se muestra en la Figura 3.8.

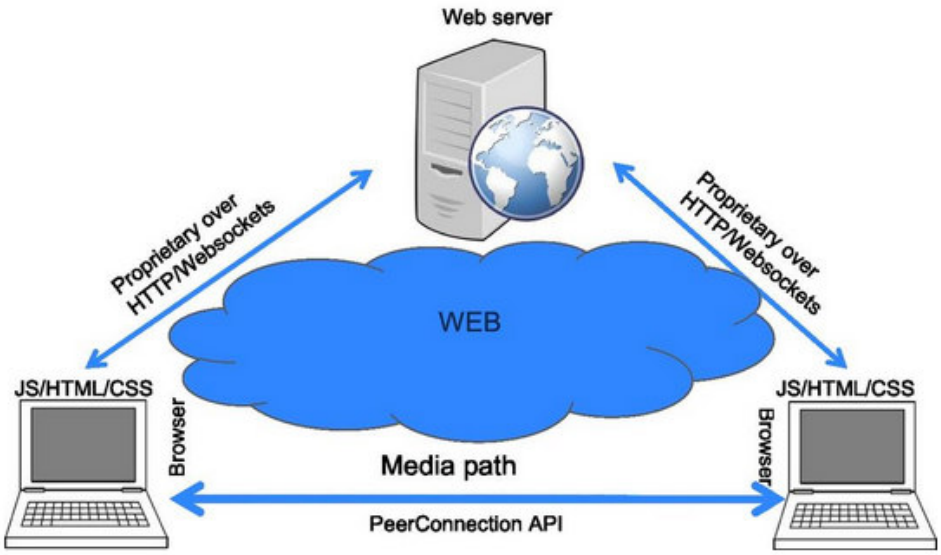


Figura 3.8: Triangulo WebRTC [8]

En el modelo de WebRTC Triangulo (Figura 3.8) ambos navegadores están ejecutando una aplicación web, desde la página web en un mismo servidor. Se establece los parámetros de la conexión entre los navegadores y el servidor, para luego la empezar la trasmisión de VoIP (en nuestro caso de estudio) directamente entre los dos navegadores; la señalización puede ir a través de HTTP o WebSockets. Vale la pena señalar que la señalización entre el navegador y el servidor no está estandarizado en WebRTC, ya que se considera que es parte de la aplicación.

Para el desarrollo de este trabajo de tesis se tendrá topologías tipo Triangulo para WebRTC.

CAPÍTULO 4

Introducción

En el capítulo 4, se detallan las pruebas realizadas y los análisis con las alternativas de servicios para implementar VoIP mediante el servicio de WebRTC en la Nube, tomando como referencia parámetros de Calidad de Servicio (QoS), y con los requerimientos de un caso típico de PYMES plantear la red que se debe tener.

4 Características de posible implementación y pruebas.

4.1 Servidores de WebRTC

Para la posible implementación de WEBRTC se eligió 3 ambientes

- Asterisk (Licencia GPL)
- 3CX(Licencia)
- Servidores Gratuitos

4.1.1 Servidor Asterisk

Utilizar software bajo licencia GPL, anteriormente basada solamente en Linux, actualmente es soportada para Windows, Mac OS y Solaris, Asterisk es una de las mejores opciones para el uso de centrales telefónicas y el uso de VOIP. Actualmente provee mejor soporte en la versión para Linux, generalmente orientado a Linux Centos aunque es compatible con Ubuntu y Debian.

Una de las características nuevas incorporadas a esta central telefónica es WebRTC, el que se encuentra documentado dentro de la página oficial de Asterisk.

La implementación requiere de características especiales para poder arrancar el servidor.

En el Apéndice A se detalla la configuración del servidor Asterisk.

4.1.2 Servidor 3CX

Es una central telefónica con licencia para Windows 7 y versiones superiores, orientado a redes telefónicas VOIP, soporta dentro de sus características a WEBRTC de manera gratuita, ya que ofrece además de ser una central telefónica también otros servicios embebidos, como WebMeeting. La central telefónica permite la configuración de manera limitada de extensiones

a menos que se active la licencia.

3CX es una opción confiable debido a que las configuraciones son sencillas, los instaladores generalmente se encargan de instalar todas las dependencias y permitir que el software esté listo para usarse, simplemente se debe configurar los parámetros necesarios para la central telefónica.

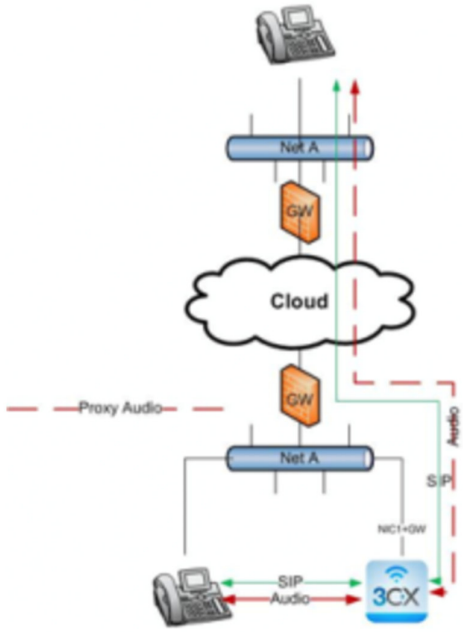


Figura 4.1. Modelo del funcionamiento del servidor 3CX. [38]

En el Apéndice B se detalla la configuración del servidor 3CX.

4.1.3 Servidores Gratuitos

WebRTC es un proyecto gratuito que permite la comunicación en tiempo real RTC es una API de bajo nivel en JavaScript que provee a los navegadores la habilidad de establecer comunicación de audio y video, pero no solamente es necesario implementar este API, en realidad como se ha analizado en las implementaciones de 3CX y Asterisk, se requiere de muchos elementos para el funcionamiento correcto.

WebRTC requiere de más característica adicionales al API entre ellas el NAT, los firewalls configurados, acceso al servidor STUN, que exista una señalización en los servidores para entregar mensajes entre los navegadores; debido a que esto puede extender y aumentar el costo de utilización de los servicios existen los servidores gratuitos de WebRTC, esta opción es bastante cómoda y económica al momento de realizar las comunicaciones mediante el navegador, únicamente es necesario acceder al servidor mediante la URL, añadir una sala e invitar a los demás participantes, es una buena opción emergente para el uso de VoIP.

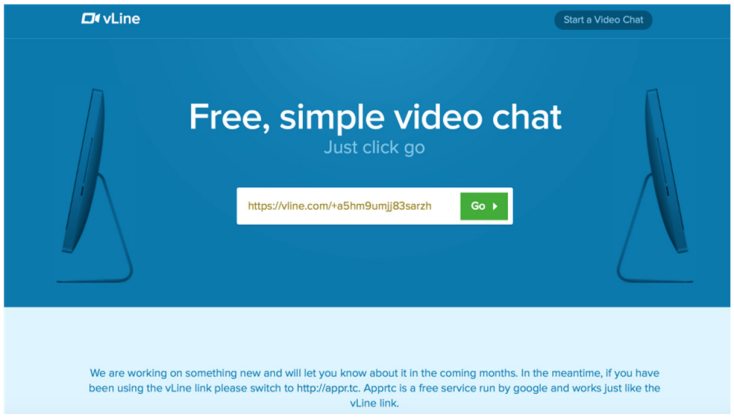


Figura 4.2 Inicio de sesión servidor Vline. (Autor de la Tesis 2015)

Existen algunos sitios web que proveen este servicio gratuito sin necesidad de crear cuentas de usuarios y contraseñas, simplemente se requiere usar el navegador compatible y utilizar el servicio, entre los servidores más populares alojados en la Nube, se tiene:

Servidor	Link
ApprRTC	https://apprtc.appspot.com/
Vline	https://vline.com
Talky	https://talky.io

Tabla 4.1 Servidores gratuitos de WebRTC. (Autor de la Tesis 2015)

Estos servidores realizan la comunicación de manera sencilla donde los clientes se conectan a un servidor y realizan la comunicación.

Figura 4.3 Comunicación WebRTC mediante Chrome y Firefox (Autor de la Tesis 2015).

Hay que destacar que estos servidores no están únicamente orientadas a VoIP, sino que también permiten al usuario aplicaciones de video y chat, de tal manera que WebRTC se vuelve muy útil al momento de realizar comunicaciones, debido a que la comunicación es en tiempo real y sencillo de implementar.

El uso es muy sencillo, simplemente una vez que se accedió a la página hay que copiar la URL que automáticamente se muestra y enviar al usuario que se desea invitar a la sala, además para identificarse se debe ingresar un NickName.

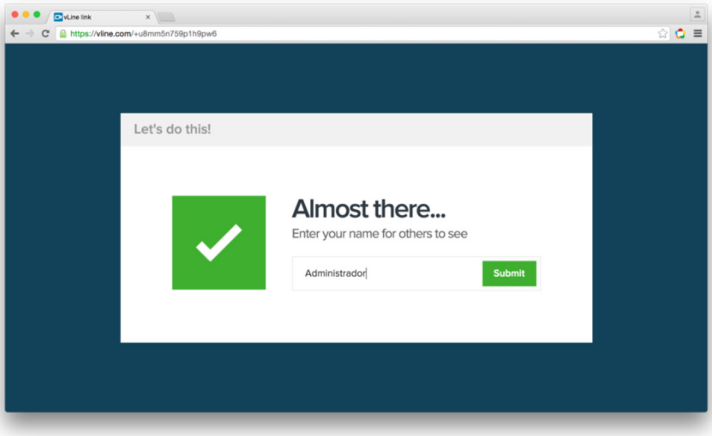


Figura 4.4 Ingreso de usuario en servidor Vline (Autor de la Tesis 2015)

Luego simplemente se comparte la dirección web que se muestra automáticamente

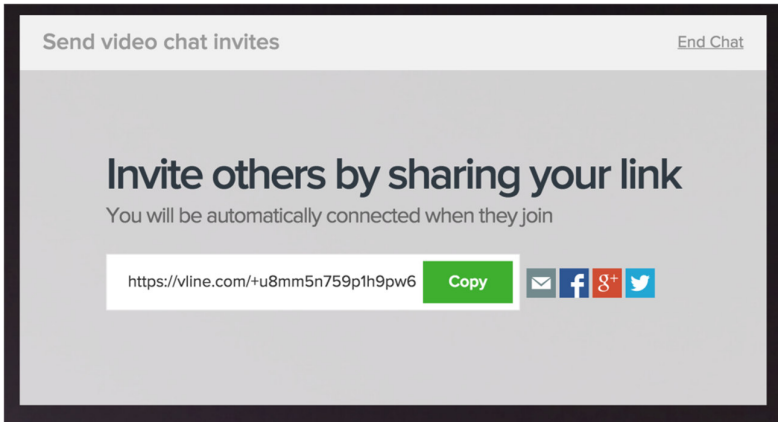


Figura 4.5 URL para conexión de clientes del servidor Vline (Autor de la Tesis 2015).

Con lo anteriormente realizado se tiene que esperar que los invitados se conecten y se podrá estar en la sala compartiendo los servicios antes mencionados.

Una de las ventajas al usar estos servidores es que no se necesita administrar la comunicación, sino que simplemente se debe agregar los invitados a la sala.

4.2 WebRTC en 3CX

3CX es una herramienta que permite con facilidad la comunicación con la tecnología WebRTC, por medio de este servidor los usuarios pueden transmitir voz y videos sobre navegadores web. La herramienta está disponible en 3CX como 3CX WebMeeting, esta herramienta puede ser instalada dentro de la LAN y también puede utilizarse como un servicio sin necesidad de configuraciones extras en los equipos, simplemente con la contratación del servicio.

Tanto la opción de instalar el servidor en la red local como la de contratar el servicio exponen versiones de prueba para que el cliente pueda hacer llamadas y probar las alternativas dentro de esta herramienta.

Se analizaran dos posibilidades para el estudio de WebRTC de 3CX

- Instalar Servidor 3CX en la red Local.
- Servidor 3CX en la nube.

4.2.1 Servidor 3CX en la red Local

La topología elegida para este tipo de configuración involucra a un router que permite segmentar la red y separarla de la red WAN en este caso Internet, mediante firewalls y habilitación de puertos.

Es importante recalcar que para el uso en una red local con cualquier servicio de WebRTC es necesario tener un ambiente de red que permita tener acceso a varias funcionalidades, como un dominio propio, servidores DNS que nos permitan apuntar al dominio y una IP pública para poder acceder desde el Internet a el servidor WebMeeting Local, si bien estas funcionalidades permiten tener el control total de la red dentro de la LAN y del Internet, se torna bastante costoso debido a que estas características son ofrecidas por proveedores de hosting y dominios; por esta razón se utilizara la herramienta WebMeeting de manera local debido a las limitaciones de la red LAN que se propone implementar como parte del estudio.

- ***Topología de la RED***

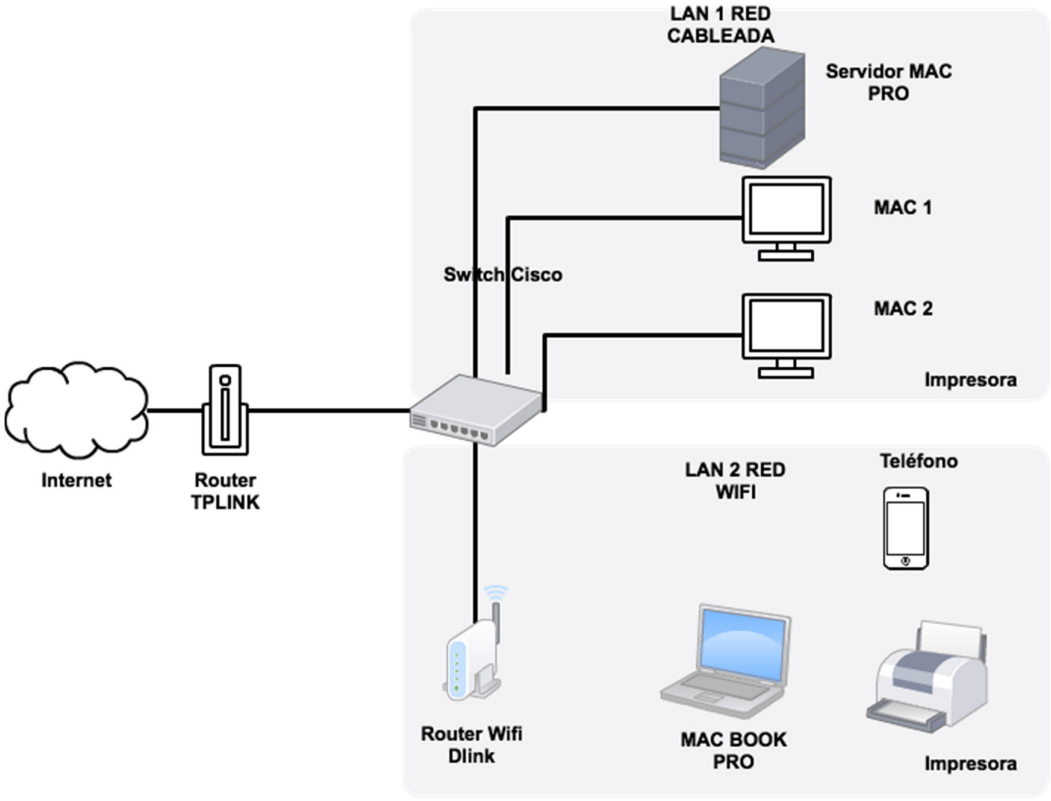


Figura 4.6 Topología de Red LAN para instalación de 3CX (Autor de la Tesis 2015)

La red LAN deriva en dos redes que se encuentran tras el router que sirve de frontera con el Internet; los equipos que se encuentran dentro de LAN1 se conectan mediante cable donde la red es Gigabit Ethernet y en la LAN2 se tiene una red inalámbrica con velocidad de 54 Mbp; como se muestra en la figura 4.6.

Los dispositivos usados para la red LAN son los siguientes:

RED	Equipo	IP
Internet	Router TP-LINK	201.125.215.211
LAN 1 Red Cableada	Router TP-LINK	192.168.2.1
	Switch Cisco	192.168.2.5 /administrable
	Servidor Mac Pro	192.168.2.11
	Máquina Virtual Windows 7	192.168.2.200
LAN2 Red Inalámbrica	Router D-Link	192.168.5.1
	Macbook Pro	192.168.5.100
	Impresora Epson	192.168.5.2

Tabla 4.2 Configuración de parámetros de red LAN para instalación de 3CX (Autor de la Tesis 2015)

WebMeeting será instalado en el Servidor Mac Pro mediante una máquina virtual con Windows 7.

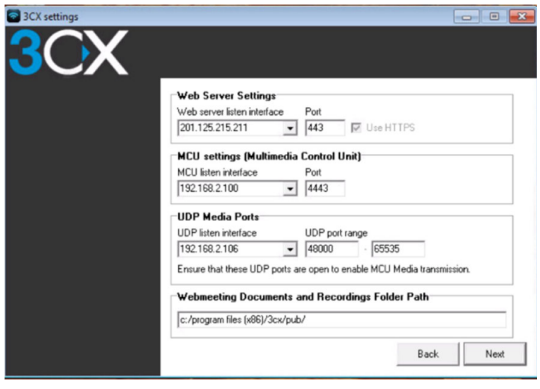


Figura 4.7 Configuración de parámetros del servidor 3CX. (Autor de la Tesis 2015)

La configuración de WebMeeting permite acceder al servidor que se encuentra en la red LAN mediante una IP pública, se debe configurar el firewall con los respectivos permisos de acceso, los puertos necesarios para la comunicación de voz están en el rango de 48000 - 65535, la IP local para el transporte de datagramas UDP que corresponde a la comunicación de voz y el puerto 4443 que permite comunicarse mediante una conexión segura con el protocolo HTTPS.

La apertura de puertos en el router permite que los clientes se conecten al servicio de manera eficiente, si no se apertura los puertos simplemente no se podrá realizar la comunicación.

443	192.168.2.100	ALL	Enabled	Modify Delete
4443	192.168.2.100	ALL	Enabled	Modify Delete
48000-65534	192.168.2.100	UDP	Enabled	Modify Delete

Figura 4.8 Puertos abiertos en router para una dirección IP privada. (Autor de la Tesis 2015)

4.2.2 3CX en la Nube

3CX también da la posibilidad de usar el servicio en la internet, la configuración es sencilla, permite administración a nivel de usuarios y se pueden crear salas de chat, la facilidad de acceso faculta que el administrador simplemente este encargado de los participante y de las salas de reuniones para acceder a la comunicación vía web.

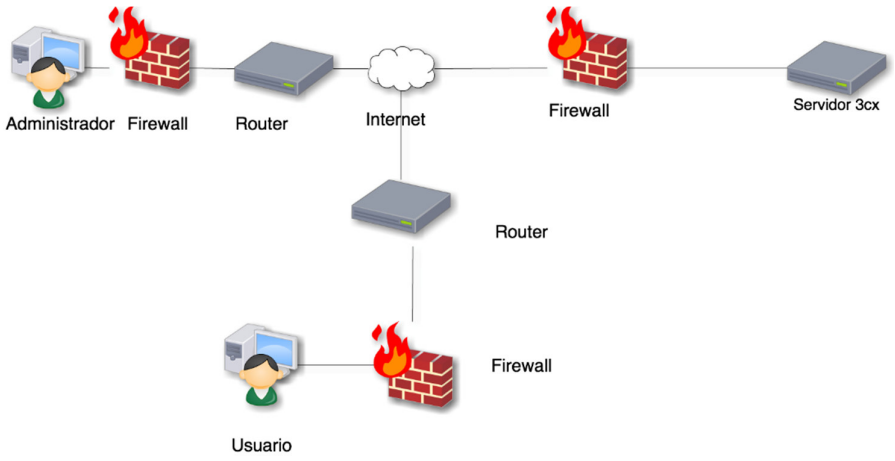


Figura 4.9 Esquema del servidor 3CX en la Nube. (Autor de la Tesis 2015)

La figura 4.9 muestra cómo se realiza la comunicación el momento de conectarse al servidor 3CX, se tiene la posibilidad de usar un servidor SMTP para el envío de las invitaciones vía correo, de tal forma que los participantes sean invitados exclusivamente mediante un correo y con una URL única.

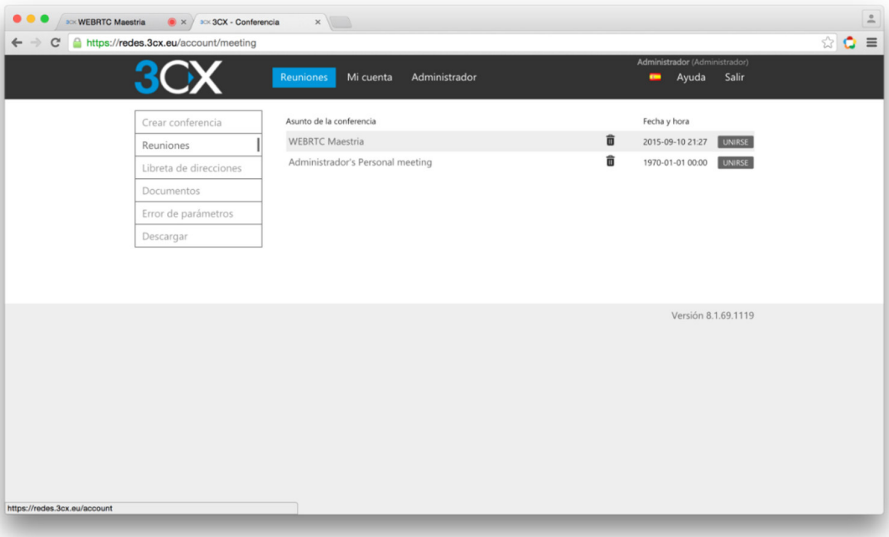


Figura 4.10 Administración de reuniones de 3CX en la Nube. (Autor de la Tesis 2015)

La conferencia puede tener un asunto según el tema a tratar y los usuarios o participantes pueden ser creados por el administrador para que puedan tener el acceso dentro de la conferencia.

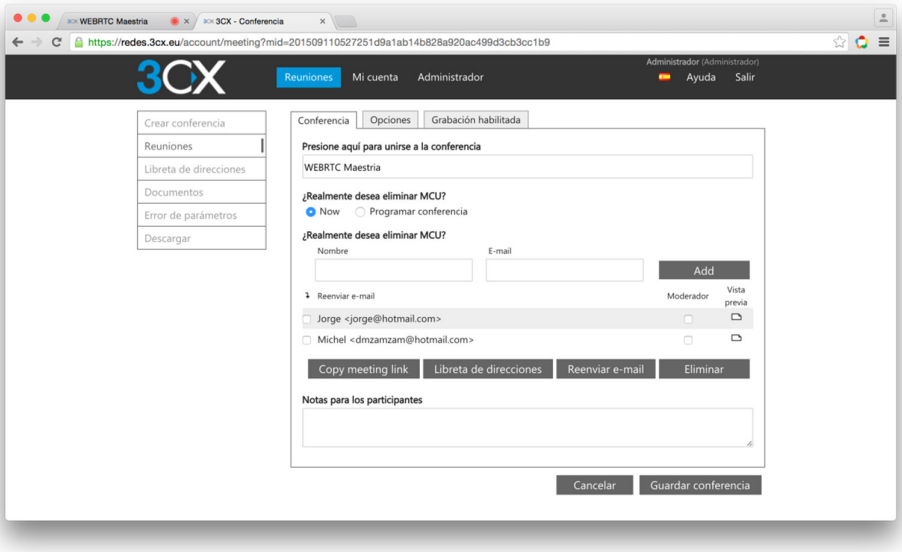


Figura 4.11 Ingreso de usuarios a conferencia 3CX en la Nube. (Autor de la Tesis 2015)

Se puede crear en la libreta de direcciones los participantes a manera de lista de contactos y estos pueden ser agregados a cualquier conferencia que sea creada, luego dentro de la sala de comunicación se puede tener aplicaciones de VoIP, Video y Chat.

4.3 Pruebas y Resultados de la LAN y Servidores WebRTC en la Nube

Las pruebas se han realizado en dos ambientes reales de red. El primer ambiente consta del servidor 3CX ubicado en la nube, específicamente el que se ha elegido está en Canadá, los clientes se comunicarán mediante varios navegadores, en diferentes lugares de Ecuador que mantengan conexión a internet, específicamente en Quito y Portoviejo.

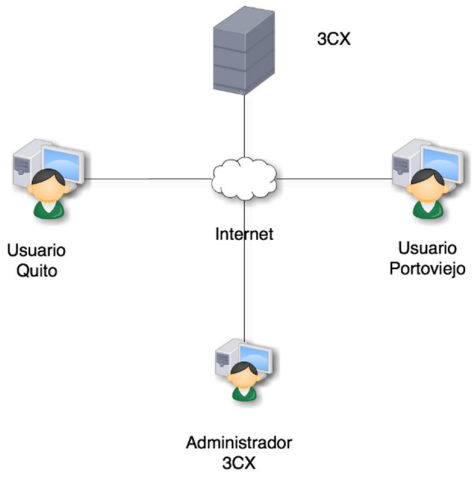


Figura 4.12 Diagrama de Red de cliente servidor de 3CX en la Nube. (Autor de la Tesis 2015)

El segundo ambiente de pruebas se realiza con tres (3) servidores gratuitos ubicados en la Nube:

- Vline
- AppRtc
- Talky

Tienen la misma topología de red de 3CX con la diferencia que no se puede administrar las salas de chat.

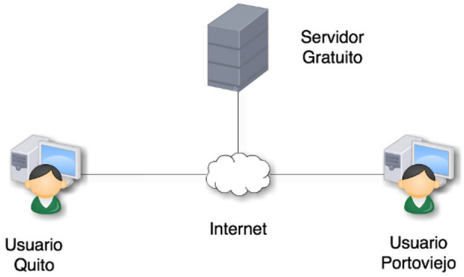


Figura 4.13 Diagrama de Red con servidor gratuitos en la Nube. (Autor de la Tesis 2015)

4.3.1 Pruebas de Laboratorio con diferentes Navegadores

WebRTC es una tecnología desarrollada por Google, siendo una herramienta de comunicación realmente nueva pero actualmente ya se encuentra soportado por otros navegadores, además de Google Chrome, como son Mozilla Firefox y Opera; los navegadores Safari e Internet Explorer no tienen compatibilidad con WebRTC por lo que en estas pruebas serán obviados.

Figura 4.14 Conexión de WebRTC mediante diferentes Navegadores. [39]

El soporte de estos navegadores también depende del proveedor del servicio de WebRTC, es decir que el proveedor indica que navegador es compatible y se puede utilizar el servicio de comunicación, por lo tanto a pesar de que WebRTC funciona con los navegadores mencionados generalmente el proveedor del servicio es quien indica como requerimiento el navegador a usar.

