

## Evaluación y comparación de códecs de video para el despliegue de un sistema de comunicación resiliente

### Evaluation and comparison of video codecs for deploying a resilient communication system

Paúl Astudillo<sup>1</sup> <https://orcid.org/0000-0001-8380-6103>,  
Christian Quinde<sup>1</sup> <https://orcid.org/0000-0002-9863-6941>, Santiago González<sup>1</sup> <https://orcid.org/0000-0001-6604-889X>, Iván Palacios<sup>1</sup> <https://orcid.org/0000-0002-3894-3341>

<sup>1</sup>*Departamento de Eléctrica, Electrónica y Telecomunicaciones, Universidad de Cuenca, Cuenca, Ecuador*

[paul.astudillo@ucuenca.edu.ec](mailto:paul.astudillo@ucuenca.edu.ec),  
[christian.quinde@ucuenca.edu.ec](mailto:christian.quinde@ucuenca.edu.ec),  
[santiago.gonzalez@ucuenca.edu.ec](mailto:santiago.gonzalez@ucuenca.edu.ec),  
[ivan.palacios@ucuenca.edu.ec](mailto:ivan.palacios@ucuenca.edu.ec)



Esta obra está bajo una licencia internacional  
Creative Commons Atribución-NoComercial 4.0.

Enviado: 2022/05/13

Aceptado: 2022/09/19

Publicado: 2022/11/30

#### Resumen

En escenarios de emergencia usualmente es difícil contar con una infraestructura de red, especialmente en lugares remotos o geográficamente complicados por el difícil acceso. Sin embargo, es necesario disponer de herramientas para comunicarse o realizar una valoración de la situación. En este sentido, las redes ad hoc surgen como una opción para solucionar esta problemática. En el presente artículo se propone el desarrollo de un sistema de comunicación de emergencia para la transmisión de tráfico multimedia en un escenario real, entre los diferentes pisos de un edificio. Para tal objetivo, se realizaron múltiples experimentos para determinar los valores de métricas fundamentales (v.g. *packet loss*, *delay* y *throughput*) para la transmisión de audio y video en tiempo real. En concreto, se desarrollaron herramientas de código abierto con capacidad de extraer las métricas requeridas y comparar de forma objetiva la calidad del video empleando mecanismos de codificación actuales (v.g. VP8, VP9, H.264, H.265). Los resultados obtenidos muestran que VP8 es el códec más adecuado para el escenario propuesto. Además, se desarrolló una herramienta de videoconferencia con capacidad de adaptar las características del video (v.g. *bitrate*) acorde a las condiciones de la red.

**Palabras clave:** Ad hoc, delay, packet loss, PSNR, redes resilientes, throughput.

**Sumario:** Introducción, Materiales y Métodos, Resultados y Discusión y Conclusiones.

**Como citar:** Astudillo, P., Quinde, C., González, S. & Palacios, I. (2022). Evaluación y comparación de códecs de video para el despliegue de un sistema de comunicación resiliente. *Revista Tecnológica - Espol*, 34(3), 12-30. <http://www.rte.espol.edu.ec/index.php/tecnologica/article/view/935>

### Abstract

In emergency scenarios, it is usually difficult to rely on network infrastructure, especially in remote or geographically difficult-to-access locations. However, it is necessary to have tools to communicate or assess the situation. Therefore, ad hoc networks emerge as an option to solve this problem. This paper proposes the development of an emergency communication system for the transmission of multimedia traffic in a real scenario, between the different floors of a building. For this purpose, multiple experiments were carried out to determine the values of fundamental metrics (e.g., packet loss, delay, and throughput) for real-time audio and video transmission. Specifically, open-source tools were developed with the ability to extract the required metrics and objectively compare video quality using current encoding mechanisms (e.g., VP8, VP9, H.264, H.265). The results obtained show that VP8 is the most suitable codec for the proposed scenario. In addition, a videoconferencing tool was developed with the ability to adapt the video characteristics (e.g., bitrate) according to the network conditions.

**Keywords:** Ad hoc, delay, packet loss, PSNR, resilient network, throughput.

### Introducción

Una red resiliente hace referencia a la capacidad de la red para proporcionar y mantener un nivel de servicio aceptable frente a diversos fallos y retos en el funcionamiento, sea cual sea la naturaleza del reto al que se enfrenta (Frnda et al., 2016). En particular, los sistemas de comunicación son susceptibles a fallos y pérdida de conectividad, usualmente ocasionados por eventos naturales como inundaciones, terremotos, entre otros problemas derivados de la tecnología y actividades humanas maliciosas (Mauthe et al., 2016).

En consecuencia, cuando los servicios de red que forman parte de una infraestructura esencial (v.g. hospitales, aeropuertos, centrales eléctricas, entre otros) dejan de estar disponibles, se generan problemas de comunicación como resultado inevitable (Mauthe et al., 2016). Para solventar este inconveniente se puede hacer uso de redes resilientes basadas en el paradigma denominado redes ad hoc, las cuales permiten la comunicación sin la necesidad de un sistema central o algún tipo de infraestructura de red preexistente (Pan et al., 2019). Por ejemplo, en (Salam et al., 2018) se propone una red ad hoc resiliente basada en drones con el propósito de mantener la comunicación durante la ocurrencia de catástrofes.

Cabe destacar que las redes ad hoc son redes descentralizadas y sin infraestructura, donde los nodos pueden operar como dispositivos terminales y además como enrutadores para el reenvío de la información hacia dispositivos que estén dentro de su radio de cobertura (Mohammed et al., 2020). En este tipo de redes, el comportamiento de métricas tales como el *delay* o el *throughput* requieren de un análisis detallado. Por ejemplo, en (González et al., 2016) se realiza una evaluación del rendimiento del *streaming* de video escalable en una red MANET (*Mobile Ad hoc Networks*), y se demuestra que a medida que aumenta el número de nodos, el *throughput* disminuye en el receptor. Además, en (Iliana et al., 2017) se establece que el *throughput* también disminuye dependiendo del número de dispositivos que ocupan el mismo canal inalámbrico.

De forma similar, en (Tolentino Medrano, 2021) se recalca que la característica dinámica y la alta movilidad ocasionan pérdida de enlaces, lo que disminuye el *throughput*. Por otro lado, en (Ahmed et al., 2020), se demuestra que el retardo de propagación (*delay*) de un enlace multisalto depende del número de repetidores y de la distancia entre repetidores adyacentes. En tal contexto, en (Castellanos et al., 2015), se plantea una estrategia de transmisión adaptativa de video en función del ancho de banda disponible en la red, lo que permitiría reducir la congestión y por tanto la pérdida de paquetes.

En lo que respecta a situaciones de emergencia, es importante dotar a las personas afectadas y al personal de emergencia la capacidad de establecer y utilizar rápidamente medios de comunicación independientes. De esta manera, en (Jagannath et al., 2019) se presenta una solución completa de extremo a extremo que puede conectar a los supervivientes de una catástrofe entre sí y con las autoridades mediante una red ad hoc autosuficiente. De la misma forma, gracias a la movilidad de los vehículos aéreos no tripulados (UAVs), es posible crear una comunicación de red aérea para emergencias que apoye rápidamente y garantice las condiciones estrictas de tiempo en la búsqueda y rescate.

En (Dinh et al., 2019), se propone una arquitectura que apoya la comunicación entre los rescatistas y las víctimas mediante el uso de la red aérea ad hoc para emergencias. Además, la mayor parte de las infraestructuras están dañadas o completamente destruidas en los escenarios posteriores a las catástrofes, por ello, se puede utilizar una red ad hoc vehicular (VANET) para llevar a cabo la operación de rescate, ya que no requiere ninguna infraestructura preexistente. En este contexto, en (Khaliq et al., 2019) se plantea y valida una forma eficaz de transmitir la información crucial mediante el desarrollo de una aplicación y el despliegue de un testbed experimental en un entorno vehicular.

Por otro lado, la vigilancia del tráfico es un aspecto importante para los Sistemas Inteligentes de Transporte (ITS) que ayudan a detectar incidentes automáticamente. En estos casos, las redes ad-hoc podrían ser una opción viable y de bajo coste, no obstante la transmisión de video manteniendo un nivel adecuado de calidad de servicio, es un desafío. Por tal motivo, (Felici-castell et al., 2021) presentan una técnica de transmisión de vídeo adaptativo en tiempo real para transportar flujos de vídeo desde las cámaras a los servidores externos de monitorización del tráfico.

