

Diseño de un sistema de comunicaciones en tiempo real en la web y su escalabilidad en la nube para consultas y seguimiento médico

José Vargas-Huamán, Kevin Quispe-Huaman, Eduardo Sutta-Gonzales,
Amarilis Tipo-Parillo, Pedro Yanque-Churo, José Sulla-Torres

Escuela Profesional de Ingeniería de Sistemas,
Universidad Nacional de San Agustín, Arequipa,
Perú

{josemvargh, kevinxpt27, laloeasg, amarilis.sussan, yanque.sis}@gmail.com,
jsulla@unsa.edu.pe

Resumen. En este artículo, se elabora un sistema de comunicaciones en tiempo real en la web con infraestructura en PubNub para las consultas y seguimiento médico por videoconferencias. El modelo se compone de dos etapas, la primera de ella es la generación de la comunicación por WebRTC y la segunda consiste en la configuración de la IaaS (Infrastructure as a Service) de PubNub y su puesta en marcha. Para evaluar el rendimiento de la aplicación se realizaron pruebas sobre usuarios en entornos de atención, obteniendo como resultado una confiabilidad del 80% sobre cinco sesiones de dos participantes cada uno. Los resultados muestran que la posibilidad de comunicarse entre los usuarios paciente-doctor fortalecen y facilitan el seguimiento de un tratamiento.

Palabras clave: WebRTC, IaaS, WebSocket, videoconferencia, PubNub.

Design of a Real-Time Communication System on the Web and its Scalability in the Cloud for Consultations and Medical Follow-Up

Abstract. In this paper, a real-time web communications system with PubNub infrastructure for consultation and medical follow-up by videoconferencing is developed. The model consists of two stages, the first of which is the generation of communication by WebRTC and the second consists of the configuration of the IaaS (Infrastructure as a Service), of PubNub and its implementation. To evaluate application performance, users were tested in care settings, resulting in 80% reliability over five sessions of two participants each. The results show that

the possibility of communicating between patient-doctor users strengthens and facilitates the follow-up of a treatment.

Keywords: WebRTC, IaaS, WebSocket, videoconference, PubNub.

1. Introducción

En los países en vía de desarrollo existe el problema relacionados con el cumplimiento o seguimiento por parte de los pacientes al tratamiento médico asignado [1]. El sistema habitual de los servicios de atención médica es a través de una interconsulta médica [2], que es cerca de 65,9 millones de atenciones por el seguro integral de salud de Perú [3]. La adecuada y oportuna atención del paciente es un factor determinante para su desarrollo [4], por lo que es sumamente importante su seguimiento y monitorización [5].

Una alternativa son los sistemas de videoconferencia, pero estos a menudo son demasiados costosos de comprar y mantener [6]. La evolución de la web nos presenta innovaciones sorprendentes y nuevas creaciones, cosa que ni siquiera imaginábamos o de plano no pensamos iban a existir. [7] La Comunicación en tiempo real para la web o WebRTC es una nueva tecnología que abre más posibilidades en nuestras aplicaciones. El desarrollo tecnológico se va incorporando a diversas situaciones, como el de un doctor y su paciente en las que las personas no disponen de tiempo, movilidad, o tienen problemas crónicos que les impide el acercamiento a una consulta, permitiendo una nueva generación de aplicaciones de Tele-salud [8]. De esta manera, tanto las plataformas online como la videoconferencia hacen posible el control, el establecimiento de diagnóstico y sus pautas terapéuticas. Al acoplar las capacidades de comunicación en tiempo real de WebRTC y las ventajas que aporta la Web of Things (WoT), se introduce el diseño de una nueva arquitectura sanitaria flexible, con el fin de proponer diversos servicios de e-salud [9], como una plataforma *web responsive* de telemedicina con videoconferencia para el seguimiento de pacientes [10], así como para módulos de videoconferencia a través de WebRTC para una plataforma de telemedicina [11].

La llamada WebRTC [12], permite utilizar HTML5 y APIs de JavaScript para crear aplicaciones que nos permitan comunicarnos vía Audio o Video, la idea es que no se necesite instalar *plugins*, para poder utilizar la tecnología. WebRTC proporciona acceso mediante programación a un vídeo en directo desde la webcam del usuario. También permite realizar conexiones punto a punto para audio y video de manera eficiente, sin embargo, para la conexión se hará uso de una infraestructura como servicio o IaaS (Infrastructure as a Service) [13].