Los desarrolladores indican en sus requerimientos con que navegador existe compatibilidad, la Tabla 4.3 resume dos características soportadas por los navegadores con respecto al uso de los servidores de WebRTC, la parte de administración indica si el servicio es administrable en cuanto a la creación de usuarios, de salas, cronogramas, anchos de banda y parámetros administrables de WebRTC, además, se muestra si el navegador permite la comunicación es decir, permite hacer llamadas ya que puede permitir administración pero no la comunicación.

Navegador	Administración	Comunicación
Chrome	3CX	3CX,Vline,Apprtc,Talky
Mozilla Firefox	3CX	Vline, Apprtc, Talky
Opera	3CX	Vline, Apprtc, Talky
Safari	N/A	N/A
Internet Explorer	N/A	N/A

Tabla 4.3 Compatibilidad de Navegadores con servidores de WebRTC. (Autor de la Tesis 2015)

Los resultados muestran que los navegadores más estables para el uso de WebRTC son Google Chrome, Opera y Firefox; los que no soportan aún WebRTC son el Internet Explorer y Safari; además se debe especificar que los navegadores Chrome y Opera proveen una API, que permite ejecutar una llamada a métodos escritos en Java para la toma de muestreo en tiempo real.

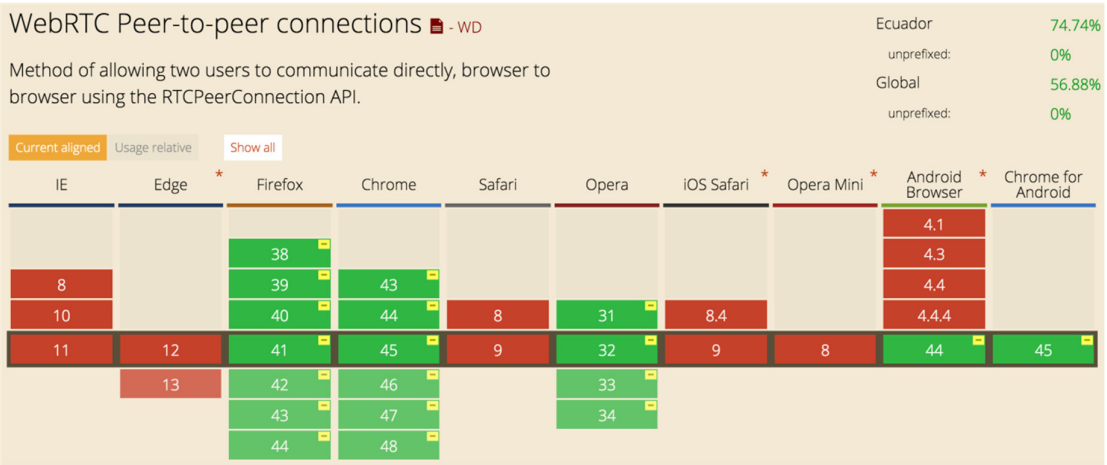


Figura 4.15 Estadística de uso de WebRTC en diferentes navegadores. [40]

Esta figura 4.15 muestra los navegadores disponibles en sus versiones y la compatibilidad con WebRTC, los que están marcados con rojo muestran los navegadores que no son compatibles y los que están de color verde indican los navegadores compatibles, la numeración que está incluida equivale a la versión el navegador, en la figura 4.15 además se observa el uso de WebRTC con conexiones "peer to peer" mediante un navegador y muestra que un 74,74% de navegadores en sus diferentes versiones son utilizados para comunicaciones en tiempo real con los navegadores compatibles en Ecuador.

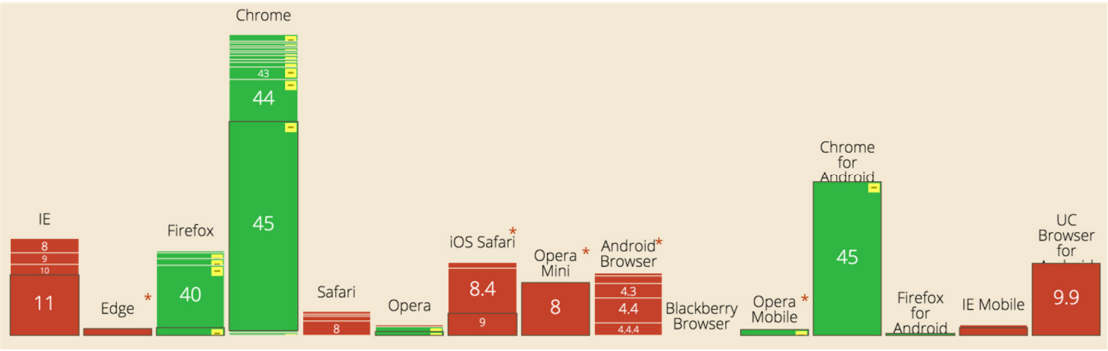


Figura 4.16 Estadísticas de uso de WebRTC en diferentes navegadores vista vertical. [40]

La figura 4.16 muestra que los usuarios prefieren Chrome, Firefox y Chrome for Android debido a que WebRTC es mejor soportado por estos navegadores.

Las pruebas de laboratorio que se han propuesto se han hecho de acuerdo a los requerimientos soportados por los servidores de WebRTC, estos se indican a continuación en la Tabla 4.4.

Navegador	Servidor Licencia en la Nube	Servidores Gratuitos en la Nube	Herramientas Captura de datos
Chrome	3CX	AppRTC	API:webrtc-internals
Firefox		Vline	Whireshark
Opera		Talky	

Tabla 4.5 Requerimientos para pruebas de laboratorio de WebRTC. (Autor de la Tesis 2015)

Se agregó una prueba de rendimiento de red, la cual permitirá evaluar inicialmente si la red es idónea para soportar paquetes UDP que son los que transportan los mensajes de voz, esta prueba da una idea de cómo podrían comportarse los paquetes UDP mientras se inyecta diferentes tamaños y cantidad de paquetes.

Para las pruebas preliminares de la red se ha elegido las siguientes herramientas:

- Herramienta de medición Iperf.
- Sistema Operativo Windows 7 Servidor
- Sistema Operativo Windows 8.1 Cliente

4.3.2 Observación y toma de valores de 3 parámetros de QoS en VoIP: RTT, Jitter y pérdida de paquetes

Las pruebas de WebRTC se harán mediante llamadas utilizando los diferentes servidores en la nube, para ello se ha incluido una tabla propuesta por CISCO, la que va a permitir comparar

los resultados de Jitter, RTT y pérdida de Paquetes.

Clasificación Calidad Voz Ip	Bueno	Aceptable	Pobre
Latencia [ms]	0ms -150ms	150ms - 300ms	> 300 ms
Jitter [ms]	0ms – 20ms	20 ms – 50 ms	> 50 ms
Perdida Paquetes [%]	0% - 0,5%	0,5% - 1,5%	> 1,5%

Tabla 4.6 Umbrales de Calidad de VoIP. [41]

Estos rangos de la Tabla 4.6 muestran los valores para la comunicación de voz en tiempo real, la toma de resultados es realizada mediante una herramienta web denominada **"WebRTC-internals/"**, la que posee una interfaz amigable y es fácil de recopilar estadísticas sin implementar la StatsAPI. Un usuario mientras el lapso de establecimiento o durante una sesión de WebRTC puede observar el rendimiento de la sesión en curso mediante la apertura de otra pestaña que apunte a chrome: // WebRTC-internals /, por ejemplo. Al final de la llamada, el usuario es capaz de exportar los registros, enviarlo por correo electrónico al proveedor de servicios (WebRTC o ISP) para su análisis.

Esta herramienta se encuentra integrada en los navegadores en los que WebRTC es compatible, particularmente con Opera y Google Chrome, permitiendo medir varios parámetros, los más importantes en el trabajo realizado son RTT, Jitter y paquetes perdidos tanto para paquetes enviados como para recibidos.

Para el análisis de los resultados utilizaremos la tabla referencial de los umbrales de QoS en VoIP sugeridos por Cisco para permitir una comunicación de calidad, además el análisis se hará en el cliente ubicado en la ciudad de Quito en modo envío.

4.3.3 Resultados de la LAN con Iperf

Se utilizara el software Iperf para hacer tomas de datos en diferentes momentos de la comunicación; Iperf trabaja en modo cliente servidor y se conecta mediante el puerto 5201, el servidor debe permitir la conexión del cliente mediante el puerto mencionado. Iperf puede ser instalado en MAC OS, Windows o Linux, el ambiente creado para las pruebas consta de un cliente Iperf con Windows 8, y el servidor Iperf con Windows 7.

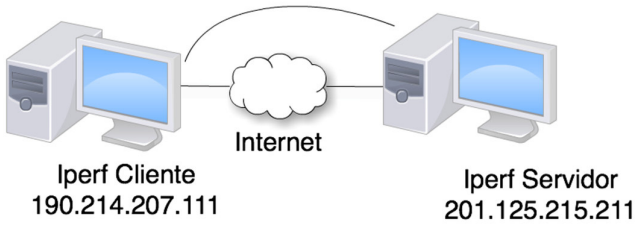


Figura 4.17 Topología de Red en la Nube para pruebas con Iperf. (Autor de la Tesis 2015)

Se ejecutó desde el cliente con Windows 8 en modo –c que equivale a modo cliente el comando –c y la IP del servidor, como resultado se observa en la figura 4.18 que por defecto se muestran diez registros de la petición que se ha realizado, además se refleja la información de ancho de banda del enlace tanto para el envío como para la recepción.

El comando utilizado es: Iperf3 –c 201.125.215.211.

La segunda prueba con el cliente, figura 4.18, utiliza el parámetro –u que permite inyectar paquetes UDP, de esta manera realiza medidas de Jitter y perdida de paquetes, además se puede argumentar el ancho de banda deseado con el parámetro –b e ingresar el valor deseado.

El comando utilizado es: Iperf3 –c 201.125.215.211 –u –b 10m.

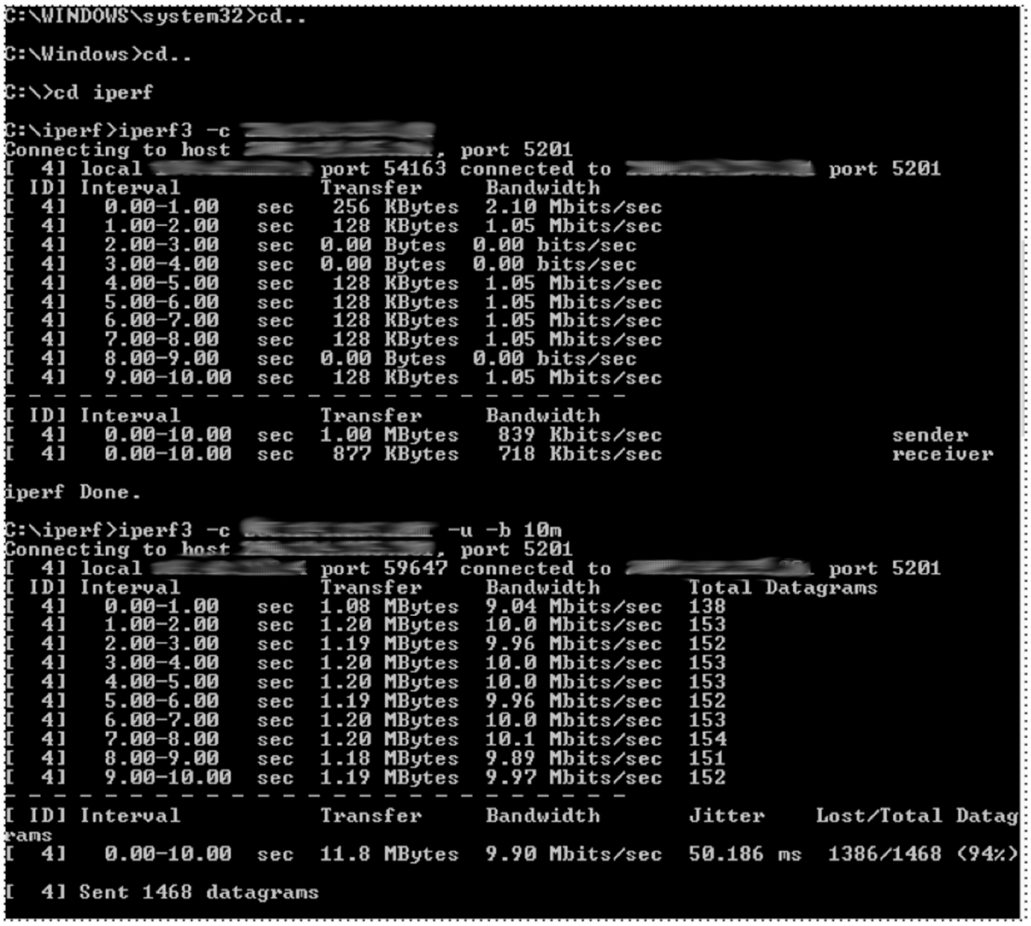


Figura 4.18 Resultados del Servidor Iperf. (Autor de la Tesis 2015)

En la figura 4.19 se observa la respuesta del servidor a la prueba de conectividad del ancho de banda de paquetes recibidos; el ancho de banda puede variar su disponibilidad por varios factores, entre ellos por el tráfico en el enlace o por la calidad de servicios de ISP, en el resultado se muestra cambios de ancho de banda pero no muestra valores muy variables por lo tanto podemos decir que la prueba de ancho de banda es aceptable y nos provee estabilidad en la conexión.

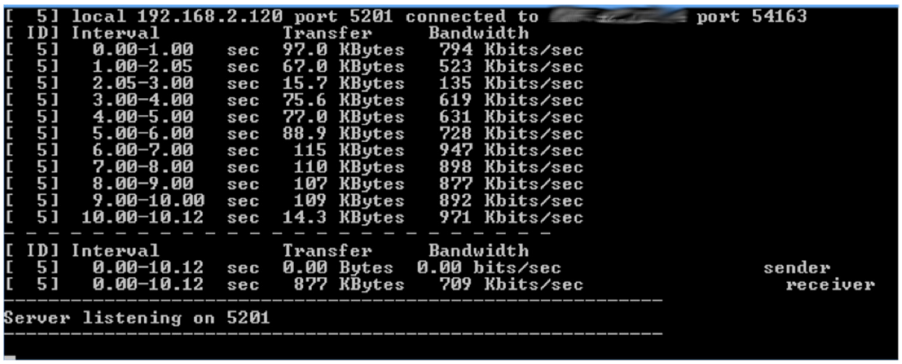


Figura 4.19 Resultados Cliente Iperf en pruebas de Ancho de Banda. (Autor de la Tesis 2015)

En la figura 4.20 se muestra la respuesta a la ejecución del segundo comando que envía paquetes UDP desde el cliente, se observa un Jitter promedio de 50,186ms que para una comunicación de voz es bastante aceptable, por lo que mediante esta prueba podemos decir que la red en la cual se realizan las pruebas debe permitir una comunicación estable. Se puede observar una pérdida de paquetes alta debido a que se envían paquetes con un ancho de banda –b de 10Mbits y en la primera prueba se obtuvo como resultado que el ancho de banda era 839 Kbits.

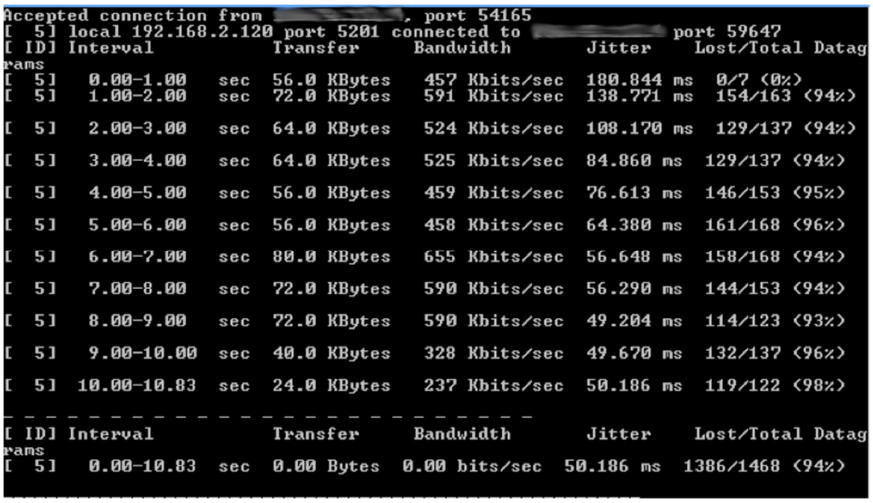


Figura 4.20 Resultados Cliente Iperf - pruebas de inyección de paquetes UDP. (Autor de la Tesis 2015)

Transferidos MB	Bandwith Mb/sec	Jitter ms	Perdidos	% Perdido	Parámetro
11,80	9,90	50,86	1386/1468	94%	10m
11,80	9,90	58,02	1402/1498	94%	10m
1,24	1,04	37,34	32/158	20%	64 bits
1,24	1,04	48,86	36/153	24%	10-64 bits
1,24	1,04	47,71	32/157	20%	
1,24	1,04	45,19	32/159	20%	10000 -64 bits
1,24	1,04	76,41	54/159	34%	
114,00	95,9	167,06	0/9	0%	10000m

Tabla 4.7. Resultados del Cliente con diferentes inyecciones de paquetes UDP. (Autor de la Tesis 2015)

La tabla 4.7 muestra los resultados de las pruebas realizadas desde el cliente al servidor, se ha sometido a varias pruebas causando un estrés a la red mediante la inyección de paquetes UDP. Los resultados muestran también que el Jitter es aceptable cuando se mantenga los parámetros de inyección de paquetes dentro del ancho de banda soportado, es decir, mientras mayor congestión se genere en la red y el ancho de banda este a su límite se tiene que el Jitter será más alto, según los resultados se puede deducir que el canal que se va a utilizar es totalmente apto para las comunicaciones debido a que el ancho de banda de subida y bajada es mayor al requerido por VoIP que es de 64Kbps según RFC 3714, además que el enlace con inyección de paquetes mayores a 64Kbps generan un Jitter menor a 150 ms el que resulta aceptable para la comunicación.

4.3.4 Resultados con el Servidor AppRTC

Como primer escenario se evaluó el servidor gratuito en la Nube AppRTC, se lo realizo mediante el API llamándola en la URL "opera:webrtc-internals"; los resultados obtenidos son los siguientes:

- **Captura Wireshark**

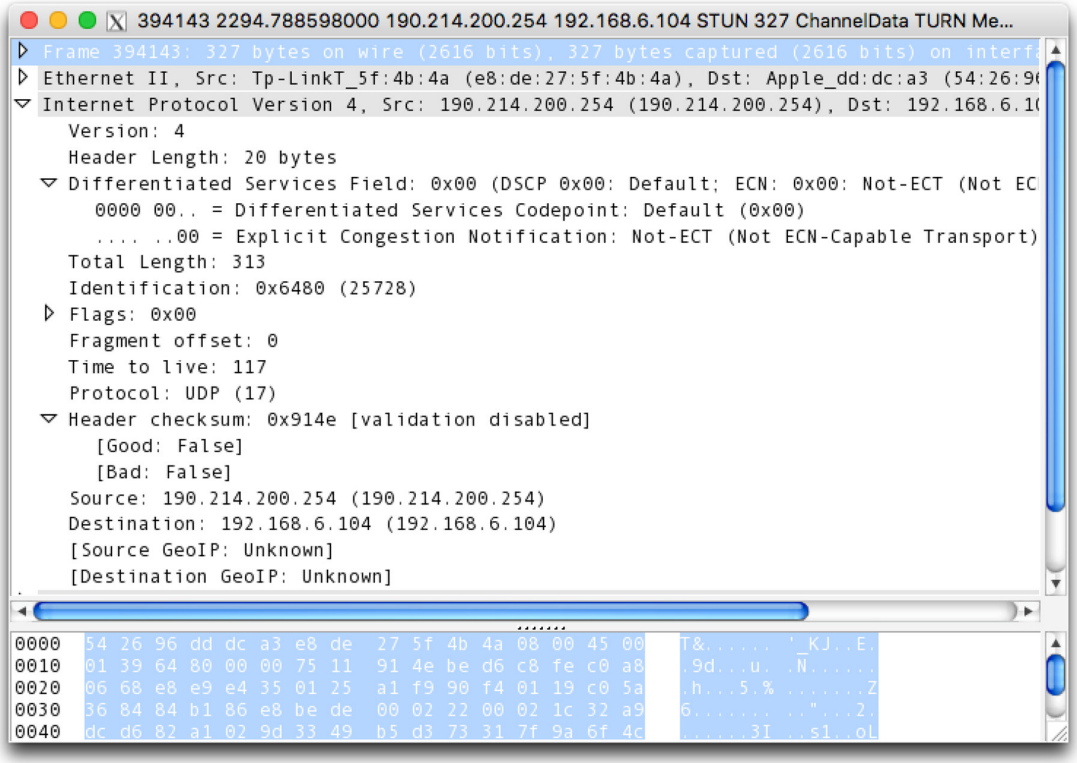


Figura 4.21 Captura paquete del servidor AppRTC mediante Wireshark. (Autor de la Tesis 2015)

La figura 4.21 muestra un paquete UDP del servidor STUN donde se muestran las direcciones IPs de Origen 190.214.200.254 y destino 192.168.6.104; la IP de origen de este paquete muestra la IP del servidor AppRTC y la de destino la IP Local del receptor ubicado en la ciudad de Quito, se puede destacar que es un paquete STUN donde se ha traducido la IP pública del receptor en una IP local de destino.

Modo Envío		Modo recepción	
Origen	2469130290	Origen	2786176042
Servidor Turn	104.155.45.181		
Servidor Webrtc	APPRTC	Servidor Webrtc	APPRTC
Bytes Enviados	16472704	Bytes Recibidos	19250924
Paquetes perdidos	1417	Paquetes perdidos	162
Paquetes enviados	171204	Paquetes recibidos	190944
CODEC	Opus	CODEC	Opus
RTT	69	RTT	0
JITTER	5	JITTER	0
IP LAN Origen	192.168.6.104	IP LAN Origen	192.168.1.7
IP Servidor	190.214.200.254	IP Servidor	190.214.200.254
IP WAN Origen	201.125.215.211	IP WAN Origen	190.214.207.111

Tabla 4.8. Captura de datos del servidor AppRTC mediante webrtc-internals. (Autor de la Tesis 2015)

Los resultados específicamente representan datos obtenidos en una línea de tiempo específica, es decir, no equivale a resultados promedios sino resultados en un momento puntual, gráficamente esta herramienta si presenta datos en diferentes momentos.

El API de Opera captura los datos de envío y de recepción tanto para audio y video, los resultados son codificados mediante un número, de tal forma que se pueden separar los resultados de audio enviados y recibidos de los resultados de video.

En la tabla 4.8, se observa el código 2469130290 que equivale a todos los paquetes enviados y el código de origen 2786176042 que equivale a los resultados de los paquetes recibidos. Una de las características del servidor AppRTC es que utiliza un servidor Turn y el modo de codificación de voz es Opus.

- ***RTT AppRTC***

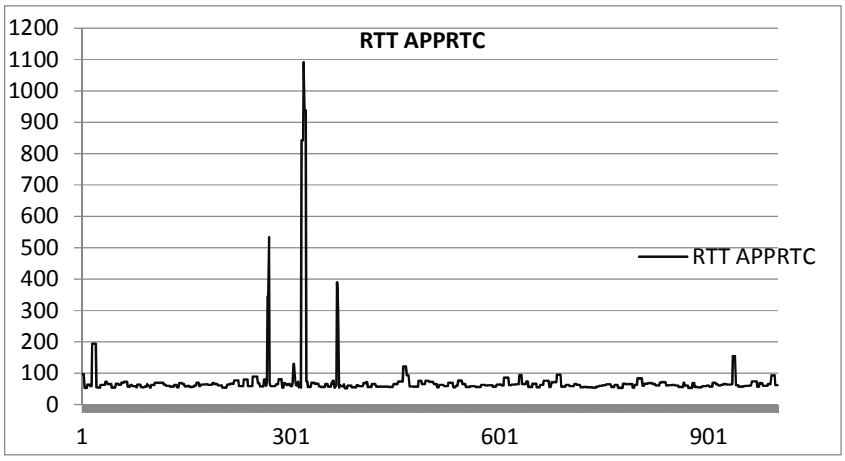


Figura 4.22 Resultados de RTT del servidor AppRTC. (Autor de la Tesis 2015)

- ***JITTER AppRTC***

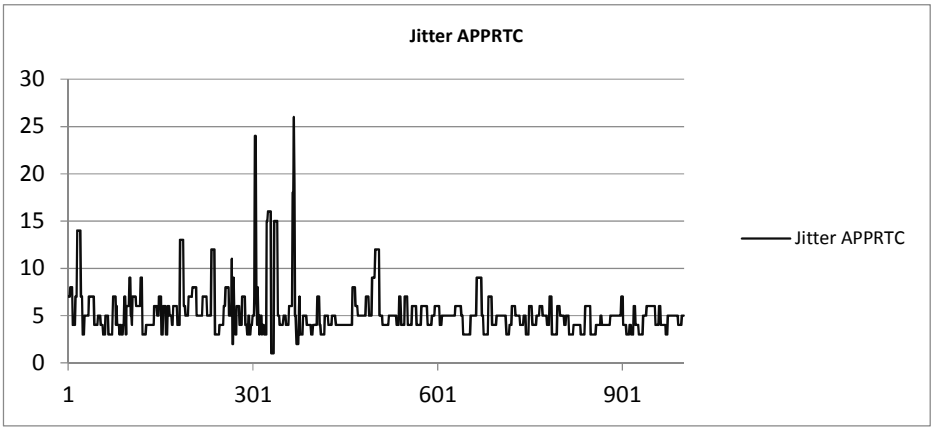


Figura 4.23 Resultados de JITTER del servidor AppRTC. (Autor de la Tesis 2015)

En las figuras 4.22 y 4.23, el Eje X representa el número de muestras que se han tomado en un intervalo de tiempo, para este caso el valor es de 990 muestras, el eje Y equivale al parámetro Jitter (figura 4.22) y RTT (figura 4.23), este eje muestra los valores en milisegundos con los cuales se parametrizan estas variables.