Por otra parte, la gran popularidad de las aplicaciones multimedia, ha impulsado un rápido crecimiento del consumo de contenidos de alta resolución, incrementando de forma dramática la demanda de ancho de banda. En tal sentido, el uso de códecs resulta fundamental ya que permiten comprimir los datos, lo que contribuye a reducir la carga de tráfico en la red, el almacenamiento requerido y por tanto mejorar la velocidad de transferencia. Cabe indicar que, el desarrollo de códecs representa una línea activa de investigación y en constante evolución (Rao et al., 2020). Entre los estándares más usados se tiene: H264, H265, VP8 y VP9 (Bienik et al., 2016).

En lo que respecta al análisis de códecs, en (Paredes et al., 2016) se indica que para aplicaciones en VANET's (*Vehicular Ad hoc Networks*) el estándar H.265/HEVC es superior en compresión a los estándares H.264/AVC y VP9. En cuanto a velocidad de codificación, en (ZhangTicao & MaoShiwen, 2019) se señala que H264/AVC es más rápido que H.265/HEVC y VP9, no obstante presenta la mitad de eficiencia de codificación.

Además, en (Grois et al., 2013) se recalca que VP9 puede presentar un tiempo de codificación de hasta 100 veces el tiempo de H.264, lo que ocasiona un retraso considerable en la emisión de un video. En (Sharma et al., 2019) se presenta un estudio donde se indica que los nuevos códecs (H.265 y VP9) proporcionan una mejora de la tasa de bits de más del 50% respecto a los anteriores H.264 y VP8, pero son considerablemente más lentos en el proceso de codificación.

Así mismo, en (Barman & Martini, 2017), se establece que el rendimiento real de los códecs de video depende en gran medida de factores tales como, el tiempo de codificación, el tipo de contenido (videos naturales/de animación/video sintético), la complejidad del video, la

resolución, los parámetros de codificación (v.g. GoP, fps, VBR/CBR) y las métricas de evaluación de la calidad (objetivo/subjetivo).

Por último, en cuanto al consumo de CPU, tanto VP9 como H.265 implementan más etapas durante la compresión que H.264, lo que aumenta la carga sobre los recursos computacionales. Sin embargo, se puede utilizar codificación basada en *hardware* para disminuir el consumo o en su defecto aumentar el número de CPU empleados. Además, el estudio descrito en (Mengzhe et al., 2015), demuestra que al aumentar el número de CPU el tiempo de codificación disminuye considerablemente, aunque como consecuencia se tiene una disminución del PSNR (*Peak Signal to Noise Ratio*).

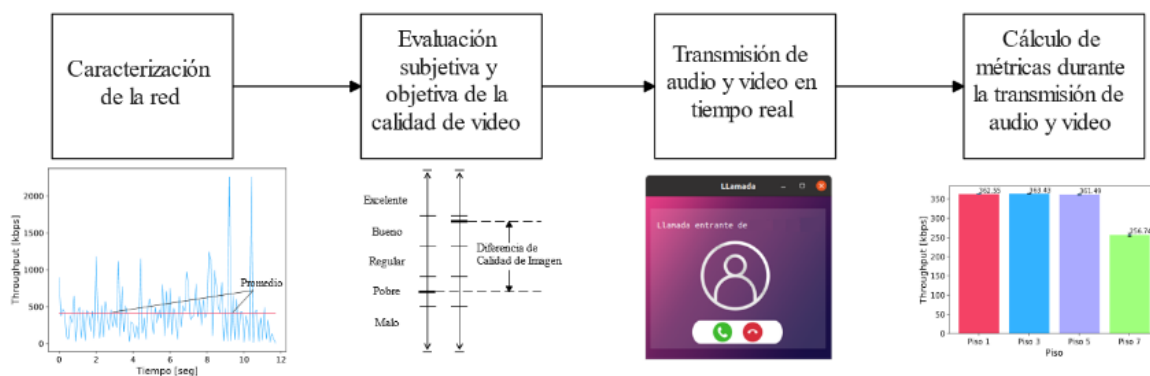
En tal contexto, en este artículo se presenta la evaluación experimental de un sistema de comunicación multimedia robusto frente a eventos de emergencia. La propuesta se basa en herramientas *hardware/software* de tipo *open source*. En particular, se presenta un desarrollo a nivel de capa de aplicación, el mismo que permite la transmisión de tráfico multimedia (audio/video) con capacidad de adaptarse a las condiciones de la red, para lo cual se analizó previamente el mejor mecanismo de compresión considerando las características de la red empleada.

Cabe indicar que el estudio fue realizado sobre un escenario real consistente en el edificio matriz de la Empresa Eléctrica Regional Centro Sur (Cuenca-Ecuador). Dicho escenario cuenta con una red ad hoc multi-salto, la misma que fue desplegada previamente con el fin de monitorizar el comportamiento dinámico del edificio (Palacios et al., 2022). Donde, la principal contribución de este artículo es el desarrollo de una metodología para caracterizar una red resiliente para comunicaciones de emergencia, y la implementación de una aplicación de videollamada con capacidades de video adaptativo sobre dispositivos con bajo procesamiento utilizando el códec VP8.

El resto del artículo se encuentra organizado de la siguiente forma. En la sección Materiales y Métodos, se presenta el procedimiento definido para caracterizar la red ad hoc, la metodología para evaluar la calidad del video empleando diferentes códecs, así como el desarrollo de una herramienta capaz de transmitir audio y video en tiempo real con capacidad de adaptarse a las características de la red. Posteriormente, en la siguiente sección se discuten los resultados obtenidos. Finalmente, se exponen las principales conclusiones del estudio.

**Figura 1**

*Metodología para la evaluación de la propuesta*



## Materiales y Métodos

En la Figura 1 se presenta la metodología empleada para la evaluación de la propuesta, la misma consta de cuatro etapas. En primer lugar, se desarrollaron un conjunto de experimentos con la finalidad de caracterizar la red ad hoc disponible en el sitio de estudio. A continuación, se procedió a analizar de forma subjetiva y objetiva la calidad del tráfico de video, en la red, dicho proceso abarcó desde la selección de un conjunto de videos hasta la comparación y análisis del códec más adecuado acorde a las características de la red.

Posteriormente, se procedió a transmitir audio y video en tiempo real para verificar la comunicación en casos de emergencia, con tal finalidad se desarrolló una solución a nivel de capa de aplicación. Para concluir, se procedió a evaluar de forma objetiva el comportamiento del tráfico multimedia transmitido en términos de *packet loss*, *delay* y *throughput*.

### Caracterización de la red ad hoc

Antes de transmitir el video es necesario verificar la capacidad de la red, mediante un análisis de métricas tales como el *packet loss*, el *delay* y el ancho de banda disponible. En tal sentido, se realizaron cuatro experimentos donde se ubicó un ordenador fijo en el sótano y otro ordenador en un piso superior (piso uno, tres, cinco y siete), tal como se esquematiza en la Figura 2. Como se puede apreciar, el experimento desarrollado entre el sótano y el piso 1 genera tres saltos (Figura 2a), el experimento con respecto al piso 3 genera cuatro saltos, el experimento en el piso 5 genera cinco saltos y finalmente el experimento entre el ordenador del sótano y el piso 7 genera seis saltos (Figura 2b). Por otro lado, cada experimento consistió en generar tráfico constante y transmitirlo desde un ordenador hacia el otro, aumentando el *bitrate* desde los 100Kbps hasta los 10Mbps para de esta forma determinar el umbral del *throughput* en cada piso.

### Metodología para la evaluación subjetiva y objetiva de la calidad de video

Para determinar el códec más adecuado para la transmisión de video sobre la red, se realizaron múltiples experimentos. En particular, se definió una metodología, la misma que se esquematiza en la Figura 3. Como se aprecia, el primer paso fue seleccionar videos en formato CIF (*Common Intermediate Format*), el mismo que es ampliamente usado en aplicaciones de videoconferencia. En tal sentido, se escogieron y descargaron las secuencias *foreman*, *mother-daughter* y *news*, disponibles en (National Science Foundation, 2000), las mismas que presentan diferentes condiciones como, por ejemplo, cambios rápidos y lentos de escena, diferente cantidad de objetos presentes y estabilidad o no de la cámara.

