UNIVERSIDAD DE CUENCA

Facultad de Ingeniería

Carrera de Electrónica y Telecomunicaciones

Sistema de Comunicación Resiliente para Escenarios de Emergencia.

Trabajo de titulación previo a la obtención del título de Ingeniero en Electrónica y Telecomunicaciones.

Autores:

Paúl Alejandro Astudillo Picón

C.I: 010570845-7

paul_alejo7@outlook.es

Christian Patricio Quinde Romero

C.I: 010532038-6

christian.quinde@outlook.com

Director:

Ing. Santiago Renán González Martínez

C.I: 010389593-4

Co-Director:

Ing. Iván Santiago Palacios Serrano

C.I: 010516942-9

Cuenca - Ecuador
10 de marzo de 2022

Resumen

En escenarios de emergencia o rescate usualmente es difícil contar con una infraestructura de red, especialmente en lugares remotos o geográficamente complicados por 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.

Dentro del 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", se implementó una red ad hoc multisalto en el edificio matriz de la Empresa Eléctrica Regional Centro Sur.

En este sentido, en el presente trabajo de tesis se propone el desarrollo de un sistema de comunicación de emergencia para la transmisión de tráfico multimedia entre los diferentes pisos del edificio empleando la red ad hoc disponible.

Con 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 *open source* 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 determinaron a VP8 como el códec más adecuado para el escenario propuesto.

Además, se desarrolló una herramienta de videoconferencia con capacidad de medir el ancho de banda disponible en la red y seleccionar el *bitrate* adecuado para mantener una calidad que permite una buena comunicación.

Para el desarrollo se empleó el protocolo Real Time Transport Protocol (RTP) para transmitir y sincronizar los flujos de audio y vídeo. Además, se empleó el protocolo Real Time Transport Control Protocol (RTCP) para la medición de métricas como el *packet loss* y el *delay* utilizando campos propios de este protocolo presente en las tramas.

Keywords : Ad hoc. RTP. RTCP. Delay. Throughput. Packet loss. PSNR.

Abstract

In emergency or rescue scenarios it is usually difficult to have a network infrastructure in place, especially in remote or geographically difficult-to-access locations. However, it is necessary to have tools to communicate or assess the situation. In this sense, ad hoc networks emerge as an option to solve this problem.

Within the research project “IoT technologies and wireless sensor networks applied to structural health monitoring in essential buildings in the city of Cuenca”, a multi-hop ad hoc network was implemented in the main building of the Empresa Eléctrica Regional Centro Sur.

In this context, this thesis work proposes the development of an emergency communication system for the transmission of multimedia traffic between the different floors of the building using the available ad hoc network.

To this end, 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 to extract the required metrics and to objectively compare video quality using current encoding mechanisms (e.g. VP8, VP9, H.264, H.265). The results obtained determined VP8 as the most suitable codec for the proposed scenario.

In addition, a videoconferencing tool was developed to measure the bandwidth available in the network and to select the appropriate bitrate to maintain an acceptable communication quality.

The Real Time Transport Protocol (RTP) was used to transmit and synchronize audio and video streams. In addition, the Real Time Transport Control Protocol (RTCP) was used to measure metrics such as packet loss and delay using fields of this protocol present in the frames.

Keywords : Ad hoc. RTP. RTCP. Delay. Throughput. Packet loss. PSNR.

Índice general

Índice de figuras

Índice de tablas

Cláusula de Propiedad Intelectual

Yo, Paúl Alejandro Astudillo Picón, autor del trabajo de titulación' "Sistema de Comunicación Resiliente para Escenarios de Emergencia", certifico que todas las ideas, opiniones y contenidos expuestos en la presente investigación son de exclusiva responsabilidad de su autor.

Cuenca, 10 de marzo de 2022

Paúl Alejandro Astudillo Picón
010570845-7

Cláusula de Propiedad Intelectual

Yo, Christian Patricio Quinde Romero, autor del trabajo de titulación' "Sistema de Comunicación Resiliente para Escenarios de Emergencia", certifico que todas las ideas, opiniones y contenidos expuestos en la presente investigación son de exclusiva responsabilidad de su autor.

Cuenca, 10 de marzo de 2022

Christian Patricio Quinde Romero
010532038-6

Cláusula de licencia y autorización para publicación en el Repositorio Institucional

Yo, Paúl Alejandro Astudillo Picón en calidad de autor y titular de los derechos morales y patrimoniales del trabajo de titulación "Sistema de Comunicación Resiliente para Escenarios de Emergencia", de conformidad con el Art. 114 del CÓDIGO ORGÁNICO DE LA ECONOMÍA SOCIAL DE LOS CONOCIMIENTOS, CREATIVIDAD E INNOVACIÓN reconozco a favor de la Universidad de Cuenca una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos.

Asimismo, autorizo a la Universidad de Cuenca para que realice la publicación de este trabajo de titulación en el repositorio institucional, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

Cuenca, 10 de marzo de 2022

Paúl Alejandro Astudillo Picón
010570845-7

Cláusula de licencia y autorización para publicación en el Repositorio Institucional

Yo, Christian Patricio Quinde Romero en calidad de autor y titular de los derechos morales y patrimoniales del trabajo de titulación "Sistema de Comunicación Resiliente para Escenarios de Emergencia", de conformidad con el Art. 114 del CÓDIGO ORGÁNICO DE LA ECONOMÍA SOCIAL DE LOS CONOCIMIENTOS, CREATIVIDAD E INNOVACIÓN reconozco a favor de la Universidad de Cuenca una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos.

Asimismo, autorizo a la Universidad de Cuenca para que realice la publicación de este trabajo de titulación en el repositorio institucional, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

Cuenca, 10 de marzo de 2022

Christian Patricio Quinde Romero
010532038-6

Dedicatoria

"Esta tesis está dedicada a:

A mis padres Marcos y Cecilia ya que muchos de mis logros se los debo a ustedes, por motivarme constantemente para alcanzar mis metas y jamás rendirme. Ustedes son mi ejemplo a seguir.

A mis hermanos Marco y Pedro, ya que siempre me apoyaron, confiaron en mí y me dieron la fuerza para seguir adelante.

A mi gran amigo y compañero Christian, por su comprensión, sacrificio y de esta manera cumplir uno de nuestros objetivos.

A nuestros directores Santiago e Iván por la confianza, el apoyo y experiencia brindada.

Finalmente, a mis demás amigos por los consejos y experiencias en todo el trayecto universitario.

Paúl Astudillo

Dedicatoria

"Esta tesis está dedicada a:

A mis padres Patricio y Silvia, quienes con su carinño, apoyo y enseñanzas me han guiado a lo largo de toda mi vida. Todo lo que soy es gracias a ustedes.

Mis hermanos Andrés y Stephany, con los que siempre he contado y hemos compartido grandes momentos juntos.

Finalmente, dedico mi tesis a mis amigos, los cuales han estado presentes y me han apoyado siempre sin dudarlo."

Christian Quinde

Agradecimientos

Se agradece a la Universidad de Cuenca por la acogida y los conocimientos adquiridos que establecieron una base para el ingreso al mundo profesional. De la misma manera, un profundo agradecimiento al proyecto de investigación aprobado por la DIUC “Tecnologías IoT y Redes Inalámbricas de Sensores Aplicados a la Monitorización de Salud Estructural en Edificios Esenciales de la Ciudad de Cuenca” que nos permitieron realizar el trabajo investigativo haciendo uso de su red.

Finalmente un agradecimiento a los directores Santiago e Iván por sus aportes, sugerencias y la guía a lo largo del desarrollo de este trabajo de tesis.

LOS AUTORES

Abreviaciones y Acrónimos

ACK Acknowledge. 70
ACR Absolute Category Rating. 41
ADST Asymmetric Discrete Sine Transform. 39
AVC Advanced Video Coding. 54

CB Codification Block. 37
CBR Constant Bit Rate. 30, 54
CIF Common Intermediate Format. 33, 59
CPU Central Processing Unit. 55, 60–62, 81, 84, 86, 93
CSV Comma Separated Values. 61
CTB Coding Tree Block. 36, 37
CTU Coding Tree Unit. 36, 37
CU Codification Unit. 37, 38

DASH Dynamic Adaptive Streaming Over HTTP. 31
DCT Discrete Cosine Transform. 35, 36, 38–40
DLSR Delay since Last Sender Report. 44, 45, 66
DoACO Delays based on Ant Colony Optimization. 52
DSCQS Double-stimulus Continuous Quality Scale. 40, 41
DSDV Destination-Sequenced Distance Vector routing. 54
DSIS Double Stimulus Impairment Scale. 40, 41, 59
DST Discrete Sine Transform. 38

ESSID Extended Service Set Identification. 43

FANET Flying Ad hoc Networks. 22, 28
FEA FANET Emergency Application. 53
fps Frames Per Second. 32

GoP Group of Pictures. 35, 73, 74
GPL GNU Public License. 42

HD High Definition. 31
HEVC High Efficiency Video Coding. 37, 39
HLS HTTP Live Streaming. 31
HTTP Hypertext Transfer Protocol. 31

ICMP Internet Control Message Protocol. 71
IDS Intrusion Detection System. 42
IoT Internet of Things. 51
IP Internet Protocol. 43

JTRS Joint Tactical Radio System. 24

LSR Last Sender Report. 44, 66

MAMC-MAC Mobility-aware Multi-hop Clustering-based MAC. 52
MANET Mobile Ad hoc Network. 22–24, 27, 28, 52–54
MB Macrobloques. 34
MOS Mean Opinion Score. 72
MP-OLSR Multipath Routing Protocol based on OLSR. 53
MPD Media Presentation Description. 31
MPEG Moving Picture Experts Group. 34
MSE Mean Squared Error. 39

NAL Network Abstraction Layer. 34
NTP Network Time Protocol. 44, 45, 61, 77

OLSR Optimized Link State Routing. 54

PB Prediction Block. 38
PBM Protocol Bridging Mechanism. 52
PCAP Packet Capture. 61
PSNR Peak Signal-to-Noise Ratio. 39, 55, 60, 62, 84–86
PU Prediction Unit. 37

QoS Quality of Service. 28, 54
QP Quantization Parameter. 35

RF Radio Frequency. 22
RR Receiver Report. 44, 45, 66
RTCP Real Time Transport Control Protocol. 1, 4, 43, 44, 50, 66, 77, 93
RTP Real Time Transport Protocol. 1, 43–45, 50, 66, 93

SB Super Block. 39
SD Standard Definition. 31
SDES Source Description. 44
SDN Software Defined Network. 51
SR Sender Report. 44, 45, 50, 66
SRW Soldier Radio Waveform. 24
SSIM Structural Similarity. 39, 40
SSRC Synchronization source. 50
SVC Scalable Video Coding. 54

TB Transform Block. 38
TCP Transmission Control Protocol. 41–43
TDD Test-Driven Development. 23
TU Transform Unit. 37, 38

UAV Unmanned Aerial Vehicle. 53
UDP User Datagram Protocol. 43, 64, 79
UHDTV Ultra High Definition Television. 37
URL Uniform Resource Locator. 31
UWSN Underwater Wireless Sensor Networks. 22

V2I Vehicle To Infrastructure. 28
V2V Vehicle to Vehicle. 28, 52
VANET Vehicular Ad hoc Network. 22, 28, 52, 54
VBR Variable Bit Rate. 30
VCEG Video Coding Experts Group. 34
VCL Video Coding Layer. 34
VOD Video on Demand. 31
VQM Video Quality Metric. 40

WHT Walsh-Hadamard Transform. 36, 39
WMM Wi-Fi MultiMedia. 54
WMN Wireless Mesh Network. 4, 26, 27
WNW Wideband Network Waveform. 24
WSN Wireless Sensor Networks. 25, 28, 51, 53

XML Extensible Markup Language. 31

YUV Luminance (Y), blue–luminance (U), red–luminance (V). 59–61, 84

CAPÍTULO 1

Introducción

1.1. Definición del problema

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 [1]. Entre los ejemplos de sistemas que deben ser resilientes se encuentran los sistemas de control de procesos industriales, las redes de comunicación que apoyan la atención médica, servicios de emergencia y rescate, y los sistemas informáticos distribuidos que sustentan el control del tráfico aéreo.