APPRTC	RTT(ms)	JITTER(ms)
Promedio	73,12	5,29
Pico Bajo	53	1
Pico Alto	1092	26
Resultado	Bueno	Bueno

Tabla 4.9. Promedio de JITTER y RTT del servidor AppRTC. (Autor de la Tesis 2015)

El servidor AppRTC muestra un desempeño realmente bueno con relación a los parámetros de RTT y JITTER, el promedio de estos parámetros se observan en la Tabla 4.9 y se puede determinar que la comunicación está dentro de los rangos buenos de la comunicación de voz, gráficamente en las figuras 4.22 y 4.23 se puede ver que la comunicación se mantiene estable, esto a pesar de los picos altos que se presentan.

- ***Perdida de Paquetes AppRTC***

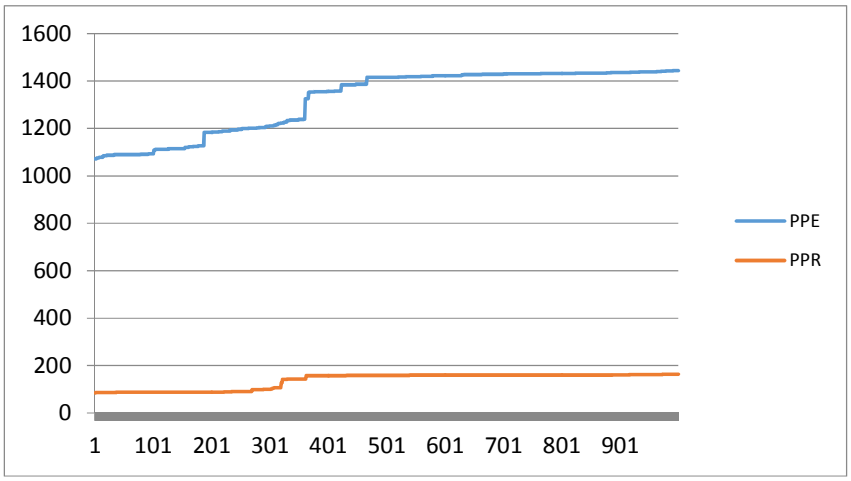


Figura 4.24. Resultado pérdida de paquetes del servidor AppRTC. (Autor de la Tesis 2015)

La pérdida de paquetes es menos del 1%, como se muestra en la tabla 4.10, lo que indica que la comunicación tiene más del 99% de paquetes entregados, es decir, que la comunicación se mantiene aceptable, el valor de RTT es de 73,12 ms, por lo que la comunicación es buena dado que debiera ser inferior a 150 ms, dado que el oído humano es capaz de detectar latencias de unos 250 ms, de la misma manera el Jitter es bueno, de acuerdo a la tabla de referencia de Cisco.

APPRTC	Paquetes	Porcentaje
P.Enviados	194620	100%
PPE	1444	0,74%
P.Recibidos	194446	100%
PPR	163	0,08%

Tabla 4.10 Porcentaje pérdida de paquetes del servidor AppRTC. (Autor de la Tesis 2015)

4.3.5 Resultados con el Servidor Vline

El segundo escenario corresponde a el servidor Vline, utilizando el navegador Google Chrome, en el que se ha obtenido los siguientes resultados mediante el API "chrome:webrtc-internals":

- **Captura Wireshark**

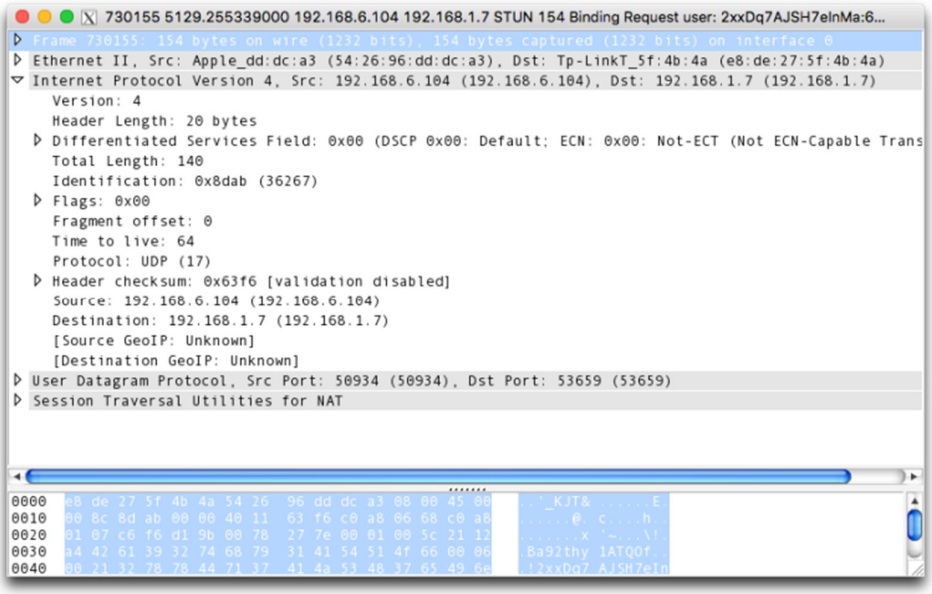


Figura 4.25. Captura paquete del servidor Vline mediante Wireshark. (Autor de la Tesis 2015)

La figura 4.25 muestra un paquete UDP donde se muestran dos direcciones IPs, la de origen 192.168.6.104 y de destino 192.168.1.7; estas direcciones muestran dos direcciones IPs privadas, las que han sido identificadas como los equipos o clientes conectados los cuales están realizando la comunicación, se puede ver las direcciones locales de la LAN gracias al servidor STUN quien se encarga de mapear las direcciones públicas con las privadas.

Modo Envio		Modo recepción	
Source	3050614237		2592777114
Servidor STUN	stun.pro.vline.com		
Servidor Webrtc	Vline	Servidor Webrtc	Vline
Bytes Enviados	867643	Bytes Recibidos	19250924
Paquetes perdidos	18	Paquetes perdidos	162
Paquetes enviados	9579	Paquetes recibidos	190944
CODEC	ISAC	CODEC	ISAC
RTT	336	RTT	0
JITTER	5	JITTER	4
IP LAN Origen	192.168.6.104	IP LAN Origen	192.168.1.7
IP LAN Destino	192.168.1.7	IP LAN Destino	192.168.6.104
IP Servidor	54.177.19.208	IP Servidor	54.177.19.208
IP WAN Origen	201.125.215.211	IP WAN Origen	190.214.207.111

Tabla 4.11 Captura de datos del servidor Vline con webrtc-internals. (Autor de la Tesis 2015)

En la Tabla 4.11, los paquetes de origen codificados mediante el numero 3050614237 y los de recepción con el numero 2592777114 muestran en un momento dado un RRT de 336 ms y un Jitter de 5 ms, de acuerdo a la tabla de referencia de Cisco tenemos que el retardo está en un rango pobre, y el Jitter en un rango bueno; el códec utilizado para esta comunicación es el ISAC. Aun no podemos concluir respecto a la calidad de comunicación debido a que los datos generados en la tabla corresponden a una toma aleatoria de la comunicación.

- ***RTT Vline***

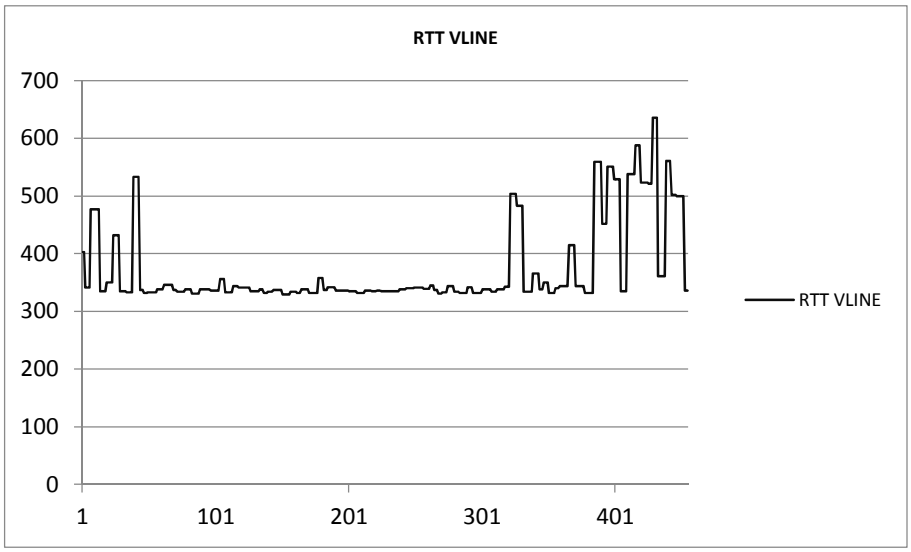


Figura 4.26 Resultado de RTT del servidor VLine. (Autor de la Tesis 2015)

- ***JITTER Vline***

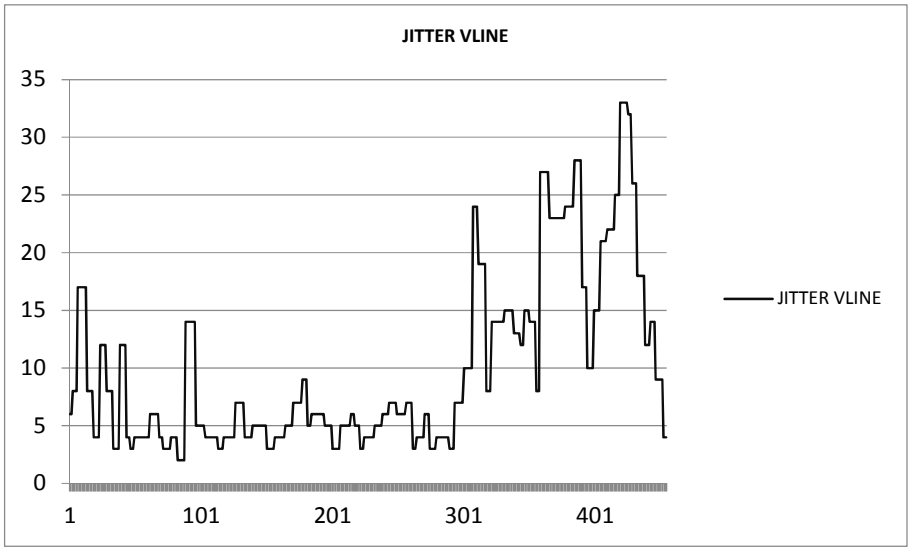


Figura 4.27. Resultado de JITTER del servidor VLine. (Autor de la Tesis 2015)

Los resultados en las figuras 4.26 y 4.27 de RTT y Jitter respectivamente, muestran que dentro de 453 muestras que corresponden al eje X los parámetros van variando en milisegundos, de manera visual se puede identificar que el RTT es bastante alto con relación a una comunicación aceptable y también el Jitter se mantiene variable debido a la variación de los retardos.

VLINE	RTT(ms)	JITTER(ms)
PROMEDIO	372,98	11,92
Pico Bajo	329	2
Pico Alto	636	33
Resultado	Pobre	Bueno

Tabla 4.12 Promedio de JITTER y RTT del servidor Vline. (Autor de la Tesis 2015)

Los promedios de las variables de la comunicación analizada, Tabla 4.12, muestran que los retardos siempre fueron mayores a el rango pobre y que el Jitter es bueno.

- ***Perdida de paquetes Vline***

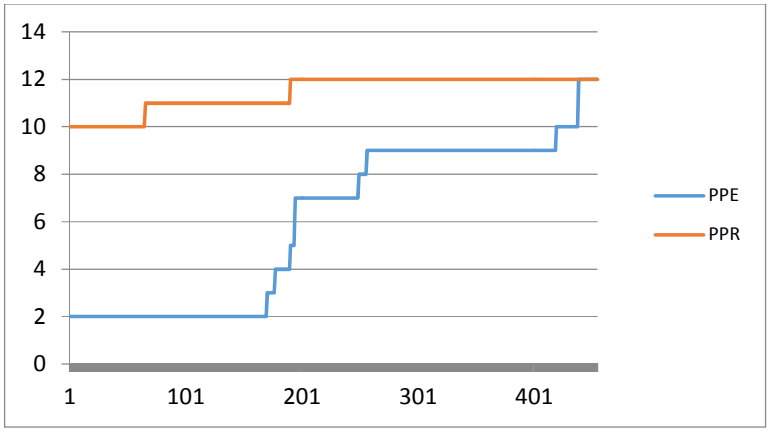


Figura 4.28 Resultado de perdida paquetes del servidor Vline. (Autor de la Tesis 2015)

La pérdida de paquetes es menor al 0,5%, Tabla 4.13, por lo que se puede deducir que los paquetes en su mayoría son entregados al destino y que la pérdida está dentro del rango bueno, con este parámetro se puede establecer que a pesar de que el RTT es alto y que podría causar problemas en la comunicación debido a que los paquetes no se entregan rápidamente, la comunicación en general está dentro del rango aceptable debido a que los valores de la pérdida de paquetes y Jitter es bueno.

Vline	Paquetes	Porcentaje
P.Enviados	16700	100%
PPE	12	0,07%
P.Recibidos	15415	100%
PPR	12	0,08%

Tabla 4.13 Porcentaje pérdida de paquetes del servidor Vline. (Autor de la Tesis 2015)

4.3.6 Resultados con el Servidor Talky

El tercer escenario corresponde a el servidor Talky, utilizando el navegador Google Chrome, en el que se ha obtenido los siguientes resultados mediante el API “chrome:webrtc-internals”.

- **Captura Wireshark**

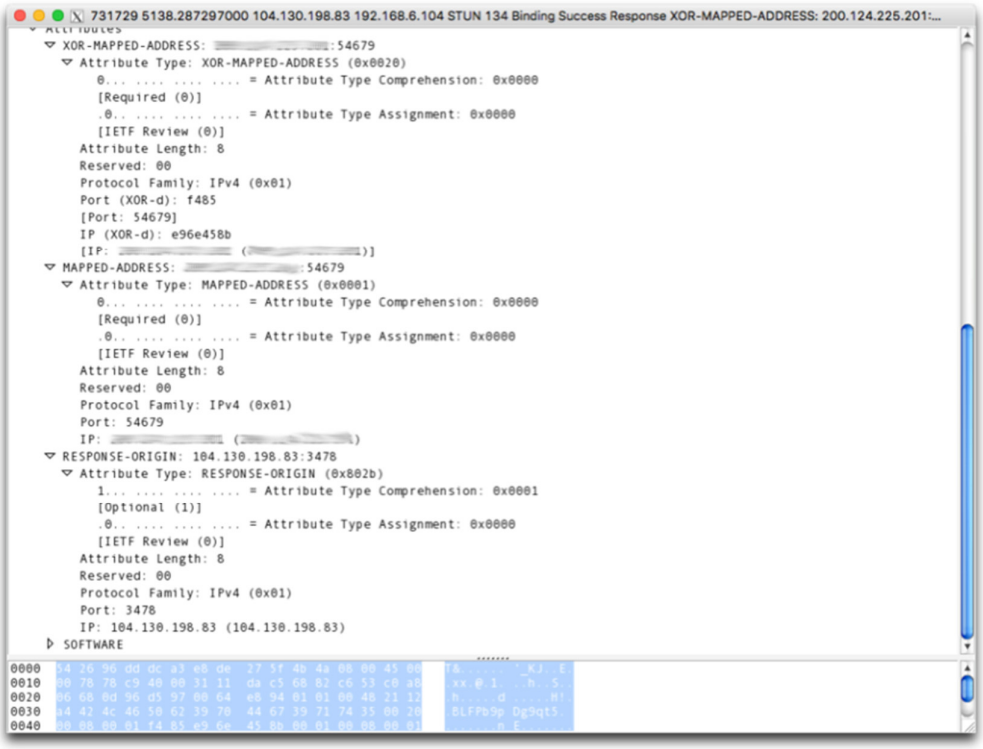


Figura 4.29 Captura paquete del servidor Talky mediante Wireshark. (Autor de la Tesis 2015)

La figura 4.29 refleja la captura de Wireshark, se muestra las direcciones IPs de Origen 201.125.215.211, la que se encuentra oculta debido a la confidencialidad del cliente emisor, y la de destino 104.130.198.83, correspondiente al servidor Talky; además se puede notar que el paquete en el encabezado contiene la dirección IP privada, la misma que esta mapeada con la dirección IP de Origen del emisor.

Modo Envio		Modo recepción	
Source	2264543506		41693592
Servidor STUN	104.130.195.95		
Servidor Webrtc	Talky	Servidor Webrtc	Talky
Bytes Enviados	1524083	Bytes Recibidos	7376233
Paquetes perdidos	70	Paquetes perdidos	70
Paquetes enviados	15320	Paquetes recibidos	72753
CODEC	OPUS	CODEC	OPUS
RTT(ms)	69	RTT	0
JITTER(ms)	9	JITTER	0
IP LAN Origen	192.168.6.104	IP LAN Origen	192.168.1.7
IP LAN Destino	192.168.1.7	IP LAN Destino	192.168.6.104
IP Servidor	104.130.198.83	IP Servidor	104.130.198.83
IP WAN Origen	201.125.215.211	IP WAN Origen	190.214.207.111

Tabla 4.14 Captura de datos del servidor Talky con webrtc-internals. (Autor de la Tesis 2015)

El Servidor Talky, Tabla 4.14, muestra en la captura de webrtc-internals que utiliza el códec OPUS, un RTT de 69 mseg., que equivale a bueno y un Jitter de 3 mseg., y está dentro del rango bueno.

- ***RTT Talky***

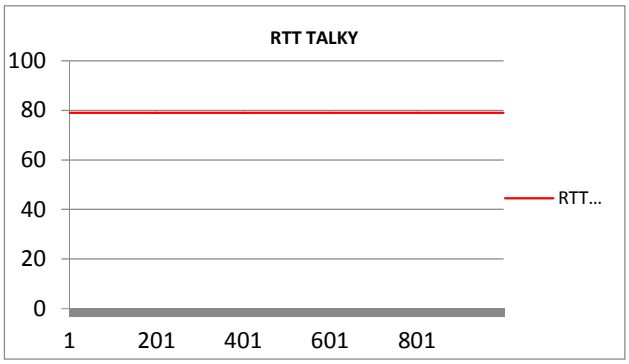


Figura 4.30 Resultado de RTT del servidor Talky. (Autor de la Tesis 2015)

- ***JITTER Talky***

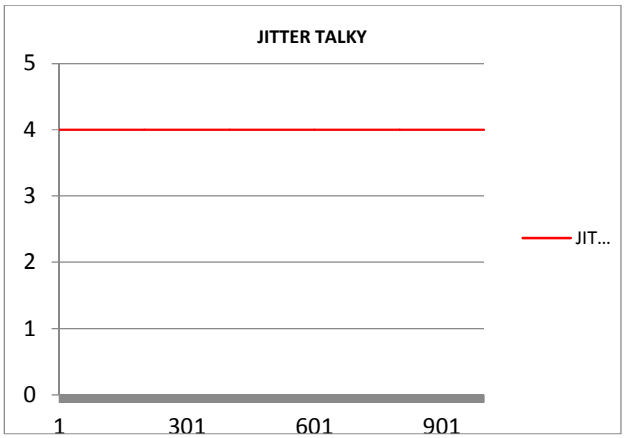


Figura 4.31 Resultado de JITTER del servidor Talky. (Autor de la Tesis 2015)

En las figuras 4.30 y 4.31 se tiene la visualización de resultados donde se observan 997 capturas de datos en milisegundos de RRT y Jitter, además se tiene que estos parámetros se mantienen constantes, el retardo puede mantenerse constante debido a tres razones, una de ellas la codificación es decir al códec de voz utilizado, otro factor es la paquetización que es

el tiempo que se toma en convertir el audio y colocarlos en paquetes IP y el último parámetro es la serialización que es el retardo de colocar los paquetes desde la capa de aplicación hasta el medio de transmisión, con este resultado se puede decir que el retardo es constante debido a estos parámetros y el Jitter también puede mantenerse constante debido al uso de memorias para almacenar temporalmente los paquetes, se muestran los valores en la Tabla 4.15.

Talky	RTT(ms)	Jitter(ms)
Promedio	79,00	4,00
Pico Bajo	79	4
Pico Alto	79	4
Resultado	Bueno	Bueno

Tabla 4.15. Promedio de JITTER y RTT del servidor Talky. (Autor de la Tesis 2015)

- ***Pérdida de Paquetes Talky***

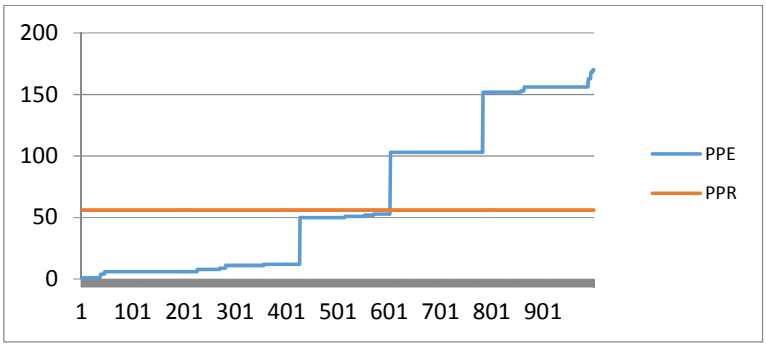


Figura 4.32 Resultado de perdida paquetes del servidor Talky. (Autor de la Tesis 2015)

Los paquetes perdidos enviados con relación a los paquetes enviados, Tabla 4.16, muestran un porcentaje bajo de pérdida que es del 0,3%, el que está en el rango bueno; esta comunicación

mediante el servidor Talky es buena debido a los valores de las variables que se han observado.

Talky	Paquetes	Porcentaje
P.Enviados	56841	100%
PPE	170	0,3%
P.Recibidos	16811	100%
PPR	56	0,33%

Tabla 4.16 Porcentaje pérdida de paquetes del servidor Talky. (Autor de la Tesis 2015)

4.3.7 Resultados con el Servidor 3CX

- **Captura Wireshark**

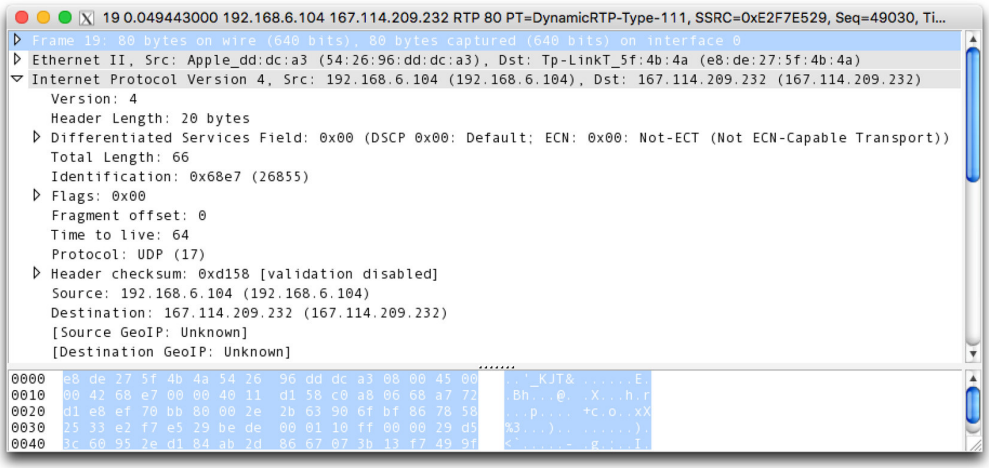


Figura 4.33 Captura de paquete del servidor 3CX mediante Wireshark. (Autor de la Tesis 2015)

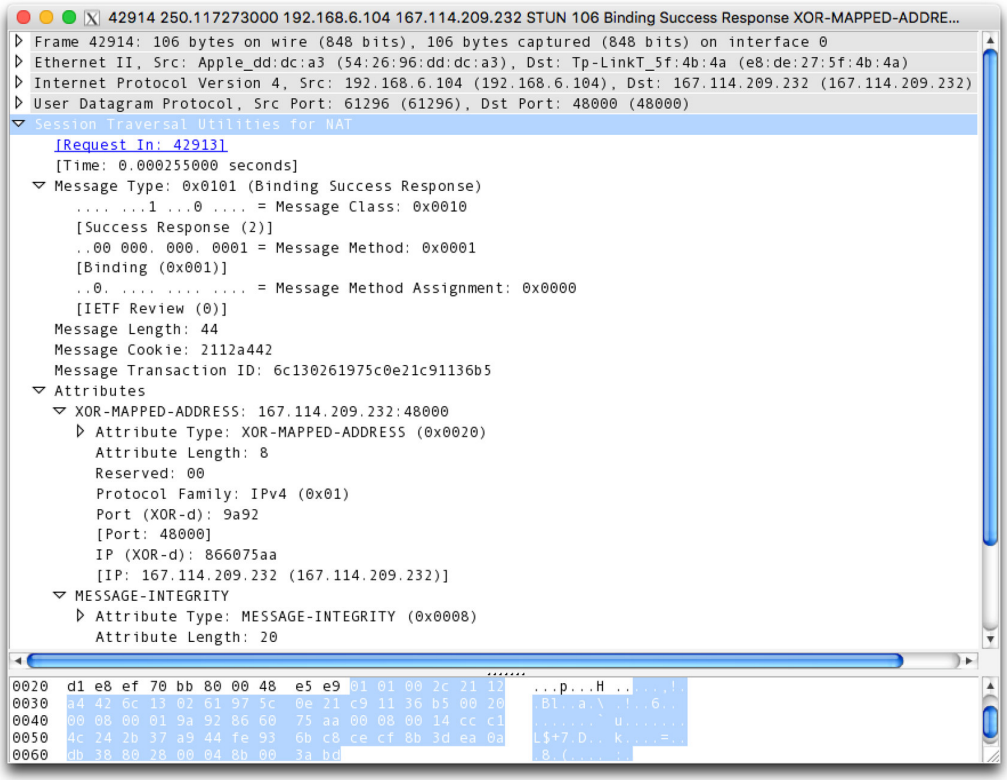


Figura 4.34. Paquete XOR-MAPPED-ADDRESS del servidor 3CX. (Autor de la Tesis 2015)

La figura 4.33 muestra la dirección IP de Origen del servidor 3CX que es 167.114.209.232, además el paquete es de tipo XOR-MAPPED, figura 4.34, el cual es un atributo encargado de responder al cliente por parte del servidor STUN, por esta razón se puede observar también la dirección IP del cliente que es 192.168.6.106, ya que XOR_MAPPED mantiene la dirección IP del cliente intacta a través de NAT.