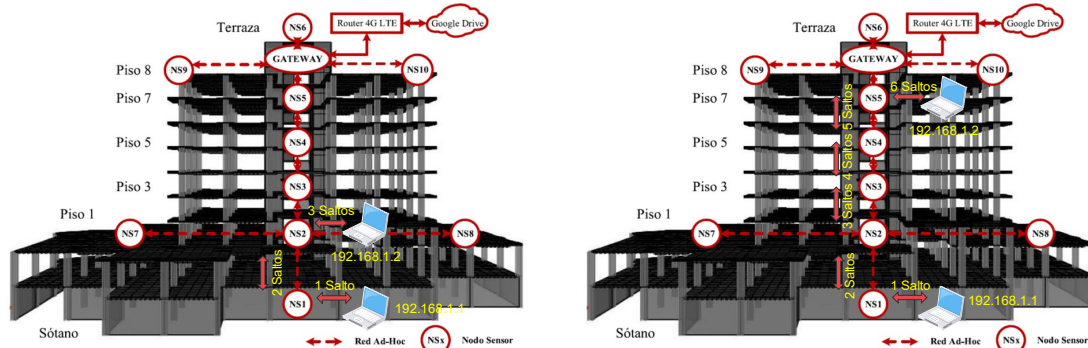
Posteriormente, se codificaron los videos empleando como parámetro principal el *bitrate* y como parámetros adicionales el GoP (*Group of Pictures*) y el *framerate*. En este punto se consideró el umbral determinado en la caracterización de la red ad hoc. El tercer paso consistió en realizar una valoración subjetiva de la calidad del video con los diferentes parámetros de codificación. Para esta evaluación se aplicó el método *Double Stimulus Impairment Scale* (DSIS), una descripción detalla de esta metodología se presenta en (Zhang & Bull, 2014). El cuarto paso fue definir de forma subjetiva la mejor opción, realizando una media entre las valoraciones de las personas encuestadas.

A continuación, utilizando la herramienta GStreamer (*GStreamer: Open Source Multimedia Framework*, 2022), se codificaron y transmitieron en tiempo real los videos sobre el propio ordenador, donde se capturaron todos los paquetes generados para obtener el perfil del tráfico de cada uno. En el sexto paso se realizó una valoración del perfil del tráfico vs. el umbral de la red ad hoc, para determinar si es posible la transmisión del video sobre la red existente. A partir del análisis, en caso de ser necesario, se ajustan nuevamente las métricas y

se regresa al paso 2. El siguiente paso tiene como objetivo definir los parámetros de *bitrate*, *GoP* y *framerate* que proporcionen los mejores resultados buscando un equilibrio entre calidad y compresión.

**Figura 2**

*Ubicación de los ordenadores para los experimentos*



a) Experimento en el piso 1 (3 saltos).

b) Experimento en el piso 7 (6 saltos).

En cuanto al paso número 8, se codificaron y transmitieron los videos en tiempo real con los códecs: H.264, H.265, VP8 y VP9. Además, se evaluaron las métricas de *throughput*, *delay*, *packet loss*, carga de CPU, tiempo de codificación y PSNR. Después, se realizaron los experimentos entre los diferentes pisos del edificio con los distintos videos seleccionados y se compararon los resultados. Finalmente, se definió el códec con mejores resultados, el cual se emplea posteriormente para la transmisión de video en tiempo real. Cabe resaltar que para el desarrollo de los experimentos se implementaron herramientas *software*, las mismas que han sido liberadas para su uso y se encuentran disponibles en <https://github.com/christianquinde/GUI> y <https://github.com/christianquinde/GUI-rx>.

**Figura 3**

*Metodología para la selección del mejor códec de video*

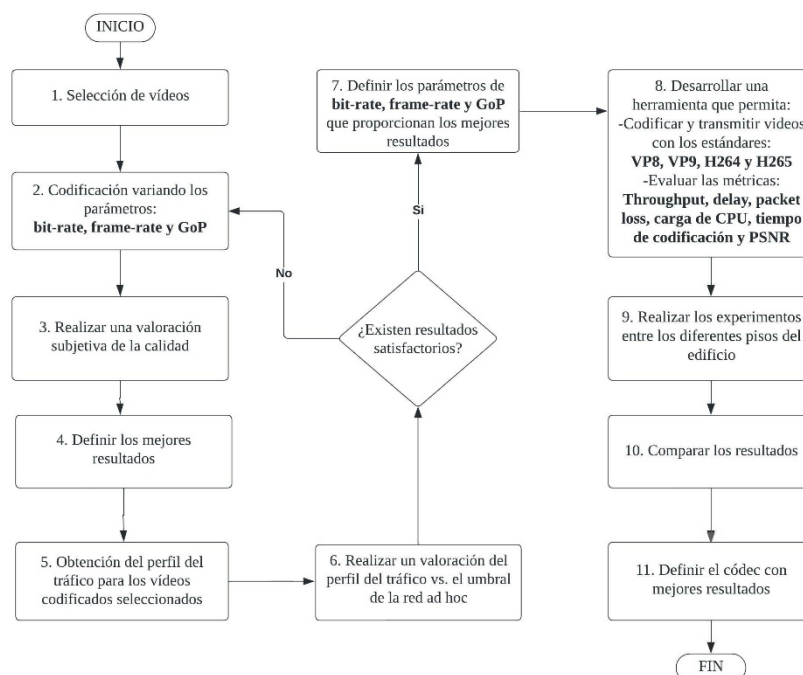
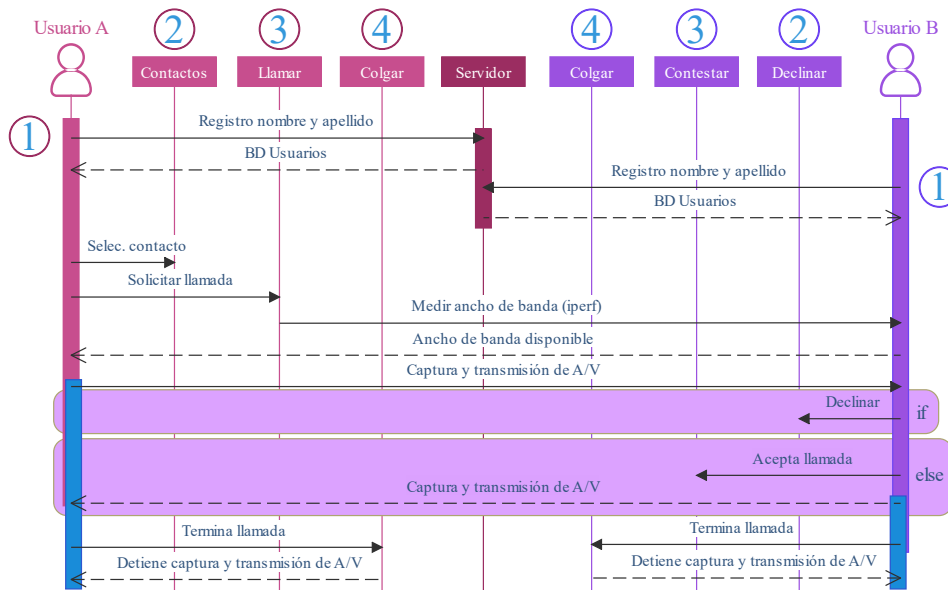


Figura 4

Diagrama secuencial para la herramienta de audio y video en tiempo real



### Transmisión de audio y video en tiempo real

En esta fase de la metodología, se desarrolló una herramienta para la transmisión de audio y video en tiempo real, basada en el protocolo RTP (*Real Time Transport Protocol*). En la Figura 4 se presenta un diagrama secuencial que describe las principales características y funcionalidades implementadas. Cabe destacar que al realizar la llamada se mide el ancho de banda de la red, lo cual permite codificar el video con diferentes valores de *bitrate*, y por lo tanto se consigue una funcionalidad de video adaptativo. En la Figura 5, se puede observar los resultados del video adaptativo, donde se evidencia que la calidad del video con 350kbps es superior al video con 250kbps, sin embargo incrementa el consumo de ancho de banda. Adicionalmente, la aplicación desarrollada ha sido liberada para su uso y se encuentra disponible en el repositorio de Github en <https://github.com/christianquinde/App-videoconferencia>.