PubNub es una red global de transmisión de dato [14]. El producto principal de PubNub es una API de mensajería publicar/suscribir en tiempo real construida sobre su red de flujo de datos global, que está compuesta por una red replicada de al menos 14 centros de datos ubicados en América del Norte, América del Sur, Europa y Asia.

La red actualmente sirve a más de 300 millones de dispositivos y transmite más de 750 mil millones de mensajes por mes, por lo que resultaría una buena opción en la implementación de la herramienta.

Como objetivo de este artículo, se ha planteado un modelo de sistema que utiliza técnicas modernas para la comunicación en tiempo real como son WebRTC y PubNub, diseñado bajo una arquitectura orientada al Cloud que provee escalabilidad para las consultas y seguimiento médico, de esta manera podrá ser una alternativa más económica y adecuada para el seguimiento médico de pacientes.

El modelo consta de dos etapas diferenciadas la primera de ellas es la generación de la comunicación por WebRTC y la segunda consiste en la configuración de la IaaS y su puesta en marcha.

A continuación, se da a conocer el estado del arte de sistemas de comunicación en tiempo real y otros trabajos tomados en cuenta en la realización de este artículo. En segundo lugar, se detalla la metodología, así como las técnicas y herramientas utilizadas para la realización de la aplicación y su evaluación. Finalmente se muestran los resultados, las conclusiones a las que se llegaron y los trabajos futuros de la investigación.

2. Trabajos relacionados

Se presenta algunos trabajos en la misma rama que anteceden este artículo.

2.1. Servicio de videoconferencia basado en WebRTC para telesalud

Este artículo desarrollado por Jang-Jaccard [6], realiza un estudio sobre los sistemas de videoconferencia existentes que se utilizan a menudo en los servicios de telesalud donde se han criticado por varias razones: (a) a menudo son demasiado costosos de comprar y mantener, (b) usan tecnologías propietarias que son incompatibles entre sí, y (c) requieren personal de TI bastante capacitado para mantener el sistema. Por lo que existe la necesidad de un sistema de videoconferencia menos costoso, compatible y fácil de usar. La comunicación web en tiempo real (WebRTC), promete ofrecer una solución al permitir navegadores web con capacidades de comunicación en tiempo real a través de API de JavaScript simples. Utilizando WebRTC, los usuarios pueden realizar llamadas de audio y video para compartir los datos a través de navegadores web sin tener que comprar o descargar software adicional.

Aunque es prometedora la perspectiva de WebRTC, no ha habido muchos casos de aplicaciones de la vida real (en particular en telesalud), que utiliza WebRTC. Jang-Jaccard presenta una experiencia práctica en el diseño y la implementación de un sistema de videoconferencia para telesalud basado en WebRTC. El sistema de videoconferencia es parte de un gran proyecto de monitoreo a distancia que se lleva a cabo en seis ubicaciones en cinco estados diferentes de Australia.

Uno de los objetivos del proyecto es evaluar si los servicios de telesalud habilitados con un alto ancho de banda, que se brindan a través de la monitorización a distancia del hogar, pueden ser rentables y mejorar los resultados en la atención de salud. Sin embargo, en este documento se centra en el sistema de videoconferencia basado en WebRTC que permite reuniones en línea entre coordinadores de atención remotamente ubicados y los pacientes en su hogar.

2.2. Solución innovadora de WebRTC para servicios de e-Health

En el trabajo de Paola Pierloni [15], presenta un estudio sobre las soluciones y servicios para e-Health y la telemedicina en el área de la salud gracias a las últimas innovaciones en electrónica, informática y telecomunicaciones. Este trabajo propone un servicio innovador para la e-Health orientada a la máxima facilidad de uso y al intercambio de signos vitales. La propuesta consiste en un servicio de teleconferencia basado en la tecnología WebRTC que permite a cualquier persona que reside de forma remota del personal médico o del hospital interactuar directamente con ellos. La solución proporciona todas las funciones comunes de WebRTC, como secuencias de video y audio en tiempo real, mensajería instantánea y uso compartido de archivos con el único requisito de un navegador web tradicional.

Más allá de eso, se implementa la transmisión y visualización en tiempo real de signos vitales y parámetros adquiridos por sensores biomédicos conectados al dispositivo personal del paciente a través del RTCDataChannel. Actualmente, la solución implica la instalación de una extensión de navegador, pero esta operación es muy simple y puede evitarse cuando las API y navegadores WebRTC admitan secuencias de medios provenientes de sensores de la misma manera que las transmisiones de audio y video. La solución demuestra cómo las tecnologías web se pueden aplicar en el sector de la salud, proporcionando servicios muy eficaces a pacientes y usuarios que por diversas razones tienen dificultades para viajar a los hospitales con el fin de recibir asistencia médica.