Modo Envio		Modo recepción	
Source	457869706		1967151048
Servidor STUN	190.214.207.111		
Servidor Webrtc	3CX	Servidor Webrtc	3CX
Bytes Enviados	610436	Bytes Recibidos	739415
Paquetes perdidos	28	Paquetes perdidos	8
Paquetes enviados	13648	Paquetes recibidos	13396
CODEC	OPUS	CODEC	OPUS
RTT (ms)	284	RTT	
JITTER (ms)	4	JITTER	0
IP LAN Origen	192.168.6.104	IP LAN Origen	192.168.1.7
IP LAN Destino	192.168.1.7	IP LAN Destino	192.168.6.104
IP Servidor	167.114.209.232	IP Servidor	167.114.209.232
IP WAN Origen	201.125.215.211	IP WAN Origen	190.214.207.111

Tabla 4.17 Captura de datos del servidor 3CX mediante webrtc-internals. (Autor de la Tesis 2015)

En la Tabla 4.17, se observan los datos del servidor 3CX mediante las capturas de “webrtc-internals”, donde se ve que utiliza para la codificación de voz el códec OPUS e indica que el

código de paquetes de origen es el 457869706 y los de recepción es el 1967151048, la muestra en un momento dado da como resultado un RRT de 284 mseg., correspondiente a un valor del rango aceptable y el Jitter de 4 mseg., en el rango bueno.

- ***RTT 3CX***

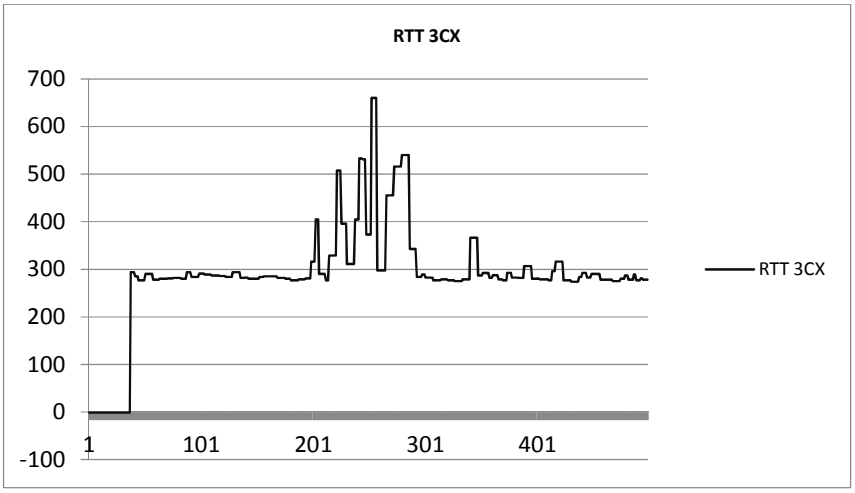


Figura 4.35. Resultado de RTT del servidor 3CX. (Autor de la Tesis 2015)

- ***JITTER 3CX***

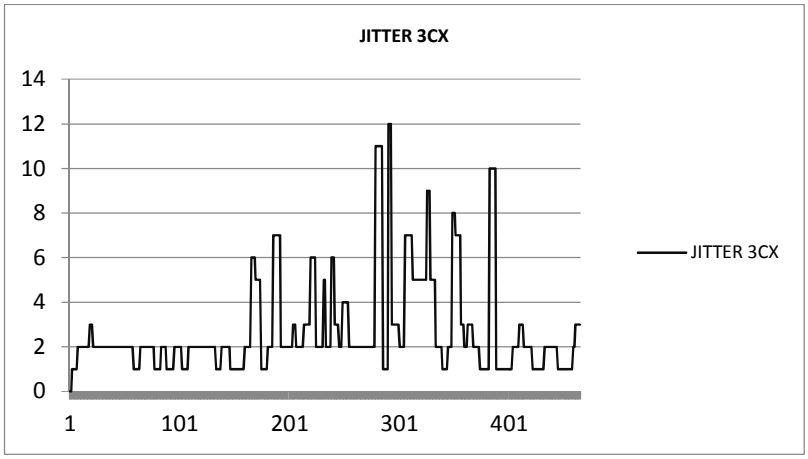


Figura4.36. Resultado de JITTER del servidor 3CX. (Autor de la Tesis 2015)

Los resultados en las figuras 4.35 y 4.36 de RTT y Jitter respectivamente, muestran que dentro de 477 muestras que corresponden al eje X los parámetros van variando en milisegundos, están en un rango bueno, mientras se mantengan los rangos buenos y aceptables, la comunicación tendrá éxito. Los valores promedios y picos se observan en la Tabla 4.18.

3CX	RTT(ms)	Jitter(ms)
Promedio	285,06	2,72
Pico Bajo	0	0
Pico Alto	757	12
Resultado	Aceptable	Bueno

Tabla 4.18. Promedio de JITTER y RTT del servidor 3CX. (Autor de la Tesis 2015)

- ***Pérdida de Paquetes 3CX***

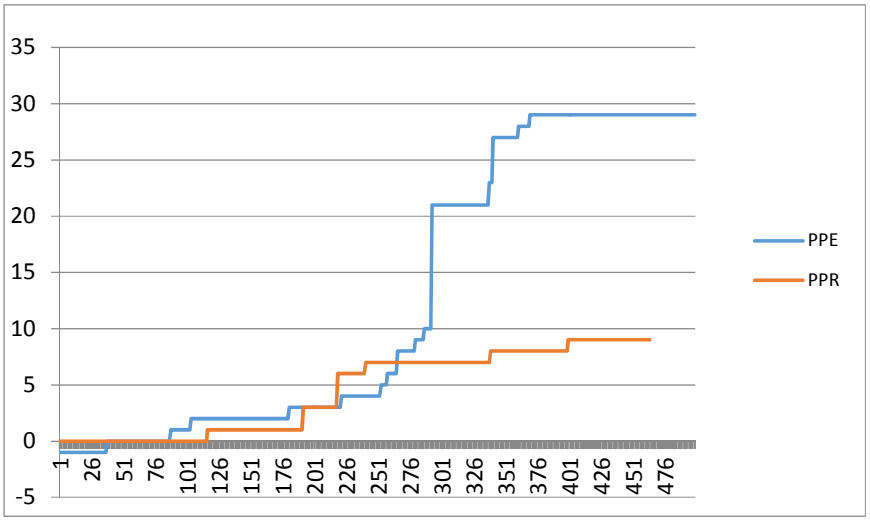


Figura 4.37 Resultado de perdida paquetes del servidor 3CX. (Autor de la Tesis 2015)

Los paquetes perdidos, Tabla 4.19, muestran que los porcentajes de pérdidas de paquetes están en el rango bueno y que la pérdida es mínima, anteriormente se mostraron los valores de Jitter y RTT los cuales estaban en un rango bueno, por esta razón se concluye que la comunicación es buena.

3CX	Paquetes	Porcentaje
P.Enviados	26102	100%
PPE	29	0,11%
P.Recibidos	23590	100%
PPR	9	0,04%

Tabla 4.19 Porcentaje pérdida de paquetes del servidor 3CX. (Autor de la Tesis 2015)

4.4 Análisis de Resultados

4.4.1 RTT

Definido como el tiempo que se tarda en llegar a su destino un paquete de datos que se enviará en la red, más tiempo que el paquete de confirmación tarda en retornar al origen, ese valor como parte del protocolo RTP se calcula, ya que se toma de referencia el paquete RTCP desde él envió hasta la llegada de la confirmación al origen, lo cual está establecido para la comunicación WebRTC.[42]

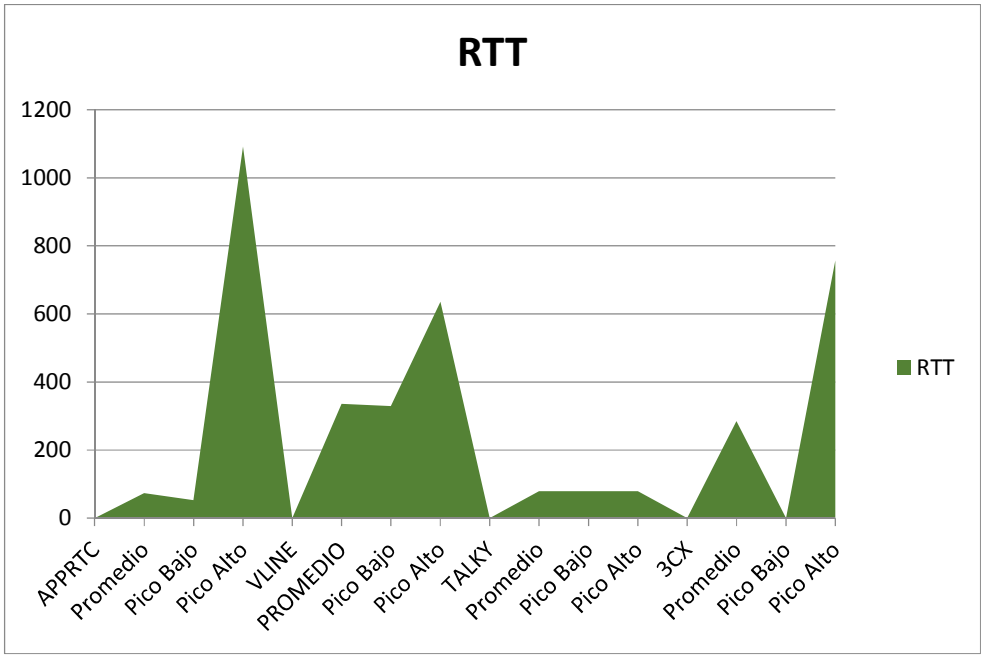


Figura 4.38. Comparativa de resultados de RTT de los servidores en la Nube. (Autor de la Tesis 2015)

El retardo o latencia es un parámetro que indica la suma de procesamientos que es sometido el paquete desde que ha sido enviado hasta que llega a su destino, el análisis con respecto las pruebas realizadas muestra que el servidor gratuito Talky ha obtenido el mejor valor de RTT, es decir que el procesamiento de los paquetes es más eficiente; generalmente cuando se obtiene valores de RTT constante se debe a que existe un buffering de Jitter el que iguala los retardos y no permite que haya variación, el buffering puede provocar que el retardo sea más alto como se muestra en la figura 4.38, debido a la espera que se produce al realizar esta acción el paquete tardará más tiempo en llegar al destino.

4.4.2 Jitter

Definido como la diferencia de tiempo entre paquetes transmitidos por la red, a la llegada de la aplicación que desea utilizar y el momento en que se debe reproducir; para el caso específico de VoIP, un buffer de Jitter se utiliza para ordenar los paquetes a la llegada de los mismos y reproducirlos en el orden correcto en el momento oportuno.

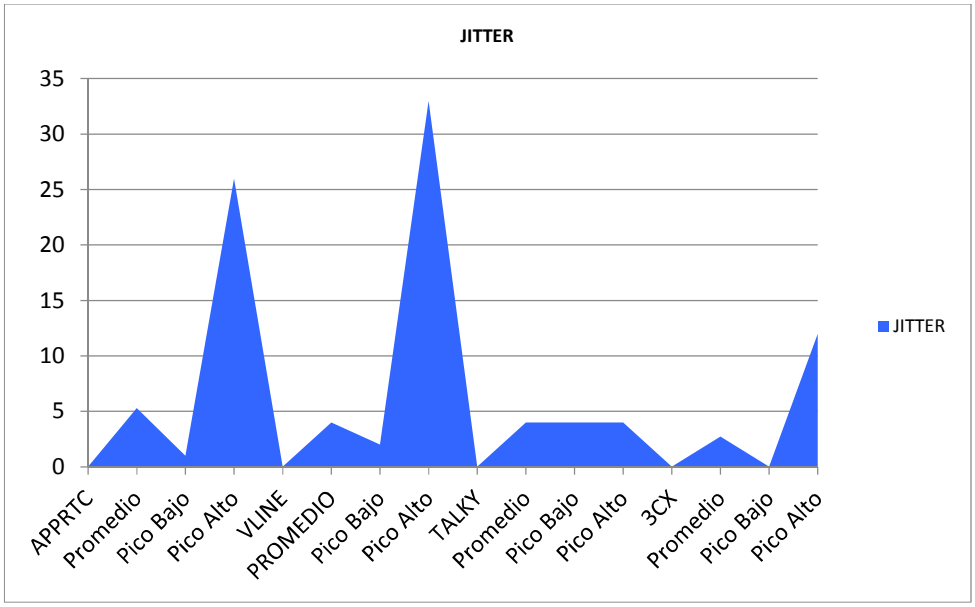


Figura 4.39 Comparativa resultados de JITTER de los servidores en la Nube. (Autor de la Tesis 2015)

En la figura 4.39 se observan los valores de JITTER, se muestra una comparación de resultados de este parámetro, en el que se aprecia que el servidor con el que se tiene una mayor estabilidad al momento de la comunicación es el servidor gratuito en la Nube Talky; el Jitter constante es derivado del buffering de Jitter, pero esto no garantiza que la comunicación sea mejor debido que el retardo será más alto y la pérdida de paquetes también podría aumentar, en el caso de análisis de resultados presentados en este trabajo el servidor gratuito Talky tiene el porcentaje más alto de pérdida de paquetes y justamente es debido a el buffering de Jitter.

4.4.3 Pérdida de Paquetes

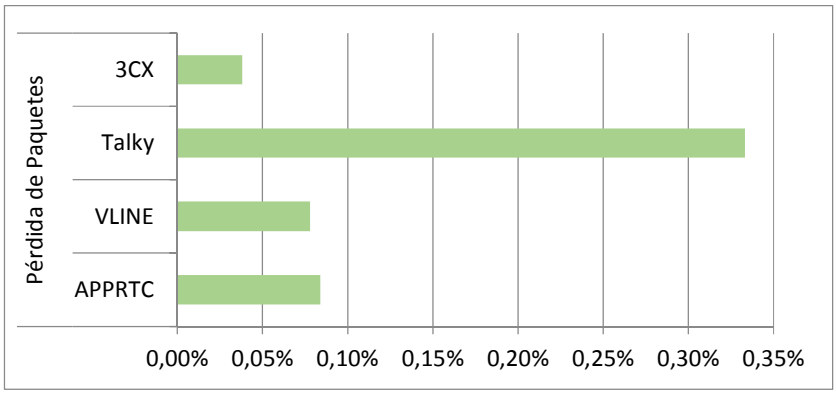


Figura 4.40 Comparativa de resultados de PP de los servidores en la Nube. (Autor de la Tesis 2015)

En la figura 4.40 se presenta los porcentajes correspondientes a la pérdida de paquetes, que está ligada con el retardo de paquetes por lo tanto con el Jitter, debido a que la voz en comunicaciones de tiempo real se transmiten mediante datagramas UDP, conociendo que el protocolo UDP no está orientada a conexión ni a comprobación de errores, existe mayor probabilidad de que los paquetes se pierdan debido a los retardos; una de las ventajas de la voz es que es bastante predictiva para el ser humano por lo que si existen pérdidas esporádicas el mensaje aún es entendible, el problema real es cuando se pierden ráfagas completas por esta razón el 1% de pérdida es un porcentaje aceptable del total de paquetes enviados.

El resultado muestra lo que los valores de Jitter y RTT han acarreado, es decir, en el servidor Talky la pérdida ha aumentado debido al buffering de Jitter, siendo este valor más alto que el de los otros servidores, pero no superior al límite permitido.

Los resultados en las gráficas corroboran la experiencia en la práctica, debido a que mientras se hacía la toma de datos se efectuaba una conversación normal en diferentes momentos, se pudo percibir la estabilidad en la comunicación de los servidores de WebRTC; si bien es cierto

se probó con Chrome, Opera y Firefox, los Proveedores del Servicio WebRTC recomiendan utilizar Google Chrome, argumentando que se podría tener un retardo más alto cuando se usa otro navegador; a pesar de estas recomendaciones, la comunicación para los 4 servidores fue exitosa, fluida y con pocas interferencias, los resultados no muestran de manera definitiva que servidor es mejor, solamente indican resultados en un lapso de tiempo determinado y con condiciones específicas del enlace de la red, los resultados pueden variar dependiendo del ancho de banda disponible, la congestión de la red y la capacidad de los Tarjetas de Red en el procesamiento y la transmisión de datos, es seguro que si se repiten las pruebas los resultados serán similares pero no exactamente iguales, por lo tanto estos resultados son solamente una referencia del momento que se realizó las llamadas.

4.5 Propuesta de Red para VoIP mediante WEBRTC

Mediante los resultados obtenidos en las pruebas realizadas en los apartados anteriores se realiza un análisis de una red empresarial típica, esto es una PYMES de 30 usuarios que tendrá dos oficinas ubicadas en las ciudades de Quito y Portoviejo; y se comunicaran mediante la aplicación VoIP por medio de la tecnología WebRTC, cuyo servicio está alojado en la Nube.

4.5.1 Red para VoIP

Para la estructura de la red se deben de considerar los siguientes aspectos:

- **Modelo de Navegador**

En cada PC de la LAN se tiene el modelo de Navegador y que ejecuta funciones para las aplicaciones en VoIP (tiempo real), como se muestra en la figura 4.41

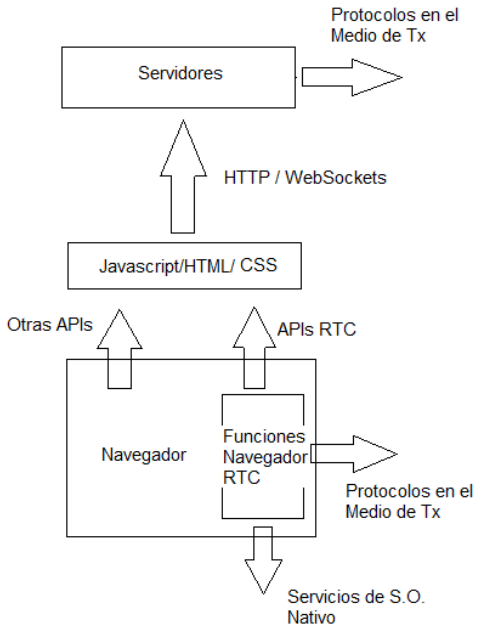


Figura 4.41 Modelo de Navegador (Autor de la Tesis 2015)

- **Pila de Protocolos Cliente en WebRTC**

En la figura 4.42 se muestra la pila de protocolos típica en el cliente que intervendrán en una llamada WebRTC

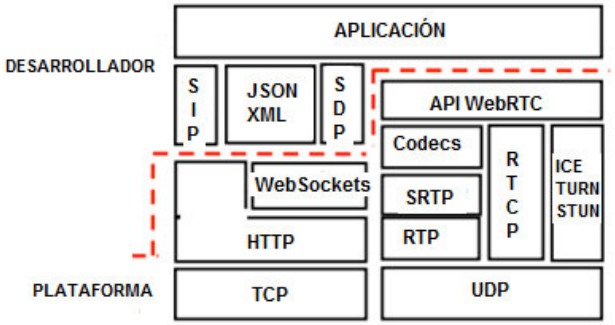


Figura 4.42 Pila de protocolos en el cliente WebRTC (Autor de la Tesis 2015)

- **Topología WebRTC en la nube**

La topología a través de la Nube que se plantea en la propuesta, cuando se realiza una llamada WebRTC entre dos clientes (PCs) desde las LANs de Quito y Portoviejo respectivamente en la PYMES, se muestra en la figura 4.43.

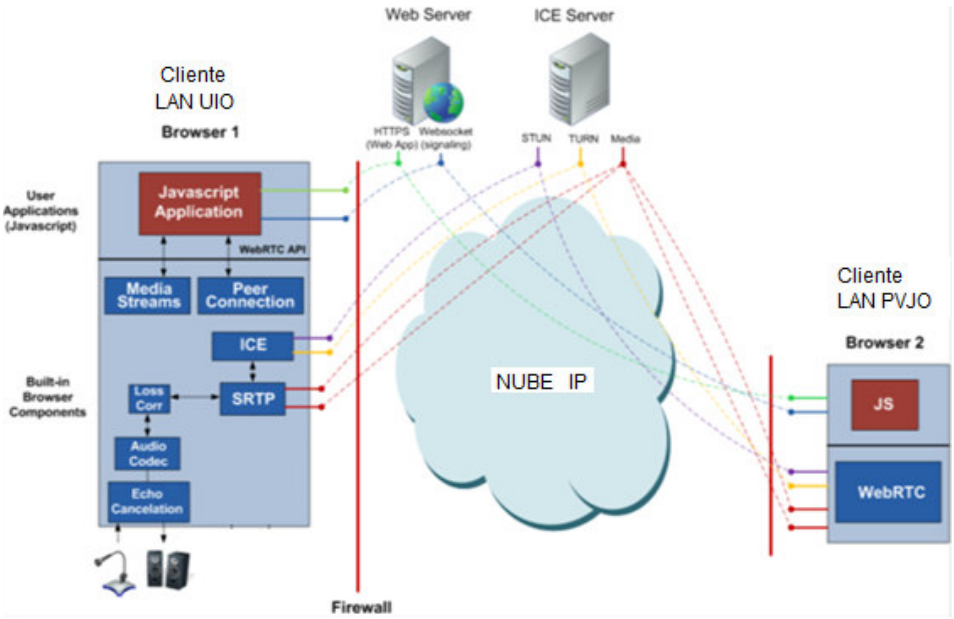


Figura 4.43 Topología de la red propuesta con servidor WebRTC en la Nube (Autor de la Tesis 2015)

4.5.2 Ancho de Banda y Códec para VoIP

El servicio ofrecido por el Servidor de WebRTC en la Nube depende en la mayoría de los casos en el número de participantes es decir, que el costo varía de acuerdo a cuantos participantes conectados al mismo tiempo se requieran, para el caso de la PYMES planteado se tiene un máximo de 30 participantes. En base al número de participantes los recursos de la red van a variar, especialmente el ancho de banda, el cual debe permitir la comunicación de 30 usuarios comunicados al mismo tiempo.

El servicio de WebRTC en la nube permite además de comunicarnos mediante la tecnología WEBRTC mediante llamadas de voz también el servicio de video llamadas, para nuestro caso únicamente tomaremos en cuenta las llamadas de voz; el utilizar el servicio de WebRTC en la nube deriva de la necesidad de un ancho de banda que permita sostener las 30 llamadas de voz al mismo tiempo.

Cuando el sistema está a su máxima capacidad se producirán un total de 15 conexiones es decir que habrá un máximo de 15 llamadas de voz el mínimo ancho de banda de una llamada de voz es de 64kbps, para este caso el ancho de banda mínimo será de 960 Kbps de subida y de bajada en los ISP de Quito y Portoviejo.

Para el servidor WebRTC que se propone, como se explicara más adelante, se sugiere cumplir con el número máximo de participantes del servidor 3CX para que estén cubierto los 50 participantes, es decir el ancho de banda mínimo de la topología será de 1920 Kbps . El número de canales en una llamada son 2, es decir para los datos de envío y recepción, en el caso de la conexión de 50 participantes tendremos que el número máximo de llamadas simultáneas es de 25 por lo que necesitaremos 50 canales.

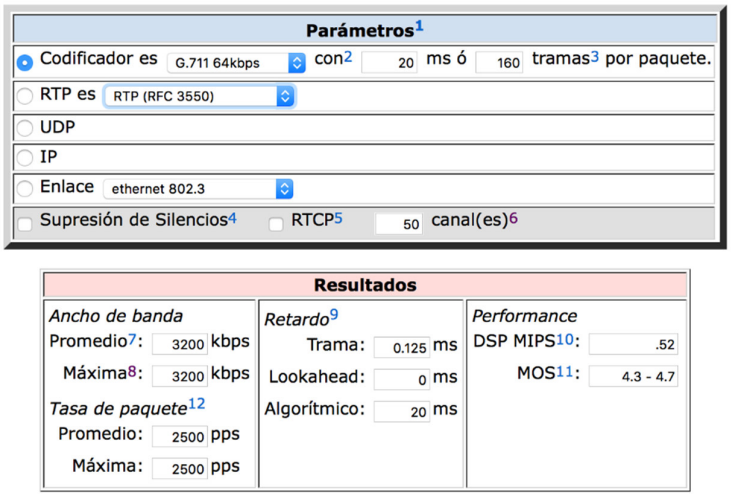


Figura 4.44 Cálculo de ancho de banda para 50 canales [43]

En la figura 4.44 se ha hecho uso de una calculadora online para conocer el ancho de banda para la comunicación de VoIP mediante WebRTC de acuerdo a los requerimientos planteados. El códec utilizado en WEBRTC es el G.711 y OPUS, la calculadora de VOIP muestra que el ancho de banda es de 3200 Kbps, este ancho de banda solo es para la comunicación mediante voz, más no se ha tomado en cuenta los requerimientos de video, por estar fuera del alcance de este estudio.

4.5.3 Servidor 3CX – WebRTC en la Nube

En la propuesta para el caso de la PYMES se ha elegido como servidor WebRTC al servidor 3CX en la Nube debido a que es robusto en cuanto a confiabilidad del servicio, con respecto a los otros servicios analizados: Asterisk, Talky, Vline y AppRTC.

El servicio 3CX en la Nube permite configurar de manera sencilla los usuarios y sobre todo el manejo de las comunicaciones, para esto solo se debe alquilar el servicio, de esta manera se garantiza que la mayor parte del tiempo se va a tener el servicio a disposición y se evita la instalación de una infraestructura completa que permita dar el servicio a los usuarios de las LANs, es decir, a pesar de que se puede instalar localmente el servidor de manera muy rápida y sencilla, los requerimientos de recursos de red y de seguridad son costosos, por lo que se pueden solventar esta necesidad alquilando el servicio.

El servicio ofrecido por 3CX en la nube, está basado en el número de participantes es decir, que el costo varía de acuerdo al número de participantes conectados al mismo tiempo que se requieran; para el caso del servicio de 3CX el número de participantes ofrecido es de 10, 25, 50, 100 y un máximo de 250. Para el caso de la PYMES el requerimiento es un máximo de 30 participantes, por lo que se ha elegido el servicio de 3CX por 50 participantes. Con este número de participantes se ha realizado en el apartado anterior el cálculo del requerimiento del ancho de banda.

Se debe tomar en consideración que el servicio de 3CX en la nube permite por medio de la tecnología WEBRTC, la comunicación mediante llamadas de voz así como también el servicio de video llamadas, para nuestro caso únicamente tomaremos en cuenta las llamadas de voz.

El servicio 3CX en la Nube es una central telefónica con licencia para Windows 7 y versiones superiores orientado a redes telefónicas VOIP, soporta dentro de sus características a WEBRTC de manera gratuita, ya que ofrece además de ser una central telefónica también otros servicios embebidos, como WebMeeting; la central telefónica permite la configuración de manera limitada de extensiones a menos que se active la licencia.