Figura 5

Video codificado con 350 kbps (izq) y 250 kbps (der)



### Cálculo de métricas durante la transmisión de audio y video en tiempo real

Para el cálculo del *delay* y de la pérdida de paquetes se empleó la información de control proporcionada por el protocolo RTCP, en concreto los paquetes *Sender Report* (SR) y *Receiver Report* (RR), los mismos que se intercambian entre los dos terminales que realizan la videollamada de emergencia. El *delay* se calcula utilizando la Ecuación 1, donde A es la hora en que se recibe el bloque RR. *Last Sender Report* (LSR) es el campo con la marca de tiempo del último SR recibido, y *Delay since Last Sender Report* (DLSR) es el tiempo entre la recepción del último paquete SR y el envío del paquete RR. Cabe mencionar que, aunque los enlaces pueden tener retrasos asimétricos, este método resulta adecuado como una medida aproximada del retardo entre los dos extremos.

$$\text{delay} = (A - \text{LSR} - \text{DLSR})/2 \quad (1)$$

Por otra parte, para obtener el porcentaje de *packet loss* se utilizó el campo *fraction lost* de los paquetes RR y el campo *cumulative number of packets lost* del último RR recibido. El primer campo representa la fracción de paquetes RTP perdidos desde que se envió el anterior paquete SR. Dicha fracción se define como el número de paquetes perdidos dividido para el número de paquetes esperados (256), el cual está definido por 8 bits dentro del campo *fraction lost* del paquete RTCP. El segundo campo utilizado contiene el número total de paquetes perdidos desde el inicio de la transmisión. Con estos valores y con el número de paquetes RR capturados se obtiene el *packet loss* mediante la Ecuación 2.

$$\text{PL}[\%] = \frac{\text{total de paquetes perdidos acumulados} \cdot 100\%}{\text{total\_RR} \cdot 256} \quad (2)$$

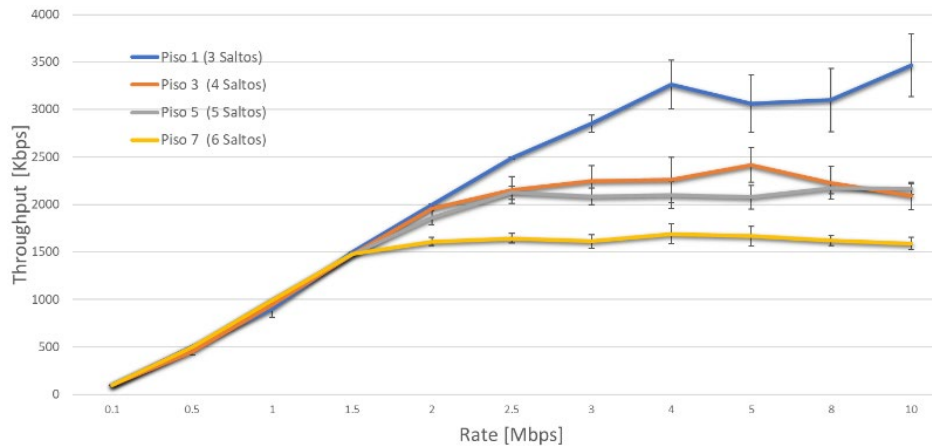
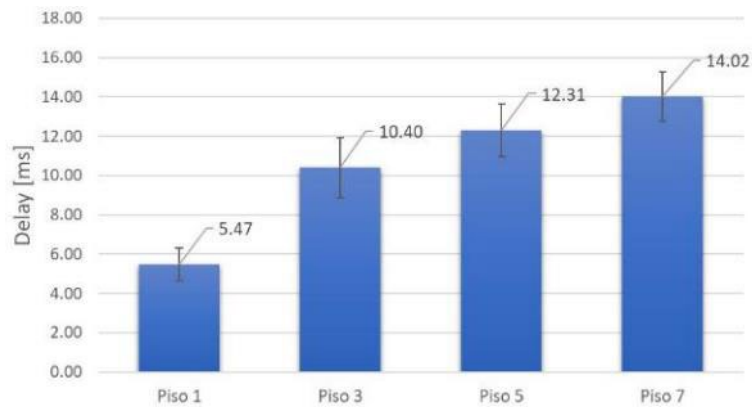
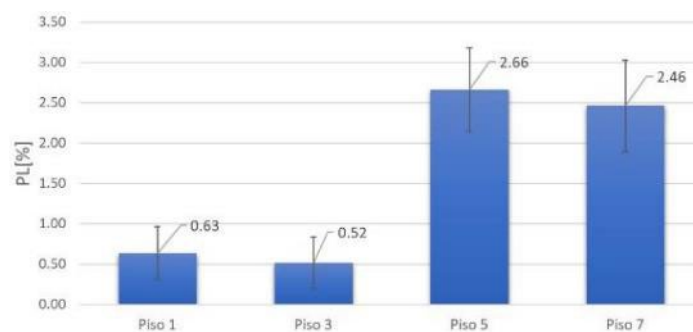
Por otro lado, para obtener el *throughput* de la transmisión, ya sea de audio o video, se empleó la herramienta *tcpstat* con un intervalo de 0.1 segundos. Es decir, a partir del tráfico capturado se realiza un filtrado por puertos para obtener el perfil del tráfico y el *throughput* promedio correspondiente a cada flujo. Cabe mencionar que se realizó un promedio de 10 transmisiones y una comparación de cada métrica por piso. De forma similar, las herramientas desarrolladas han sido liberadas para su uso y pueden descargarse en <https://github.com/christianquinde/herramienta-medicion> y <https://github.com/christianquinde/server>.

## Resultados y Discusión

### Caracterización de la red ad hoc

Con el objetivo de caracterizar la red ad hoc, se realizaron varios experimentos transmitiendo tráfico constante entre los diferentes pisos del edificio. Para cada tasa de transmisión se repitió el experimento un total de 10 veces con el objetivo de efectuar un análisis estadístico. Los resultados obtenidos se presentan en la Figura 6, donde se evidencia que conforme se incrementa el número de saltos disminuye el *throughput* de la red. Además, se puede observar que el *throughput* máximo alcanzado es de 3.5 Mbps y el mínimo de 1.6 Mbps para los pisos 1 (3 saltos) y 7 (6 saltos) respectivamente. Cabe resaltar que a partir de los resultados obtenidos se estableció el umbral de la red con el valor de 1.6 Mbps.



**Figura 6***Throughput [Kbps] vs data rate deseado [Mbps]***Figura 7***Retardo vs número de piso***Figura 8***Pérdida de paquetes vs número de piso*

### Análisis del delay y packet loss

Para obtener el retardo de la red entre los ordenadores (es decir entre los nodos extremo), se realizaron múltiples experimentos transmitiendo mensajes tipo *Internet Control Message Protocol* (ICMP), de los cuales se obtuvo el número de paquetes perdidos y el tiempo requerido para cada uno. Los resultados que se presentan en la Figura 7 demuestran que el retardo aumenta conforme se incrementa el número de saltos, siendo 14.02 ms y 5.47 ms para el piso 7 y 1 respectivamente.

Por otro lado, para obtener el porcentaje de *packet loss* se realizó un promedio de las transmisiones realizadas. En la Figura 8 se presentan los valores obtenidos para cada piso, donde se observa que existe una mayor pérdida de paquetes en los pisos más altos. Los resultados indican que el piso 7 tiene menor pérdida de paquetes que el piso 5, esto se debe a que existe variabilidad debido a que se trata de un escenario real y se debe tener presente la naturaleza del medio inalámbrico. Por lo cual, a partir de los intervalos de confianza se tienen resultados similares para los pisos 5 y 7.

### Evaluación subjetiva y objetiva de la calidad de video

Siguiendo el esquema de la Figura 3, la valoración subjetiva de la calidad de video se determinó mediante la métrica MOS (*Mean Opinion Score*), los resultados se presentan en la Tabla 1. Además, se caracterizó el perfil del tráfico de las secuencias de video para compararlo frente al valor umbral del *throughput* obtenido en la red (1.6 Mbps).

**Tabla 1**

*Mejores resultados del análisis subjetivo con bitrate = 350 kbps*

VIDEO	GoP	Fps
Foreman	30	20, 25
Mother-daughter		
News		
Foreman	60	20, 25
Mother-daughter		
News		

En este contexto, para los tres videos seleccionados se escogieron las métricas que aseguran una mayor calidad y fluidez del video, es decir un tamaño de GoP igual a 30 y un valor de 25 fps (*Frames per Second*), teniendo en cuenta que en la mayoría de casos no se superó el umbral establecido de la red ad hoc.