2.3. WebRTC: entrega de telesalud en el navegador

Arin Sime [8], explica como WebRTC está habilitando una nueva generación de aplicaciones de Telesalud y será una parte importante del futuro de Telesalud. WebRTC permite que las aplicaciones web controlen el micrófono y la cámara de video del usuario desde el navegador. En este punto de vista, el autor presenta los pros y los contras de WebRTC para aplicaciones de telesalud.

La telesalud es una parte del cuidado de la salud en rápida expansión a nivel mundial, con el potencial de ahorrar costos y servir mejor a los pacientes con atención especializada, sin importar dónde viven.

Un desafío final a tener en cuenta con WebRTC es que puede ser difícil establecer una llamada con éxito detrás de algunas redes de hospitales. Si su hospital tiene una política de seguridad de red muy restrictiva, puede ser difícil que dos navegadores web que usan WebRTC establezcan la conexión P2P necesaria para realizar una llamada.

2.4. Telemedicina para la gestión de emergencias mediante WebRTC.

Con el rápido avance y desarrollo en el campo de las comunicaciones en tiempo real, la telemedicina ha alcanzado grandes alturas y ha ayudado a salvar millones de vidas durante situaciones de emergencia. [16] El uso de la telemedicina es de larga data, pero

su aplicación en la gestión de atención de emergencia está todavía en su etapa de desarrollo.

En la actualidad hay varios sistemas de telemedicina de emergencia disponibles en el mercado que utiliza la electrónica del vehículo hasta la fecha, la última tecnología de telecomunicaciones y software especializado. Sin embargo, estos sistemas son altamente sofisticados, voluminosos y caros y son empleados por muy pocos centros de salud. Por lo tanto, estos servicios salvavidas no están disponibles para gran parte de la población, especialmente los que viven en las zonas rurales. El objetivo de este trabajo es presentar una nueva idea en la que estos servicios *Tele Emergency* puedan ser implementados de una manera mucho más eficiente, económica y menos sofisticada, para que estos servicios puedan ser ampliamente proporcionados. Aquí proponemos una nueva aplicación de Telemedicina de emergencia para la gestión de atención de emergencia que utiliza WebRTC para la comunicación en tiempo real. Este sistema sólo requiere un dispositivo móvil con conexión a Internet con Chrome o Mozilla instalado en él. El dispositivo se transporta dentro de la ambulancia para llevar a cabo una evaluación inicial del paciente y luego se lleva al centro de salud más cercano donde se lleva a cabo el tratamiento adicional con la ayuda de especialistas cuya telepresencia es proporcionada por dispositivos habilitados para WebRTC.

2.5. Arquitectura de cuidado de la salud inteligente usando WebRTC y WoT

Saad El Jaouhari [9], presenta el diseño de una nueva arquitectura de salud flexible, para proponer diversos servicios de salud electrónica al unir las capacidades de comunicación en tiempo real de WebRTC y las ventajas traídas por la *Web of Things* (WoT), En este trabajo se centran principalmente en los servicios relacionados con la atención médica remota de pacientes y personas mayores. Presenta la arquitectura a través del análisis de tres casos de uso principales: un monitoreo remoto y continuo de personas mayores y un examen médico remoto de pacientes e intervención de emergencia en caso de un accidente.

2.6. Revisión a la aplicación de cloud computing en p2p video streaming

La computación en nube se ha introducido como una solución a varios problemas del tradicional sistema de e-learning basado en la web, tales como un almacenamiento limitado, un elevado coste de mantenimiento de la infraestructura y una baja interoperabilidad entre el componente de aprendizaje electrónico basado en un sistema web. Sin embargo, el rendimiento del sistema de computación en nube puede deteriorarse con el aumento del número de usuarios y empeorar cuando muchos usuarios acceden a la transmisión de video desde el sistema en la nube. Esto se debe a la arquitectura centralizada de la nube que puede generar congestión de tráfico de red y cuello de botella en los servidores de la nube.

Se ha propuesto la arquitectura *peer to peer* para superar este problema. Utilizando P2P todos los nodos del sistema de nube pueden actuar tanto como servidores como clientes al mismo tiempo para reducir la congestión y el cuello de botella del sistema.

En la actualidad, hay pocas revisiones en *streaming* de vídeo P2P, aunque se han realizado estudios intensivos sobre el desarrollo del sistema.

Con esta condición, la identificación y la comprensión del desarrollo de la transmisión de vídeo P2P será costosa, requiere mucho tiempo y físicamente agotador.