La propia Internet es una infraestructura crítica, que soporta algunos servicios que pueden considerarse críticos [2]. Por lo tanto, los sistemas de comunicaciones, que son fundamentales en todos los ámbitos de desarrollo están expuestos a una serie de fallos y pérdidas de conectividad, como por ejemplo, ocasionados por eventos naturales (v.g. inundaciones, terremotos, entre otros), problemas derivados de la tecnología y actividades humanas maliciosas [3].

En consecuencia, es necesario enfrentar continuamente un amplio conjunto de retos que a menudo provocan fallos en los enlaces o nodos y en otros componentes del sistema, como se discute en [4]. En particular, de acuerdo a las estadísticas, alrededor del 60-70 % de los fallos se refieren a cortes de enlaces de comunicación individuales, por ejemplo, debido a cortes involuntarios de los enlaces por parte de terceros. Cuando los servicios de red que forman parte de una infraestructura crítica dejan de estar disponibles, se generan problemas de comunicación como resultado inevitable [3]. Para solventar este problema se puede hacer uso de redes resilientes basadas en el paradigma denominado redes ad hoc. Donde, acorde a [5], cada nodo cumple las funciones de enrutador y dispositivo terminal, es decir, un sistema descentralizado en el que cada nodo participa activamente en la red para reenviar los datos a los demás nodos, por lo que la determinación de los nodos que reenvían los datos se realiza dinámicamente en función de la conectividad de la red.

Por otra parte, debido a la gran diversidad de aplicaciones de usuario disponibles en la actualidad y en especial aquellas que generan tráfico multimedia (v.g. audio, video), los mecanismos de compresión de los

datos resultan fundamentales para optimizar los recursos de una red de comunicación. Por ejemplo, una mayor cantidad de tráfico generado por un video de alta calidad, requiere un mayor ancho de banda durante la transmisión y adicionalmente mayor espacio en disco para su almacenamiento, lo que resulta en un incremento de los costes. Por tal motivo, el desarrollo de códecs representa una línea activa de investigación y en constante evolución [6]. En la actualidad existen diversos códecs disponibles para diferentes casos de uso que compensan parámetros clave como la calidad, la latencia, el tamaño, el consumo de energía y el coste [7]. Los avances en la tecnología de compresión de video, contribuyen a reducir la demanda de recursos del sistema, como el tiempo de procesamiento, el uso de la memoria, el ancho de banda de la red y la duración de la batería. Esto es posible reduciendo la complejidad de los códecs de video sin comprometer la calidad del video de salida [8].

La presente propuesta de tesis se enmarca dentro del proyecto de investigación aprobado por la DIUC "Tecnologías IoT y Redes Inalámbricas de Sensores Aplicados a la Monitorización de Salud Estructural en Edificios Esenciales de la Ciudad de Cuenca". El trabajo se enfoca en la evaluación experimental de un sistema de comunicación robusto frente a eventos de emergencia. En particular, se propone un desarrollo a nivel de capa de aplicación que permita la transmisión de tráfico multimedia (audio/video). El escenario de estudio consistirá en el edificio matriz de la Empresa Eléctrica Regional Centro Sur, institución colaboradora en el mencionado proyecto de investigación, sobre el cual actualmente se encuentra desplegada una red ad hoc con topología multisalto.

El desarrollo del trabajo se basó en una metodología experimental, mediante el uso de herramientas de hardware/software de tipo open source. En concreto, se definieron las siguientes cincos etapas. Para empezar, se realizó un análisis de la literatura respecto a redes ad hoc multisalto. A continuación, se desarrollaron experimentos para caracterizar una red ad hoc multisalto en cuanto a métricas de red (v.g. *throughput*, *delay*, *packet loss*). En tercer lugar, se llevó a cabo un análisis de la literatura respecto a los mecanismos de compresión de video actuales (VP8, VP9, H264, H265) y métricas para la evaluación. Posteriormente, se diseñó una herramienta de software que permita la evaluación objetiva de la calidad de video, a partir de lo cual sea posible comparar y seleccionar el mecanismo de compresión más adecuado considerando las características de la red empleada. Finalmente, se implementó una aplicación para la transmisión de tráfico de audio y video en tiempo real con capacidad de adaptarse a las condiciones de la red y una segunda aplicación que permite evaluar objetivamente la calidad de una videollamada.

1.2. Objetivos

1.2.1. Objetivo general

Implementar un sistema de comunicación de emergencia para la transmisión de tráfico de audio/video.

1.2.2. Objetivos específicos

El presente trabajo de tesis tiene los siguientes objetivos específicos:

- Evaluar el rendimiento de una red descentralizada (ad hoc multisalto) para la transmisión de tráfico de video.
- Comparar la funcionalidad y rendimiento de los códecs H.264, H.265, VP8 y VP9.
- Desarrollar experimentos para la evaluación objetiva de la calidad del tráfico multimedia.
- Implementar una herramienta para la transmisión de audio y video en tiempo real.

1.3. Estructura del documento y contribuciones

La presente memoria se encuentra organizada en seis capítulos. En el Capítulo 2, Marco teórico, se presentan los conceptos fundamentales relacionados con el trabajo de tesis tales como, los fundamentos de redes ad hoc, tráfico multimedia, descripción de herramientas para el análisis de tráfico, los protocolos para la transmisión de flujos multimedia y los *frameworks* multimedia. A continuación, en el Capítulo 3, Trabajos relacionados, se describen los estudios más relevantes disponibles en la literatura, donde se discuten y analizan propuestas relacionadas al ámbito de redes de sensores y redes ad hoc, sistemas de comunicación resilientes, soporte de tráfico multimedia sobre redes ad hoc y mecanismos de codificación de video. Eventualmente, en el Capítulo 4, Metodología, se presenta el procedimiento utilizado para caracterizar la red ad hoc, evaluar objetiva y subjetivamente la calidad del video transmitido utilizando diferentes códecs de video, el desarrollo de una herramienta capaz de transmitir audio y video en tiempo real con la capacidad de adaptarse a las características de la red y una herramienta para medir métricas de la red y realizar un análisis estadístico. Posteriormente, en el Capítulo 5, Evaluación Experimental y Resultados, se presentan los resultados de la caracterización de la red ad hoc, la evaluación subjetiva y objetiva de la calidad de video, se define el códec más adecuado y la evaluación de la herramienta de transmisión de audio y video en tiempo real. Finalmente, en el Capítulo 6, Conclusiones, se detallan las principales conclusiones obtenidas al realizar este trabajo de tesis.

En cuanto a las contribuciones del trabajo de titulación, a continuación se resaltan los principales aportes:

- Caracterización de una red ad hoc multisalto sobre un escenario real en cuanto a métricas de *delay*, *throughput* y *packet loss*.
- Caracterización del tráfico multimedia sobre una red multisalto empleando mecanismos de codificación actuales.
- Desarrollo de una herramienta para transmitir video con las herramientas de GStreamer y FFmpeg.
- Desarrollo de una herramienta de software para evaluar objetivamente la calidad de video empleando métricas como el *delay*, *throughput*, *packet loss*, porcentaje de uso de CPU, PSNR y tiempo de envío y codificación.
- Desarrollo de una herramienta de software de videoconferencia con capacidad de análisis del ancho de banda disponible en la red, realizada con GStreamer.
- Desarrollo de una herramienta de software para la medición del *delay*, *throughput* y *packet loss* del tráfico de audio y video generado por la aplicación de videoconferencia usando los protocolos RTP y RTCP.

Capítulo 2

Marco teórico

En este capítulo se presentan los conceptos fundamentales relacionados con el trabajo de titulación. Se inicia con una descripción del paradigma de comunicación ad hoc, sus características, aplicaciones y tipos. Más adelante, se presentan los conceptos correspondientes al tráfico multimedia, sus características, tipos, mecanismos de compresión y evaluación de calidad. Seguidamente, se describen las herramientas para el análisis de tráfico y redes. Posteriormente, se describen algunos protocolos para la transmisión de flujos multimedia. Finalmente se describen ciertos *frameworks* multimedia.

2.1. Redes ad hoc

Las redes ad hoc están formadas por un conjunto de nodos que se comunican a través de un medio inalámbrico común. Además la comunicación se realiza sin una infraestructura, como una estación base o un punto de acceso cableado. Adicionalmente, el establecimiento de las redes debe hacerse de forma distribuida y descentralizada. Por lo tanto, la complejidad de las redes está en los propios nodos [9], los cuales deben ser capaces de resolver los problemas de la red como el enrutamiento y la seguridad.

2.1.1. Características generales

Como se mencionó anteriormente, las redes ad hoc son descentralizadas y no requieren de infraestructura para funcionar. Estas redes tienen un bajo coste y necesitan de un reducido tiempo para su despliegue. Además, a diferencia de las redes centralizadas, el diseño y la planificación son mínimos, o inclusive, en ciertos casos, no son necesarios. Los nodos actúan como *hosts* y también como dispositivos de enrutamiento, por lo que la red es multi-salto y los recursos son limitados [10]. La red puede hacerse operativa en diversos escenarios, redes MANET, Flying Ad hoc Networks (FANET), Underwater Wireless Sensor Networks (UWSN), Vehicular Ad hoc Network (VANET), etc [11].