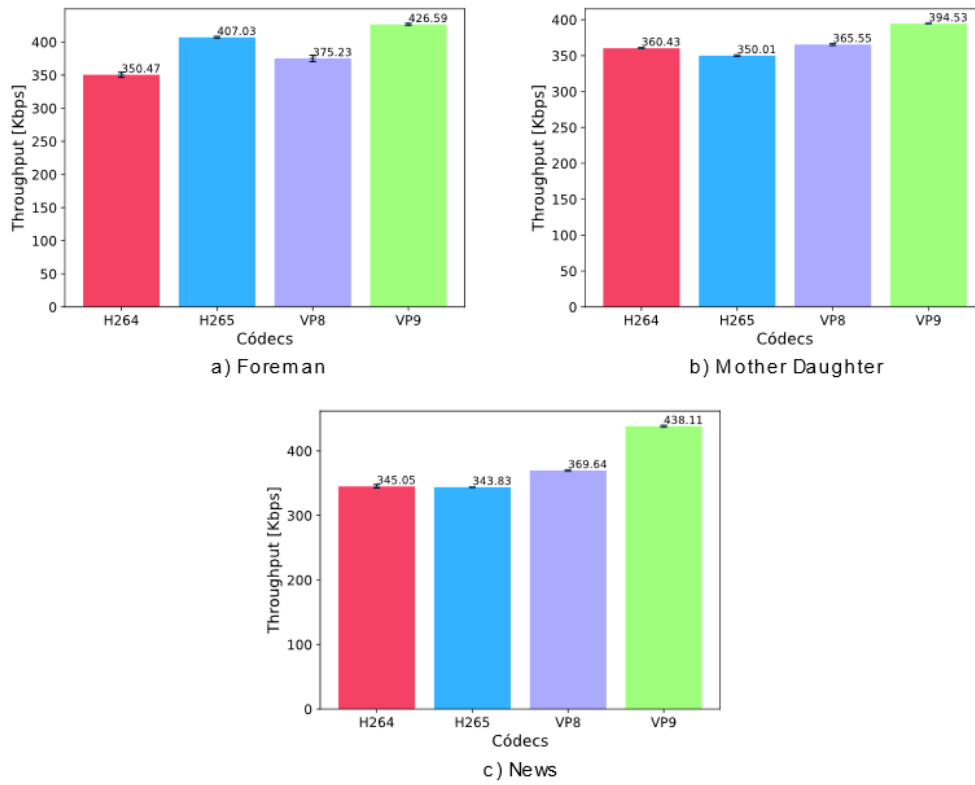
### Evaluación del *throughput*

Con el objetivo de analizar el *throughput* se realizaron cinco transmisiones con cada códec de video en todos los pisos. En la Figura 9 se presentan los resultados obtenidos para el caso más crítico, es decir cuando se requiere un mayor número de saltos (piso 7). Donde se observa que VP9 requiere mayor ancho de banda. Sin embargo, el *throughput* máximo está por debajo del umbral establecido en la caracterización de la red ad hoc, por lo que el video se reproduce sin problemas.

Además, en la Figura 10, se presentan los perfiles del tráfico de una transmisión del video "News" con cada códec en el piso 7. Como era de esperarse, el perfil del tráfico es variable y presenta ráfagas, las mismas que al superar el umbral de la red generan pérdida de paquetes. Con lo cual, los mejores resultados se obtuvieron con los códecs VP8 y H265, seguidos de H264 y por último VP9.

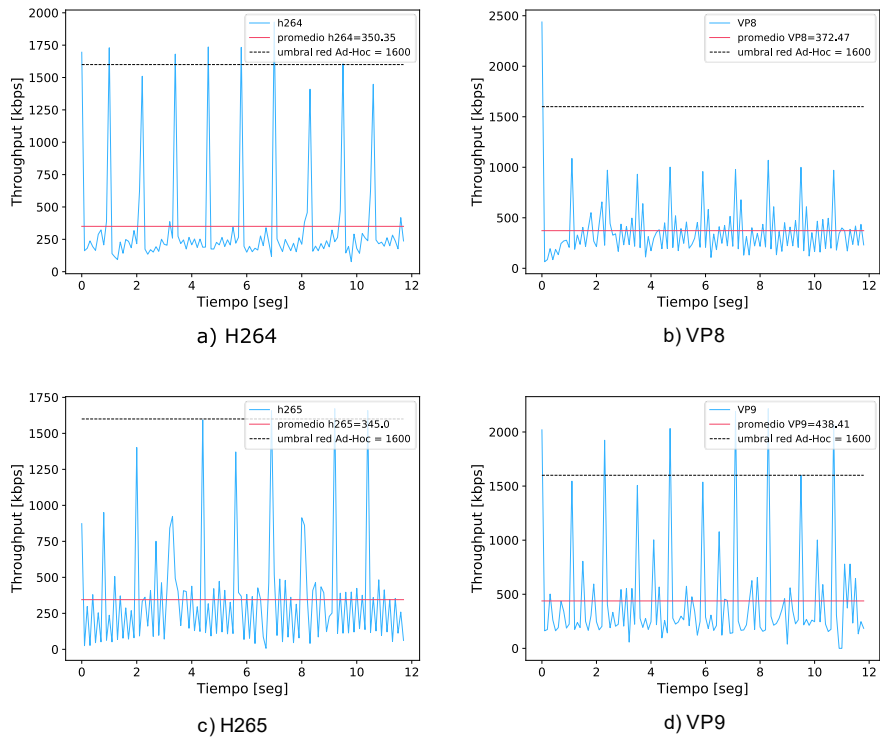
**Figura 9**

*Throughput promedio en el piso 7*



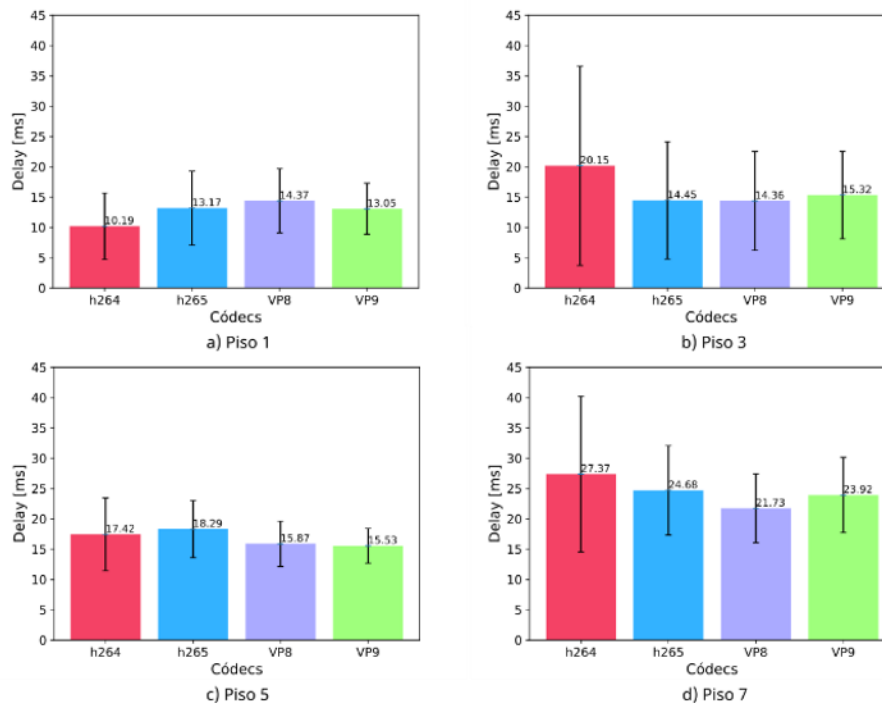
**Figura 10**

*Throughput para el video “News” en el piso 7*



**Figura 11**

*Delay promedio obtenido para el video “News” en cada piso*



### ***Evaluación del retardo***

Para determinar el valor del *delay* se restan los tiempos entre el transmisor y receptor con ayuda del protocolo RTCP y un servidor NTP que sincroniza a las dos terminales. Se puede notar que al subir de piso el *delay* se incrementa. Cabe recalcar que los tiempos de *delay* varían dependiendo de la hora del día (nivel de interferencia) y el nivel de precisión que tiene el cliente al sincronizarse con el servidor NTP.

En la Figura 11, se presentan los resultados para el video “News” y se observa que en el caso más extremo (seis saltos), la aplicación cumple con los requisitos establecidos en la recomendación UIT-T G.114 (Tsbmail, 2003), la cual recomienda un retardo menor a 150ms para una transmisión en tiempo real.