El objetivo de este documento es revisar el último desarrollo de *streaming* de vídeo P2P basado en *cloud computing*. Un método de revisión narrativa se ha utilizado como la metodología para investigar los artículos de *streaming* de vídeo P2P de 2009 a 2014. Los resultados de esta investigación muestran que el 90% de *e-learning* basado en la nube se integran con P2P cuando se trata de *streaming* de vídeo.

3. Metodología

3.1. Técnicas y herramientas

WebRTC. WebRTC [17] es un estándar en desarrollo por el Internet Engineering Task Force (IETF) y el World Wide Web Consortium (W3C) que pretende definir un framework, protocolos e interfaces de programación que proveerán comunicaciones interactivas y en tiempo real de audio, vídeo y datos a las aplicaciones en los navegadores web.

Los APIs de los que consta el protocolo son los siguientes:

- *getUserMedia*: se utiliza para acceder a los recursos multimedia del usuario como su cámara o su micrófono. El resultado será un *MediaStream*.
- *MediaStream*: es un conjunto de pistas de flujos de audio o vídeo con los recursos multimedia del usuario.
- *RTCPeerConnection*: representa un canal de conexión con otro cliente. Un cliente añade a un *RTCPeerConnection* uno o más *MediaStreams* para compartirlos con el otro cliente.

Cloud computing. Es un modelo para proveer acceso bajo demanda a recursos de computación como redes, servidores, almacenamiento, aplicaciones y servicios de forma configurable y a través de la red [18].

Las características del modelo de Cloud Computing son las siguientes:

- *On-demand self-service*: Los usuarios pueden disponer de capacidades de computación (CPU, ancho de banda, almacenamiento, etc.) según lo necesiten.
- *Broad network access*: Estas capacidades están disponibles en la red en diferentes posiciones y son servidas a través de mecanismos estándar.
- *Resource pooling*: El Cloud sigue un modelo *multi-tenant* por el cual pueden asignarse recursos a diferentes usuarios.

PubNub. Es una red global de transmisión de datos (DSN) y una compañía de infraestructura en tiempo real (IaaS), el producto principal de PubNub es una API de mensajería para publicar y/o suscribir en tiempo real, construida sobre su red de flujo de datos global.

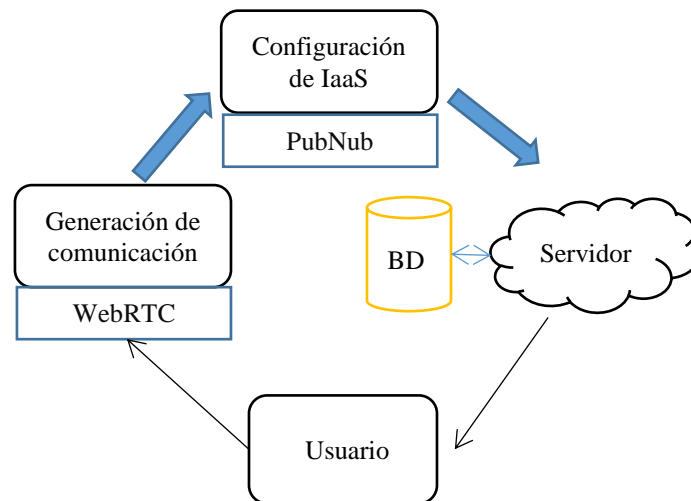


Fig. 1. Diseño propuesto de comunicación en tiempo real.

La mensajería proporciona *streaming* de datos en tiempo real y señalización de dispositivos, e incluye cifrado AES incorporado y cifrado TLS / SSL opcional.

Los componentes atómicos que componen un flujo de datos son Claves API, Mensajes y Canales.

WebSocket (BackEnd PubNub). La parte del desarrollo e implementación del WebSocket es una parte fundamental del proyecto para que los usuarios puedan comunicarse entre ellos y enviar y recibir información de forma transparente.

PubNub le ofrece el soporte completo de RFC 6455 para HTML5 WebSocket Client Specification. PubNub WebSockets permite que cualquier navegador (moderno o no) soporte las API estándar de HTML5 WebSocket.

3.2. Análisis del sistema

El sistema se ha desarrollado con la tecnología WebRTC ya que ofrece las herramientas necesarias y un API en lenguaje JavaScript fácil de utilizar y programar para este propósito. Las tecnologías utilizadas en la Sala Principal donde los usuarios se conectan y realizan llamadas entre ellos, se ha optado por utilizar en el servidor un *servlet* y un *websocket*, y en el lado del cliente JavaScript, HTML5 y CSS para renderizar la interfaz y enviar/recibir datos.

