

Estudio e Implementación de aplicaciones soportadas en WebRTC que permitan mejorar la Calidad de Experiencia (QoE) del Audio sobre IP (AoIP)

Omar Augusto Rincón Rojas
Maestría en Redes y Sistemas de
Comunicación
Universidad Santo Tomás, Colombia
omar.rincon@vorticestore.com

Oscar Enrique Hurtado Camacho
Vortice Store
Bucaramanga, Colombia
ingenieria@vorticestore.com

Tito Raúl Vargas Hernández
Maestría en Redes y Sistemas de
Comunicación
Universidad Santo Tomás, Colombia
tivarher@ustabuca.edu.co

Abstract— WebRTC is a new project for the standardization of the real time communication of video, audio and data for the web browsers and mobile applications, without the use of plug-ins or any additional software. The quality of the audio streaming through the use of WebRTC is an important aspect to be measured, taking in consideration the services and applications it allows to create. For that reason, this work presents a study of the quality of service and experience of the audio streaming utilizing WebRTC using a developed application prototype which allows to set an audio codec from Opus, iSAC and G.722 to be used in the streaming. The results of the study provide information about the performance of the audio streaming through WebRTC in a local network, internet, and with limited bandwidth for each of the considered audio codecs, and an assessment of the quality of experience based on the recommendations of the ITU-T standard using the measurement of the *Mean Opinion Score* (MOS), that shows that the Opus audio codec is the preferred codec for the audio quality it provides, and besides it requires an average use of bandwidth in comparison with the rest of the codecs considered.

Keywords— *WebRTC, audio streaming, quality of experience.*

Resumen— WebRTC es un nuevo proyecto para estandarizar la comunicación en tiempo de real de audio, video y datos de forma nativa en los navegadores web y aplicaciones móviles, sin el uso de programas adicionales. La medición de la calidad de la transmisión de audio bajo la tecnología que WebRTC ofrece es de importancia de cara a los servicios y aplicaciones que permite crear. Es por esto que en este trabajo se presenta un estudio de la evaluación de la calidad del servicio y experiencia de la transferencia de audio mediante WebRTC por medio de un prototipo de aplicación desarrollado que permite establecer el códec a utilizarse en la transmisión de entre Opus, iSAC y G.722. Se obtuvo información acerca del desempeño en una red local, internet y de ancho de banda limitado de la transmisión de audio mediante WebRTC para cada uno de los codecs y una evaluación de la calidad de la experiencia mediante la medición del *Mean Opinion Score* (MOS) según la norma ITU-T, en donde se evidencia que el códec Opus es el más preferido por la calidad que ofrece y además requiere un uso de ancho de banda promedio en la relación a los demás codecs considerados.

Palabras claves—*WebRTC, streaming de audio, calidad de la experiencia.*

I. INTRODUCCIÓN

El desarrollo de Internet y su suite de protocolos ha permitido la creación de aplicaciones que conectan computadores en cualquier lugar del mundo, y en donde la demanda de información y contenido requiere de respuestas inmediatas. Los servicios que permiten la participación, colaboración, y comunicación en tiempo real son una necesidad de los consumidores y obligan a los proveedores de soluciones tecnológicas ofrecer servicios de calidad e impulsar el desarrollo de nuevos estándares y un uso eficiente de las tecnologías de red ya implementadas.

Como respuesta a esta demanda, específicamente de contenidos de audio y video, han surgido tecnologías como el Audio sobre IP y WebRTC.

Audio sobre IP permite la distribución de audio digital utilizando la misma infraestructura ya existente de redes IP para Internet y para voz sobre IP (VoIP) [1], [2]. El audio sobre IP ha sido diseñado para proporcionar una alta calidad de audio y tiene aplicaciones actualmente como refuerzo de voz, música de fondo y de primer plano e incluso por las estaciones de radio por Internet, sin embargo, audio sobre IP requiere más ancho de banda que VoIP, lo que lo hace inadecuado para aplicaciones donde el ancho de banda es limitado [3].

Por otra parte, WebRTC (Comunicación en tiempo real para la web), es un nuevo proyecto en desarrollo que tiene código abierto y está patrocinado por google y se basa en estándares abiertos determinados por el W3C (World Wide Web Consortium) y por el IETF (Internet Engineering Task Force), el cual permite estandarizar la comunicación en tiempo real vía audio, video o datos entre navegadores web y aplicaciones móviles de manera nativa mediante el uso de HTML5 y APIs de JavaScript sin necesidad de plug-ins [4].