### ***Evaluación de la pérdida de paquetes***

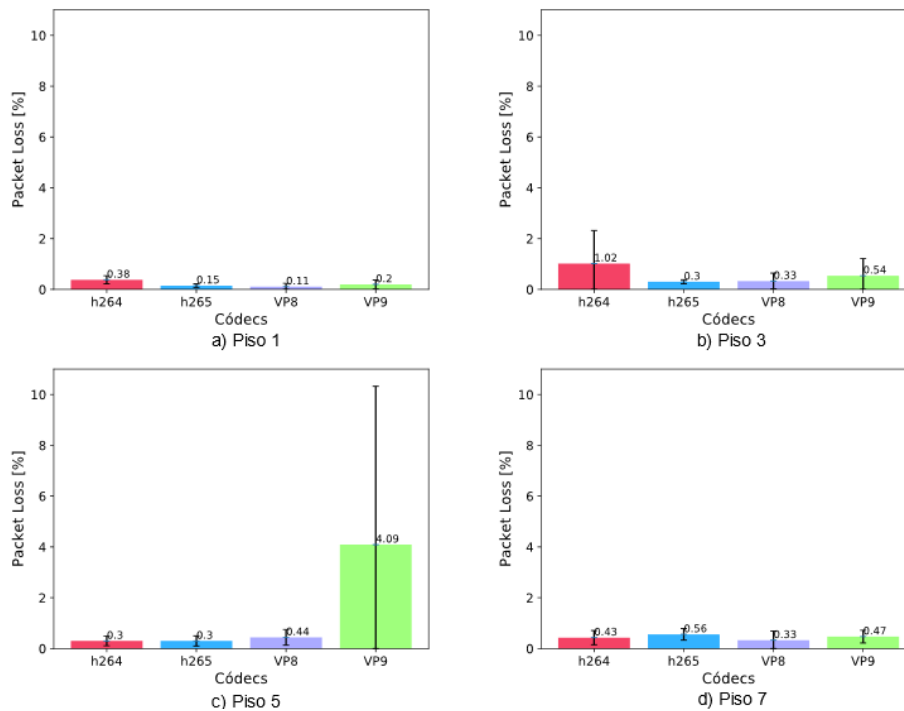
La siguiente métrica evaluada es el porcentaje de pérdida de paquetes. Para obtener este valor se compara el número de paquetes generados en el transmisor con los paquetes obtenidos en el receptor empleando la Ecuación 3. Para que la comparación sea adecuada se realiza un filtrado de los paquetes capturados por dirección IP de origen y destino, número de puerto y protocolo de transporte (UDP, *User Datagram Protocol*).

En la Figura 12, se presentan los resultados obtenidos para el video “News” en cada piso. Donde, a partir de la gran amplitud de las barras de error se nota que existe una gran variabilidad en los resultados entre cada una de las transmisiones, esto se debe a las interferencias y las condiciones propias del canal inalámbrico. Sin embargo, se observa que en promedio, la pérdida de paquetes es baja y no supera el 5 %, estando en la mayoría de casos por debajo del 1 %, lo cual se considera una pérdida de paquetes “aceptable” para una llamada de voz y video (Vyopta, 2019).

$$\text{Packet Loss \%} = (\text{Paquetes enviados} - \text{Paquetes recibidos}) / \text{Paquetes enviados} \quad (3)$$

Figura 12

Porcentaje de packet loss para el video "News" en cada piso



### Evaluación del consumo de CPU

Para obtener el consumo de CPU, en cada transmisión se empleó la herramienta *top* de Linux. Los resultados se presentan en la Figura 13, donde se puede destacar una tendencia similar en todos los pisos para el caso del video *mother-daughter*, siendo el códec VP8 el más eficiente, ya que cuenta con el menor consumo de CPU. En cuanto a los códecos VP9 y H264 tienen un consumo equitativo alrededor del 100 %, y por último H265 es el códec con mayor consumo de CPU. Cabe destacar que el comando *top* muestra el consumo como un porcentaje de una sola CPU.

En los sistemas multinúcleo, se pueden tener porcentajes superiores al 100 %, como es el caso de H265. Es decir, el 100% representa un solo núcleo funcionando a su máxima capacidad. Cabe indicar que, se fijó el uso de CPU de VP8 y VP9 en 2 y 4 núcleos respectivamente para poder realizar una transmisión en tiempo real y ajustarse al tiempo de duración del video. Los códecos H264 y H265 utilizan el *preset speed-preset* en su valor por defecto (*Medium*) que mantiene un equilibrio entre calidad y velocidad de codificación.

### Evaluación del tiempo de codificación y transmisión

Para esta métrica, al modificar los valores por defecto de los códecos, se establece que el tiempo de codificación y tiempo de transmisión sean iguales al tiempo de video para poder realizar una transmisión en tiempo real.

### Análisis del PSNR

En la Figura 14 se muestran los resultados de la métrica PSNR para cada video, dependiendo del tipo de códec. Respecto a los resultados, se evidencia H265 como el códec con mejor calidad debido a que tiene un mayor valor de PSNR, a éste le siguen H264 y VP9 manteniendo resultados similares y por último está el códec VP8.

Figura 13

Consumo de CPU promedio para el video "News" en cada piso

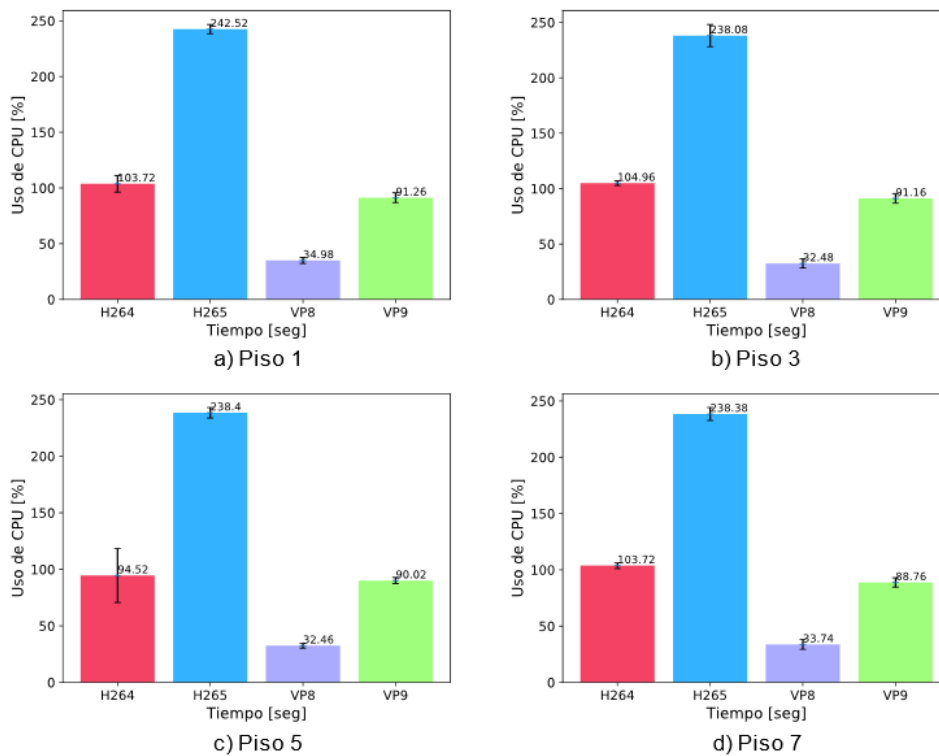
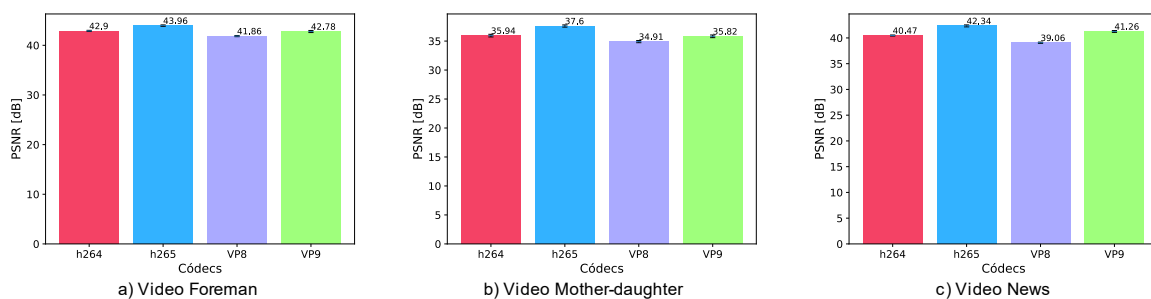


Figura 14

Valor de PSNR de los videos

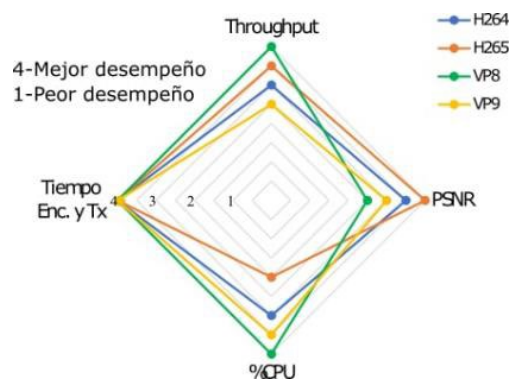


### Definición del códec más adecuado

Basándose en el análisis realizado, en la Figura 15 se presenta un resumen de los mejores resultados para cada códec. En tal sentido se seleccionó a VP8 como el códec más adecuado para la aplicación requerida. Concretamente porque en términos de *throughput* es el códec que no presenta ráfagas que superen el umbral establecido y esto permite que la pérdida de paquetes sea menor. A su vez, es el códec que demanda menos CPU, permitiendo hacer uso de equipos con menos recursos de *hardware*. Finalmente, en términos de PSNR tiene resultados similares a los demás códecs.