Los lenguajes de programación y tecnologías que utiliza esta aplicación son:

- En el lado del cliente (FrontEnd): HTML5, JavaScript (WebSocket), CSS, WebRTC.
- En el lado del servidor (BackEnd): Java, WebSocket, NodeJS, PubNub.
- En la base de datos: mongoDB.

3.3. Diseño del software

Modelo cliente-servidor. El sistema está basado en una arquitectura cliente-servidor, (véase la figura 1).

Los aspectos novedosos e importantes del diseño propuesto son la integración de las diferentes tecnologías que permitan la monitorización y consultas en los pacientes para una atención médica y que no existe en la mayoría de los servicios de salud en el Perú.

3.4. Implementación del proyecto

El desarrollo de la aplicación se ha realizado en MongoDB, ExpressJS, AngularJS, NodeJS. También es conocido como Stack MEAN, que nos permite el desarrollo de aplicaciones, y páginas web dinámicas, que están basadas cada una de estas en JavaScript. A esto se agregó las librerías WebRTC y PubNub, que también están desarrolladas en JavaScript. Gracias a esta característica, la integración de estas librerías y *frameworks*, el sistema resultó exitosamente auto-suficiente.

En la figura 1, se muestra el diagrama de funcionamiento del sistema. Que se compone de Usuario, servidor y base de datos. En la Base de datos se almacena información de suscripción del usuario, en la que se encripta los datos confidenciales del usuario, en este caso solo contraseña. El servicio de alojamiento de base de datos en la nube que hemos utilizado es MongoLab, que una vez creada la base de datos nos provee una cadena de conexión con nuestra base de datos.

En el Servidor que se muestra en la figura 1, se desarrolló la autenticación de usuarios. El Cliente (Navegador web) envía información de validación de usuario al servidor, este obtiene la información de la base de datos y envía 3 cadenas de caracteres separados por un punto, conocido como Json Web Token (JWT). Es por medio de esto que se realiza la seguridad en cliente/servidor. El servicio de alojamiento en la nube que hemos utilizado es Heroku, ya que nos provee de servicio de seguridad SSL para la ejecución de las librerías WebRTC y PubNub, además provee seguridad en la conexión con la base de datos, a través de la configuración de variables de entorno.

En el lado de Cliente (navegador), *AngularJS* se complementa perfectamente con WebRTC y PubNub, para realizar la conexión y comunicación peer-to-peer entre dos usuarios. Para Ello las librerías nos especifica crear una variable PHONE (teléfono), en la que pondremos: un número, que se será *username* de usuario; una clave pública y una clave de suscripción, que no provee PubNub; y por último se activa el certificado de seguridad SSL, sin esto no funciona la aplicación. Posteriormente se implementa las funciones de *phone.dial*, *phone.receive*, *session.connected*, entre otras.

De esta forma explicamos, de forma general, el funcionamiento del sistema. Hay muchas cosas más que se necesitan explicar sobre lo que realiza por dentro, en su código, PubNub y WebRTC, pero hay que entender una sola una cosa, que estas librerías nos ofrecen una abstracción para realizar una comunicación peer-to-peer. Que muy en el fondo estas librerías están haciendo uso de *websockets* para crear una red de comunicación punto a punto entre dos usuarios como se muestra en la figura 2.

Tabla 1. Parámetros configuración WebRTC

Componente	Parámetros	Descripción
HTML5	<video>	Para reproducir y capturar vídeo de la cámara y mostrarlo en pantalla.
	<canvas>	Para crear imágenes o renderizaciones dinámicas.
JavaScript	XMLHttpRequest	Para obtener información de una URL sin tener que recargar la página completa.
	JSON	Para intercambio de datos.
	onopen()	Método que se ejecuta cuando el websocket se abre por un cliente.
	send()	Método que envía un mensaje por el websocket.
	onmessage()	Método que se ejecuta cuando se recibe un mensaje
	onclose()	Método que se ejecuta cuando se cierra el websocket.
	onerror()	Método que se ejecuta cuando se produce un error en el websocket.
WebSocket PubNub	Publish / .Subscribe	PubNub proporciona una manera fácil de publicar datos en tiempo real, ya sea dispositivos individuales o grupos grandes de ellos.
	Presence	Da visibilidad en tiempo real de quién está suscrito en un "canal"
	Storage / Playback	Registra flujos de datos y proporciona acceso instantáneo y consultas en datos que se publicaron en el pasado
	Analytics	Obtiene opiniones en tiempo real y en tiempo real sobre el tráfico y el uso en tiempo real
	Security	Agrega cifrado AES a tiempo real, así como un marco de concesión / revocación completa para garantizar que sólo los usuarios autorizados puedan suscribirse

3.5. Configuración WebRTC y Pubnub

Al ejecutar los diferentes módulos que hacen falta para el funcionamiento de un servicio basado en WebRTC y PubNub y que se han explicado en la arquitectura son necesarios algunos parámetros de configuración. Estos parámetros son utilizados por los diferentes módulos para establecer algunas configuraciones en sus componentes.