En cuanto al alcance de la comunicación, la extensión de una red ad hoc se limita generalmente a su ámbito de aplicación, el mismo puede ser un área operativa, un edificio, un campus o una región geográfica muy pequeña para alguna aplicación específica. En las redes ad hoc los nodos tienen interfaces Radio Frequency (RF) y todas

las transmisiones y recepciones se realizan en la misma banda de frecuencia, es decir, ningún nodo realiza una traducción de frecuencias. Además todas las redes ad hoc operan en modo Test-Driven Development (TDD).

2.1.2. Ventajas

La independencia de cualquier infraestructura es la principal ventaja de las redes ad hoc. Por tal motivo, es posible desplegar una red ad hoc en cualquier situación complicada, como por ejemplo en casos de emergencia, como conflictos armados o desastres naturales. A continuación se mencionan algunas de las ventajas de este tipo de redes [9].

1. *Sin infraestructura y con menor coste:* Existen situaciones en las que un usuario de un sistema de comunicación no puede depender de una infraestructura. Además, desplegar una infraestructura puede resultar costoso para aplicaciones específicas en zonas con muy baja densidad, como el desierto, la montaña, una zona aislada o una edificación histórica o patrimonial. En consecuencia, al no existir una infraestructura tampoco hay costos de mantenimiento.
2. *Movilidad:* A diferencia de las redes con infraestructura, en las que el movimiento de los nodos está restringido por la frontera de la celda, en las redes ad hoc, un nodo puede moverse libremente mientras tenga conexión con otro nodo dentro de la red. Además, puede comunicarse con otros nodos pertenecientes a la misma red. Así pues, en situaciones como operaciones de emergencia, rescate o casos de catástrofe, no se puede confiar en la conectividad centralizada. Por ejemplo, las redes MANET admiten la movilidad de sus nodos y de esta manera es posible mantener la comunicación entre dispositivos móviles, siempre que el destino sea alcanzable.
3. *Descentralizada y robusta:* Otra ventaja de las redes ad hoc es su robustez. En este sentido, en las redes centralizadas, si por alguna razón una de las estaciones base no funciona, entonces, todos los usuarios de esa estación base perderán la conectividad con otras redes. En tal caso, es evidente la complejidad que implica reparar o sustituir de forma inmediata la infraestructura averiada, viéndose por tanto afectados los servicios y la conectividad. Por el contrario, en las redes ad hoc se puede evitar este problema. En particular, si un nodo abandona la red o simplemente no funciona correctamente, por ejemplo debido al agotamiento de energía o a un ataque informático, la conectividad en la red se puede mantener siempre que exista al menos un nodo que permita una ruta hacia el dispositivo de destino.

2.1.3. Desventajas

Los puntos débiles de los enlaces inalámbricos afectan directamente a una red tipo ad hoc. Por ejemplo, la baja velocidad de tranmisión, la seguridad y el control de acceso al medio son problemas comunes en las comunicaciones inalámbricas. A continuación se presentan algunas de las desventajas de dichas redes.

1. *Tasa de error más alta:* A diferencia de la transmisión por cable, la comunicación inalámbrica es susceptible a errores (v.g., pérdida de paquetes, alteración de datos) durante la propagación de las ondas. Por ejemplo, un obstáculo puede causar *shadowing*, reflexión, dispersión, desvanecimiento, refracción y difracción de la onda [12].
2. *Velocidad de datos más baja:* Uno de los mayores problemas de las redes ad hoc es la reducida velocidad de transmisión de datos. Las características propias del canal inalámbrico utilizado para la comunicación impiden que se transmitan los datos a la misma velocidad que en la comunicación por cable. Por otra parte, una frecuencia más alta permite transmitir datos a mayores velocidades; sin embargo, es más vulnerable a las interferencias y se reduce el alcance.

3. *Topología dinámica y escalabilidad:* Dado que las redes ad hoc no permiten los mismos protocolos de enrutamiento disponibles en las redes de Internet, son vulnerables a problemas de escalabilidad, lo que hace que las tablas de enrutamiento sean más grandes [13]. Como los nodos de las MANET son móviles, la topología cambia a medida que los nodos se desplazan, y por tanto, se requiere enviar mensajes de actualización de rutas con mayor frecuencia. Tal aumento del número de mensajes de control incrementa el volumen de tráfico en la red. En tal sentido, los protocolos de enrutamiento aplicados a redes ad hoc suelen estar diseñados para reducir el número de mensajes de control. Un buen algoritmo para redes ad hoc debe ser capaz de evaluar y comparar la escalabilidad relativa de las redes ante el aumento del número de nodos y la movilidad de los mismos. Es muy importante saber cuántos mensajes de control se necesitan para poder administrar el uso de ancho de banda [9].
4. *Seguridad:* Debido a la naturaleza dinámica y distribuida, sin infraestructura, y a la falta de puntos de control centralizados, las redes ad hoc son vulnerables a varios tipos de ataques. A diferencia del canal por cable, el canal inalámbrico es accesible tanto para los usuarios legítimos de la red como para los atacantes malintencionados. Por lo tanto, las redes ad hoc son susceptibles a sufrir ataques que van desde los pasivos, como las escuchas, hasta los activos, como las interferencias. Especialmente en el caso de las redes MANET, el consumo de energía y las capacidades de cálculo limitadas, debido a la limitación de la energía, hacen que no se puedan ejecutar algoritmos que requieran mucho cálculo, como los algoritmos de clave pública. El ataque pasivo significa que el atacante no envía ningún mensaje, el atacante solo escucha el canal, por lo que es casi imposible detectar este ataque. Por el contrario, los ataques activos modifican, borran los paquetes, inyectan paquetes a destinos no válidos.
5. *Consumo de energía:* Una red MANET permite a los nodos móviles comunicarse en ausencia de una infraestructura fija. Por lo tanto, funcionan con la energía de la batería. Debido a esta limitación, deben contar con algoritmos que sean eficientes desde el punto de vista energético, así como operar con recursos limitados de procesamiento y memoria.

2.1.4. Aplicaciones

En esta sección se destacan los principales ámbitos de aplicación de las redes ad hoc.

2.1.4.a. Segmento militar

Estas redes ad hoc suelen ser muy grandes en cuanto al número de nodos participantes, por tanto es necesario garantizar la escalabilidad del protocolo en cuanto al número de nodos. Además, se requiere considerar métricas tales como el retardo y el consumo de energía. Pero lo más importante es tener en cuenta la naturaleza de la propia red, que suele estar formada por "subredes". Por ejemplo, las redes inalámbricas tácticas construidas con el Joint Tactical Radio System (JTRS) tienen capas de subredes. En tal sentido, existe el nivel de Soldier Radio Waveform (SRW), el mismo que puede tener dos subniveles, uno para las comunicaciones entre soldados y otro para la conexión en red de los sensores. Por encima, está el nivel Wideband Network Waveform (WNW), que a su vez tiene dos subniveles; uno forma subredes locales para las comunicaciones de vehículo a vehículo, y el otro es para la conectividad global, para generar una única subred en todo el territorio [14].

2.1.4.b. Emergencias y rescate

Este tipo de red se usa en situaciones de emergencia cuando la red convencional o la comunicación móvil falla o se pierde debido a un desastre natural. Es de gran utilidad para comunicarse con las personas que necesitan

algún tipo de asistencia o rescate [15].

2.1.4.c. Redes de sensores

Las Wireless Sensor Networks (WSN) representan un tipo especializado de red ad hoc, sus principales campos de aplicación se esquematizan en la Figura 2.1.

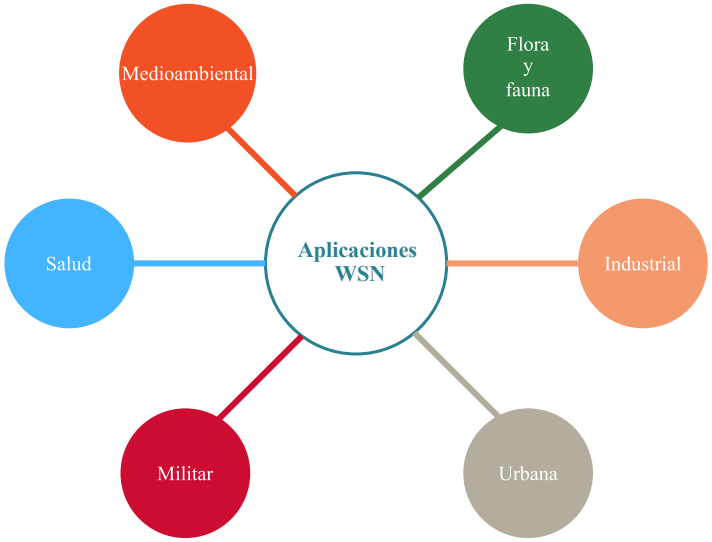


Figura 2.1: Categorías más populares de las aplicaciones de WSN.

En el ámbito de la salud, las WSN utilizan sensores médicos avanzados para monitorizar a los pacientes dentro de un centro de salud, como un hospital o dentro de su casa, así como para proporcionar una monitorización en tiempo real de las señales vitales de los pacientes mediante el uso de hardware portátil. Por otro lado, las aplicaciones medioambientales que exigen una monitorización continua de las condiciones ambientales en zonas hostiles y remotas pueden mejorarse con la utilización de las WSN.

Las principales subcategorías de aplicaciones medioambientales de las WSN, son la monitorización de la calidad del agua, del aire y la alerta de emergencia en casos de sismos, actividad volcánica o detección de Tsunami. Adicionalmente, las WSN se pueden emplear en diversas aplicaciones industriales. Las principales subcategorías de aplicaciones industriales en cuanto a lógistica, robótica y la monitorización del estado de la maquinaria.

En el ámbito urbano, la variedad de capacidades de detección que ofrecen las WSN también brinda la oportunidad de obtener un nivel de información sobre un área de interés, ya sea en ambientes interiores como por ejemplo edificaciones o en exterior. En particular, las WSN constituyen una base tecnológica para la caracterización de ambientes urbanos, lo que ofrece un número ilimitado de aplicaciones [16]. Adicionalmente, existes diversas aplicaciones donde los vehículos constituyen los nodos de la red, entre las más destacables se tiene:

2.1.4.d. Seguridad en las carreteras

Estas aplicaciones son aquellas que reducen el número de víctimas relacionadas con la conducción y los accidentes. Algunos ejemplos son la respuesta a emergencias, la asistencia de cambio de carril, el aviso de colisión frontal, luz de freno de emergencia electrónica, la notificación automática de accidentes, el seguimiento de vehículos robados, seguimiento de delincuentes conocidos.

2.1.4.e. Eficiencia vial

Son aquellas que reducen la dificultad de la tarea de conducir, entre las cuales están el sistema de notificación de tráfico, el diagnóstico remoto de vehículos, la predicción de la topología de las carreteras, la evaluación del entorno, el cobro automático de peajes y la localización de plazas de aparcamiento.