Figura 15

Desempeño de cada códec por métrica



### Resultados de la transmisión de audio y video en tiempo real

Para transmitir audio y video entre los diferentes pisos del edificio, primero se estableció un umbral de 1.6 Mbps para realizar diferentes codificaciones con distintos *bitrates*. Debido al estado de la red, en algunos instantes al codificar con 350 kbps se generaron ráfagas que superan los 1.6 Mbps, ocasionando gran pérdida de paquetes y por lo tanto una degradación de la calidad. A partir de lo observado, cuando la red disponía de un *throughput* por encima del umbral el video se codificó con un *bitrate* igual a 350 kbps y por debajo de este umbral se codificó con 250 kbps. De esta manera se obtiene siempre una videollamada fluida.

En la Tabla 2, se presenta el ancho de banda medido antes de iniciar la transmisión en cada piso y también el *bitrate* seleccionado automáticamente en cada uno, teniendo en cuenta el umbral establecido. Donde se observa que el piso más crítico (piso 7) fue el único en codificar el video con 250 kbps ya que en ese instante hubo sobrecarga en la red.

Tabla 2

Ancho de banda medido por piso para la selección del *bitrate* adecuado

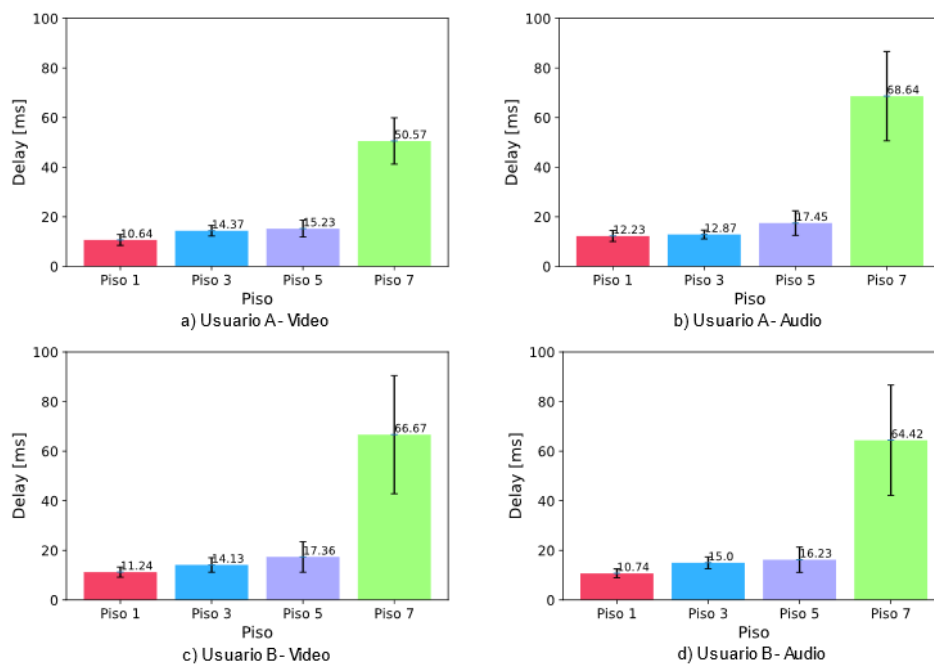
PISO	SALTOS	ANCHO DE BANDA [Mbps]	BITRATE [Kbps]
1	3	2.82	350
3	4	2.72	350
5	5	2.5	350
7	6	1.59	250

Para evaluar objetivamente la calidad de audio y video, se calcularon las métricas a partir del tráfico capturado en tiempo real una vez establecida la videollamada. Cabe indicar que se realizó un total de 10 capturas con una duración de dos minutos cada una con el objetivo de efectuar un análisis estadístico de los resultados. Al finalizar las transmisiones en todos los pisos, se promediaron las tres métricas (*throughput*, *delay*, *packet loss*) tanto para audio como para video.

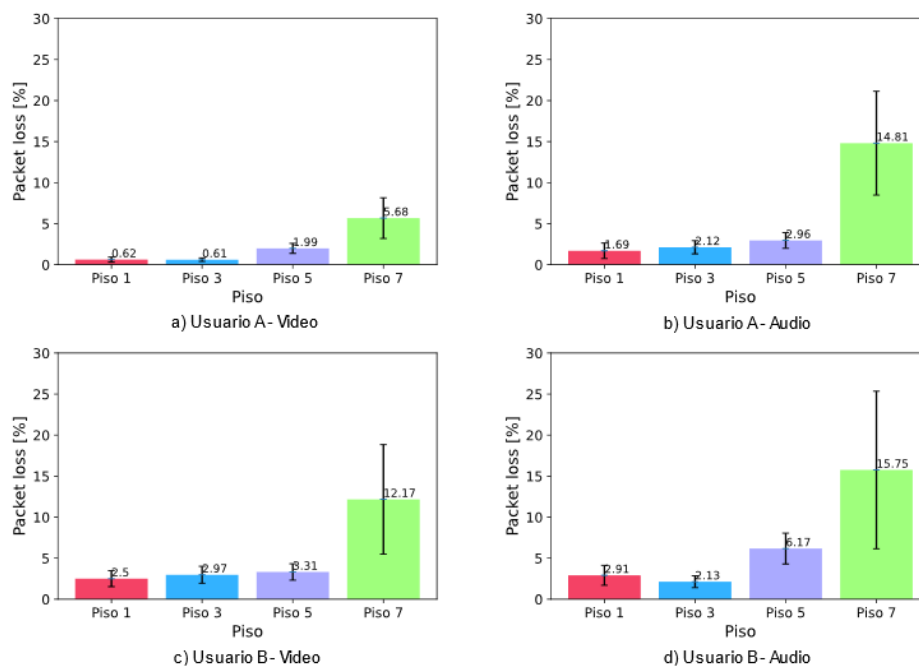
Además, en la Figura 16 correspondiente al *delay*, se comprobó que, al incrementar el número de saltos, el *delay* aumentó tanto para el audio como para el video. Finalmente, para la Figura 17 correspondiente al *packet loss* se verificó que el piso 7 tiene mayor pérdida de paquetes. Cabe recalcar que el *packet loss* y el *delay* son dependientes de la sobrecarga o interferencia que existe en ese momento en la red.

**Figura 16**

Resultados para la métrica delay de audio y video del usuario A y B

**Figura 17**

Resultados para la métrica packet loss de audio y video del usuario A y B



## Conclusiones

La caracterización de la red ad hoc demostró que, al existir interferencias, las cuales aumentan o disminuyen dependiendo de la hora del día al variar el número de redes wifi que afectan al canal 7, y conforme se incrementa el número de saltos, la red se sobrecarga reduciendo el ancho de banda disponible. Debido a esto, es de gran importancia contar con un



sistema de transmisión de video capaz de comprobar el estado de la red en el momento que se desea realizar la videollamada y escoger el *bitrate* más adecuado en ese instante.

Además, permitió establecer el umbral máximo de ancho de banda en 1.6 Mbps en el piso más crítico, siendo este la referencia para mantener una comunicación aceptable. Cabe mencionar que la metodología empleada consistió en colocar nodos fijos dentro de un área limitada del edificio y para obtener el mayor ancho de banda se eligió un intervalo del día con la menor interferencia.

En cuanto a los experimentos realizados para obtener el códec óptimo, se comprobó que VP8 permite la transmisión de un mismo video con ráfagas y con una amplitud menor al umbral de la red, alrededor de 1Mbps, mientras que los demás códecs tuvieron ráfagas que superan los 1.6Mbps. Otro aspecto importante es que con VP8 se requirió tan solo un máximo dos CPUs, a diferencia de VP9 que utilizó 4 y tanto H264 como H265 utilizaron 8. Siendo estas las razones principales para descartar a los demás códecs estudiados. De esta forma, en caso de emergencia, cualquier persona puede hacer uso de la herramienta de videollamada incluso con dispositivos limitados en hardware.

Cabe destacar que, la aplicación de video de emergencia desarrollada permite una comunicación bidireccional de audio y video entre dos usuarios dentro de la red ad hoc y además con la capacidad de adaptarse a las condiciones de la red para garantizar una videoconferencia fluida.

Por último, las pruebas realizadas confirmaron el correcto funcionamiento y establecimiento de una videollamada entre los pisos más alejados, en donde se presentan las condiciones más críticas en cuanto a *delay*, *throughput* y *packet loss*. Con lo cual, se estableció un *bitrate* de 250 Kbps para obtener una videollamada fluida en el piso más crítico, generando un *delay* promedio de 50ms y un *packet loss* promedio de 5.68% en uno de los usuarios. En conclusión, por medio de la herramienta desarrollada se estableció una comunicación de emergencia en tiempo real aprovechando una red ad hoc desplegada.