WebRTC está diseñado para alto rendimiento, una comunicación de alta calidad de video, audio y datos arbitrarios. Las aplicaciones WebRTC necesitan un servicio a través del cual pueden intercambiar metadatos de la red y los medios de comunicación, un proceso conocido como la señalización. Sin embargo, una vez que ha tenido lugar la señalización, video, audio, datos, se transmiten directamente entre los clientes, evitando el uso de un servidor intermediario [5].

Así, WebRTC suministra comunicación entre el usuario final, es decir el usuario que ejecuta el navegador, y el servidor donde se encuentra alojado el servicio web, transmite la información que permite comenzar una comunicación del cliente con el servidor, facilitando la conexión también con otros clientes que estén accediendo a este mismo servicio web [6].

Debido a la naturaleza de las aplicaciones WebRTC, en tiempo real o interactivas, los códecs que utiliza están específicamente diseñados para estos propósitos, es decir, deben tener latencias bajas y cómputo operacional pequeño. Los códecs de audio que utiliza actualmente WebRTC para audio son Opus, iSAC, iLBC y G.722, sin embargo los navegadores web por ahora solo implementan, según las especificaciones definidas en la RFC 7874, Opus con frecuencia de muestreo de 48 kHz, iSAC con frecuencias de muestreo de 16 y 32 kHz, y G.722.

Hasta ahora, todas las plataformas disponibles de comunicación en tiempo real que incluían capacidad para flujos más allá del texto (i.e. audio, video y datos) estaban basadas en clientes dedicados, o en plug-ins (generalmente java o flash) para el navegador, como es el caso de, Skype, entre otros. Sin embargo, las aplicaciones web basadas en WebRTC no requieren de ninguno de esos complementos, por lo que se podría decir que no existe ninguna otra plataforma que sustituya la utilidad proporcionada por el nuevo estándar propuesto por WebRTC. Es por esta razón que existe una oportunidad importante para establecer información y datos de la calidad del servicio de la arquitectura web basada en WebRTC a pesar de su etapa temprana de desarrollo. Esta información permite a las empresas del sector de comunicaciones, que ofrezcan comunicaciones en tiempo real, de mejorar la experiencia que brindan a sus clientes en la red, permitiendo ofrecer un servicio que satisfaga a los consumidores.

En la literatura existe un interés por obtener información acerca de la calidad de contenidos audiovisuales transferidos usando WebRTC y su impacto en la calidad de la experiencia (QoE) [7], [8], incluso de herramientas que permitan una evaluación automatizada de la calidad de contenido audiovisual en aplicaciones desarrolladas mediante WebRTC [9]. Sin embargo, no existe en la literatura un estudio detallado de la calidad del audio en aplicaciones que usan WebRTC.

Es por todo lo anterior que en este trabajo se presenta un estudio del manejo de audio sobre IP empleando WebRTC que permita mejorar la calidad de experiencia por medio del uso de un prototipo de aplicación. Se presenta una descripción detallada del prototipo de aplicación desarrollado para las pruebas y de la topología de pruebas implementada. Se presentan además los resultados de las pruebas y conclusiones del estudio y el trabajo futuro.

II. METODOLOGÍA

En esta sección se presenta la metodología utilizada para hacer el estudio de audio sobre IP empleando WebRTC mediante un prototipo de aplicación. Teniendo en cuenta la arquitectura y los protocolos utilizados para la comunicación entre pares mediante WebRTC, se han analizado los recursos

necesarios para generar y estudiar tráfico de audio sobre IP generado y transmitido mediante WebRTC.

Así, se ha construido el prototipo de aplicación para pruebas que permite la transmisión de audio entre dos clientes mediante WebRTC y permite cambiar antes de la comunicación el códec de audio a utilizar por el navegador web.

WebRTC permite el acceso a el audio en tiempo de real de los dispositivos conectados de los clientes, a las aplicaciones construidas con su API para su transmisión. Sin embargo, se necesita de un mecanismo para coordinar la transmisión y enviar mensajes de control en un proceso conocido como señalización.