2.1.4.f. Información y entretenimiento

Las aplicaciones de publicidad que aconsejan al conductor sobre tiendas locales y opciones de entretenimiento, así como las aplicaciones que proporcionan algún tipo de entretenimiento a los ocupantes del vehículo, es decir, se crea una red ad hoc entre las tiendas y los vehículos [17] [18].

2.2. Tipos de redes ad hoc

En esta sección se presenta una descripción de los tipos de redes ad hoc con sus características, ventajas, desventajas y aplicaciones. En la Figura 2.2, se observa la clasificación de estas redes.

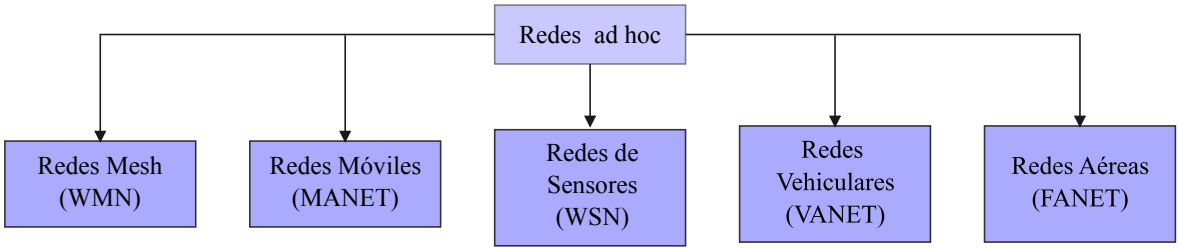


Figura 2.2: Clasificación de redes ad hoc.

2.2.1. Red Mesh (WMN)

La red inalámbrica *Mesh* (WMN) es una red que utiliza múltiples saltos para enviar paquetes entre el origen y el destino. Específicamente, se compone de routers tipo *mesh* y clientes mesh, tal como se observa en la Figura 2.3. Estas redes contienen un *gateway* con acceso a Internet, haciéndolas compatibles con tecnologías existentes como redes cableadas o inalámbricas y proporcionando simultáneamente comunicaciones *peer-to-peer*, accesos *backhaul* y servicios de otras redes inalámbricas. La WMN puede construirse utilizando varios tipos de tecnologías de radio, incluyendo el estándar IEEE 802.11 [19].

Los *routers* de malla tienen una movilidad mínima y ayudan a transferir paquetes hacia y desde los clientes, formando una red troncal para los clientes. Por otro lado, los clientes pueden funcionar como *routers* pero carecen de la funcionalidad de un puente *gateway* considerándolos dispositivos móviles o estacionarios de usuario final como pueden ser ordenadores portátiles o teléfonos móviles [20].

La reducción de costes en tarjetas inalámbricas ha impulsado el despliegue de redes tipo *mesh*, esto sumado a la facilidad en cuanto a mantenimiento, cobertura fiable, comunicación robusta y redundante, representan algunas de sus principales ventajas [20]. Otra ventaja valiosa de las redes tipo *mesh* es que son fáciles de construir en un entorno cambiante, debido a que se puede actualizar la red de comunicaciones existente. Específicamente, la actualización se realiza agregando nodos adicionales a la red de comunicación existente. Por otra parte, se presentan algunos desafíos entre los cuales están por ejemplo, la potencia de transmisión la cual debe ser mayor o igual que la de las redes ya existentes, medición y modelado de interferencias, asignación de canales/radios,

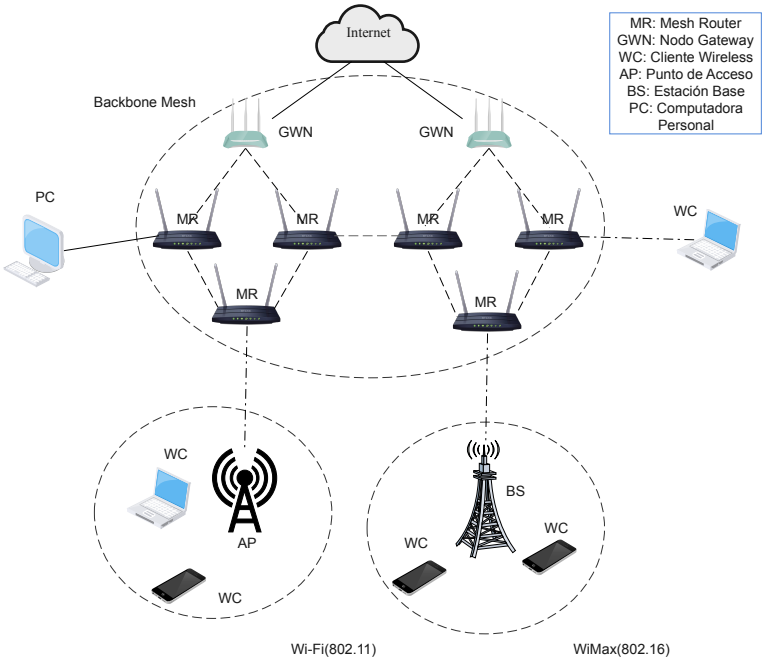


Figura 2.3: Arquitectura de una red WMN.

modelado de rendimiento y análisis de capacidad [21].

Además, cabe mencionar que, dado que las WMN proporcionan una conectividad fiable, rápida e inalámbrica, las mismas pueden utilizarse en lugares donde se encuentran grandes multitudes. Entre algunos escenarios de aplicación, se pueden destacar:

- Red doméstica de banda ancha
- Información sobre catástrofes y redes de emergencia
- Red empresarial
- Red de área metropolitana
- Automatización de edificios
- Sistema de transporte
- Salud y ciencias médicas
- Seguridad y vigilancia

2.2.2. Red móvil (MANET)

Una Mobile Ad hoc Network (MANET) es un grupo autónomo de usuarios móviles que se comunican a través de enlaces inalámbricos. La topología de la red puede variar rápidamente y de forma aleatoria, ya que los nodos son móviles. Esta red es autoconfigurada y descentralizada ya que los nodos operan como *routers* y *hosts* permitiendo el descubrimiento de la topología y la entrega de mensajes. Como se observa en la Figura 2.4, los nodos móviles pueden comunicarse directamente con los nodos que están en el rango de cobertura de cada uno, mientras que otros nodos necesitan la ayuda de nodos intermedios para enrutar sus paquetes, con lo que se puede realizar varios saltos. En estas redes es necesario un procedimiento de enrutamiento para encontrar un camino que permita reenviar los paquetes de forma adecuada entre el origen y el destino [22].

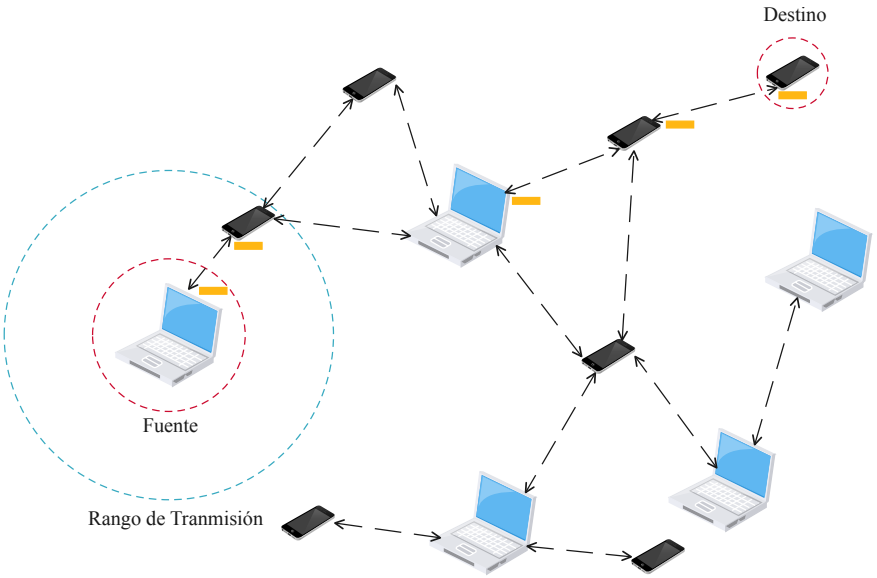


Figura 2.4: Arquitectura de una red MANET.

La principal ventaja de este tipo de redes es que están totalmente distribuidas y pueden funcionar en cualquier lugar sin la ayuda de una infraestructura. Esta propiedad hace que estas redes sean muy robustas, y además cuentan con una instalación rápida con la menor intervención posible del usuario. A su vez, el mayor desafío es el enrutamiento, debido al rápido cambio en la topología de la red y con diferentes velocidades de movilidad. Por lo que dichas redes requieren un control de Quality of Service (QoS), especialmente en el caso de tráfico multimedia. Así mismo, otro desafío de estas redes es la seguridad ya que los datos transferidos de un nodo a otro deben enviarse de forma segura y completa [23]. En particular, las MANET varían según su aplicación:

- Red vehicular ad hoc (VANET)
  Los vehículos se utilizan como nodos para crear una red móvil proporcionando información relacionada y gestionando el tráfico. Hay dos tipos de comunicación en las VANET: de Vehicle to Vehicle (V2V) y Vehicle To Infrastructure (V2I) [24].
- Red ad hoc aérea (FANET)
  Las FANET están formadas por drones que proporcionan movilidad y conectividad a zonas remotas.

2.2.3. Red de Sensores (WSN)

Es un tipo especial de red ad hoc, consistente en un grupo de nodos sensores dispersos, los cuales están interconectados mediante el uso de comunicación inalámbrica orientada a la captura y monitorización de eventos y variables físicas en tiempo real.

En cuanto a la arquitectura, tal como se observa en la Figura 2.5, contempla un elemento de enlace o *gateway* que permite la comunicación entre la aplicación anfitriona y los dispositivos de campo [25].