En la Tabla 1, pueden observarse estos parámetros junto con una pequeña descripción de su funcionalidad.

4. Resultados y pruebas

A continuación, se explica un ejemplo de las pruebas del sistema realizado en el marco de esta investigación y utilizando la unidad de control en una máquina con características similares a la *t2 medium*, del proveedor Amazon EC2 (Procesador de 4 núcleos y 4gb de RAM). Se han diseñado tres escenarios que son los más comunes en

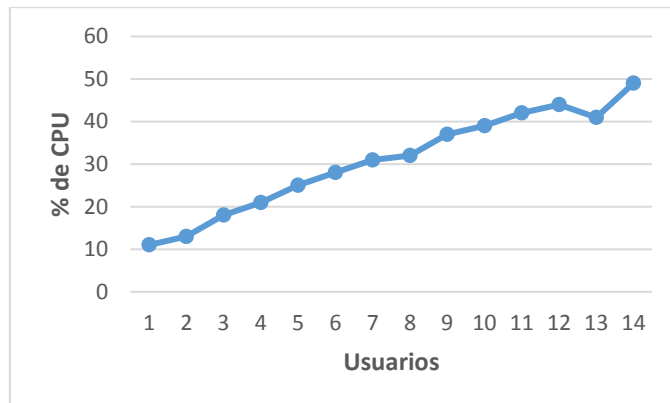


Fig. 2. Uso de CPU en Streaming.

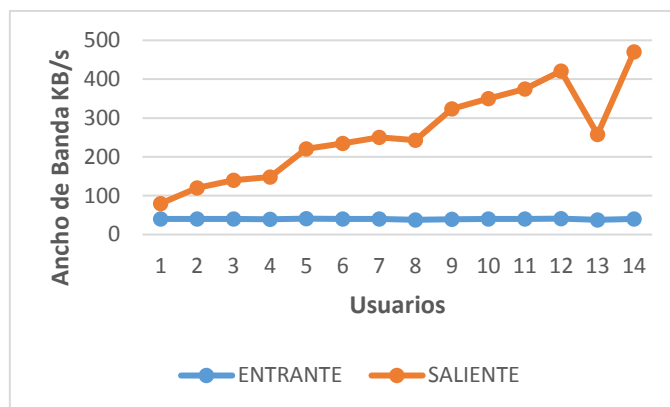


Fig. 3. Uso de ancho de banda en Streaming.

sistemas de videoconferencias: un *streaming* de vídeo en tiempo real, una videoconferencia multiusuario y múltiples sesiones de dos participantes cada uno (paciente-doctor). En los tres sistemas se ha realizado una monitorización del consumo de memoria y ancho de banda, así como del uso de ancho de banda entrante y saliente.

En el primer escenario, *streaming* en vivo, uno de los usuarios está publicando su flujo de audio y vídeo en la sesión y los clientes que se suscriben a él, van añadiéndose de forma gradual. En la Figura 2 podemos observar como el uso de CPU aumenta de forma lineal con el incremento de usuarios que se suscriben al *streaming*.

Esto ocurre porque el estándar WebRTC, como ya se ha explicado, utiliza SRTP para la transmisión de paquetes y por lo tanto se tiene que desproteger y volver a proteger los paquetes para realizar la transmisión desde el usuario que publica hacia los que se suscriben.

También podemos observar en la Figura 3 como el ancho de banda entrante es constante durante toda la sesión y el saliente aumenta también de forma lineal debido a



Fig. 4. Uso de CPU en Videoconferencia.