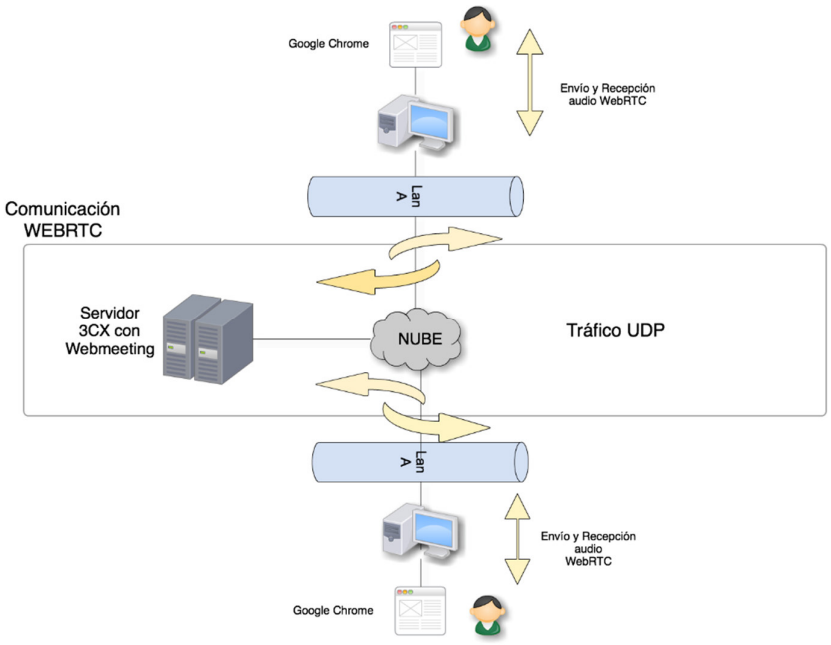


Figura 4.45 Modelo de funcionamiento de servidor 3CX (Autor de la Tesis 2015)

La comunicación se despliega desde el Navegador Web ubicado en los equipos terminales correspondientes a los PCs de las LANs, el mensaje es captado en la capa de aplicación para luego a nivel de capa de transporte los bits de VoiP son encapsulados en datagramas UDP, para ser enviados al receptor; como se muestra en la figura 4.45.

El servicio 3CX fue creado específicamente para un ambiente de Windows y esta soportada desde Windows 7 32bits y 64bits en adelante, al ser instalado se puede manejar de manera sencilla mediante su interfaz web; a través de la interfaz se puede configurar todos los parámetros de la central telefónica de manera amigable e intuitiva, soporta NAT y conectividad con redes telefónicas públicas PSTN y es compatible con IP PBX de Asterisk.

Los requerimientos de hardware mínimos son:

Equipo	Característica
Memoria	1GB
CPU	Intel core i3
Servidor web	Abyss
Disco	SATA 30GB
Red	100/1000/Mbit/s

Tabla 19. Requerimientos de hardware para instalación de 3CX. [44]

Debido a la versatilidad del servidor 3CX, permite realizar llamadas VoIP en un entorno de red local o en la nube, además ofrece el servicio de llamadas VoIP mediante WebRTC, esta característica del servidor 3CX, permite que los usuarios o participantes de la LAN o de la Nube puedan hacer llamadas con mayor facilidad sin necesidad de usar software adicional, solamente con el navegador compatible para realizar la llamada, debido a la carencia de recursos en cuanto a infraestructura de la LAN propuesta se puede utilizar un servicio que 3CX ofrece en la nube, cuya característica es la utilización de la infraestructura como servicio (IASS) el que solventa los requerimientos de instalar el servidor en la LAN

El servicio 3CX ofrece varios productos para comunicaciones entre ellos:

- 3CX Phone System, alojada en la LAN con la posibilidad de habilitar WebMeeting.
- Teléfonos SIP para llamadas VoIP.
- 3CX WebMeeting, para video conferencia y llamadas VOIP mediante WebRTC en la Nube.

La tecnología WebRTC está disponible en la central 3CX Phone System que es una central PBX para llamadas de VoIP y permite la activación de WebMeeting para llamadas WebRTC; o también está presente como un módulo individual en WebMeeting, mediante este módulo se puede especificar de manera sencilla los parámetros y características que se desean utilizar, entre ellas:

- Compartir pantallas
- Control remoto del servidor en la Nube
- Asistencia remota
- Usuarios ilimitados
- Grabación de las conversaciones en audio y video
- Conferencias de voz y video en un solo clic

WebMeeting es una herramienta de pago que permite usarlo en la nube mediante una subscripción anual o perpetúa, a continuación se detalla la opción de WebMeeting en la nube, que se ha tomado de la página oficial de 3CX.

3CX WebMeeting

Launch web meetings from a web portal.

	Product Code	Euro	USD	Buy
3CX WebMeeting 10 Participants	3CXWMS10	475	495	Via Partner
3CX WebMeeting 25 Participants	3CXWMS25	895	950	Via Partner
3CX WebMeeting 50 Participants	3CXWMS50	1,495	1,595	Via Partner
3CX WebMeeting 100 Participants	3CXWMS100	2,395	2,495	Via Partner

3CX WebMeeting for 3CX Phone System

Launch web meetings from your 3CXPhone (requires 3CX Phone System).

	Product Code	Euro	USD	Buy
3CX WebMeeting 25 Participants	3CXWM25	495	595	Via Partner
3CX WebMeeting 50 Participants	3CXWM50	975	1,095	Via Partner
3CX WebMeeting 100 Participants	3CXWM100	1,975	2,195	Via Partner

3CX WebMeeting Server

Host meetings on-premise. Pricing is for a perpetual license and includes one year of maintenance.

3CX WebMeeting On-Premise	Product Code	Euro	USD	Buy
3CX WebMeeting 25 Participants	3CXWMOP25	1,995	2,275	Via Partner
3CX WebMeeting 50 Participants	3CXWMOP50	3,500	3,995	Via Partner
3CX WebMeeting 100 Participants	3CXWMOP100	5,995	6,750	Via Partner
3CX WebMeeting 250 Participants	3CXWMOP250	9,995	11,500	Via Partner

Figura 4.46 Precios de WebMeeting alojado en la nube mediante pago por el servicio perpetuo [45]

En la figura 4.46, se describe una de las opciones que ofrece 3CX al momento de requerir el servicio de realizar llamadas de VoIP mediante WebRTC.

Para la propuesta de comunicación de VoIP mediante WebRTC de la PYMES, en lo referente al servicio WebRTC en la Nube se plantea la siguiente opción:

- ***3CX WebMeeting (IAAS - SAAS)***

Esta versión corresponde a la utilización de software como servicio, es decir, que WebMeeting está ubicado en la nube y el cliente o participante que accede al servicio mediante una interface Web, donde puede darse de alta en el servidor y administrar los parámetros y configuraciones

del servidor, además este no requiere la instalación, solamente que los participantes utilicen el navegador Google Chrome, el costo varía según el número de participantes.

Debido a que este servicio se encuentra en la nube podemos identificar dos tipos de servicios en la Nube: IAAS y SAAS, esto se debe a que el servicio de 3CX al cliente final o participante es tipo SAAS, es decir, el participante usa el software como servicio y simplemente se accede de manera rápida a través de un pago, mientras que al realizar las pruebas con el servidor 3CX y buscar donde este se halla alojado mediante la dirección IPv4 se obtuvo lo siguiente:

Figura 4.47. Resultado de la herramienta WHOIS para obtener datos de IP Pública [46]

En la figura 4.47 se muestra que el servidor 3CX se encuentra alojado en Canadá y que está ligado a una organización llamada OVH Hosting, la misma que ofrece servicios de alojamiento de máquinas virtuales y servidores, es decir que 3CX mediante un pago usa la infraestructura como servicio de un proveedor, por lo que este tipo de Servicio en la Nube es de tipo IAAS. De las pruebas realizadas se pudo constatar que los servicios de 3CX se alojan

en otras locaciones y utiliza otros proveedores de este servicio de infraestructura como por ejemplo Amazon.

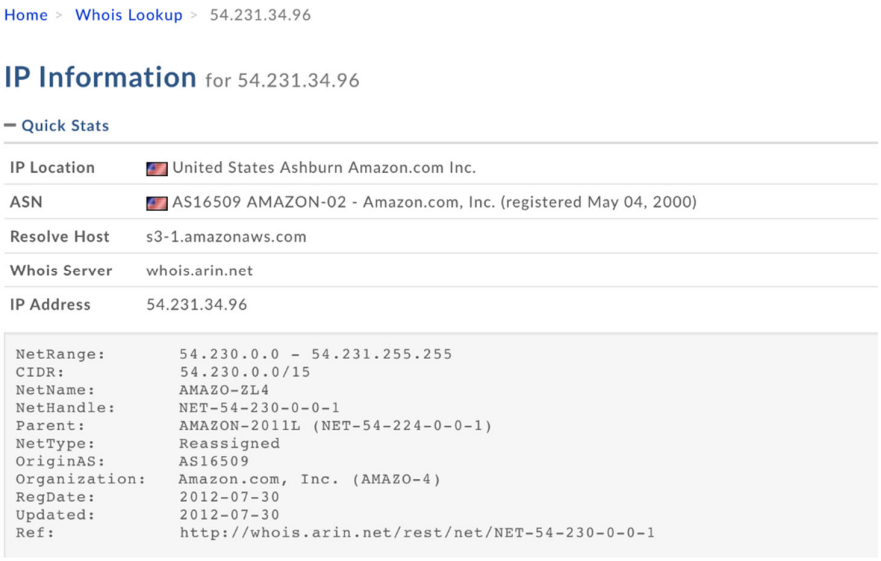


Figura 4.48. Resultado de la herramienta WHOIS para obtener datos de IP Pública [46]

En esta figura 4.46 se observa que el servidor se encuentra en EEUU y que el proveedor IAAS es Amazon.com. Las direcciones IPs públicas han sido obtenidas en el momento de realizar las pruebas mediante la herramienta Whireshark y webrtc-internals.

4.5.4 Topología de Red y Tabla de Direccionamiento

En la figura 4.49 se muestran la topología de red para la comunicación de VoIP por medio de la tecnología WebRTC en una PYMES de 30 usuarios que tendrá dos oficinas ubicadas en las ciudades de Quito y Portoviejo.

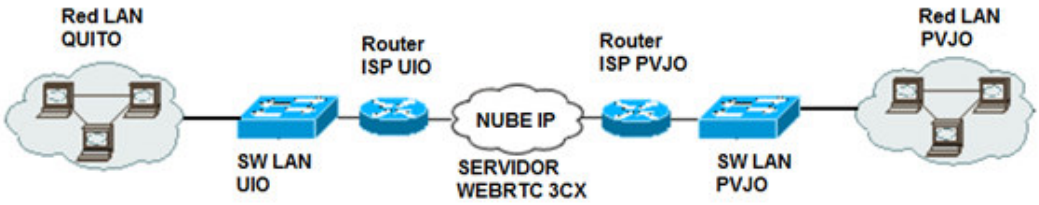


Figura 4.49 Topología de Red – PYMES 30 usuarios (Autor de la Tesis 2015)

El direccionamiento mediante direcciones IPv4 se muestra en la tabla 4.20.

Red	**IP**	**Router**	**Participantes**	**Navegador**
LAN	192.168.6.0 / 24	Quito	15	Google Chrome
LAN	192.168.7.0 /24	Portoviejo	15	Google Chrome
WAN (ISP)	201.125.215.211	Quito		
WAN (ISP)	190.214.207.111	Portoviejo		
Servicio 3CX en la Nube (Servidor Web y Señalización)	167.114.209.232	Canadá	50	Chrome Opera Firefox

Tabla 4.20 Direccionamiento IPv4 y servidor 3CX. (Autor de la Tesis 2015)

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

- Es viable la implementación de servicios de VoIP en la Nube, tanto con servidores gratuitos como con licencia mediante la tecnología de WebRTC, ya que los valores obtenidos en las pruebas realizadas están dentro de los rangos estandarizados en los parámetros de Jitter, RTT y pérdidas de paquetes, esto es Calidad de Servicio (QoS).
- La mejor práctica para el uso de la tecnología WebRTC es el uso de los servidores probados que ofrezcan soporte y estabilidad, es decir, el usar un servicio alojado en la Nube exime de la administración de servicios, servidores, puertos y firewalls; desde el punto de vista empresarial (PYMES) se debe usar un servidor de pago ya que provee más estabilidad que los gratuitos que se encuentran en la Nube.
- Al momento el uso de servidores que se deban de configurar e instalar en una LAN ya sea con GPL o Licencia, demanda de costos elevados en la compra de equipos que sean compatibles con el servicio y en el talento humano que pueda administrar el servidor, si bien los servidores con licencia son muy estables, esto exige que la red y seguridades de la misma sean robustas; al mismo tiempo que dependan de un administrador de red que pueda tener siempre el servicio activo y se encargue de solucionar problemas causados por tráfico, seguridades y los usuarios, en este caso la administración en sí se vuelve costosa ya que si el número de usuarios del servicio crece a futuro también deberá crecer la capacidad de la LAN y por tanto demandará también de costos altos, por lo cual es mejor utilizar el servicio en la Nube en el que ya se encuentra todo configurado y solo dependerá del ancho de banda con el que acceda a Internet y un moderador que cree cuentas para los usuarios.
- La comunicación de voz mediante WebRTC fue totalmente estable y de alta calidad, en las pruebas realizadas se pudo constatar que todos los servidores ofrecen una buena calidad en el servicio y que a pesar de usar distintos Codecs de audio la comunicación siempre en todos los casos fue entre buena y aceptable.

- Si se instala un servidor GPL o con licencia en una red local es importante contar con los equipos adecuados, como firewalls con políticas de acceso muy seguras debido a que al abrir puertos específicos los hackers se basan en estos puertos para el uso de sus ataques, el primer problema que se tiene al abrir los puertos es el ataque constante hacia el servidor desde los puertos abiertos, para solucionar esto es importante contar con un firewall en el router y en el sistema operativo, una de las mejores prácticas es bloquear IPs externas e internas después de tres intentos fallidos.
- Las características que se requiere al momento de comunicaciones de VoIP son que la disponibilidad del servicio sea siempre el 100% equivalente a 24/7 y que las comunicaciones tengan la mayor parte del tiempo buena calidad es decir que el Jitter y la latencia no sean factores que se eleven al momento de alto tráfico, en cuanto a la disponibilidad los servidores gratuitos alojados en la Nube no sucede esto, puesto que no siempre están disponibles para su uso y tampoco muestran la mejor QoS.
- Los resultados tanto en la práctica como con la herramienta "webrtc-internals", en los valores obtenidos se muestra que están dentro de los rangos permitidos de una comunicación estable o como se define en la tabla de valores de referencia de Cisco (tomadas para este estudio) la comunicación en la mayoría del tiempo está entre buena y aceptable, no se ha obtenido un RTT o un Jitter dentro de los rangos pobres por eso la comunicación mediante WebRTC es totalmente garantizada.

RECOMENDACIONES

- Para la Implementación en la LAN de 3CX y exponer en la Nube al servidor se recomienda tomar todas las medidas de seguridad del caso, debido a que cuando se expone al Internet el servidor WebRTC, la apertura de puertos generan riesgos y posibles ataques de hacker como se lo experimento en el desarrollo de este trabajo académico, por lo tanto para futuros trabajos se debe investigar sobre las seguridades que son indispensables para garantizar la efectividad del servicio.
- Se recomienda para futuros trabajos en lo referente a WebRTC se analicen los protocolos de señalización los cuales no están definidos en un estándar para esta tecnología.
- Otro tema que se podría analizar e investigar es seguridad en WebRTC referente al tráfico de los paquetes de voz y a la vulnerabilidad que filtra las direcciones IP reales de los usuarios.
- Se debería implementar el servicio de WebRTC para las empresas en sus propias páginas web, permitiendo a los clientes conectarse al servicio, de esta manera se podría tener una atención al cliente más personalizada y fluida con relación a el soporte que generalmente se ofrece mediante una sala de Chat.
- Se deberían realizar estudios sobre nuevos protocolos en la arquitectura WebRTC como por ejemplo el protocolo QUIC.

BIBLIOGRAFIA

[**1**] Rubiano. J., Mena. A., Hernández J. WebRTC - Una nueva tecnología web al servicio de la educación. Caso en VirtualNet 2.0. Recuperado el 01 de agosto 2014 de http://repositorio.redclara.net/handle/10786/623.

[**2**] Portal de WebRTC. ORG [En línea]. Recuperado el 05 de agosto de 2014 de http://www.webrtc.org/.

[**3**] Garcia. A., Fernandez. I., Desarrollo de un cliente de vídeo bajo demanda para redes con plano de control SIP/IMS. Recuperado el 06 de agosto de 2014 de http://e-archivo.uc3m.es/handle/10016/18684

[**4**] Narayana, R. y Mahadevan, G.: Event driven architecture using html5 web sockets for wireless sensor networks. Recuperado el 05 de agosto de 2014 de http://www.techrepublic.com/resource-library/whitepapers/event-driven-architecture-using-html5-web-sockets-for-wireless-sensor-networks/.

[**5**] Portal de WebRTC.ORG. WebRTC Code and API, General Overview. Recuperado el 22 de agosto de 2014 de http://www.webrtc.org/reference/architecture

[**6**] Rojano E., ¿Sustituirá WebRTC al protocolo SIP en VoIP?. Recuperado el 19 de agosto de 2014 de https://www.sinologic.net/blog/2014-07/sustituira-webrtc-al-protocolo-sip-en-VoIP.html#.VChA_fl5P_w

[**7**] Baz. I., Millan. J., Video conferencia y presentación en VOIP2DAY 2013, Tema WebRTC: ¿aún sigues pensando en webphones?. Recuperado el 12 de agosto de 2014 de http://www.VoIP2day.com/2013/es/conferencias-webrtc

[**8**] Loreto. S., Romano S. P. (2014), Real-Time Communication with WebRTC: Peer-to-Peer in the Browser (pp. 1-10), O'Reilly Media, Inc. ,

[**9**] Millán. J., WebRTC (Web Real-Time Communications) Publicado en BIT nº 197, COIT & AEIT,2014. Recuperado el 28 de agosto de 2014 de http://www.ramonmillan.com/tutoriales/webrtcbeneficios.php

[**10**] Johnston. A, Burnett. D. (2014), WebRTC: APIs and RTCWEB Protocols of the HTML5 Real-Time Web, Third Edition, (pp. 1-6),

[**11**] JOYANES, Luis (2012). Revista del Instituto Español de Estudios Estratégicos, COMPUTACIÓN EN LA NUBE, tomado de http://revista.ieee.es/index.php/ieee/article/viewFile/10/8]; último acceso: 18 de diciembre de 2014

[**12**] MELL Peter – GRANCE [2011], The NIST Definition of Cloud Computing, National Institute of Standards and Technology tomado de http://faculty.winthrop.edu/domanm/csci411/Handouts/NIST.pdf, último acceso: 18 de diciembre de 2014

[**13**] ONTSI [2012], Cloud Computing, retos y oportunidades, El Estudio Cloud Computing. Retos y Oportunidades ha sido elaborado por los siguientes componentes del equipo de Estudios del ONTSI, tomado de http://www.ontsi.red.es/ontsi/sites/default/files/1-

_estudio_cloud_computing_retos_y_oportunidades_vdef.pdf; último acceso: 18 de diciembre de 2014

[**14**] KAMARAJU A., NICOLAS P. [2009], "Cloud Storage," Storage Networking Industry Association, tomado de http://www.luisespino.com/pub/cloud_computing_luis_espino.pdf, último acceso: 27 de diciembre de 2014

[**15**] MENDOZA A.[2007], Utility Computing Technologies, Standards, and Strategies. United States of America: Artech House, Inc., tomado de http://www.luisespino.com/pub/cloud_computing_luis_espino.pdf, último acceso: 27 de diciembre de 2014

[**16**] Observatorio Regional de Sociedad de la Información (ORSI) (2010), Cloud Computing. La Tecnología como Servicio, tomado de http://www.orsi.jcyl.es, último acceso: 30 de diciembre de 2014

[**17**] SING T., KUMAR P. [2009], "Smart Metering the Clouds," in 18th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises, United States of America, pp. 66-71; tomado de http://www.luisespino.com/pub/cloud_computing_luis_espino.pdf, último acceso: 19 de diciembre de 2014

[**18**] JOYANES, Luis [2011], Computación en la Nube e innovaciones tecnológicas El nuevo paradigma de la Sociedad del Conocimiento, tomado de https://gissic.files.wordpress.com/2011/07/computacion_en_nube_revista_paraguay_luis_joyanes.pdf, último acceso: 19 de diciembre de 2014

[**19**] AVILA, Oscar [2011], Computación en la nube, tomado de http://www.izt.uam.mx/newpage/contactos/anterior/n80ne/nube.pdf, último acceso: 19 de diciembre de 2014

[**20**] HARTPENCE, Bruce (2013), Packet Guide to Voice over IP, O'Reilly Media, Inc.

[**21**] http://es.slideshare.net/francescperezfdez/introduccin-a-los-protocolos-de-enrutamiento-din; último acceso: 18 de FEBRERO de 2015

[**22**] Balchunas, A (2007), Routing Protocol Comparison v1.01, extraído desde http://www.routeralley.com/ra/docs/routing_protocol_comparison.pdf, 03 FEBRERO 2015

[**23**] YÁGÜEZ García Javier [2011], Arquitectura de Redes de Comunicaciones, tomado de http://pegaso.ls.fi.upm.es/arquitectura_redes/clase4-CUARTOENCAMINADINAMICO-MPLS-9noviembre2011.pdf, último acceso: 18 de marzo de 2015

[24] AHMED Addel, MADANI Habib, SIDDIQUI Talal [2009], VoIP Performance Management and Optimization: Managing VoIP Networks, tomado de http://www.ciscopress.com/articles/article.asp?p=1339559&seqNum=7, último acceso: 18 de marzo de 2015

[25] VoIP Performance Management and Optimization: Managing VoIP Networks[2013], tomado de http://elastixtech.com/qos-calidad-de-servicio-para-VoIP/, último acceso: 18 de marzo de 2015

[**26**] Entendiendo HTML5: guía para principiantes [2013], tomado de http://hipertextual.com/archivo/2013/05/entendiendo-html5-guia-para-principiantes/, último acceso: 23 de marzo de 2015

[**27**] NAVAJAS OJEDA, Antonio [2012], Guía Completa de CSS3, tomada de http://www.etnassoft.com/biblioteca/guia-completa-de-css3/, último acceso: 23 de marzo de 2015

[**28**] MERINO, MARCOS [2014], ¿Qué es una API y para qué sirve?, tomada de http://www.ticbeat.com/tecnologias/que-es-una-api-para-que-sirve/, último acceso: 23 de marzo de 2015

[**29**] ALEGSA, Leandro [2014], Definición de Script, tomado de http://www.alegsa.com.ar/Dic/script.php, último acceso: 23 de marzo de 2015

[**30**] ¿Qué es JavaScript? - Definición de Javascript, tomada de https://www.masadelante.com/faqs/javascript, último acceso: 25 de marzo de 2015

[**31**] GRIGORIK Ilya [2013], High Performance Browser Networking, O'Reilly Media, Inc.

[**32**] RISTIC Dan [2015], Learning WebRTC, Packt Publishing

[**33**] WebRTC Arquitecture, tomada de http://www.webrtc.org/architecture, último acceso: 23 de marzo de 2015

[**34**] MILLAN, Ramón Jesús [2014], WebRTC: comunicaciones en tiempo real en el navegador Web, tomado de http://www.ramonmillan.com/tutoriales/webrealtimecommunications.php, último acceso: 23 de mayo de 2015

[**35**] A Study of WebRTC Security [2015], tomada de http://webrtc-security.github.io/, último acceso: 16 de mayo de 2015

[**36**] Turn, tomada de http://wikitel.info/wiki/TURN, último acceso: 16 de mayo de 2015

[**37**] LEVI Tsahi Levent [2014], What is the Difference Between a Signaling Protocol and a Transport Protocol?, tomada de https://bloggeek.me/signaling-vs-transport-protocol/, último acceso: 16 de mayo de 2015

[**38**] DOS SANTOS Afonso[2015], Configuraciones de red soportadas por la Central Telefónica 3CX, tomada de http://www.3cx.es/blog/configuraciones-red-soportadas; último acceso: 16 de julio de 2015

[**39**] http://www.3cx.es/webrtc/que-es-webrtc; último acceso: 10 de septiembre de 2015

[**40**] ADAM Bergkvist Ericsson [2015], WebRTC Peer-to-peer connections, tomada de http://caniuse.com/#search=webrtc, último acceso: 10 de septiembre de 2015

[**41**] AHMED Adeel [2009], VoIP Performance Management and Optimization: Managing VoIP

Networks, tomada de http://www.ciscopress.com/articles/article.asp?p=1339559&seqNum=7, último acceso: 10 de septiembre de 2015

[42] https://tools.ietf.org/html/draft-ietf-rtcweb-rtp-usage-25, último acceso: 10 de septiembre de 2015

[43] http://www.bandcalc.com/es/#note6, último acceso: 10 de octubre de 2015

[44] http://www.3cx.es/, último acceso: 10 de octubre de 2015

[45] http://www.3cx.com/ordering/pricing/webmeeting/, último acceso: 10 de octubre de 2015

[46] http://whois.domaintools.com, último acceso: 10 de octubre de 2015

GLOSARIO DE TÉRMINOS

API:	Application Programming Interface
RTT:	Round-trip delay time
W3C:	World Wide Web
WebRTC:	Web Real-Time Communication
RTP:	Real-Time Transport Protocol,
UDP:	User Datagram Protocol
LAN:	Local Area Network
WAN:	Wilde Area Network
VoIP:	Voice Over Internet Protocol
URL:	Uniform Resource Locator
TLS:	Transport Layer
TURN:	Traversal Using Relays Around
STUN:	Session Traversal Utilities
SRTP:	Secure Real-time Transport Protocol
SIP:	Session Initiation Protocol
HTTP:	HyperText Transfer Protocol
HTTPS:	HyperText Transfer Protocol Secure
QoS:	Quality of Service
GNU:	General Public License

Apéndices

APENDICE A

CONFIGURACION DEL SERVIDOR ASTERISK

Para el estudio que se plantea los requerimientos para la configuración del servidor Asterisk son los siguientes:

Virtual Box

Una manera de simplificar la implementación es virtualizar un sistema operativo dentro otro sistema operativo, independientemente del cual sea nuestro sistema base, se puede instalar un nuevo sistema que correrá sobre el nativo en una PC, debido a los requerimientos mínimos de Centos 7 los recursos del PC que se convierta en servidor serán compartidos y no afectará el desempeño sobre el servidor Asterisk y el PC.