Los métodos y protocolos para llevar a cabo el proceso de señalización no son parte de la especificación de WebRTC ya que se considera que es parte de la aplicación.

En el prototipo de aplicación creado el proceso de señalización fue desarrollado con la librería para Javascript Socket.io que usa principalmente el protocolo WebSocket y permite el intercambio de mensajes de control de sesión, configuración de red de ambos clientes, y de negociación de formato de medios a utilizar en la sesión. La página web del prototipo de la aplicación es ofrecida usando el servidor web Nginx.

En la Figura 1 se describe la topología de pruebas del prototipo de aplicación creado que se denomina el modelo triangular de WebRTC [10], el cual se compone de un sitio ofrecido por un servidor web y de señalización que por medio de WebRTC permite la comunicación entre 2 clientes por medio de una red entre pares (P2P, por sus siglas en inglés).

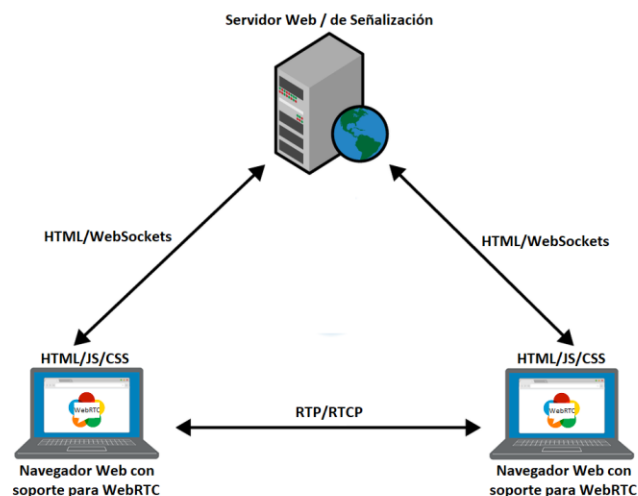


Figura 1. Topología de pruebas con el prototipo realizado.

El prototipo de aplicación permite, mediante la manipulación de los mensajes del protocolo SDP, establecer la preferencia, antes de establecer una comunicación entre los clientes, del códec de audio a utilizar de entre Opus, iSAC y G.722, esto si están disponibles por parte de los navegadores involucrados. Además, permite visualizar que códec está activo en la comunicación, la dirección IP del otro par y su número de

puerto, aparte de otros datos estadísticos de la conexión proveídos por la API de WebRTC.

WebRTC mediante el uso de Interactive Connectivity Establishment (ICE) determina la mejor ruta para intercambiar los paquetes de la transmisión. ICE se basa en los protocolos STUN y TURN para esta tarea. En el caso del prototipo de aplicación servidores Session Traversal Utilities for NAT (STUN) de libre uso de Google para resolver el potencial problema de NAT traversal y permitir el descubrimiento entre los clientes, con el fin de establecer la comunicación. En la Figura 2 se puede apreciar un diagrama funcional de la arquitectura del prototipo realizado.

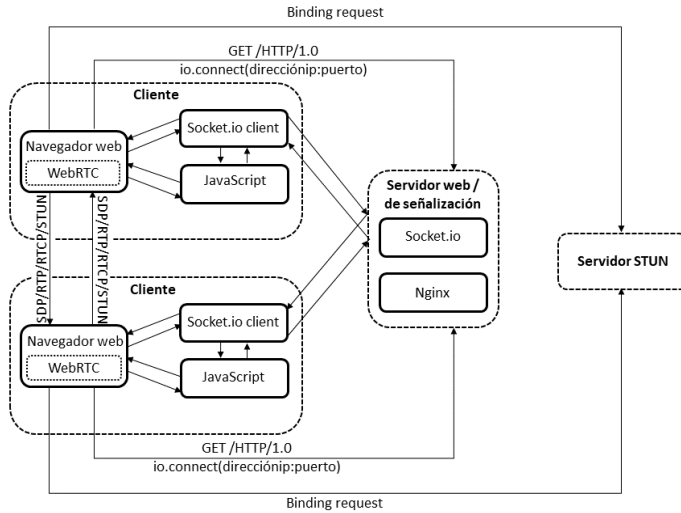


Figura 2. Diagrama funcional de la arquitectura del prototipo desarrollado.