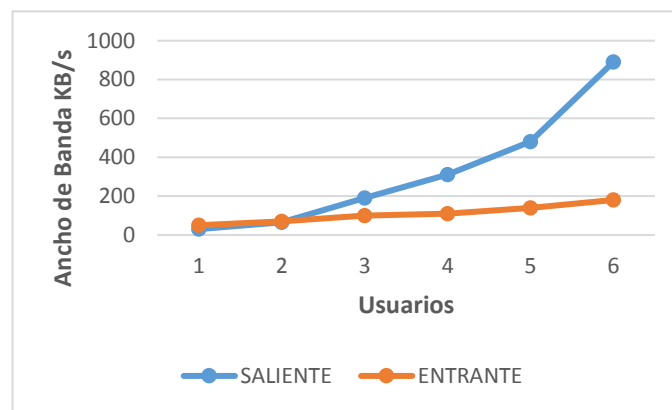


Fig. 5. Uso de ancho de banda en Videoconferencias.

que cada nuevo cliente conectado implica un nuevo flujo de salida. Acerca de la memoria utilizada aumenta también de forma lineal, pero con una variación mínima durante la sesión (de unos 10 MB). Finalmente pueden observarse pequeñas anomalías cuando se conectan más de 6 usuarios, pero probablemente sea debido a un error en el rendimiento del cliente que publica los datos.

En el segundo escenario, la videoconferencia multiusuario, cada usuario que se conecta a la sesión pública su flujo de audio y vídeo y además se suscribe al resto de usuarios que estaban conectados previamente. Se ha establecido un límite de seis usuarios ya que es el número que comúnmente se utiliza en este tipo de sesiones.

En la Figura 4 y la Figura 5 podemos observar como el consumo de ancho de banda entrante aumenta linealmente con el incremento de usuarios en la sesión debido al hecho de que cada nuevo usuario publica su flujo en el sistema. Sin embargo, el ancho de banda saliente y el consumo de CPU se incrementan de forma exponencial debido a

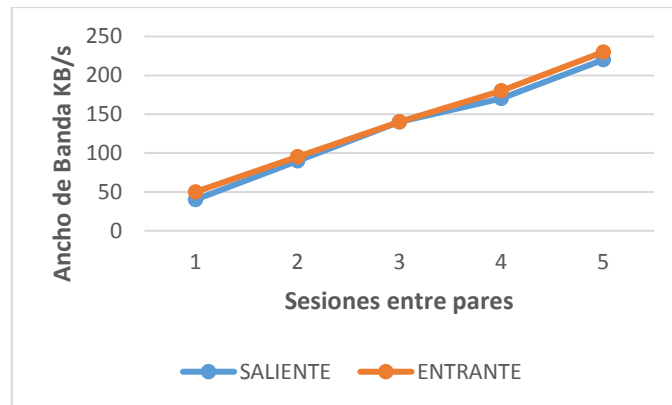


Fig. 6. Uso de ancho de banda en sesiones de pares (2 participantes).

que por cada nuevo usuario la unidad de control debe redirigir el nuevo flujo al resto de usuarios. Por lo tanto, el número de flujos de salida aumentará siguiendo la Ec. (1):

$$N = n(n - 1), \quad (1)$$

donde n es el número de usuarios en la sala. El uso de memoria en este escenario también aumenta de forma exponencial, pero de nuevo con una variación insignificante para este estudio.

En la Figura 6 podemos observar como el consumo de ancho de banda tanto entrante como saliente aumenta linealmente con el incremento de sesiones de pares de usuarios debido al hecho de que cada nueva sesión pública su flujo en el sistema.

Considerando los tres escenarios, del total de pruebas realizadas se obtienen resultados razonablemente precisos y aceptables que corresponden al 80% de las pruebas hechas en desfase y duración sin cortes para redes de confiabilidad medio-baja (inferior a los 300 KB/s). El orden de complejidad del sistema de control es lineal con respecto a la cantidad de componentes (usuarios), de la red. Para redes muy confiables (cliente estable en 500 KB/s aprox.), deben realizarse gran cantidad de sesiones sin presentar alteraciones, conservando resultados aceptables, por lo que el tiempo de ejecución aumenta en función de la confiabilidad de la red.

5. Conclusiones

En el trabajo Se ha diseñado una arquitectura del sistema orientada al *Cloud* que permite proveer servicios avanzados de comunicaciones en tiempo real en los navegadores web tales como videoconferencia entre muchos usuarios. Con el particular caso que incorporaba a dos participantes paciente-médico en el seguimiento de un tratamiento.

Para el caso del streaming, el resultado del uso de cpu y el ancho de banda utilizado han sido aceptables en la comunicación paciente-médico. Para el ancho de banda

entrante se mantiene lineal, mientras que la salida tiene unos picos de crecimiento aceptable.

Para el caso de la videoconferencia, el resultado del uso de cpu y el ancho de banda saliente tiene un crecimiento exponencial en la comunicación paciente-médico, debido al hecho de que cada nuevo usuario publica su flujo en el sistema.