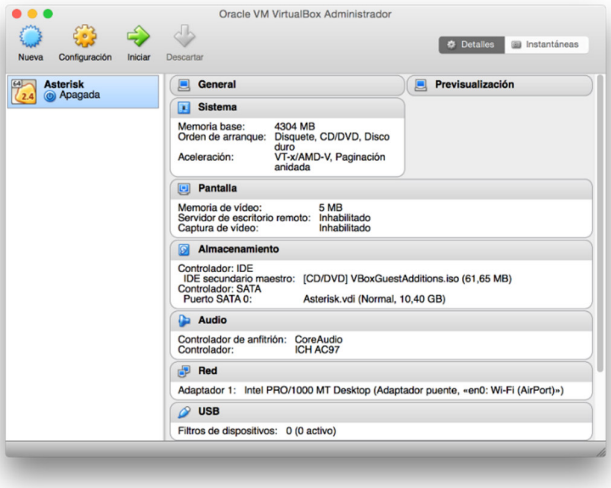


Figura A. 1. Máquina Virtual de Asterisk. (Autor de la Tesis 2015)

Es importante la configuración de VirtualBox especialmente la configuración de la tarjeta de red, la que debe estar configurada como un puente para que se pueda tener acceso a la red donde el PC principal se encuentra conectado.

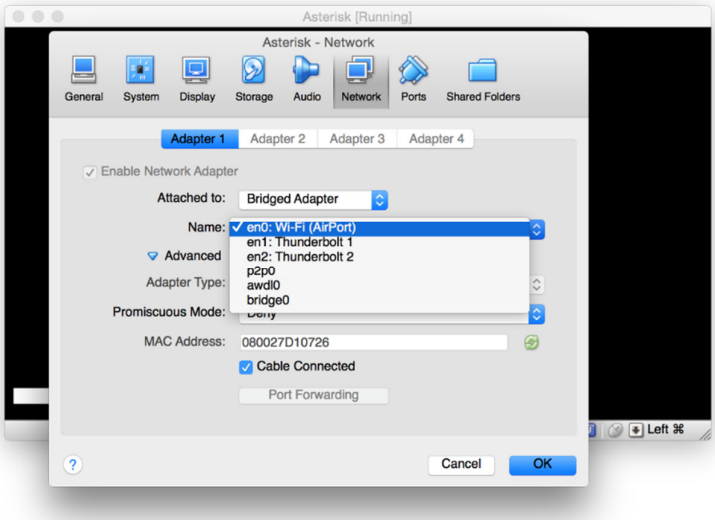


Figura A.2. Configuración de Tarjeta de Red en Virtual BOX. (Autor de la Tesis 2015)

La configuración de Bridged Adapter permitirá hacer puente con cualquiera de las interfaces que posee la PC.

Es importante configurar la dirección IP de la máquina virtual de forma estática ya que luego permitirá utilizar los SOCKETS para el intercambio de datos entre dos aplicaciones. Esta dirección estática permitirá abrir puertos relacionándolos con la máquina virtual.

La configuración de red de Linux se puede realizar mediante el comando:

nano /etc/sysconfig/network-scripts

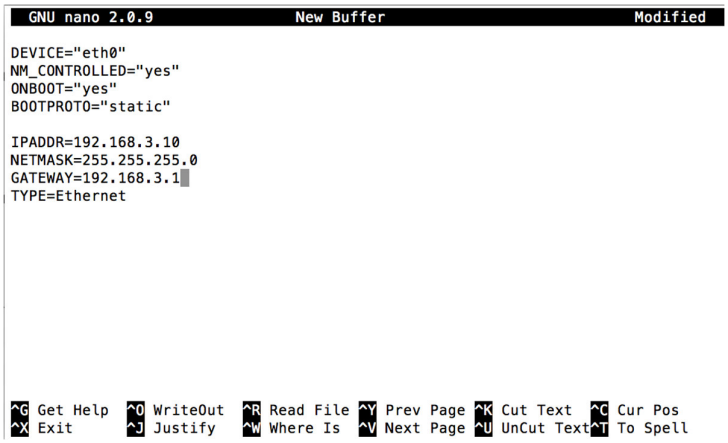


Figura A. 3. Configuración de IP de la interface de red en Centos. (Autor de la Tesis 2015)

Una vez configurada la dirección IP estática se debe reiniciar el servicio de red, para instalar

Asterisk con WebRTC.

Linux Centos 7 .iso

La imagen de Linux Centos permite agregar los repositorios de Asterisk y las librerías, para la instalación de WebRTC se deben instalar todas las librerías de las que depende.
Para esto se continúa con el siguiente procedimiento:

- yum update
- yum install gcc-c++ make gnutls-devel kernel-devel libxml2-devel ncurses-devel subversion doxygen texinfo curl-devel net-snmp-devel neon-devel
- yum install uuid-devel libuuid-devel sqlite-devel sqlite git speex-devel gsm-devel

Mediante estas librerías se tendrán todas las dependencias para poder instalar **Asterisk y WebRTC**.

Debido a la exposición de la comunicación en la red también es importante que se instale libsrtp la que permitirá codificar los datos en la red ya que se trata de un protocolo de seguridad. Esta librería viene disponible para compilar manualmente por lo que se recomienda verificar la versión compatible con el sistema operativo y con la versión de Asterisk.

- #cd /usr/src/
- #wgethttp://downloads.asterisk.org/pub/telephony/asterisk/asterisk-13-current.tar.gz
- #tar –xzvf asterisk-13-current.tar.gz
- #cd /usr/src/asterisk-13.1.0 && make clean./configure –with-crypto –with-ssl –with-srtp=/usr/local/lib
- #contrib/scripts/get_mp3_source.sh
- #make menuselect.makeopts
- #menuselect/menuselect –enable format_mp3 –enable res_config_mysql –enable app_mysql –enable app_saycountpl –enable cdr_mysql –enable EXTRA-SOUNDS-EN-GSM

Con esta línea de comandos se podrá compilar manualmente la instalación de Asterisk. Se debe tomar en cuenta que la versión que se ha instalado es la Asterisk 13.4.0 y que las librerias y dependencias deben ser compatibles.

Configurar extensiones

Una vez instalado Asterisk se prosigue con la configuración básica de las extensiones SIP, que servirán para realizar llamadas.
Los archivos a configurar son

rtp.conf

Se configura el rango de protocolos para RTP entre 10000 y 20000 y el servidor STUN que permita realizar NAT, además el parámetro icesupport debe ser activado ya que posibilita la comunicación entre agentes corriendo detrás de NAT y firewalls

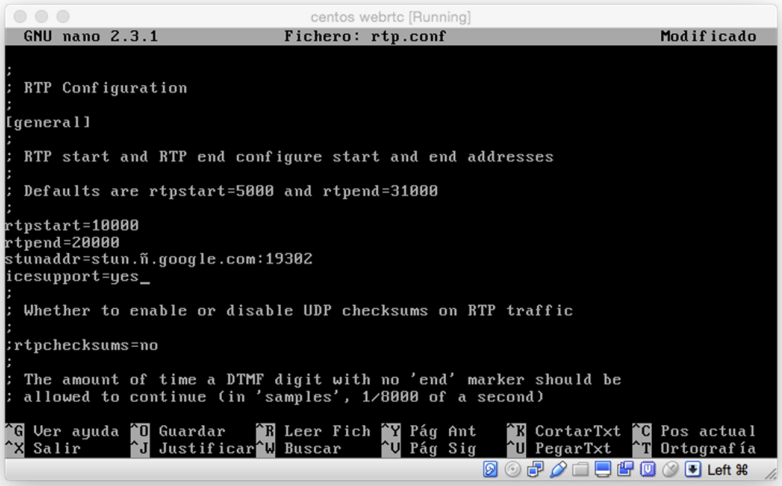


Figura A.4. Configuración de puertos RTP y servidor STUN. (Autor de la Tesis 2015)

http.conf

Asterisk contiene un servidor HTTP que se debe habilitar para que WebRTC pueda conectarse mediante el navegador al el servidor. El puerto asignado para http es el 8088. Además se tiene que asignar una dirección IP para que se identifique con el servidor o dejarla en 0.0.0.0 para permitir el acceso a cualquier host.

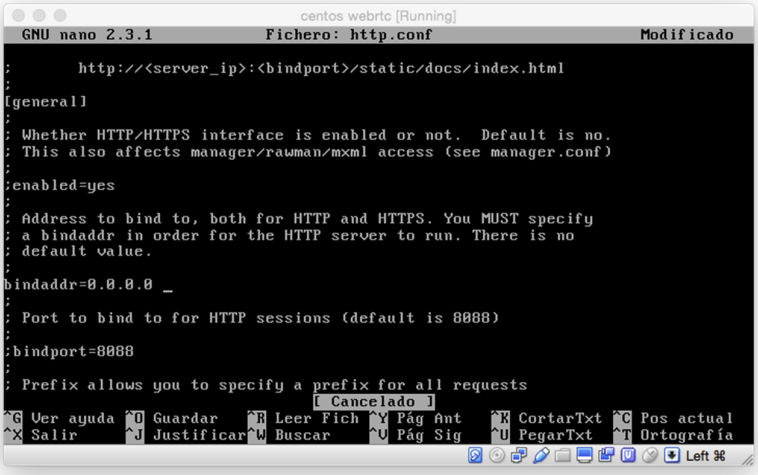


Figura A.5. Configuración de puerto para comunicación. (Autor de la Tesis 2015)

sip.conf

La configuración SIP permite configurar el socket de acceso 0.0.0.0:5060, esto permitirá a los softphones conectarse mediante este puerto al servidor, otro parámetro importante que maneja este archivo es la dirección IP pública la que posibilita direccionar los paquetes fuera de la LAN.

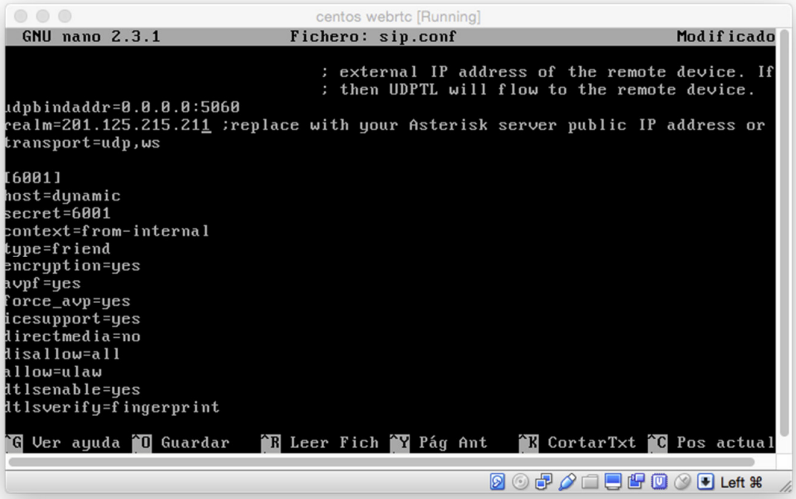


Figura A.6. Configuración de parámetros SIP para conexión de llamadas. (Autor de la Tesis 2015)

Se puede incluir en este archivo la configuración para el login del usuario de la telefonía WebRTC, [6001] equivale al usuario y secret=6001 al password que se utilizara para iniciar la sesión.

- pjsip.conf
- extensións.conf

En este archivo de configuración se añaden las extensiones SIP para hacer llamadas entre terminales.

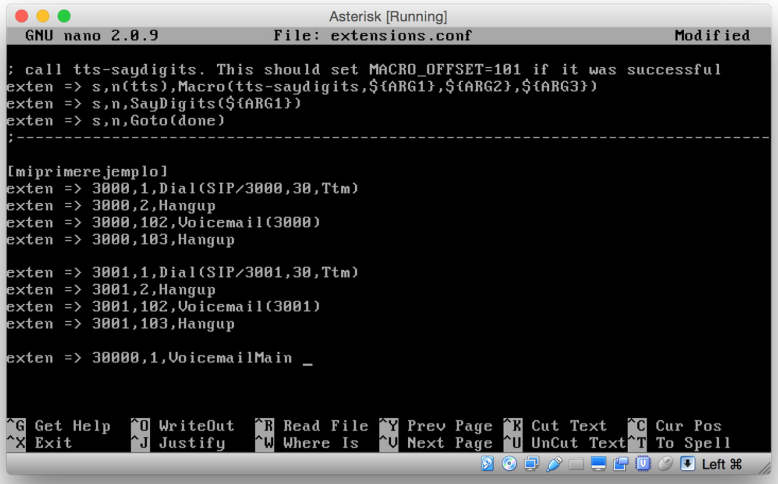


Figura A.7. Configuración de extensiones para realizar llamadas. (Autor de la Tesis 2015)

Las extensiones 3000 y 3001 permitirán hacer llamadas mediante teléfono IP con soporte SIP, softphones y mediante navegadores como Chrome.

Manejo de Certificados

El manejo de certificados web permitirá evaluar la seguridad de las páginas cumpliendo estándares de sitios seguros.

Es mejor utilizar una comunicación encriptada, con el uso de certificados podemos mitigar este problema, debido que las redes LAN expuestas al Internet tienden a ser víctimas de ataques o de pinchazos comúnmente llamados.

El protocolo SRTP (Secure Real-time Transport Protocols) permitirá cifrar y autenticar paquetes protegiendo la integridad del paquete, ya que está desarrollado para proteger datos en tiempo real como los de VOIP.

El protocolo TLS (Transport Layer Security) provee seguridad a nivel de capa de transporte, permite autenticar mediante certificados con una llave simétrica intercambiada con quien se está comunicando. Utilizado comúnmente en VoIP, navegación web y correo electrónico.

Es importante el uso de ambos protocolos porque a pesar de que con TLS se haya encriptado y salvaguardado la conexión, los datos transmitidos no lo están; por lo que van de la mano TLS y SRTP.

Asterisk también permite generar certificados mediante su aplicación web, se debe acceder vía web a la dirección del servidor Asterisk. El manejo de certificados se lo puede hacer en la siguiente sección

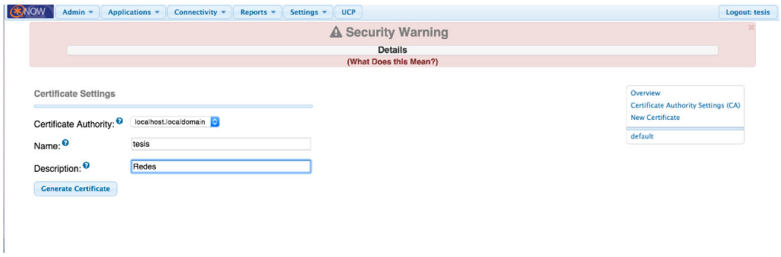


Figura A.8. Formulario para creación de Certificados vía Asterisk. (Autor de la Tesis 2015)

Firewall configurado

Uno de los puntos más sensibles al momento de exponer el servidor Asterisk a Internet es el firewall, se debe conocer todos los puertos que WebRTC necesita. Los puertos son mandatorios; es decir que se deben abrir todos los puertos que se necesitan, si alguno se omite la comunicación no podrá ser efectuada. Este punto es muy sensible debido a que los puertos que se abren son conocidos y al exponer el servidor en el Router al Internet se puede ser víctimas de ataques.

Dos seguridades a tomar en cuenta antes de que la comunicación se realice, deben ser la apertura de puertos en el router y la otra es verificar el firewall en el sistema operativo. Centos maneja su propio firewall con IPtables, si los puertos se encuentran cerrados en el firewall de Centos la comunicación será rechazada y el servidor no será transparente a la conexión con los clientes o host que necesitan el uso del servicio de VoIP.

```
[root@localhost ~]# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
fail2ban-FTP  tcp  --  anywhere             anywhere            tcp dpt:ftp
fail2ban-apache-auth  tcp  --  anywhere             anywhere
fail2ban-SIP  all  --  anywhere             anywhere
fail2ban-PBX-GUI  all  --  anywhere             anywhere
fail2ban-BadBots  tcp  --  anywhere             anywhere            multiport dports
 http,https
fail2ban-SSH  tcp  --  anywhere             anywhere            tcp dpt:ssh
fail2ban-recidive  all  --  anywhere             anywhere

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
```

Figura A.9. Políticas de IPTables en Linux Centos. (Autor de la Tesis 2015)

En la figura 9 de IPtables se muestra las políticas de seguridad de puertos, también se podría verificar o listar las conexiones abiertas con netstat.

Se puede verificar en Centos los puertos que se están escuchando y también los puertos cerrados, figura 10.

```
[root@localhost ~]# netstat
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address               Foreign Address             State

tcp        0      0 localhost:5038              localhost:53840             ESTABLIS
HED
tcp        0      0 localhost:54756             localhost:5038              TIME_WAI
T
tcp        0      0 localhost:54754             localhost:5038              TIME_WAI
T
tcp        0     48 192.168.2.104:ssh           192.168.2.70:57738          ESTABLIS
HED
tcp        0      0 localhost:53840             localhost:5038              ESTABLIS
HED
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags       Type       State         I-Node Path
unix  2      [ ]         DGRAM                    6871   @/org/kernel/udev/udevd
unix  2      [ ]         DGRAM                    8752   @/org/freedesktop/hal/udev_
event
```

Figura A.10. Conexiones abiertas con Linux Centos. (Autor de la Tesis 2015)

Los puertos que deben estar abiertos para el uso de WebRTC se han configurado en el servidor Asterisk y son los siguientes:

- 5060 permite la comunicación te softphones
- 8088 permite la conexión segura HTTP
- 10000-20000 rango de protocolos que permite RTP

IP

La exposición del servidor Asterisk a Internet exige que se deban configurar tanto la IP privada para el servidor y la IP Pública, para que mediante puertos se pueda visualizar en el Internet el servidor. A esta combinación se le llama sockets y mediante estos se puede acceder desde cualquier parte del mundo al servidor.

Esta configuración de WebRTC en Centos permite utilizar la tecnología HTML5 SIP client; Google provee un cliente basado en javascript para la integración en redes sociales como Twitter, Google+ y Facebook, juegos en línea y la mayoría de servicios que se den en la web, la facilidad de esta utilidad es que no necesita de plugins ni programaciones extra.

Este cliente web de Google permite conectar al servidor de Asterisk configurado para WebRTC y realizar comunicación entre clientes conectados al servidor, además este cliente brinda características de realizar llamadas de voz, de video y mensajería instantánea.

Para acceder al cliente se debe ingresar a http://sipml5.org/; a continuación se detalla la configuración de los parámetros y como se realiza un test como llamada.

El nombre para identificarse con el cliente al que se quiere comunicar.

Los datos de SIP, es decir la extensión 3000 y los datos que están configurados en sip.conf , el usuario, la dirección IP pública y el password.

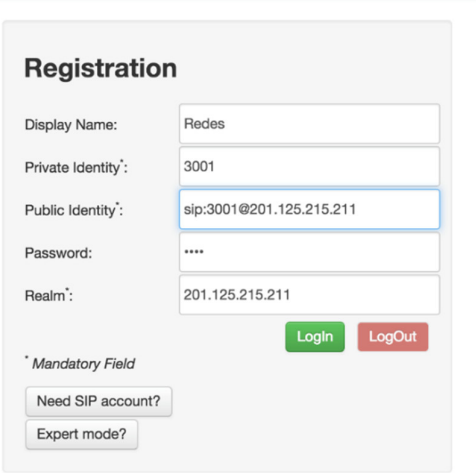


Figura A. 11. Formulario para la conexión de Cliente SIP al servidor Asterisk. (Autor de la Tesis 2015)

Luego en modo experto se configuran los parámetros siguientes que están dentro del servidor Asterisk.

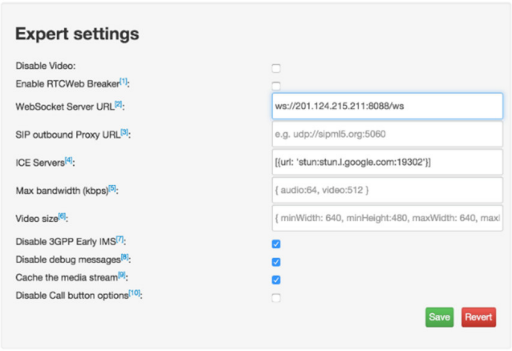


Figura A.12. Formulario para conexión del Cliente al Servidor WebRTC mediante STUN. (Autor de la Tesis 2015)

Se configura el web socket con el puerto 8088 y la dirección IP pública, además se configura el servidor STUN de Google “tun:stun.l.google.com:19302”, que permitirá realizar NAT.

APENDICE B

CONFIGURACION DEL SERVIDOR 3CX

Para el estudio que se plantea los requerimientos para la configuración del servidor 3CX son los siguientes:

Los requerimientos de hardware mínimos son:

Requerimientos	Características
Memoria	1GB
CPU	Intel Core i3
Servidor web	Abyss
Disco	SATA 30GB
Red	100/1000/Mbit/s

Tabla B.1. Requerimientos de hardware para instalación de 3CX. (Fuente: http://www.3cx.es/)

Figura B.1. Interfaz de Inicio de 3CX. (Autor de la Tesis 2015)

Al igual que Asterisk también se puede instalar 3CX dentro de una máquina virtual de Windows y realizar un puente con la interface de red y la máquina virtualizada.

Extensiones

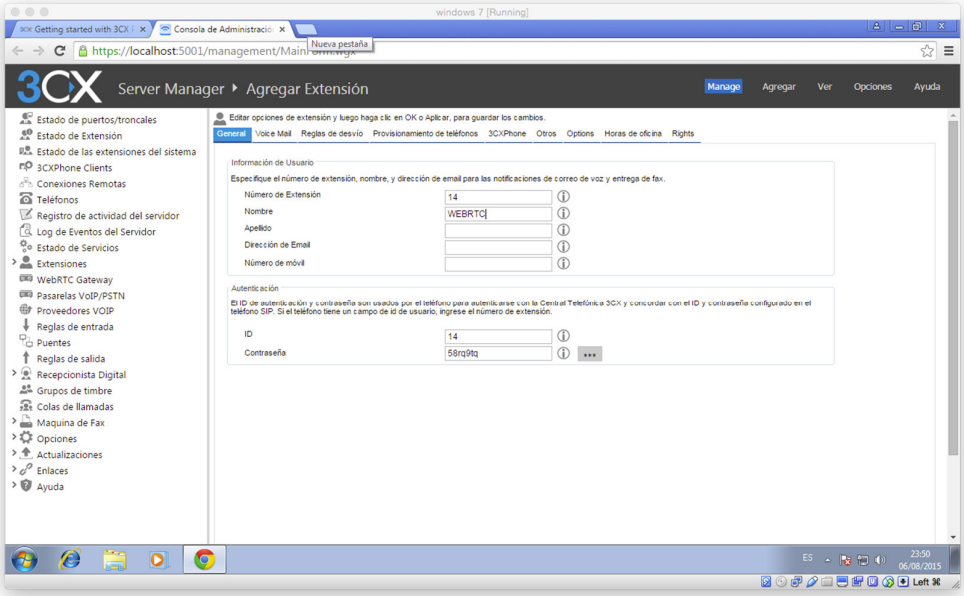


Figura B.2. Configuración de extensiones SIP en 3CX. (Autor de la Tesis 2015)

Con 3CX se pueden crear extensiones de manera sencilla, simplemente en la pestaña de agregar se da un clic a extensiones y se configura la misma, primeramente el número de extensión, previamente se había configurado al momento de la instalación la cantidad de dígitos que debe de tener el número de la extensión, en este caso se eligieron dos y para el ejemplo se ha elegido el número 14. El nombre, apellido, correo y número telefónico también son campos importantes al momento de la administración de las extensiones aunque opcional pero permite identificar a quien pertenece la extensión.

EL ID permite conectarse a la central telefónica 3CX, el ID necesita ser validado con un password; 3CX permite agregar todas las extensiones que se necesitan, al ser un software con licencia en la versión gratuita permite solamente dos llamadas simultaneas y en la versión PRO hasta 1024, también dependerá del hardware para implementar las versiones con licencia.

3CX permite controlar otras opciones como el correo de voz para las llamadas que no se puedan atender, reglas de desvío cuando el usuario no pueda atender las llamadas, se pueden hacer un traspaso de extensión y otras opciones de mucha utilidad que pueden ser configuradas dependiendo de lo que la extensión requiera.

WEBRTC

WebRTC está ganando mucha importancia en internet y los navegadores como Google Chrome, Mozilla Firefox y Opera están soportando este estándar; Apple trata de implementar su propia tecnología pero la vista a futuro es que se una a el estándar WebRTC.

3CX es compatible actualmente con WebRTC y permite conectarse a la central telefónica desde cualquier locación geográfica mediante Internet, sin usar ninguna aplicación extra solamente uno de los navegadores web que se ha mencionado.

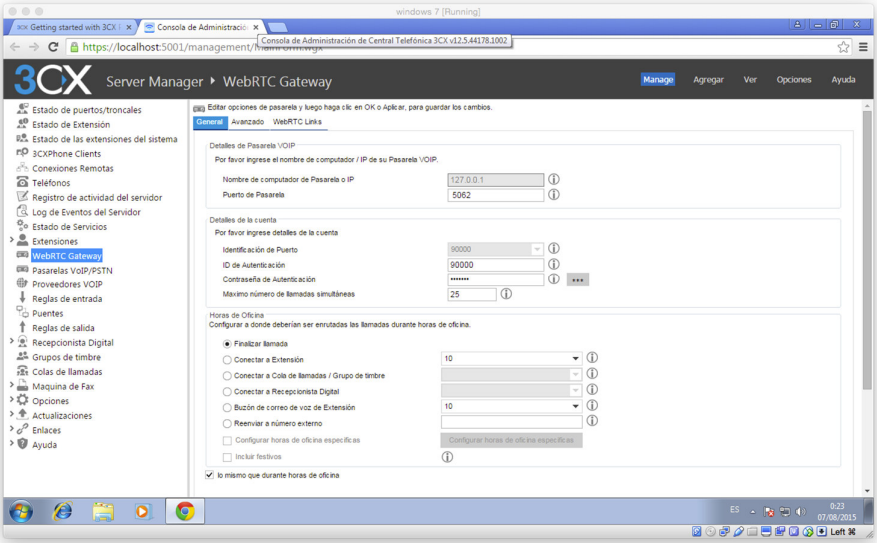


Figura B.3. Configuración de puertos WebRTC en 3CX. (Autor de la Tesis 2015)

Para implementar WebRTC en 3CX se configura el puerto VoIP de pasarela con el valor 5061, este puerto debe estar abierto en el firewall del sistema operativo y del router. También se puede configurar el ID y contraseña de la cuenta WebRTC.