Además, para generar un entorno real de interconexión entre los clientes y los servidores, como lo sería Internet, para simular momentos de congestión, cambios de ancho de banda disponible, y provocar pérdidas de paquetes que afecten la calidad del servicio, se ha incluido a la topología la herramienta Dummynet [11].

Esta herramienta se instala y se activa un proceso en los clientes que permite asignar canales de transmisión del tráfico generado, en este caso el tráfico de WebRTC entre los clientes, con la particularidad de tener la posibilidad de controlar el ancho de banda del canal asignado para la transmisión. Utilizando la anterior topología descrita en la Figura 1, se han realizado pruebas de transmisión de audio en tiempo real, las pruebas realizadas y sus resultados son presentados en la siguiente sección.

III. RESULTADOS

A continuación, se presentan resultados del estudio de la transferencia de audio sobre IP empleando WebRTC. Se utilizó el navegador web Google Chrome para las pruebas del prototipo desarrollado. Se realizaron mediciones de la calidad de servicio (QoS) como el retardo promedio, jitter, paquetes perdidos y tasa de transferencia efectiva (en inglés throughput), y una medición de manera subjetiva en términos del Mean Opinion Score (MOS), para así analizar la relación que existe

entre la calidad de experiencia del usuario (QoE) y la calidad de servicio (QoS).

Cabe resaltar que el Mean Opinion Score (MOS) es una prueba subjetiva basada en las recomendaciones de la norma ITU-T, que permite medir la percepción de cada individuo en el rango de 1 a 5, donde 1 es la más baja calidad percibida, y 5 es la percepción de más alta.

Haciendo uso de la topología presentada en la sección anterior, se han configurado y puesto en marcha tres escenarios de transmisión de audio.

A. Escenario A

En una red local se realizaron capturas del tráfico de red en ambos clientes para transmisiones de audio en tiempo real con cada uno de los codecs considerados. Las duraciones de las transmisiones fueron de 5 minutos. En la Figura 3 se puede observar el escenario considerado.

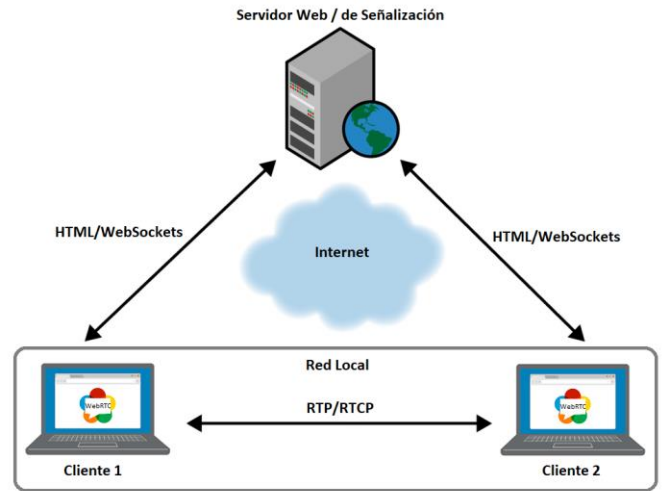


Figura 3. Topología de pruebas del escenario A.

Se presentan a continuación en las Tablas I y II los aspectos de calidad de servicio observados para los dos clientes involucrados en una transmisión de audio por medio de la aplicación desarrollada de audio sobre IP con WebRTC.

TABLA I. Aspectos de calidad de servicio en una red local para el cliente 1.

CLIENTE 1				
	Opus	iSAC 16 kHz	iSAC 32 kHz	G.722
Retardo promedio (ms)	3,905	3,195	5,638	3,998
Paquetes perdidos	0,02%	0%	0%	0,026%
Jitter (μs)	42	62	51	45
Throughput (bytes/s)	7105,7	3387,6	4359,3	11297,5

TABLA II. Aspectos de calidad de servicio en una red local para el cliente 2.

CLIENTE 2				
	Opus	iSAC 16 kHz	iSAC 32 kHz	G.722
Retardo promedio (ms)	5,460	4,075	3,267	2,497
Paquetes perdidos	0,046%	0,039%	0,108%	0,020%
Jitter (μs)	35	96	6	34
Throughput (bytes/s)	7361,2	5882,4	5663,3	11298,3

Se puede observar en este escenario un buen desempeño y calidad de la conexión con un retardo promedio de paquetes inferior a 6ms y pérdida de paquetes inferior al 0.05%. Además, se puede observar los valores de throughput típicos de cada uno de los códec utilizados en la transmisión de audio.