La principal ventaja de este tipo de redes es que permite a los usuarios e investigadores acceder fácilmente a los datos y experimentar con diferentes configuraciones de la infraestructura objeto de estudio, dicha monitorización se ejecuta durante un largo tiempo (hasta varios años) [26]. Otro aspecto de estas redes, es que los nodos sensores necesitan energía para detectar, procesar y transmitir información, pero su energía es limitada ya que se alimentan con baterías, siendo uno de los desafíos de estas redes [27]. Además, las WSN se componen de

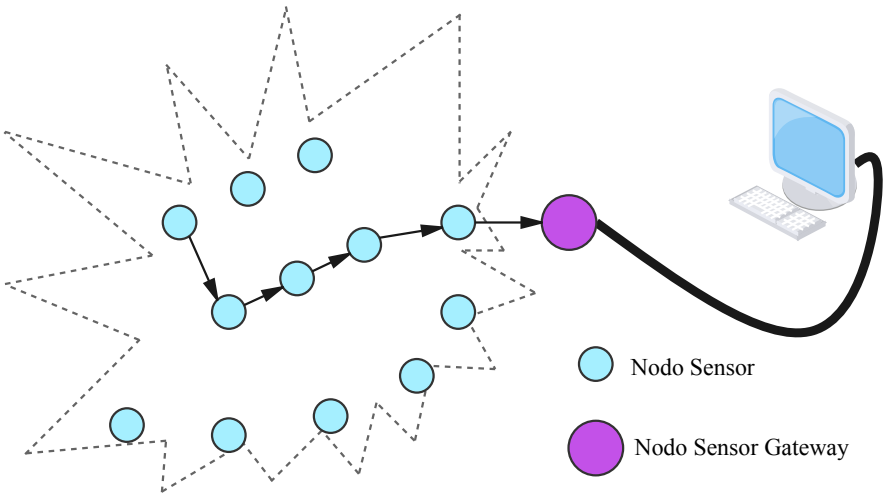


Figura 2.5: Arquitectura WSN.

múltiples sensores que pueden monitorear situaciones ambientales como por ejemplo:

- Humedad
- Presión
- Luz
- Nivel de audio
- Temperatura

También interviene en escenarios tales como:

- Aplicaciones militares
  - Equipo de control y munición
- Aplicaciones medioambientales
  - Seguimiento de las condiciones que afectan a la plantación
- Aplicaciones de salud
  - Datos de telemonitorización relacionados con la fisiología humana
- Aplicaciones en el hogar
  - Automatización inteligente
- Aplicaciones comerciales
  - Vigilancia de la zona de la catástrofe

2.3. Tráfico multimedia

En este apartado se describen conceptos fundamentales relacionados al ámbito de las comunicaciones multimedia requeridos en el presente trabajo de titulación.

2.3.1. Características del tráfico de video

La tasa de bits constante (Constant Bit Rate (CBR)) es el tipo de tráfico más común en las redes inalámbricas. En CBR, la fuente crea paquetes de datos de tamaño fijo a una velocidad constante. Por el otro lado, con la tasa de bits variable (Variable Bit Rate (VBR)), el tamaño de los paquetes y la tasa de transmisión varían según los requisitos del escenario multimedia/video. Es importante resaltar que el tráfico de video es variable y usualmente se caracteriza por intervalos de envío de una gran cantidad de tráfico a manera de ráfagas (*burst*), seguido de intervalos con tráfico reducido, dicho comportamiento se aprecia en la Figura 2.6.

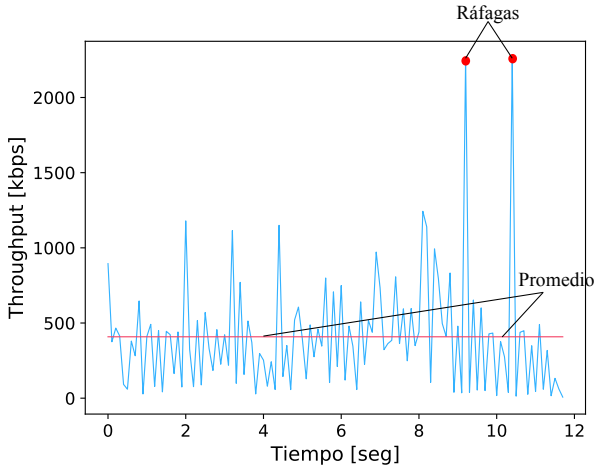


Figura 2.6: Comportamiento de una transmisión de video utilizando VBR.

2.3.2. Tipos de aplicaciones multimedia

Dentro de las aplicaciones multimedia existen diferentes tipos de servicios de transmisión de video. Estos se diferencian de acuerdo a la aplicación específica a la que van dirigidos [28]. A continuación se listan algunos de los más relevantes.

- *Live Action:* En este caso la secuencia de video corresponde a una grabación de objetos fijos o en movimiento. Se trata de personas y lugares reales, en vez de ser creados digitalmente.
- *Animación:* La animación es un estilo de video realizado a partir de muchas imágenes. Cuando las imágenes se reproducen una tras otra dan la ilusión de movimiento.
- *Live Streaming:* Consiste en la transmisión de video en vivo a los espectadores a través de plataformas tales como Youtube, Facebook, Instagram, entre otras. En la actualidad este tipo de servicio genera un volumen considerable del tráfico en Internet, con previsiones de que alcanzarán más del 60 % del tráfico mundial en 2022 [29].

2.3.3. Requerimientos de ancho de banda

Uno de los aspectos que convierten el tráfico de video en un reto tanto para los proveedores de contenidos como para la infraestructura de red, es su exigencia de ancho de banda. En la Tabla 2.1, se muestran los distintos anchos de banda recomendados para diferentes resoluciones de video [29].

Tabla 2.1: Ancho de banda recomendado para diferentes resoluciones de video.

Resolución	**Ancho de Banda [Mbps]**
Standard Definition (SD)	3
High Definition (HD)	5
Ultra HD	25

En respuesta a estos requisitos, los principales proveedores de *streaming* como Youtube y Netflix han invertido en una nueva tecnología denominada *streaming* adaptativo. La transmisión dinámica adaptativa sobre Hypertext Transfer Protocol (HTTP) o Dynamic Adaptive Streaming Over HTTP (DASH), ha surgido como un estándar para la transmisión de video por Internet. Este cuenta con una serie de mecanismos de adaptación del bitrate para los sistemas DASH con el fin de ofrecer una calidad de video que se ajuste al *throughput* de las condiciones dinámicas de la red para obtener una mejor calidad de experiencia [30]. DASH define dos formatos básicos: el Media Presentation Description (MPD), que utiliza Extensible Markup Language (XML) para proporcionar un manifiesto del contenido disponible, sus diversas alternativas, sus direcciones Uniform Resource Locator (URL) y otras características; y los *Segments*, que contienen los flujos de datos multimedia segmentados en forma de trozos ya sea en ficheros individuales o múltiples [31].

Por otro lado, Apple desarrolló su propio protocolo de transmisión adaptativo en 2009, denominado HTTP Live Streaming (HLS), el cual envía audio y video a través de HTTP desde un servidor web ordinario para su reproducción en dispositivos basados en iOS como iPhone, iPad, iPod touch y Apple TV y en ordenadores de escritorio. Utilizando el mismo protocolo que impulsa la web, HLS despliega contenidos mediante el uso servidores web ordinarios y redes de distribución de contenidos. HLS está diseñado para ser fiable y se adapta dinámicamente a las condiciones de la red optimizando la reproducción para la velocidad disponible de las conexiones por cable e inalámbricas [32]. Además, HLS es compatible con lo siguiente:

- Emisiones en directo y contenidos pregrabados (video bajo demanda o VOD)
- Múltiples flujos alternativos a diferentes velocidades de bits
- Conmutación inteligente de flujos en respuesta a los cambios de ancho de banda de la red
- Encriptación de medios y autenticación de usuarios

En la Figura 2.7 se muestran los componentes de una transmisión HLS.

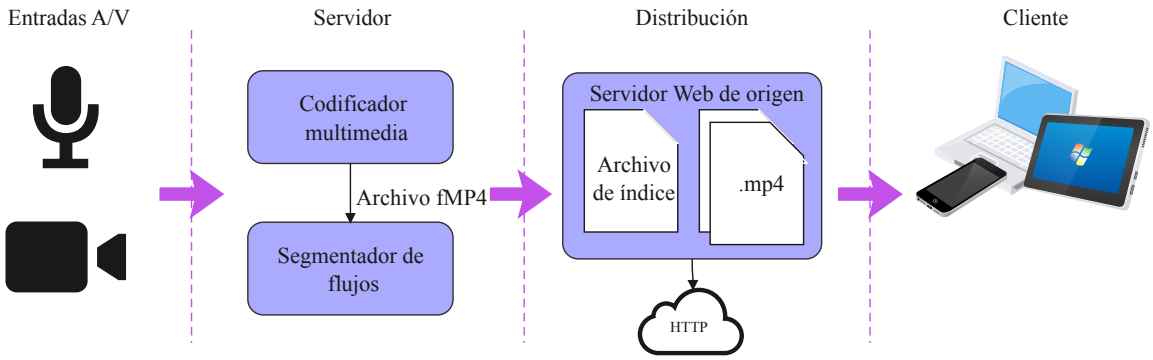


Figura 2.7: Componentes de una transmisión HLS.

2.3.4. Mecanismos de compresión de video

Un video consiste en una serie de imágenes digitales discretas exhibidas en una sucesión rápida con una magnitud fija. En los videos la magnitud con la que se muestran los fotogramas se mide en cuadros por segundo o Frames Per Second (fps). Un fotograma es una imagen digital que se compone de tramas de píxeles. Un píxel es un pequeño cuadrado con una sola propiedad llamada color. Por lo tanto, un cuadro con W píxeles de ancho y de H píxeles de alto tiene un tamaño de cuadro WxH píxeles [33]. Por otro lado, la compresión de video es el proceso de convertir el video digital en un formato que ocupa menos capacidad cuando se almacena o transmite [34]. Para lo cual existen dos tipos de técnicas de compresión de video: compresión con pérdidas y compresión sin pérdidas.

En la compresión sin pérdidas, según [35] se considera un flujo de símbolos $x_1, \ldots, x_N$, donde cada x_i es un elemento del mismo conjunto finito $\mathcal{X}$. El flujo se obtiene extrayendo cada símbolo x_i de forma independiente de la misma distribución $\tilde{p}$, es decir, las x_i son i.i.d (independientes e idénticamente distribuidas) según $\tilde{p}$. Se necesita codificar el flujo de símbolos en un flujo de bits, de forma que podamos recuperar los símbolos exactos mediante decodificación. En esta configuración, la entropía de $\tilde{p}$ es igual al número esperado de bits necesarios para codificar cada x_i :

$$H(\tilde{p}) = \text{bits}\,(x_i) = \mathbb{E}_{x_i \sim \tilde{p}}\left[-\log_2 \tilde{p}\,(x_i)\right] \tag{2.1}$$

En general, sin embargo, el $\tilde{p}$ exacto es desconocido, y en su lugar se considera la configuración en la que existe un modelo aproximado p. Entonces, la tasa de bits esperada será igual a la entropía cruzada entre $\tilde{p}$ y p, dada por:

$$H(\tilde{p}, p) = \mathbb{E}_{x_i \sim \tilde{p}}\left[-\log_2 p\,(x_i)\right] \tag{2.2}$$

De aquí se deduce que, cuanto mayor sea la diferencia entre el modelo p utilizado para la codificación y el modelo real $\tilde{p}$, entonces se necesitarán más bits para codificar los datos que realmente se distribuyen según $\tilde{p}$.

Por el contrario, la compresión con pérdidas puede conseguir una mayor tasa de compresión eliminando información innecesaria que no es obvia para los espectadores y que no cambiará la calidad subjetiva de la señal de video decodificada. Se emplea en datos que contienen muchas redundancias y son insensibles a las pérdidas. En la compresión con pérdidas se destruye parte de la información, es decir, los datos originales no se pueden recuperar con esta técnica [33].