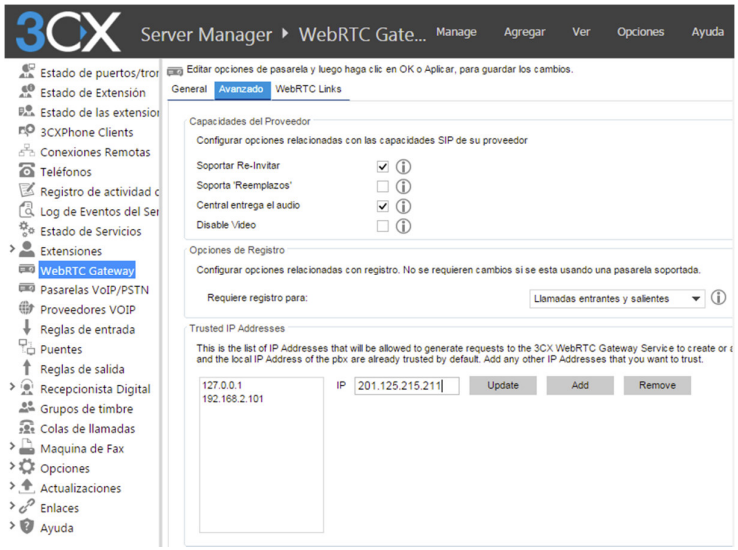


Figura B.4. Configuración de IPs para comunicación de WebRTC. (Autor de la Tesis 2015)

En las configuraciones avanzadas se agrega la dirección IP del Gateway que permitirá acceder remotamente y localmente al servidor.

Para acceder a las llamadas desde el navegador es necesario una URL que permita direccionar el tráfico hacia el servidor, esta es generada automáticamente con los parámetros que se han configurado al momento de la instalación de WebRTC, lo que se necesita es un socket, es decir la dirección IP pública y el puerto de acceso, por defecto los puertos son 80 y 443 para este caso se ha configurado el 5000 para HTTP y 5001 para HTTPS.

En la opción WebRTC Links aparecen todas las extensiones configuradas, aquí se generan las URL para el acceso a la llamada vía web, se selecciona la extensión que se le quiere dar acceso y se habilita el link.

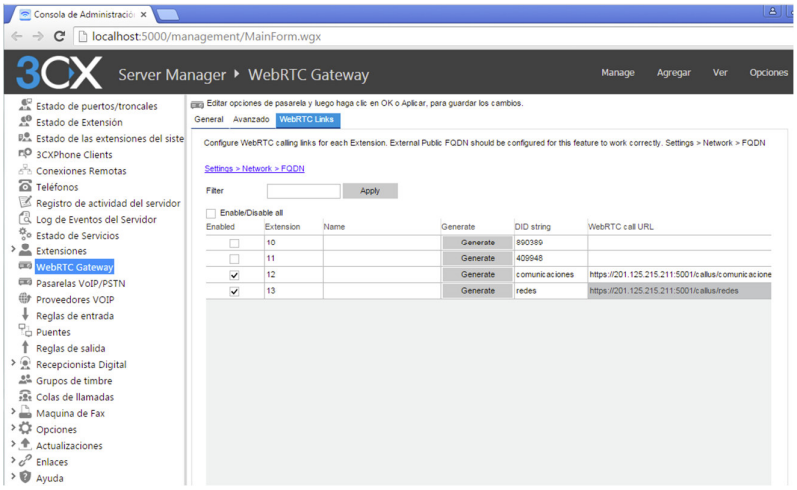


Figura B. 5. Configuración de Links para llamadas WebRTC. (Autor de la Tesis 2015)

Las extensiones generadas son:

- https://201.125.215.211:5001/call/comunicaciones
- https://201.125.215.211:5001/call/redes

Estas URL corresponden a las direcciones que se usan, para el proyecto, como extensiones; y permite realizar las llamadas en el navegador, la opción DID string permite configurar los nombres a las extensiones para poder identificarlos de mejor manera.

Firewall

Uno de los requisitos principales para que la central telefónica funcione es que se encuentre ubicada detrás de un firewall, cuando se realizan conexiones remotas se debe actuar sobre el firewall y configurarlo, de esta manera la central 3CX pueda comunicarse con los clientes remotos de manera segura y que la conexión tenga éxito.

Al usar extensiones SIP se deben abrir los siguientes puertos para que permitan la conexión:

El puerto UDP 5060 que es utilizado para las conexiones SIP.

El puerto TCP 5061 para comunicaciones TLS en conexiones seguras de SIP

Los puertos UDP 9000-9199 que permiten las conexiones RTP, para la transmisión del audio en la llamada.

Figura B.6. Representación de la comunicación con WebRTC y a través de Firewall.
(http://www.3cx.es/docs/configuracion-router-firewall-VoIP)

El 3CX Phone System equivale al servidor de WebRTC y se encuentra detrás del firewall, al realizar la llamada desde el VoIP Provider inicialmente se establece la conexión SIP y luego la comunicación de audio o video.

Se debe configurar las extensiones remotas para poder realizar las llamadas fuera de la red local, para conectarse necesitan del túnel 3CX y abrir los puertos siguientes:

El puerto 5090 UDP y TCP

Puertos 80 HTTP y 443 HTTPS

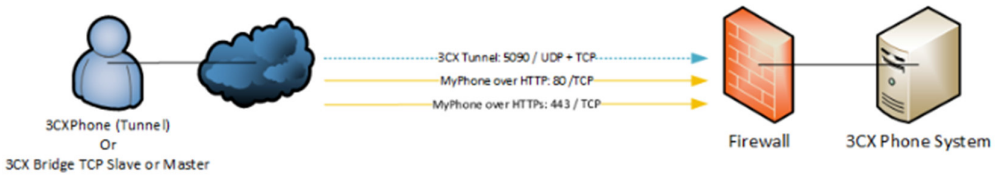


Figura B.7. Representación de la comunicación con WebRTC mediante puertos TCP y UDP. (http://www.3cx.es/docs/configuracion-router-firewall-VoIP)

En este caso las llamadas remotas se harán a través de conexiones mediante socket; la conexión al servidor se lo hace mediante la URL: "https://dominiowebrtc:443/" o mediante http://dominiowebrtc:80/; el puerto puede variar debido a que al momento de instalar 3CX también permite configurar y cambiar puertos de 5000 a 5001.

Certificados con IIS Web Server

Se requiere el uso de certificados adecuados para que los usuarios finales puedan comprobar la identidad del servidor, es necesario crear los certificados con entidades que emitan certificados de confianza, también se puede obtener certificados en línea en www.cacert.org; si se omite este paso al momento que los clientes remotos quieran autenticarse con el servidor no lo podrán realizar.

Para crear los certificados se debe usar IIS Manager, el que permite convertir un host en un servidor web, de esta manera se puede publicar páginas web de manera local o remota, también crear los certificados que permitirán autenticarse a los usuarios remotos.

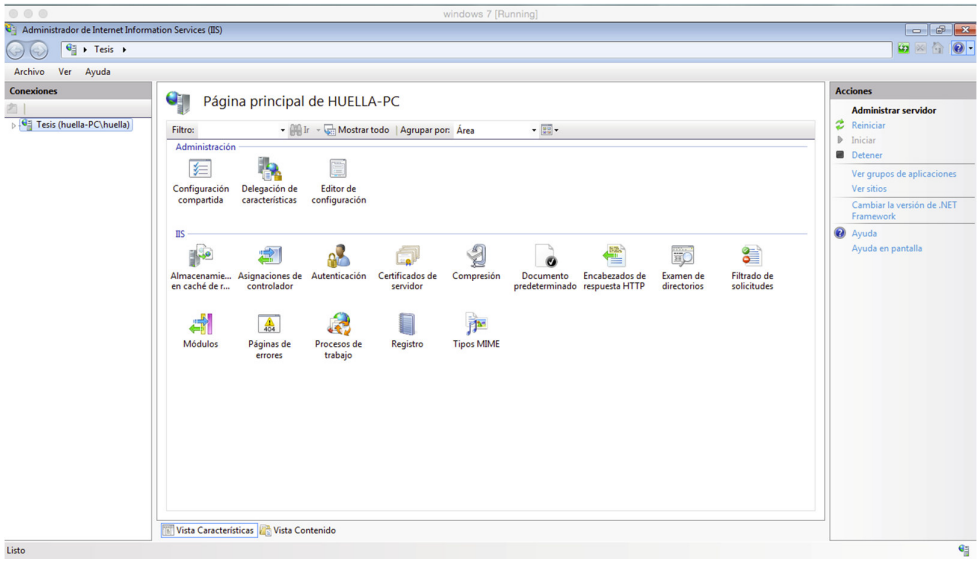


Figura B.8. Configuración en IIS para creación de certificados. (Autor de la Tesis 2015)

Para crear el certificado se da un clic en Certificados de Servidor y luego se llena el siguiente

formulario

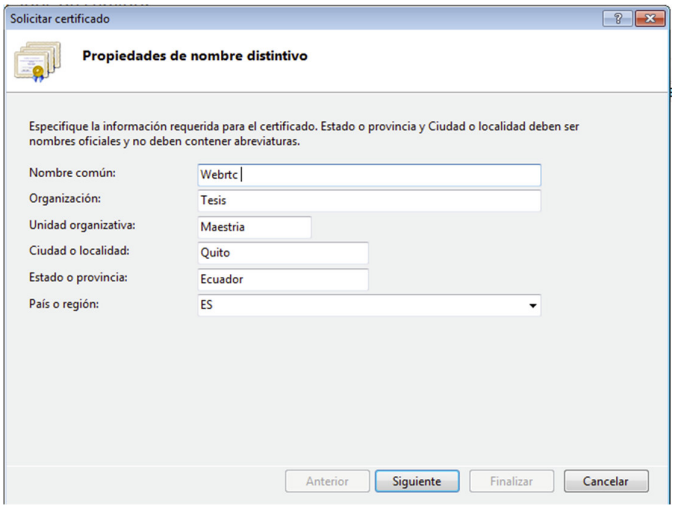


Figura B.9. Formulario de IIS para creación de certificados para WebRTC. (Autor de la Tesis 2015)

Al finalizar el formulario se genera un archivo que se puede exportar a un archivo con extensión .txt; este archivo se puede validar luego en la web www.cacert.org

El resultado es el siguiente:

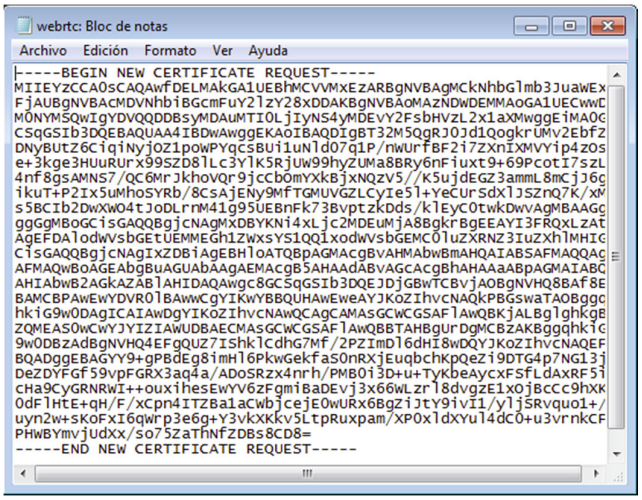


Figura B.10. Texto de codificación de Certificado para WebRTC. (Autor de la Tesis 2015)

Este archivo se lo debe validar en línea en la página de cacert y luego cargarlo al servidor ISS para que permita la autenticación con el servidor del proyecto.

APENDICE C

Captura de las pruebas realizadas

A continuación se muestran algunas capturas del proceso de toma de datos en la comunicación WebRTC en varias llamadas, con las distintas herramientas utilizadas.

- Toma de datos en comunicación mediante servidor Talky, con herramienta "webrtc-internals"

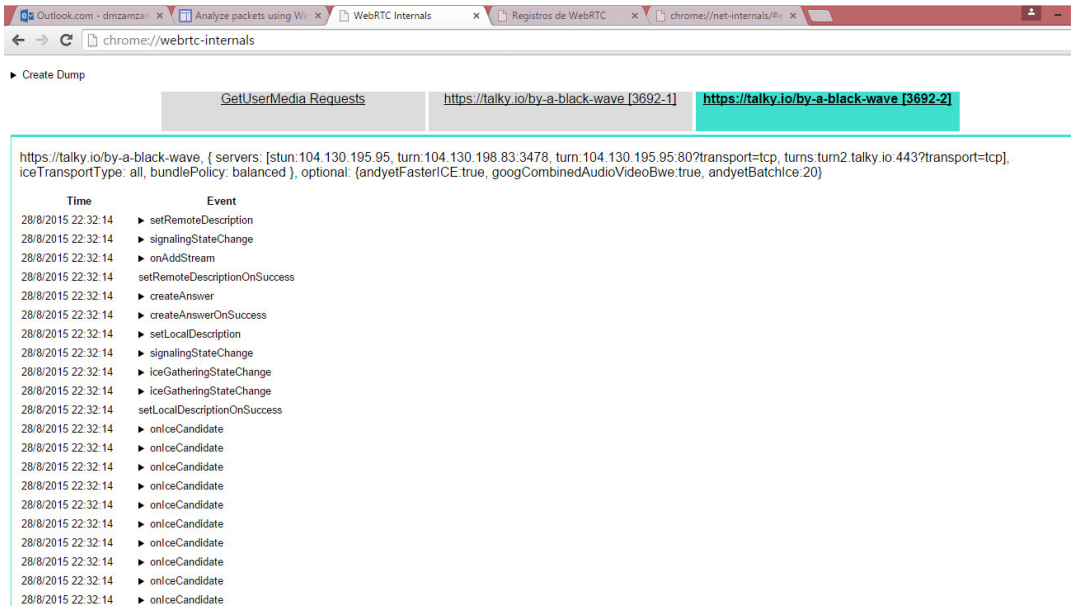


Figura C.1. webrtc-internals en servidor Talky. (Autor de la Tesis 2015)

- Verificación de Conectividad con ICMP-PING, y captura de paquetes con herramienta Wireshark

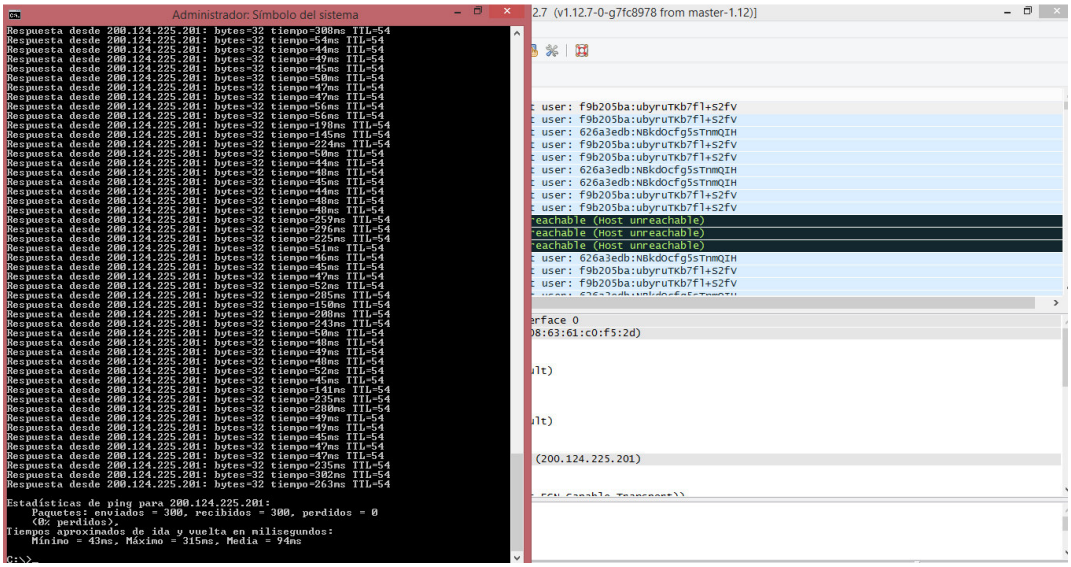


Figura C.2. Ping al cliente 2 webrtc y captura de comunicación con Wireshark (Autor de la Tesis 2015)

- Toma de datos en comunicación mediante servidor 3CX, con herramienta "webrtc-internals" y visualización de gráficos de algunas de las variables analizadas

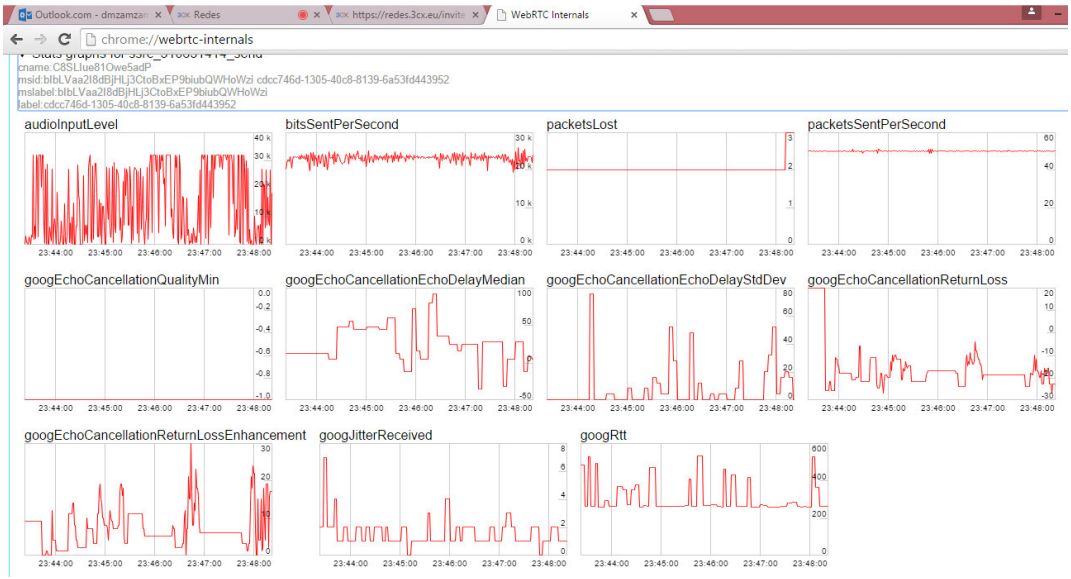


Figura C.3. Gráficos de webrtc-internals en servidor 3CX. (Autor de la Tesis 2015)

- Verificación grafica de ubicación geográfica de servidor WebRTC gratuito en la nube

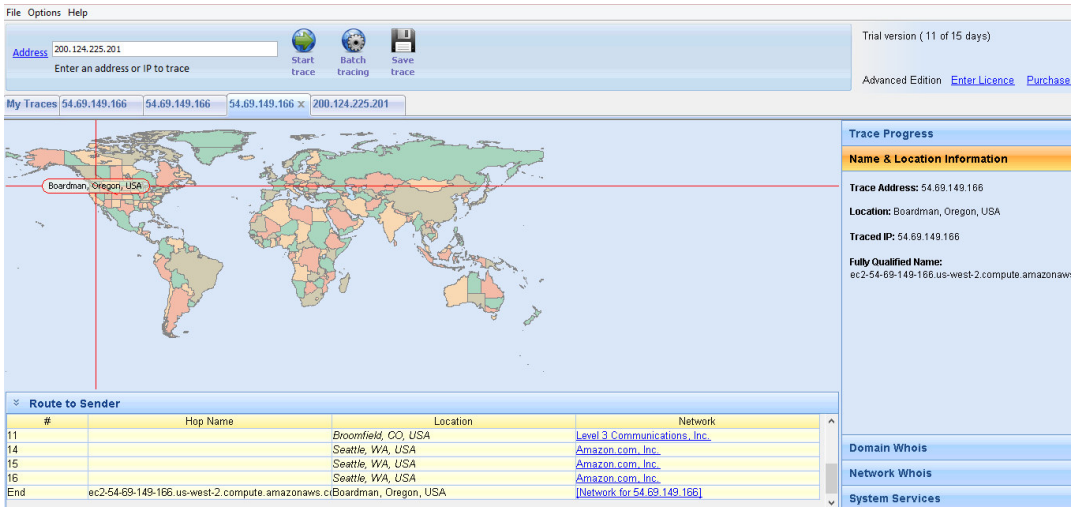


Figura C.4. Ubicación Geográfica de servidor WebRTC en la Nube (Autor de la Tesis 2015)

- Análisis de paquete RTP con herramienta Wireshark, en la comunicación WebRTC

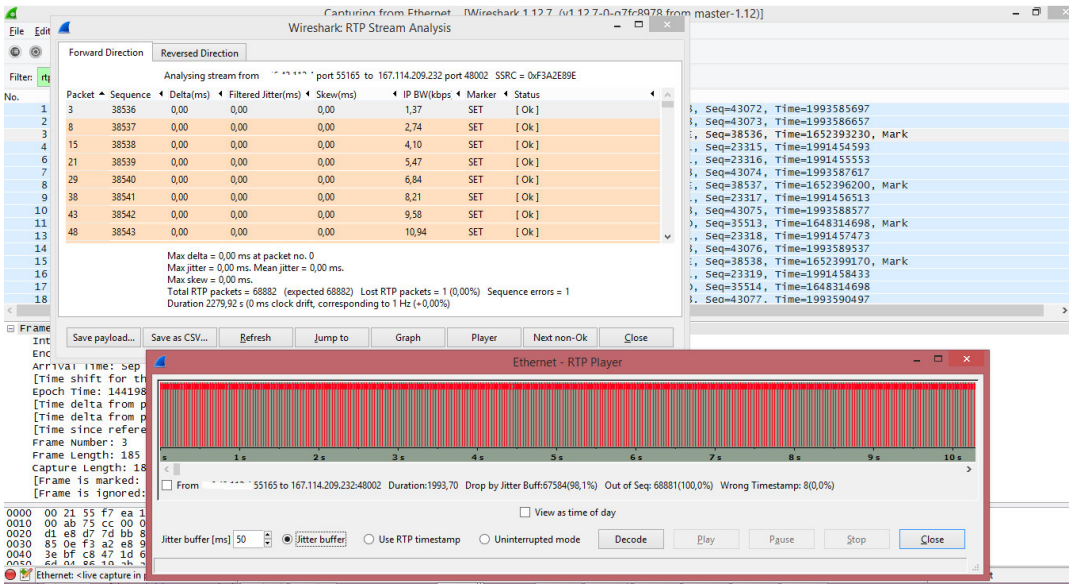


Figura C.5. Análisis de paquete RTP con Wireshark. (Autor de la Tesis 2015)

- Llamada WebRTC mediante el servidor 3CX, analizada con la herramienta "webrtc-internals"

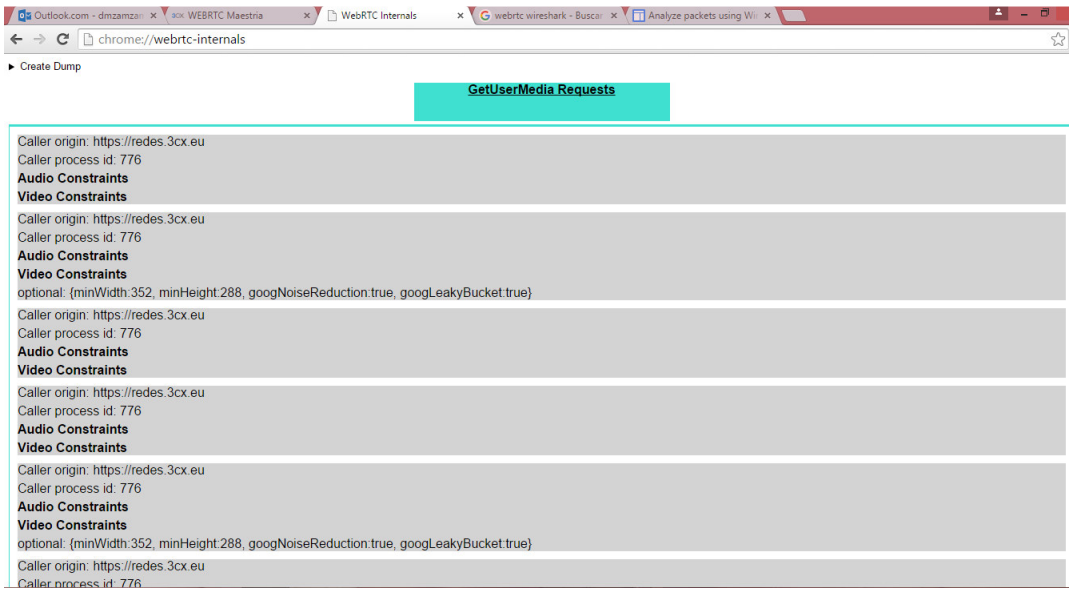


Figura C.6. Llamada WebRTC en servidor 3CX con "webrtc-internals". (Autor de la Tesis 2015)

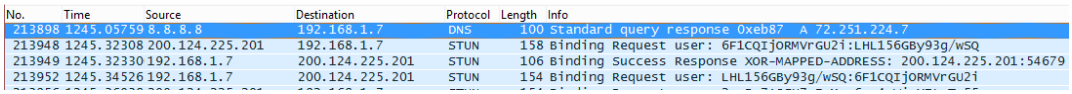
APENDICE D

Análisis de API PeerConnection para comunicación de VoIP mediante WebRTC

La API PeerConnection permite que dos usuarios se comuniquen directamente, de un navegador a otro, esto representa una asociación con el par remoto, que suele ser otra instancia de la misma aplicación JavaScript que se ejecuta en el navegador del otro extremo, a continuación se muestra este proceso con la herramienta Wireshark ejecutada en uno de los clientes de la comunicación.

El análisis se lo realiza con la captura de una llamada WebRTC mediante el servidor Talky.

1.- Consulta al servidor DNS y correspondencia con servidor STUN

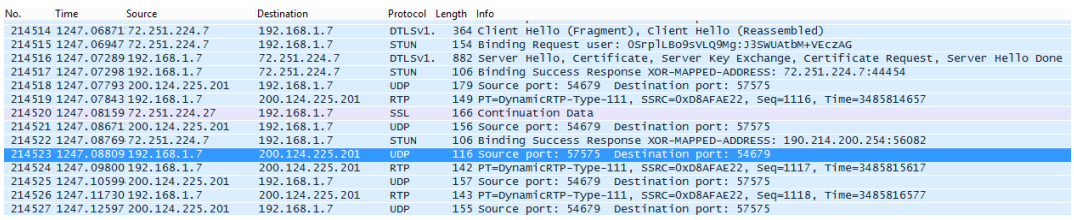


No.	Time	Source	Destination	Protocol	Length	Info
213898	1245.05759	8.8.8.8	192.168.1.7	DNS	100	Standard query response 0xeb87 A 72.251.224.7
213948	1245.32308	200.124.225.201	192.168.1.7	STUN	158	Binding Request user: 6F1CQIjORMVrGU2i:LHL156GBy93g/wSQ
213949	1245.32330	192.168.1.7	200.124.225.201	STUN	106	Binding Success Response XOR-MAPPED-ADDRESS: 200.124.225.201:54679
213952	1245.34526	192.168.1.7	200.124.225.201	STUN	154	Binding Request user: LHL156GBy93g/wSQ:6F1CQIjORMVrGU2i

Figura D.1. Consulta DNS y servidor STUN. (Autor de la Tesis 2015)

En la figura anterior el cliente ha iniciado la aplicación, con dirección IP 192.168.1.7, y ha recibido la respuesta del servidor DNS con información del servidor STUN con dirección IP 72.251.224.7, el cual se encarga de mapear IP pública del router del lado del cliente con un puerto que se usará en el tráfico de la llamada de VoIP.