B. Escenario B

En la red local se realizó una limitación del ancho de banda en uno de los clientes con la herramienta Dummynet, para luego realizar transmisiones de audio en tiempo real con cada uno de los codecs considerados. Esto con el propósito de conocer el efecto en la transmisión de audio de una conexión con ancho de banda limitado.

El ancho de banda se limitó en el rango de 8 a 128 Kilobit/s incrementando el ancho de banda de a 8 Kilobits/s para cada nueva transmisión con cada códec utilizado. En la Figura 4 se puede observar el escenario considerado.

Se observó que al limitar la transmisión por debajo de un valor de ancho de banda específico para cada códec la transmisión de audio se detiene por parte del navegador web.

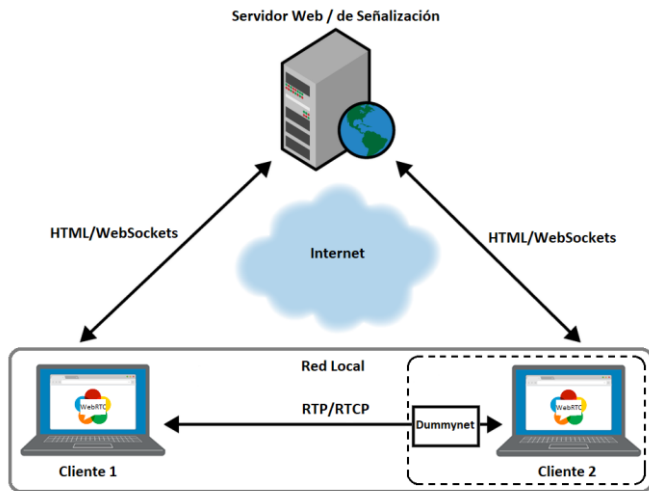


Figura 4. Topología de pruebas del escenario B.

En las Tablas III y IV se observan las pruebas de transmisión de audio en las que el ancho de banda no es suficiente para mantener que la transmisión de audio se realice por un tiempo considerable, y en donde el cliente 2 realiza la

limitación de ancho de banda. Para valores menores de ancho de banda a los considerados en las tablas la transmisión de audio no es posible para cada uno de los códec, y para valores mayores que los considerados la transmisión es satisfactoria. Las transmisiones de audio en la mayoría de los casos en las que no es satisfactoria dura menos de 10 segundos, pero excepcionalmente en otros casos la transmisión puede seguir con un retardo que se incrementa con el tiempo hasta que la transmisión se detiene.

TABLA III. Pruebas con limitación de ancho de banda del cliente 1.

CLIENTE 1						
Códec	Ancho de banda (Kbit/s)	Retardo promedio (s)	Paquetes perdidos	Jitter (μs)	Throughput (bytes/s)	Tiempo de transmisión (s)
Opus	56	0,160	0,068%	33	6822,2	146
	64	0,037	0,036%	37	6862,1	441
	72	0,024	0,295%	39	6876,9	1834
iSAC 16kHz	24	1,224	0,820%	21406	3230,0	2
	32	0,897	0,392%	86	4277,1	6
	40	0,039	0,029%	7	4220,4	1834
iSAC 32kHz	24	1,119	0,794%	72	3241,1	2
	32	0,635	0%	77	4269,0	7
	40	0,215	0,014%	74	4535,3	624
G.722	64	0,722	0,543%	38	8396,6	2
	72	0,667	0,350%	37	9464,0	5
	80	0,529	0,381%	34	10421,7	9

TABLA IV. Pruebas con limitación de ancho de banda del cliente 2.