2.3.4.a. Espacios de color

La mayoría de las aplicaciones de video digital dependen de la visualización de video en color, por lo que necesitan un mecanismo para capturar y representar la información en color. Una imagen monocromática solo necesita un número para indicar el brillo o la luminancia de cada muestra espacial. Las imágenes en color, en cambio, necesitan al menos tres números por posición de píxel para representar el color con precisión. El método elegido para representar el brillo, la luminancia o la luma y el color se describe como un espacio de color [36], el cual puede ser RGB o YCrCb. En el espacio de color RGB, una muestra de imagen en color se representa con tres números que indican las proporciones relativas de rojo, verde y azul, los tres colores primarios aditivos de la luz. En este contexto, al combinar el rojo, el verde y el azul en distintas proporciones se puede crear cualquier color.

Sin embargo, el sistema visual humano es menos sensible al color que a la luminancia. En el espacio de color RGB, los tres colores tienen la misma importancia, por lo que suelen almacenarse con la misma resolución, pero es posible representar una imagen en color de forma más eficiente separando la luminancia de la información de color y representando la luma con una resolución mayor que el color. El espacio de color Y:Cr:Cb es una forma popular de representar eficazmente las imágenes en color. En particular, "Y" es el componente de luminancia y puede calcularse como una media ponderada de R, G y B:

$$Y = k_r R + k_g G + k_b B \tag{2.3}$$

Donde k son los factores de ponderación. La información de color puede representarse como componentes de diferencia de color (crominancia o croma), donde cada componente de crominancia es la diferencia entre R, G o B y la luminancia Y:

$$\begin{aligned} Cr &= R - Y \\ Cb &= B - Y \\ Cg &= G - Y \end{aligned} \tag{2.4}$$

La descripción completa de una imagen en color viene dada por Y, el componente de luminancia, y tres diferencias de color Cr, Cb y Cg que representan la discrepancia entre la intensidad del color y la luminancia media de cada muestra de la imagen. Sin embargo, Cr+Cb+Cg es una constante, por lo que solo es necesario almacenar o transmitir dos de los tres componentes de crominancia, ya que el tercer componente siempre puede calcularse a partir de los otros dos.

Por otro lado, Common Intermediate Format (CIF) es un formato estandarizado para la resolución de imagen, la frecuencia de imagen, el espacio de color y el submuestreo de color de las secuencias de video digital utilizadas en los sistemas de videoconferencia. Se definió por primera vez en la norma H.261 en 1988. CIF es una forma de codificación de video de menor resolución y una opción para aplicaciones de menor resolución, en contraste con los resultados de mayor resolución en megapíxeles [37].

2.3.4.b. Reducción de la redundancia

La señal de video digital contiene información similar y correlacionada entre los píxeles y fotogramas vecinos, lo cual es ideal para la compresión mediante la eliminación o reducción de la redundancia [38]. Entre las formas de redundancia se tiene:

- Redundancia espacial
  La redundancia espacial es la consecuencia de la correlación en las dimensiones espaciales horizontal y vertical entre los valores de los píxeles vecinos dentro de la misma imagen o fotograma de video, también conocida como correlación *intraframe*.
- Redundancia temporal
  La redundancia temporal se debe a la correlación entre diferentes imágenes o cuadros de un video, también conocida como correlación entre imágenes o *interframe*. Es decir, los cambios producidos entre imágenes son mínimos.
  En la Figura 2.8 se muestra un ejemplo de redundancia espacial y temporal.

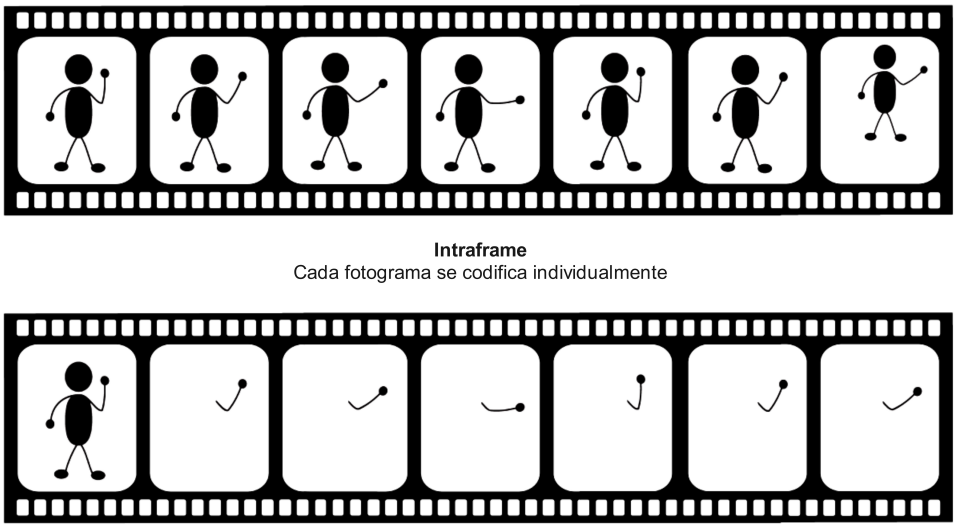


Figura 2.8: Redundancia espacial y temporal.

A continuación se detallan las principales diferencias entre los códecs H.264/AVC, VP8, H.265/HEVC y VP9, los cuales utilizan compresión con pérdidas.

2.3.4.c. Códec H.264/AVC

H.264 es el estándar de codificación de video más importante. El mismo se desarrolló en el periodo comprendido entre 1999 y 2003, y luego se amplió de 2003 a 2009. Dicho códec fue desarrollado por el ITU-T Video Coding Experts Group (VCEG) junto con el ISO/IEC JTC1 Moving Picture Experts Group (MPEG) [39]. En H.264 la capa Video Coding Layer (VCL) representa de manera eficiente el contenido de los datos de video. Mientras que la capa Network Abstraction Layer (NAL) se encarga de dar formato a esos datos y agregar información de cabecera de forma adecuada para su transporte o almacenamiento [34]. H.264 utiliza la codificación de video híbrida con compensación de movimiento basada en bloques, pero con algunas diferencias importantes respecto a los estándares anteriores. A continuación se presentan algunas de las diferencias más importantes:

- Capacidad de predicción de movimiento mejorada
- Uso de una transformación de coincidencia exacta con un tamaño de bloque pequeño
- Filtro de desbloqueo adaptativo en bucle
- Métodos de codificación de entropía mejorados

En la Figura 2.9 se muestra el proceso de codificación y decodificación utilizando el estándar H.264. Este proceso consiste de tres etapas: Predicción, Transformación y Codificación. Después de la transmisión o almacenamiento es necesario realizar el proceso inverso empezando por la decodificación, transformación inversa y reconstrucción. Cabe mencionar que la versión decodificada, en general no es idéntica a la secuencia original debido a que H.264 es un mecanismo de compresión con pérdidas.

Con respecto a la primera etapa, el códec H.264 realiza predicciones intra-trama aprovechando las redundancias espaciales que existen. Se emplea tamaños de Macrobloques (MB) de 16x16 y de 4x4 muestras, donde

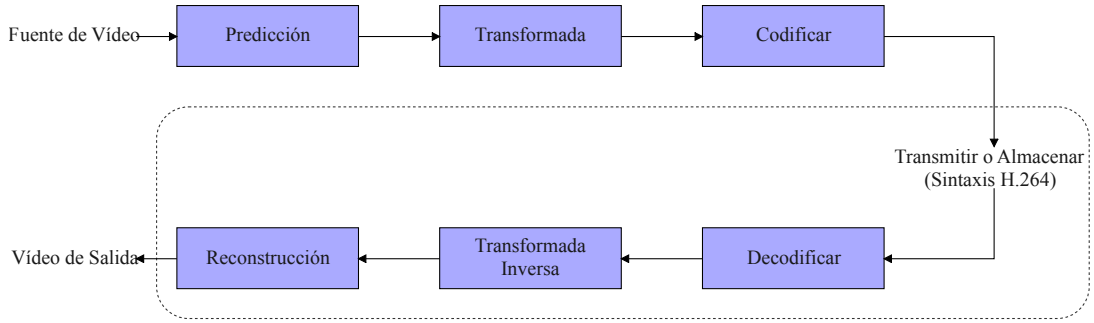


Figura 2.9: Proceso de codificación y decodificación utilizando el estándar H.264.

el tamaño del MB se define acorde a las características de la imagen (homogeneidad) [36]. En cambio, las predicciones inter-trama, utilizan las tramas previas o futuras. El estándar H.264 utiliza tamaños de bloque variables para compensar el movimiento, referencias de múltiples fotogramas para la predicción, predicción generalizada de fotogramas B y uso de fotogramas B como referencias [34].

En cuanto a la segunda etapa, el bloque *Transformada*, las muestras son transformadas utilizando bloques de 8x8 o 4x4 haciendo uso de la Discrete Cosine Transform (DCT). La DCT es una transformación muy conocida en matemáticas, que esencialmente transforma una señal del dominio espacial al dominio de la frecuencia. Descompone una señal en una serie de funciones armónicas de coseno. Debido a su propiedad de descorrelación y a su mejor compactación de la energía, se ha utilizado en muchas aplicaciones, como la compresión de datos y el reconocimiento de patrones [40]. A continuación, los coeficientes del bloque transformado son cuantificados. Para ello se divide cada coeficiente por un valor entero denominado QP o tamaño de paso del cuantificador [41]. Por tanto, la cuantificación reduce la precisión de los coeficientes transformados acorde al parámetro QP.

En el bloque *Codificar* se realiza la codificación generando una cadena de bits comprimida. Para generar la cadena de bits, los mecanismos usuales en H.264 son: *Variable Length Coding* y/o *Arithmetic Coding* [42]. Además, el estándar H.264 especifica un conjunto de perfiles, donde cada uno define parámetros, herramientas y características disponibles en el estándar. Un perfil establece límites sobre las capacidades del codificador y decodificador. Por lo tanto, cada perfil se relaciona con un tipo de aplicación [36].

- *Baseline Profile*: Aplicaciones de videoconferencia con bajo retardo.
- *Main Profile*: Aplicaciones de entretenimiento como servicios de TV (SD).
- *High Profiles*: Aplicaciones de video en alta resolución (HD, FHD, UHD).

Ahora, dependiendo del tipo de perfil H.264 el codificador hace uso de tres tipos de fotogramas: “I”, “P” y “B” o Group of Pictures (GoP). Donde el fotograma “I” contiene la información completa de una imagen muestreada, por lo que no requiere de otros fotogramas de video para decodificarlo. El fotograma “P” utiliza los datos de los fotogramas anteriores para descomprimir. Por último, el fotograma “B” utiliza los fotogramas anteriores y los posteriores como referencia de datos para comprimir la mayor cantidad de datos [43].