Para una sesión de pares, el resultado del uso de cpu y el ancho de banda tiene incrementos lineales aceptables.

En base a los tres escenarios se ha obtenido un 80% de aceptabilidad por lo que resulta confiable en una comunicación paciente-médico.

Para futuros estudios, se deben tomar en consideración, para escenarios similares, el límite de número de usuarios soportados por el sistema, así como, que se debe de hacer para incrementar ese límite.

Referencias

1. Martín-Alfonso, L.: Repercusiones para la salud pública de la adherencia terapéutica deficiente. *Revista Cubana de Salud Pública*. 32(3) (2006)
2. Montero-Ruiz, E., López-Álvarez, J.: La interconsulta médica: problemas y soluciones. *Medicina Clínica*, 136, pp. 488–490 (2011)
3. Velásquez, A., Suarez, D., Nepo-Linares, E.: Health sector reform in Peru: Law, governance, universal coverage, and responses to health risks. *Revista peruana de medicina experimental y salud publica*, 33(3), pp. 546–555 (2016)
4. O' Shea-Cuevas, G., Rizzoli-Córdoba, A., Aceves-Villagrán, D., Villagrán-Muñoz, V., Carrasco-Mendoza, J., Halley-Castillo, E., Delgado-Ginebra, I., Pizarro-Castellanos, M., Vargas-López, G., Antillón-Ocampo, F., Villasís-Keever, M., Muñoz-Hernández, O.: Sistema de Protección Social en Salud para la detección y atención oportuna de problemas del desarrollo infantil en México. *Boletín Médico del Hospital Infantil de México*, 72, pp. 429–437 (2015)
5. Fernández-Lozano, I.; Toquero-Ramos, J., Castro-Urda, V., Marín, Alonso-Pulpón, L.: Monitorización remota: una visión crítica. *Cuadernos de Estimulación Cardiaca*, 4, pp. 37–42 (2011)
6. Jang-Jaccard, J., Nepal, S., Celler, B., Yan, B.: WebRTC-based video conferencing service for telehealth. *Computing*, 98, pp. 169–193 (2016)
7. González, A.: Diseño de un sistema de comunicaciones. Tesis Master, 73, (2015)
8. Sime, A.W.: WebRTC: delivering telehealth in the browser. *mHealth*, 98, pp. 169–193 (2016)
9. El-Jaouhari, S., Bouabdallah, A., Bonnin, J.M., Lemlouma, T.: Toward a Smart Health-care Architecture Using WebRTC and WoT. *World Conference on Information Systems and Technologies*, Springer, pp. 531–540 (2017)
10. Cáceres-Taladriz, C., Pérez-Silva, J., Chausa, P., León, A., García-Alcaide, F., Gómez-Aguilera, E.J.: Auditoría y mejoras en la seguridad de la aplicación Hospital VIHrtual: plataforma web responsive de telemedicina para el seguimiento de pacientes con VIH. *Libro de actas*, 109 (2015)
11. Malla, C., Patricio, D.: Desarrollo e implementación de un módulo de videoconferencia a través de webrtc para una plataforma de telemedicina rural. Universidad Politécnica de Madrid, Madrid, (2014)
12. Johnston, A., Burnett, D.: WebRTC. Digital Codex LLC, St. Louis, MO (2014)

13. Sotomayor, B., Montero, R., Llorente, I., Foster, I.: Virtual Infrastructure Management in Private and Hybrid Clouds. *IEEE Internet Computing*, 13, pp. 14–22 (2009)
14. Burnett, D.: WebRTC: Handling Media on the Web. *Multimodal Interaction with W3C Standards*, Springer Link, pp. 155–169 (2017)
15. Pierleoni, P., Pernini, L., Palma, L., Belli, A., Valenti, S., Maurizi, L., Sabbatini, L., Marroni, A.: An innovative webRTC solution for e-health services. *e-Health Networking, Applications and Services (Healthcom), IEEE 18th International Conference*, pp. 1–6 (2016)
16. Vidul, A., Hari, S., Pranave, K., Vysakh, K., Archana, K.: Telemedicine for emergency care management using WebRTC. *Advances in Computing, Communications and Informatics (ICACCI), International Conference*, pp. 1741–1745 (2015)
17. Bergkvist, A., Burnett, D.C., Jennings, C., Narayanan, A.: Webrtc 1.0: Real-time communication between browsers. Working draft, W3C. 91, (2012)
18. Mell, P., Grance, T.: The NIST definition of cloud computing. Computer Security Division. Information Technology Laboratory, National Institute of Standards and Technology Gaithersburg (2011)