CLIENTE 2						
Códec	Ancho de banda (Kbit/s)	Retardo promedio (s)	Paquetes perdidos	Jitter (μs)	Throughput (bytes/s)	Tiempo de transmisión (s)
Opus	56	0,018	0%	42	7209,7	146
	64	0,046	0,031%	40	7292,3	441
	72	0,032	0,093%	39	7297,9	1834
iSAC 16kHz	24	1,943	28,346%	1452	3111,7	2
	32	1,664	22,753%	233	3977,8	6
	40	1,360	5,236%	81	5071,7	1834
iSAC 32kHz	24	2,243	36,829%	397	3070,4	2
	32	1,634	21,575%	219	4012,6	7
	40	1,361	8,370%	80	5078,6	624
G.722	64	0,999	20,388%	79	8112,4	2
	72	0,831	13,219%	61	9464,0	5
	80	0,662	5,177%	54	10162,6	9

A partir esta información se puede observar que la implementación actual de WebRTC para transmisión de audio no realiza una adaptación al ancho de banda disponible por parte del cliente, además de que los requerimientos de ancho de banda varían para cada códec, Opus requiere aproximadamente de al menos 80 Kbps para funcionar bien, iSAC de 48 Kbps aproximadamente y G.722 de 88 Kbps aproximadamente. Esto coloca a G.722 como el códec utilizado que más ancho de banda requiere para funcionar satisfactoriamente.

C. Escenario C

Se realizaron transmisiones de audio en tiempo real en internet con cada uno de los códec. Para esto se contó con doce (12) voluntarios los cuales usaron cada uno de los códec para conversar con otro de los voluntarios por una duración aproximada de 2 minutos y al final de la transmisión midieron el MOS de la misma, además se realizaron capturas del tráfico de red en ambos clientes para medir la calidad de servicio. En la Figura 5 se puede observar el escenario considerado.

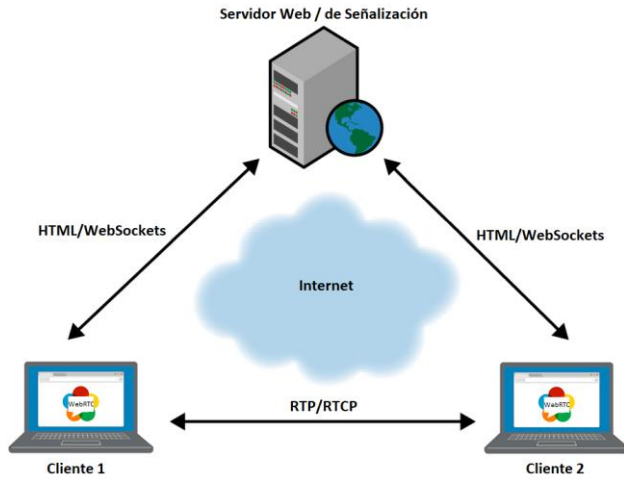


Figura 5. Topología de pruebas del escenario C.

En la Tabla V se presentan los datos promedio de la calidad de servicio observada en este escenario para las transmisiones de audio efectuadas por cada uno de los voluntarios.

TABLA V. Aspectos de calidad de servicio para pruebas en internet.

Número de usuarios	Códec	Retardo promedio (ms)	Paquetes perdidos promedio	Jitter promedio (µs)	Throughput promedio (bytes/s)
12	Opus	32,38	0,128%	33	7129,8
	iSAC 16 kHz	32,95	0,380%	67	5529,3
	iSAC 32 kHz	34,64	0,862%	67	5486,4
	G.722	33,75	0,394%	51	11258,1

Además en la Figura 6 se puede observar el comportamiento de las evaluaciones del MOS para cada una de las transmisiones de audio con los codecs evaluados.

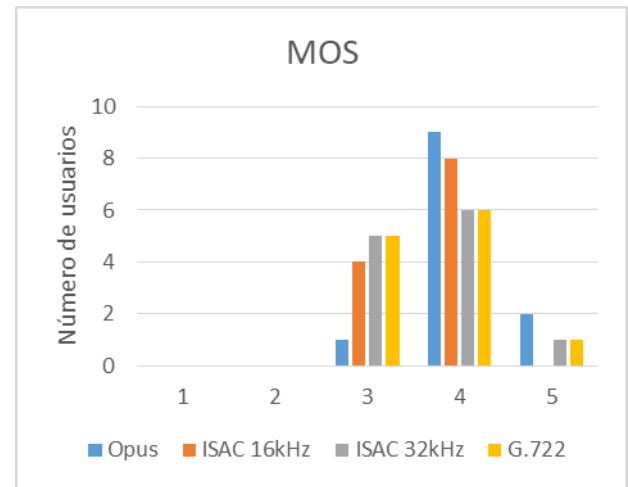


Figura 6. Mean Opinion Score para cada uno de los códec.