2.3.4.d. Códec VP8

En comparación con otros formatos de codificación de video, VP8 tiene muchas características técnicas distintivas que le permiten lograr una alta eficiencia de compresión y una baja complejidad computacional para

la decodificación al mismo tiempo. En mayo de 2010, Google anunció el inicio de un nuevo proyecto de medios abiertos "WebM", dedicado a desarrollar un formato de medios abiertos de alta calidad para la web que esté disponible de forma gratuita para todos los usuarios [44]. De aquí surgió el nuevo formato de compresión de video de código abierto, VP8.

Los desarrolladores de VP8 se enfocaron en aplicaciones de video basadas en Internet, De manera que, desde el comienzo se plantearon algunos requerimientos básicos. El primero es tener un bajo ancho de banda, ya que uno de los supuestos básicos del diseño es que, en un futuro previsible, el ancho de banda disponible en la red será limitado. El segundo requerimiento es que el cliente puede tener un *hardware* heterogéneo. Es decir, crear implementaciones eficientes para una amplia gama de dispositivos. El último requerimiento es la utilización del formato de video web. La búsqueda de la eficiencia de la compresión y la simplicidad del decodificador condujo a una serie de características técnicas distintivas en VP8, en relación con otros formatos de compresión de video como H.264/AVC. A continuación, se destacan las innovaciones técnicas de VP8 [45] y [46]:

- **Transformación híbrida con cuantificación adaptativa:** VP8 utiliza una transformada de coseno discreto (DCT) de 4x4 bloques para toda la señal residual de luma y croma. Dependiendo del modo de predicción, los coeficientes DCT de un macrobloque de 16x16 pueden someterse a una transformación Walsh-Hadamard de 4x4. Mientras que H.264/AVC solo utiliza coeficientes cuyo valor absoluto es cero o una potencia de dos, VP8 modela coeficientes DCT con mayor precisión, lo que lleva a coeficientes como, por ejemplo, 20091 o 35468.
- **Marcos de referencia flexibles:** VP8 utiliza tres marcos de referencia para la interpredicción, pero el esquema es algo diferente del esquema de compensación de movimiento de referencia múltiple visto en otros formatos. El diseño de VP8 limita el tamaño del búfer a tres búferes de marcos de referencia y mantiene una descorrelación efectiva en la compensación de movimiento.
- **Tipos de Fotogramas** El equivalente en VP8 a los conocidos fotogramas "I" se denomina fotogramas clave. Además, solo se definen los fotogramas que coinciden con los fotogramas tipo "P". La ausencia de fotogramas "B" distingue a VP8 de H.264 .
- **Macrobloques y modos de predicción:** H.264/AVC proporciona una predicción ponderada de un solo MB basada en múltiples fotogramas, mientras que VP8 solo admite la referencia de un fotograma por MB. A diferencia de H.264/AVC, que ofrece una precisión de subpíxeles de 1/4, VP8 ofrece una precisión de 1/8 píxeles para el componente de luma que aumenta con la tasa de submuestreo de los componentes de croma.

El proceso de codificación de VP8 sigue cinco pasos, tal como se observa en la Figura 2.10. En primer lugar, el macrobloque a codificar recibe un predictor. El predictor se resta del macrobloque y a continuación, el macrobloque se transforma utilizando la Transformada Discreta del Coseno (DCT) o la Transformada de Walsh-Hadamard Transform (WHT), dependiendo del modo de predicción. Tras la transformación, el paso de cuantificación toma los resultados del proceso de transformación y los restringe a un determinado rango de salida [47].

2.3.4.e. Códec H.265/HEVC

A diferencia de H.264, H.265 desarrolla un modelo de predicción espacio-temporal híbrido mejorado que utiliza 35 modos direccionales con una unidad de árbol de codificación (Coding Tree Unit (CTU)) de un tamaño de bloque de hasta 64×64. La CTU se divide a su vez en bloques de 64×64, 32×32 o 16×16. El (Coding

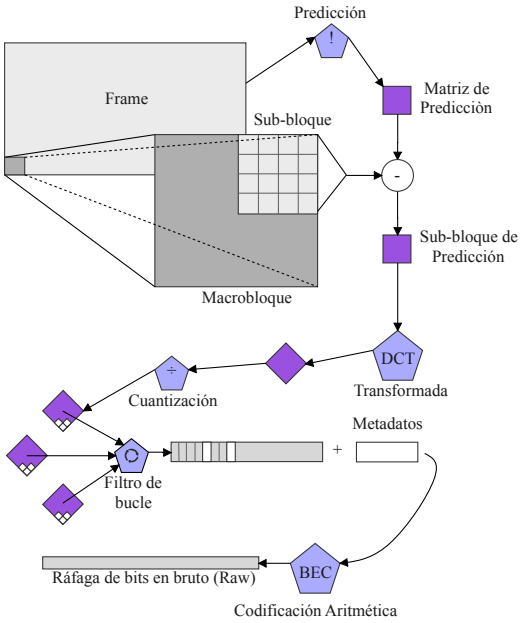


Figura 2.10: Proceso de codificación de VP8.

Tree Block (CTB)) se divide en una unidad de codificación Codification Unit (CU), después de lo cual la CU se divide en una unidad de predicción (Prediction Unit (PU)) para la predicción interframe o intraframe y la CU se divide también en una unidad de transformación (Transform Unit (TU)) para codificar la predicción residual. El códec soporta hasta 8K UHDTV (8192×4320) con fps de hasta 300 [48]. Además, H.264 es superado en compresión por H.265/HEVC en un 50 %, pero al mismo tiempo es 5869 veces más complejo [49].

El principal objetivo del High Efficiency Video Coding (HEVC) es la reducción de la tasa de bits. La reducción media de la tasa de bits de HEVC es del 52 % para 480p, del 56 % para 720p, del 62 % para 1080p y del 64 % para 4K UHD. Un codificador HEVC procede a dividir una imagen en regiones formadas por bloques para la primera imagen. La Figura 2.11 muestra el típico codificador de video HEVC que podría hacer un flujo de bits para el estándar HEVC. A continuación se presentan algunas de las características destacadas de H.265 [50] y [39]:

- **Unidad de árbol de codificación (CTU):** La unidad de árbol de codificación (CTU), tiene un tamaño seleccionado por el codificador y puede ser mayor que un macrobloque tradicional. La CTU está formada por un CTB de luma y los correspondientes CTB de croma y elementos de sintaxis. El tamaño L×L de un CTB de luma puede elegirse como L = 16, 32 o 64 muestras, y los tamaños más grandes suelen permitir una mejor compresión. Por tanto, HEVC admite una partición de los CTB en bloques más pequeños utilizando una estructura de árbol y una señalización de tipo cuádruple.
- **Unidades de codificación (CU) y bloques de codificación (CB):** La sintaxis de árbol cuádruple, de las unidades de árbol de codificación, especifica el tamaño de los píxeles y las posiciones de sus CBs de luma y croma. La raíz del árbol cuádruple está vinculada a la CTU. Por lo tanto, el tamaño del CTB de luma es el mayor tamaño admitido para un CB de luma. La división de una CTU en CBs de luma y croma se señala conjuntamente. Una CB de luma y normalmente dos CB de croma, junto con la sintaxis asociada, forman una unidad de codificación (CU). Un CTB puede contener solo una CU o puede dividirse para formar múltiples CUs, y cada CU tiene una partición asociada en unidades de predicción (PUs) y un árbol

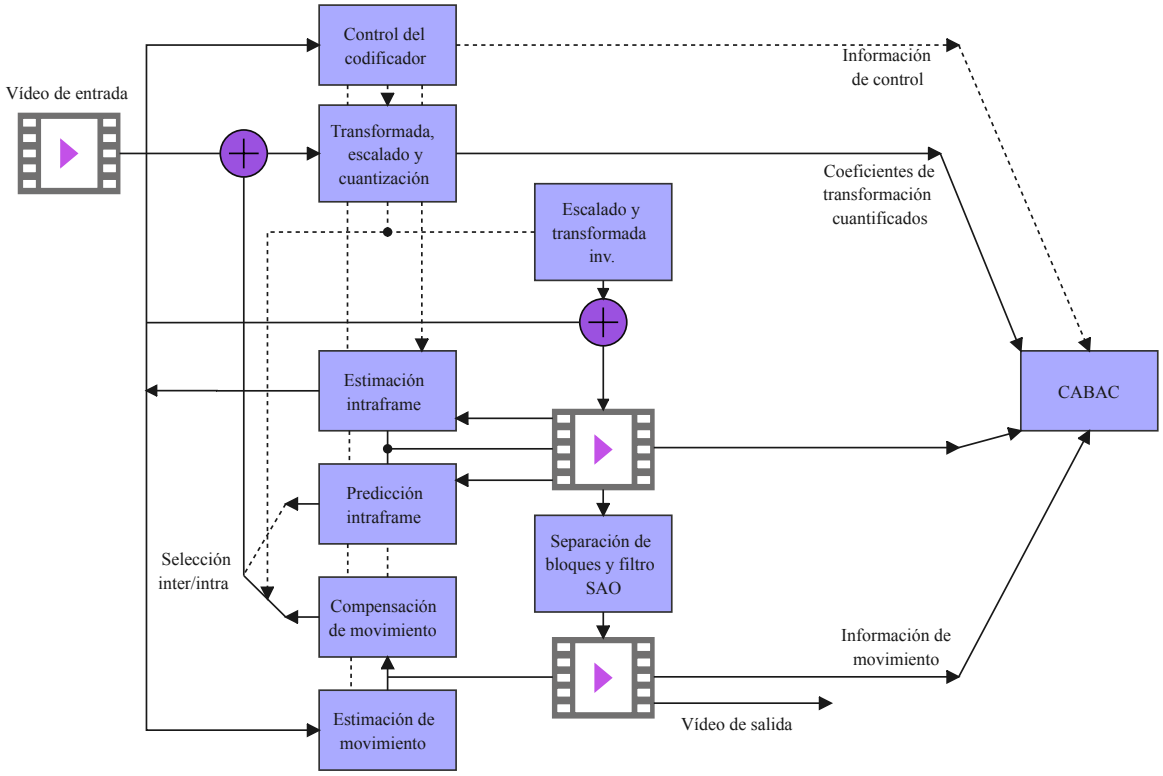


Figura 2.11: Codificador de video H.265/HEVC.

de unidades de transformación (TUs).