2.- Protocolo STUN y negociación DTLS



No.	Time	Source	Destination	Protocol	Length	Info
214514	1247.06871	72.251.224.7	192.168.1.7	DTLSv1.	364	Client Hello (Fragment), Client Hello (Reassembled)
214515	1247.06947	72.251.224.7	192.168.1.7	STUN	154	Binding Request user: OSrplLBo9sVLQ9Mg:J3SWUAtbM+VEczAG
214516	1247.07289	192.168.1.7	72.251.224.7	DTLSv1.	882	Server Hello, Certificate, Server Key Exchange, Certificate Request, Server Hello Done
214517	1247.07298	192.168.1.7	72.251.224.7	STUN	106	Binding Success Response XOR-MAPPED-ADDRESS: 72.251.224.7:44454
214518	1247.07793	200.124.225.201	192.168.1.7	UDP	179	Source port: 54679 Destination port: 57575
214519	1247.07843	192.168.1.7	200.124.225.201	RTP	149	PT=DynamicRTP-Type-111, SSRC=0xD8AFAE22, Seq=1116, Time=3485814657
214520	1247.08159	72.251.224.27	192.168.1.7	SSL	166	Continuation Data
214521	1247.08671	200.124.225.201	192.168.1.7	UDP	156	Source port: 54679 Destination port: 57575
214522	1247.08769	72.251.224.7	192.168.1.7	STUN	106	Binding Success Response XOR-MAPPED-ADDRESS: 190.214.200.254:56082
214523	1247.08809	192.168.1.7	200.124.225.201	UDP	116	Source port: 57575 Destination port: 54679
214524	1247.09800	192.168.1.7	200.124.225.201	RTP	142	PT=DynamicRTP-Type-111, SSRC=0xD8AFAE22, Seq=1117, Time=3485815617
214525	1247.10599	200.124.225.201	192.168.1.7	UDP	157	Source port: 54679 Destination port: 57575
214526	1247.11730	192.168.1.7	200.124.225.201	RTP	143	PT=DynamicRTP-Type-111, SSRC=0xD8AFAE22, Seq=1118, Time=3485816577
214527	1247.12597	200.124.225.201	192.168.1.7	UDP	155	Source port: 54679 Destination port: 57575

Figura D.2. Protocolos STUN Y DTLS. (Autor de la Tesis 2015)

3.- Paquete STUN, Binding Request User

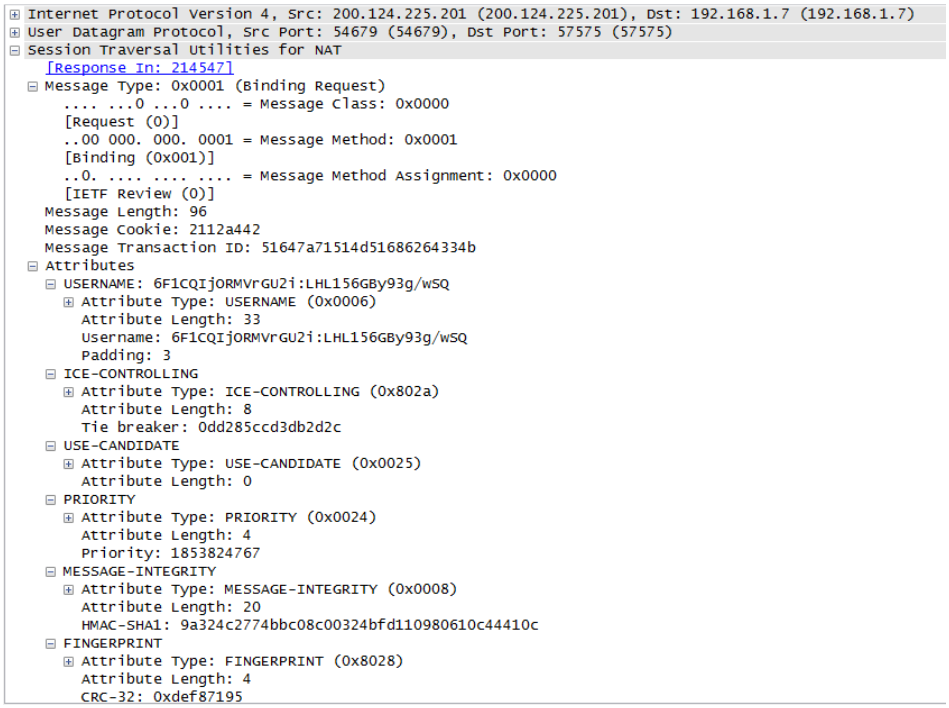


Figura D.3. Paquete STUN - Binding Request User . (Autor de la Tesis 2015)

4.- Paquete STUN, Binding Success Response

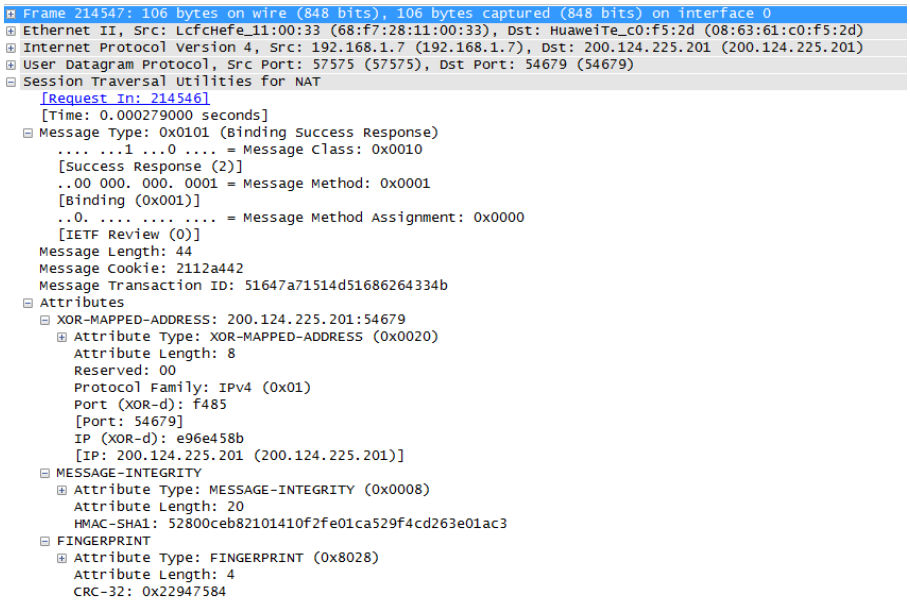


Figura D.4. Paquete STUN - Binding Success Response . (Autor de la Tesis 2015)

5.- Paquete DTLS, Client Hello

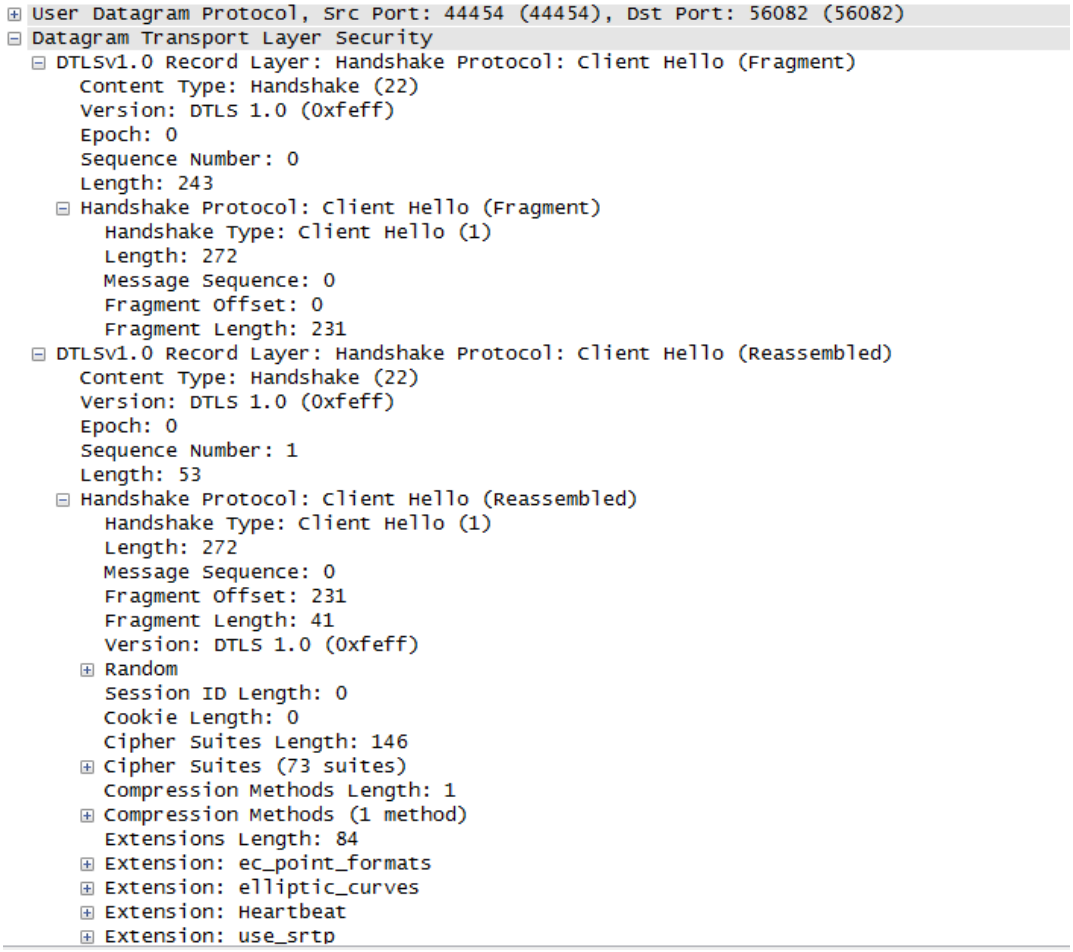


Figura D.5. Paquete DTLS, Client Hello (Autor de la Tesis 2015)

6.- Paquete DTLS, Server Hello

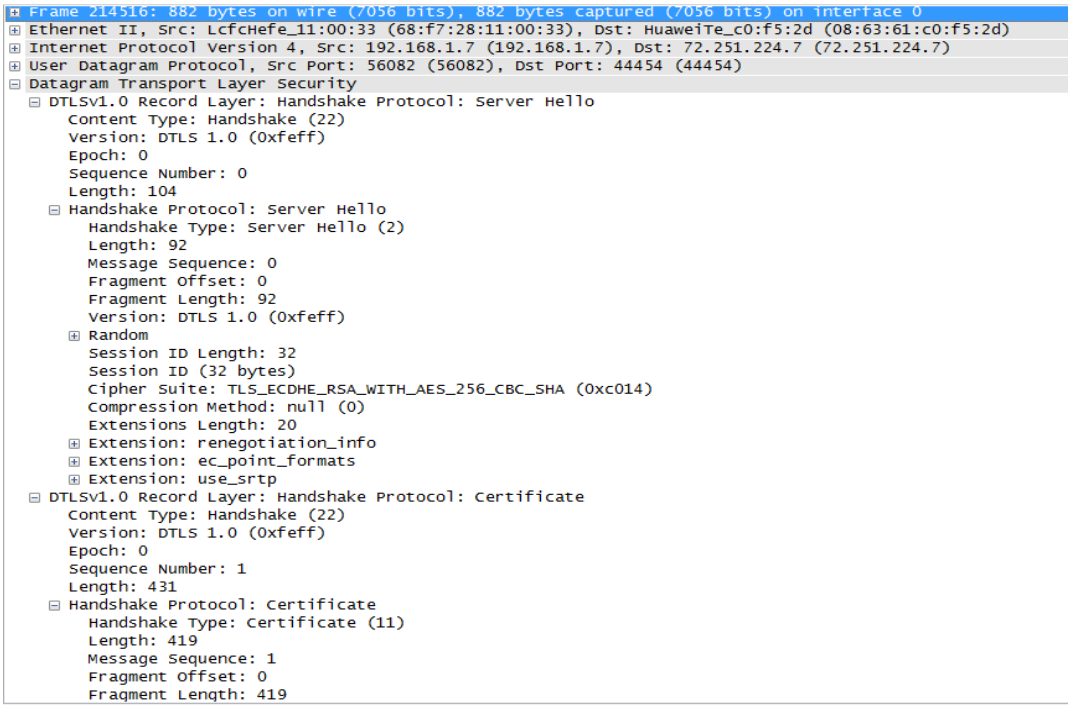


Figura D.6. Paquete DTLS, Server Hello (Autor de la Tesis 2015)

7.- Paquete DTLS, Change Cipher Spec

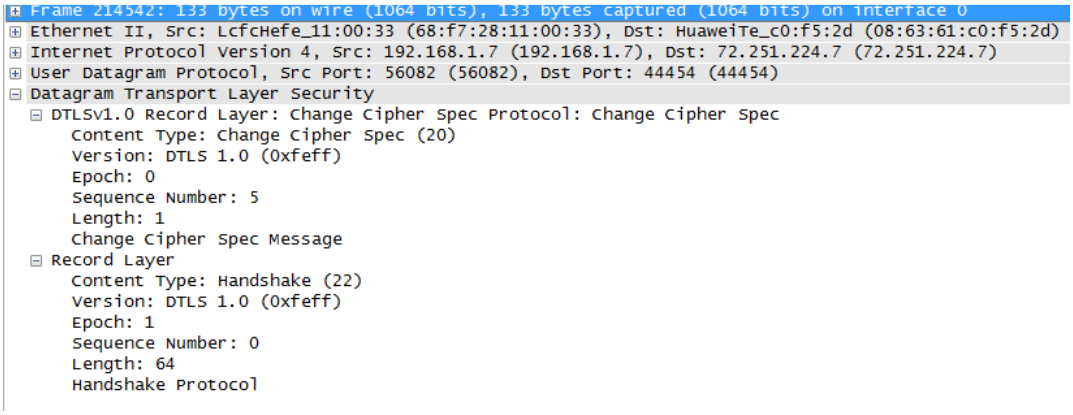


Figura D.7. Paquete DTLS, Change Cipher Spec (Autor de la Tesis 2015)

8.- Flujo de paquetes RTP

No.	Time	Source	Destination	Protocol	Length	Info
1	0.00000000	192.168.1.7	200.124.225.201	RTP	143	PT=DynamicRTP-Type-111, SSRC=0xD8AFAE22, Seq=4298, Time=3425954817
3	0.01922500	192.168.1.7	200.124.225.201	RTP	147	PT=DynamicRTP-Type-111, SSRC=0xD8AFAE22, Seq=4299, Time=3425955777
6	0.02189200	192.168.1.7	200.124.225.201	RTP	179	PT=DynamicRTP-Type-116, SSRC=0xCD30174A, Seq=23727, Time=2014018634, Mark
9	0.03959500	192.168.1.7	200.124.225.201	RTP	151	PT=DynamicRTP-Type-111, SSRC=0xD8AFAE22, Seq=4300, Time=3425956737
11	0.05953600	192.168.1.7	200.124.225.201	RTP	141	PT=DynamicRTP-Type-111, SSRC=0xD8AFAE22, Seq=4301, Time=3425957697
15	0.07876700	192.168.1.7	200.124.225.201	RTP	141	PT=DynamicRTP-Type-111, SSRC=0xD8AFAE22, Seq=4302, Time=3425958657
18	0.08694200	192.168.1.7	200.124.225.201	RTP	179	PT=DynamicRTP-Type-116, SSRC=0xCD30174A, Seq=23728, Time=2014024574, Mark
20	0.09999800	192.168.1.7	200.124.225.201	RTP	142	PT=DynamicRTP-Type-111, SSRC=0xD8AFAE22, Seq=4303, Time=3425959617
23	0.11968100	192.168.1.7	200.124.225.201	RTP	142	PT=DynamicRTP-Type-111, SSRC=0xD8AFAE22, Seq=4304, Time=3425960577
25	0.13938600	192.168.1.7	200.124.225.201	RTP	138	PT=DynamicRTP-Type-111, SSRC=0xD8AFAE22, Seq=4305, Time=3425961537
27	0.15606900	192.168.1.7	200.124.225.201	RTP	179	PT=DynamicRTP-Type-116, SSRC=0xCD30174A, Seq=23729, Time=2014030784, Mark
28	0.15941700	192.168.1.7	200.124.225.201	RTP	142	PT=DynamicRTP-Type-111, SSRC=0xD8AFAE22, Seq=4306, Time=3425962497
30	0.17933700	192.168.1.7	200.124.225.201	RTP	138	PT=DynamicRTP-Type-111, SSRC=0xD8AFAE22, Seq=4307, Time=3425963457
33	0.19947000	192.168.1.7	200.124.225.201	RTP	148	PT=DynamicRTP-Type-111, SSRC=0xD8AFAE22, Seq=4308, Time=3425964417
36	0.21903600	192.168.1.7	200.124.225.201	RTP	143	PT=DynamicRTP-Type-111, SSRC=0xD8AFAE22, Seq=4309, Time=3425965377
38	0.22308400	192.168.1.7	200.124.225.201	RTP	179	PT=DynamicRTP-Type-116, SSRC=0xCD30174A, Seq=23730, Time=2014036724, Mark
40	0.23881700	192.168.1.7	200.124.225.201	RTP	144	PT=DynamicRTP-Type-111, SSRC=0xD8AFAE22, Seq=4310, Time=3425966337
44	0.25944300	192.168.1.7	200.124.225.201	RTP	141	PT=DynamicRTP-Type-111, SSRC=0xD8AFAE22, Seq=4311, Time=3425967297
48	0.27956100	192.168.1.7	200.124.225.201	RTP	139	PT=DynamicRTP-Type-111, SSRC=0xD8AFAE22, Seq=4312, Time=3425968257
52	0.29113100	192.168.1.7	200.124.225.201	RTP	136	PT=DynamicRTP-Type-116, SSRC=0xCD30174A, Seq=23731, Time=2014042664, Mark
53	0.29892500	192.168.1.7	200.124.225.201	RTP	154	PT=DynamicRTP-Type-111, SSRC=0xD8AFAE22, Seq=4313, Time=3425969217
56	0.31876400	192.168.1.7	200.124.225.201	RTP	153	PT=DynamicRTP-Type-111, SSRC=0xD8AFAE22, Seq=4314, Time=3425970177
57	0.33960200	192.168.1.7	200.124.225.201	RTP	146	PT=DynamicRTP-Type-111, SSRC=0xD8AFAE22, Seq=4315, Time=3425971137
62	0.35422100	192.168.1.7	200.124.225.201	RTP	179	PT=DynamicRTP-Type-116, SSRC=0xCD30174A, Seq=23732, Time=2014048694, Mark
63	0.35936300	192.168.1.7	200.124.225.201	RTP	148	PT=DynamicRTP-Type-111, SSRC=0xD8AFAE22, Seq=4316, Time=3425972097
67	0.37908100	192.168.1.7	200.124.225.201	RTP	149	PT=DynamicRTP-Type-111, SSRC=0xD8AFAE22, Seq=4317, Time=3425973057
69	0.39929600	192.168.1.7	200.124.225.201	RTP	147	PT=DynamicRTP-Type-111, SSRC=0xD8AFAE22, Seq=4318, Time=3425974017
71	0.41888100	192.168.1.7	200.124.225.201	RTP	144	PT=DynamicRTP-Type-111, SSRC=0xD8AFAE22, Seq=4319, Time=3425974977

Figura D.8. Flujo de paquetes RTP (Autor de la Tesis 2015)

9.- Paquete RTP

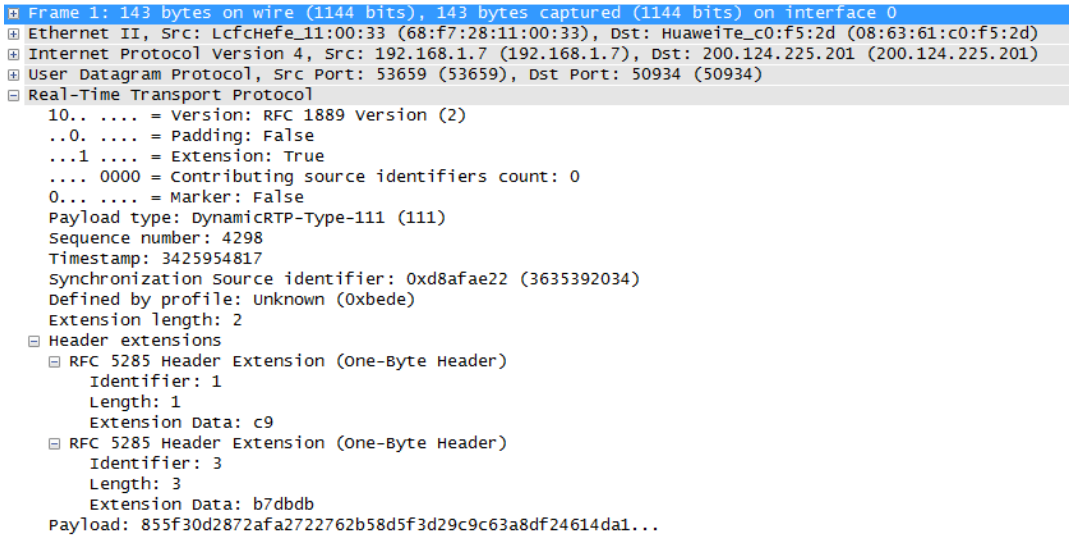


Figura D.9. Paquete RTP (Autor de la Tesis 2015)

10.- Captura de paquete RTCP, encapsulado en UDP

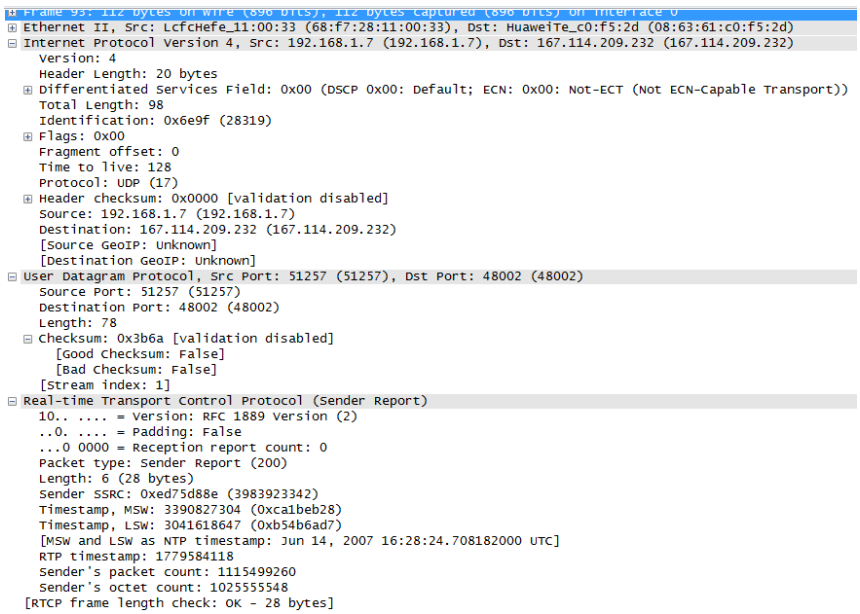


Figura D.10. Paquete RTCP (Autor de la Tesis 2015)

11.- Estadísticas de los protocolos

Protocol	% Packets	Packets	% Bytes	Bytes	Mbit/s	End Packets	End Bytes	End Mbit/s
Frame	100,00 %	313583	100,00 %	50301749	0,226	0	0	0,000
Ethernet	100,00 %	313583	100,00 %	50301749	0,226	0	0	0,000
Internet Protocol Version 4	99,76 %	312835	99,83 %	50218399	0,226	0	0	0,000
User Datagram Protocol	97,12 %	304554	91,96 %	46258119	0,208	5	346	0,000
Real-Time Transport Protocol	39,86 %	124986	45,10 %	22683613	0,102	120982	18693428	0,084
Data	43,44 %	136209	35,80 %	18010304	0,081	136209	18010304	0,081
QUIC (Quick UDP Internet Connections)	2,45 %	7682	1,96 %	984572	0,004	7682	984572	0,004
Session Traversal Utilities for NAT	10,50 %	32913	8,39 %	4218437	0,019	32912	4218364	0,019
Data	0,00 %	1	0,00 %	73	0,000	1	73	0,000
Real-time Transport Control Protocol	0,72 %	2257	0,50 %	252047	0,001	2063	230472	0,001
Domain Name Service	0,11 %	331	0,11 %	54378	0,000	331	54378	0,000
Malformed Packet	0,01 %	29	0,00 %	1740	0,000	29	1740	0,000
NetBIOS Datagram Service	0,00 %	4	0,00 %	980	0,000	0	0	0,000
Bootstrap Protocol	0,00 %	12	0,01 %	4050	0,000	12	4050	0,000
Hypertext Transfer Protocol	0,04 %	112	0,09 %	44220	0,000	112	44220	0,000
NetBIOS Name Service	0,00 %	9	0,00 %	828	0,000	9	828	0,000
Datagram Transport Layer Security	0,00 %	5	0,01 %	2604	0,000	5	2604	0,000
Transmission Control Protocol	2,60 %	8152	7,85 %	3949709	0,018	5539	2696129	0,012
Secure Sockets Layer	0,57 %	1779	2,20 %	1106906	0,005	1656	977629	0,004
Secure Sockets Layer	0,04 %	123	0,26 %	129277	0,001	123	129277	0,001
Hypertext Transfer Protocol	0,05 %	146	0,17 %	85786	0,000	87	48681	0,000
Compuserve GIF	0,01 %	21	0,02 %	12491	0,000	21	12491	0,000
Line-based text data	0,01 %	22	0,03 %	14037	0,000	22	14037	0,000
Media Type	0,00 %	8	0,01 %	5873	0,000	8	5873	0,000
JPEG File Interchange Format	0,00 %	2	0,00 %	1883	0,000	2	1883	0,000
Portable Network Graphics	0,00 %	1	0,00 %	676	0,000	1	676	0,000
eXtensible Markup Language	0,00 %	4	0,00 %	829	0,000	4	829	0,000
JavaScript Object Notation	0,00 %	1	0,00 %	1316	0,000	0	0	0,000

Figura D.11. Estadísticas de los protocolos (Autor de la Tesis 2015)