Se puede observar que el códec mejor evaluado de los considerados es el Opus y el peor evaluado el iSAC de 16 kHz teniendo en cuenta que los usuarios expresaron la aparición de ecos en la transmisión con dicho códec.

IV. CONCLUSIONES

Este trabajo permitió analizar múltiples escenarios de transmisión de audio empleando WebRTC. El uso de usuarios permitió mediante el uso de las pruebas MOS según la norma ITU-T vislumbrar la preferencia por el códec Opus. La limitación de ancho de banda para la transmisión de audio permitió observar que WebRTC tiene el potencial para desempeñarse bien en redes de área local y corporativas, en donde el ancho de banda disponible supera por mucho los requerimientos mínimos de los codecs para poder funcionar satisfactoriamente, sin embargo, en internet es posible que se presenten momentos de congestión en la red que puedan ocasionar una detención de la transmisión de audio. Estas situaciones deben ser manejadas por aplicación haciendo uso de la API de WebRTC para detectar una degradación de la conexión y así responder de acuerdo.

Es por esto que como trabajo futuro se tiene la implementación de un prototipo que permita probar las estrategias para manejo de la degradación de la conexión en la transmisión de audio mediante WebRTC. A su vez de una transmisión de archivos de audio por los canales de audio de WebRTC, y la grabación en un archivo de audio de la transmisión de audio recibida remotamente que permita una evaluación con referencia de la calidad de la experiencia.

AGRADECIMIENTOS

Este trabajo fue desarrollado gracias al apoyo de la Universidad Santo Tomás, el programa de Maestría en Redes y Sistemas de Comunicación y la empresa Vortice Store.

REFERENCIAS

- [1] S. Church y S. Pizzi, *Audio Over IP: Building Pro AoIP Systems with Livewire*. Focal Press, 2009.

- [2] P. Stevens y M. Zemack, «Standardising Audio Contribution over IP Communications», *British Broadcasting Corporation*, oct. 2008.
- [3] «A guide to Audio Over IP», 2016. [En línea]. Disponible en: <https://shop.cie-group.com/blog/8/audio-over-i-p-aqip>. [Accedido: 03-jun-2018].
- [4] A. Bergkvist, D. C. Burnett, C. Jennings, A. Narayanan, y B. Aboba, «WebRTC 1.0: Real-time communication between browsers», *Working draft, W3C*, vol. 91, 2012.
- [5] J. V. Rodrigo Delgado, «Estudio de WebRTC y su implementación con J2EE/JavaEE», Universidad de Valladolid, 2016.
- [6] J. Nightingale, Q. Wang, C. Grecos, y S. Goma, «Modeling QoE for streamed H.265/HEVC content under adverse network conditions», en *5th IET International Conference on Wireless, Mobile and Multimedia Networks (ICWMMN 2013)*, 2013, pp. 249-253.
- [7] D. Vučić, L. Skorin-Kapov, y M. Sužnjević, «The impact of bandwidth limitations and video resolution size on QoE for WebRTC-based mobile multi-party video conferencing», presentado en PQS 2016 5th ISCA/DEGA Workshop on Perceptual Quality of Systems, Berlin, Germany, pp. 59-63.
- [8] K. D. Moor, S. Arndt, D. Ammar, J. N. Voigt-Antons, A. Perkis, y P. E. Heegaard, «Exploring diverse measures for evaluating QoE in the context of WebRTC», en *2017 Ninth International Conference on Quality of Multimedia Experience (QoMEX)*, 2017, pp. 1-3.
- [9] B. Garcia, L. Lopez-Fernandez, F. Gortazar, y M. Gallego, «Analysis of Video Quality and End-to-End Latency in WebRTC», en *2016 IEEE Globecom Workshops (GC Wkshps)*, 2016, pp. 1-6.
- [10] S. Loreto y S. Romano, *Real-Time Communication with WebRTC: Peer-to-Peer in the Browser*. O'Reilly Media, 2014.
- [11] M. Carbone y L. Rizzo, «Dummynet Revisited», *SIGCOMM Comput. Commun. Rev.*, vol. 40, n.º 2, pp. 12–20, abr. 2010.