Como futuras investigaciones se plantea aplicar el sistema a una red con nodos móviles y un protocolo de enrutamiento dinámico para cubrir un área mayor dentro del edificio. Además, se puede utilizar un método donde el canal usado sea escogido de forma dinámica para utilizar el mayor ancho de banda disponible.

### Reconocimientos

Los autores de este trabajo agradecen al Vicerrectorado de Investigación de la Universidad de Cuenca y al proyecto de investigación “Tecnologías IoT y Redes Inalámbricas de Sensores Aplicados a la Monitorización de Salud Estructural en Edificios Esenciales de la Ciudad de Cuenca” que permitieron realizar el trabajo de investigación haciendo uso de su red.

### Referencias

- Ahmed, Z., Naz, S., & Ahmed, J. (2020). Minimizing transmission delays in vehicular ad hoc networks by optimized placement of road-side unit. *Wireless Networks 2020* 26:4, 26(4), 2905–2914. <https://doi.org/10.1007/S11276-019-02198-X>
- Barman, N., & Martini, M. G. (2017). H.264/MPEG-AVC, H.265/MPEG-HEVC and VP9 codec comparison for live gaming video streaming. *2017 9th International Conference on Quality of Multimedia Experience, QoMEX 2017*. <https://doi.org/10.1109/QoMEX.2017.7965686>

- Bienik, J., Uhrina, M., Kuba, M., & Vaculik, M. (2016). Performance of H. 264, H. 265, VP8 and VP9 Compression Standards for High Resolutions. *2016 19th International Conference on Network-Based Information Systems (NBIS)*, 246–252.
- Castellanos, W., Guzmán, P., Arce, P., & Guerri, J. C. (2015). Mechanisms for improving the scalable video streaming in mobile Ad hoc networks. *PE-WASUN 2015 - Proceedings of the 12th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks*, 33–40. <https://doi.org/10.1145/2810379.2810391>
- Dinh, T. D., Le, D. T., Tran, T. T. T., & Kirichek, R. (2019). Flying Ad-Hoc Network for Emergency Based on IEEE 802.11p Multichannel MAC Protocol. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11965 LNCS, 479–494. [https://doi.org/10.1007/978-3-030-36614-8\\_37/COVER](https://doi.org/10.1007/978-3-030-36614-8_37/COVER)
- Felici-castell, S., García-pineda, M., Segura-garcia, J., Fayos-jordan, R., & Lopez-ballester, J. (2021). Adaptive live video streaming on low-cost wireless multihop networks for road traffic surveillance in smart cities. *Future Generation Computer Systems*, 115, 741–755. <https://doi.org/10.1016/j.future.2020.10.010>
- Frnda, J., Voznak, M., & Sevcik, L. (2016). Impact of packet loss and delay variation on the quality of real-time video streaming. *Telecommunication Systems*, 62(2), 265–275.
- González, S., Castellanos, W., Guzmán, P., Arce, P., & Guerri, J. C. (2016). Simulation and experimental testbed for adaptive video streaming in ad hoc networks. *Ad Hoc Networks*, 52, 89–105. <https://doi.org/10.1016/j.adhoc.2016.07.007>
- Grois, D., Marpe, D., Mulyoff, A., Itzhaky, B., & Hadar, O. (2013). Performance comparison of H.265/MPEG-HEVC, VP9, and H.264/MPEG-AVC encoders. *2013 Picture Coding Symposium, PCS 2013 - Proceedings*, 394–397. <https://doi.org/10.1109/PCS.2013.6737766>
- GStreamer: open source multimedia framework.* (2022). <https://gstreamer.freedesktop.org/>
- Iliana, E., Ángel, M., & Roberto Carlos, A. (2017). Análisis del parámetro Throughput en una red Ad hoc y MANET en el estándar 802.11ac. *Marzo*, 3, 1–9. [www.ecorfan.org/spain](http://www.ecorfan.org/spain)
- Jagannath, J., Furman, S., Jagannath, A., Ling, L., Burger, A., & Drozd, A. (2019). HELPER: Heterogeneous Efficient Low Power Radio for enabling ad hoc emergency public safety networks. *Ad Hoc Networks*, 89, 218–235. <https://doi.org/10.1016/J.ADHOC.2019.03.010>
- Khaliq, K. A., Chughtai, O., Shahwani, A., Qayyum, A., & Pannek, J. (2019). An Emergency Response System: Construction, Validation, and Experiments for Disaster Management in a Vehicular Environment. *Sensors 2019, Vol. 19, Page 1150, 19(5)*, 1150. <https://doi.org/10.3390/S19051150>
- Mauthe, A., Hutchison, D., Cetinkaya, E. K., Ganchev, I., Rak, J., Sterbenz, J. P. G., Gunkelk, M., Smith, P., & Gomes, T. (2016). Disaster-resilient communication networks: Principles and best practices. *2016 8th International Workshop on Resilient Networks Design and Modeling (RNDM)*, 1–10.
- Mengzhe, L., Xiuhua, J., & Xiaohua, L. (2015). Analysis of H.265/HEVC, H.264 and VP9 coding efficiency based on video content complexity. *2015 IEEE International Conference on Computer and Communications (ICCC)*, 420–424. <https://doi.org/10.1109/CompComm.2015.7387608>
- Mohammed, A. S., Balaji B, S., S, S. B. M., N, A. P., & K, V. (2020). FCO — Fuzzy constraints applied Cluster Optimization technique for Wireless AdHoc Networks. *Computer Communications*, 154, 501–508. <https://doi.org/10.1016/J.COMCOM.2020.02.079>
- National Science Foundation, A. S. U. (2000). *YUV Sequences*. <http://trace.eas.asu.edu/yuv/>
- Palacios, I., Placencia, J., Muñoz, M., Samaniego, V., González, S., & Jiménez, J. (2022). MQTT Based Event Detection System for Structural Health Monitoring of Buildings. *Lecture Notes in Networks and Systems*, 405 LNNS, 56–70. [https://doi.org/10.1007/978-3-030-96043-8\\_5](https://doi.org/10.1007/978-3-030-96043-8_5)
- Pan, Z., Qin, H., Yi, X., Zheng, Y., & Khan, A. (2019). Low complexity versatile video coding for traffic

- surveillance system. *International Journal of Sensor Networks*, 30(2), 116–125. <https://doi.org/10.1504/IJSNET.2019.099473>
- Paredes, C. I., Mezher, A. M., & Igartua, M. A. (2016). Performance Comparison of H.265/HEVC, H.264/AVC and VP9 Encoders in Video Dissemination over VANETs. *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST, 195 LNICST*, 51–60. [https://doi.org/10.1007/978-3-319-61949-1\\_6](https://doi.org/10.1007/978-3-319-61949-1_6)
- Rao, K. R., Bredow, J. W., Manry, M. T., Devarajan, V., Karthikeyan, R., & Dillon, W. E. (2020). *Heterogeneous Transcoding for Next Generation Multimedia Video Codecs for Efficient Communication The members of the Committee approve the doctoral dissertation of SHREYANKA SUBBARAYAPPA*.
- Salam, H. U., Memo, S., Das, L., Rehman, A. U., & Hussain, Z. (2018). Drone Based Resilient Network Architecture for Survivals in Earthquake Zones in Pakistan. *Sindh University Research Journal -Science Series*, 50(001), 175–182. <https://doi.org/10.26692/surj/2018.01.0031>
- Sharma, J., Choudhury, T., Satapathy, S. C., & Sabitha, A. S. (2019). Study on H.265/HEVC against VP9 and H.264 : on space and time complexity for codecs. *Proceedings of the 2018 International Conference On Communication, Computing and Internet of Things, IC3IoT 2018*, 106–110. <https://doi.org/10.1109/IC3IOT.2018.8668132>
- Tolentino Medrano, N. A. (2021). *Diseño e implementación de un nodo VANET considerando un sistema de control disparado por eventos*.
- Tsbmail. (2003). *G.114 : Tiempo de transmisión en un sentido*. <https://www.itu.int/rec/T-REC-G.114-200305-I/es>
- Vyopta. (2019). *What's an Acceptable Amount of Packet Loss in 2019?* <https://www.vyopta.com/blog/video-conferencing/understanding-packet-loss/>
- Zhang, F., & Bull, D. R. (2014). *Measuring Video Quality*. 5, 227–249. <https://doi.org/10.1016/B978-0-12-420149-1.00007-7>
- ZhangTicao, & MaoShiwen. (2019). An Overview of Emerging Video Coding Standards. *GetMobile: Mobile Computing and Communications*, 22(4), 13–20. <https://doi.org/10.1145/3325867.3325873>