- **Unidades de predicción y bloques de predicción (PB):** Una estructura de partición de la PU tiene su raíz en el nivel de la CU. Dependiendo de la decisión del tipo de predicción básica. H.265/HEVC admite tamaños de PB variables desde 64×64 hasta 4×4 muestras.
- **TUs y bloques de transformación:** Una estructura de árbol de la TU tiene su raíz en el nivel de la CU. Se definen funciones base enteras similares a las de una transformada coseno discreta (DCT) para los tamaños de TB cuadrada de 4×4, 8×8, 16×16 y 32×32. Para la transformada 4×4 de los residuos de predicción intra-imagen de luma, se especifica alternativamente una transformada entera derivada de una forma de transformada discreta del seno (DST).
- **Desplazamiento adaptativo de la muestra (SAO):** En el bucle de predicción entre imágenes se introduce un mapeo de amplitud no lineal después del filtro de desbloqueo. Su objetivo es reconstruir mejor las amplitudes de la señal original utilizando una tabla de búsqueda descrita por unos pocos parámetros adicionales que pueden determinarse mediante el análisis del histograma en el lado del codificador.

Algunas caracteríscas como: señalización de vectores de movimiento, compensación de movimiento, predicción intraimagen, control de cuantificación, codificación de entropía y filtrado de desbloqueo en bucle son similares a las usadas en H.264 pero mejoradas.

2.3.4.f. Códec VP9

En 2011 después de que Google adquiriera On2 Technologies, se comenzó con el desarrollo del cócec VP9 y se presentó como producto final en el año 2013. Este códec fue creado con bases del códec VP8 y de igual manera de código abierto. El objetivo de su creación consistió en que con solo un modesto aumento de la

complejidad de la descodificación se alcance un flujo de bits mucho más compacto que su predecesor VP8 [51].

Por lo general VP9 se ofrece como una alternativa a HEVC, aunque dependiendo del escenario y tipo de configuración uno es mejor que el otro [52].

Entre algunas características se tiene:

- La incorporación de bloques de predicción de mayor tamaño mejora la eficiencia de la codificación en VP9. Este códec introduce superbloques (SB) de tamaños de hasta 64x64 y permite el desglose hasta llegar a bloques de 4x4.
- VP9 admite un conjunto de 10 modos de predicción interna para tamaños de bloque que van desde 4x4 hasta 32x32, también admite 4 modos de interpredicción para tamaños de bloque que van desde 4x4 hasta 64x64 píxeles.
- VP9 utiliza tres tipos de transformaciones: Transformación discreta del coseno (DCT), la transformada asimétrica de seno discreto (ADST) y la transformada Walsh-Hadamard (WHT) .
- Además de hacer previsiones para decodificar múltiples cuadros en paralelo, VP9 también tiene soporte para decodificar cuadros individuales usando múltiples hilos.
- VP9 admite un indicador de modo resistente a errores a nivel de trama que, cuando se activa, solo permite modos de codificación en los que es posible conseguirlo [51].

2.3.5. Mecanismos de evaluación de calidad de video

En esta sección se describen los dos tipos de mecanismos utilizados para la evaluación de la calidad de video y los principales algoritmos utilizados.

2.3.5.a. Evaluación objetiva

- Relación pico-señal-ruido (PSNR): El PSNR indica la aproximación de calidad entre la imagen original y la imagen distorsionada [53]. Para obtener el PSNR, se debe obtener primero el error cuadrático medio (MSE) tal como se muestra en la Ecuación 2.5.

$$\text{MSE} = \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} \left[(f(i,j) - F(i,j)]^2 \right.}{M \cdot N} \tag{2.5}$$

  Donde f(i,j) es la señal original en el píxel (i, j), F(i, j) es la señal reconstruida, y M x N es el tamaño de la imagen. Finalmente, se establece el PSNR en la Ecuación 2.6.

$$\text{PSNR}(dB) = 10 \cdot \log_{10} \frac{L^2}{\text{MSE}} \tag{2.6}$$

  Donde, L es el rango dinámico de los valores de los píxeles (por ejemplo para 8 bits es $2^8 - 1 = 255$). Cabe señalar que un valor alto de PSNR significa una mayor aproximación de calidad al video original.

- Índice de similitud estructural (SSIM)
  La métrica SSIM compara dos imágenes extraídas de la misma ubicación espacial con la ayuda de la media, la varianza y la covarianza. Donde la media y la varianza se consideran aproximadamente como estimaciones de la luminancia y el contraste de la señal. Mientras que la covarianza se considera como una medida de cuánto cambia una señal de forma no lineal con respecto a otra señal. Esta métrica se define en la Ecuación 2.7 [54].

$$\text{SSIM}(x,y) = \frac{\left(2\mu_x\mu_y + C_1\right)\left(2\sigma_{xy} + C_2\right)}{\left(\mu_x^2 + \mu_y^2 + C_1\right)\left(\sigma_x^2 + \sigma_y^2 + C_2\right)} \tag{2.7}$$

C1 y C2 son constantes que se definen como:

$$C_1 = (K_1 L)^2$$
$$C_2 = (K_2 L)^2$$

Donde, L es el rango dinámico de los valores de los píxeles, y K1 y K2 son dos constantes pequeñas. El índice SSIM se encuentra en el rango de [-1, 1] y un valor alto indica una mejor calidad.

- Métricas de calidad de video (VQM)
  La métrica VQM mide los efectos perceptivos de las deficiencias del video, como el desenfoque, el movimiento espasmódico/no natural, el ruido global, la distorsión en bloque y la distorsión del color, y los une en una sola métrica. Para tal objetivo, calcula la visibilidad de los artefactos expresados en el dominio DCT. Cuanto mayor sea el VQM, más grave será la distorsión y peor la visibilidad. En la Figura 2.12 se muestra el diagrama de bloques de la métrica, el cual consta de 9 pasos [55].

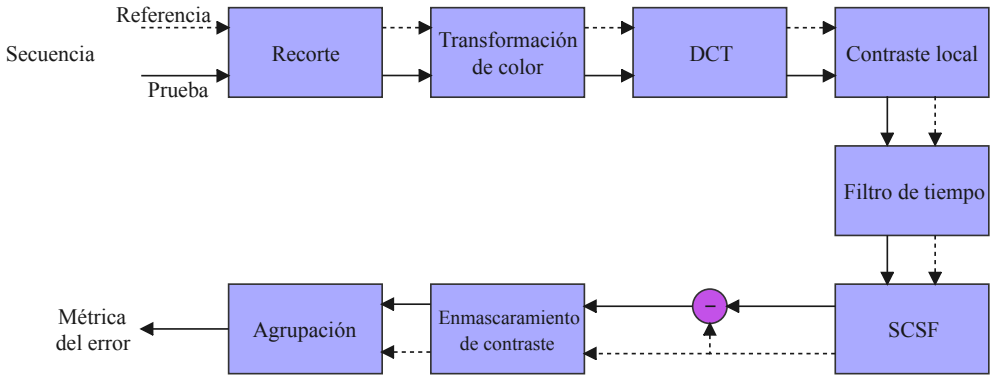


Figura 2.12: Diagrama de bloques de la métrica VQM.

Además, la métrica VQM tiene una alta correlación con la evaluación subjetiva de la calidad de video

2.3.5.b. Evaluación subjetiva

- Método DSCQS
  El método de escala de calidad continua de doble estímulo (Double-stimulus Continuous Quality Scale (DSCQS)) comprende un video de referencia y un video de evaluación el cual se presenta a un espectador en dos ocasiones, realizando la evaluación en la segunda vez. Además, no se indica a los sujetos cuál es el video de referencia y estos definen una calificación utilizando una escala discreta de cinco niveles como se muestra en la Figura 2.13. Después los valores se promedian entre todos los participantes para obtener el valor de DSCQS [56].
- Método DSIS
  El método de escala de Deterioro del Estímulo Doble Double Stimulus Impairment Scale (DSIS) presenta las secuencias una sola vez a diferencia del método DSCQS con lo que se reduce el tiempo del test. Después, los sujetos realizan una calificación general utilizando una escala discreta de cinco niveles. Por lo

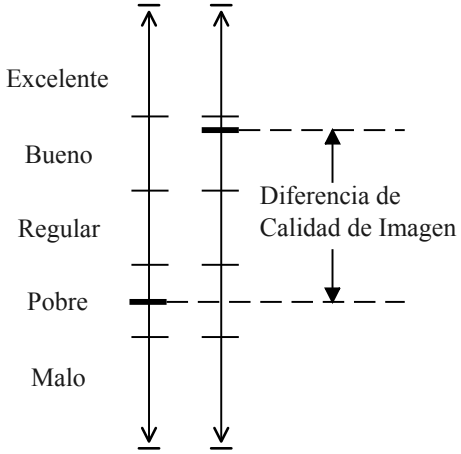


Figura 2.13: Categorías de evaluación.

general, este método es recomendado para evaluar la diferencia de la secuencia de prueba en comparación con la de referencia.

- Método ACR
  El método de clasificación absoluta por categorías Absolute Category Rating (ACR) es un método de estímulo único. Los sujetos solo ven el video objeto de la prueba, sin la referencia, y realizan una calificación para su calidad general utilizando una escala discreta de cinco niveles. Este método es muy eficiente en comparación con el DSIS o el DSCQS ya que al no mostrar la secuencia de referencia el tiempo de calificación es menor [57].

2.4. Herramientas para el análisis de tráfico y redes

En este apartado se describen las principales herramientas *open source* empleadas en el trabajo de titulación para el desarrollo de los experimentos.

2.4.1. Iperf

Iperf es una herramienta que mide el ancho de banda de extremo a extremo. Para tal objetivo hace uso de flujos TCP, permitiendo variaciones en parámetros como el tamaño de la ventana TCP y el número de flujos paralelos. El ancho de banda alcanzable de extremo a extremo es el ancho de banda con el que una aplicación en un *host* puede enviar datos a una aplicación en otro *host* destino. En particular, iperf aproxima el ancho de banda acumulado (el total de datos transferidos entre los *hosts* finales durante el periodo total de transferencia) al ancho de banda alcanzable de extremo a extremo [58].

2.4.2. Analizadores de red

La captura de paquetes es un término de redes informáticas para interceptar un paquete de datos que está cruzando o moviéndose por una red informática específica. Una vez que se captura un paquete, se almacena temporalmente para poder analizarlo. El paquete se inspecciona para ayudar a diagnosticar, resolver los problemas de la red y determinar si se cumplen las políticas de seguridad de la red. Los piratas informáticos también pueden utilizar técnicas de captura de paquetes para robar los datos que se transmiten a través de una red [59].

2.4.2.a. Tcpdump

Una de las herramientas de captura de tráfico de red más precisas y ligeras disponibles de forma gratuita es Tcpdump. Este analizador de red provee muchas opciones donde los detalles de los paquetes capturados pueden ser vistos en varios formatos como la opción "-t" que da la salida de la marca de tiempo legible para el ser humano, "-X" muestra el contenido de los paquetes en Hex y ASCII, "-v" "-vv" y "-vvv" incrementan la información de los paquetes a ser mostrada [60].

2.4.2.b. Wireshark

Wireshark es un proyecto de software de código abierto y se publica bajo la Licencia Pública General de GNU GPL. Este software cuenta con un front-end gráfico, además de algunas opciones de clasificación y filtrado integradas. Además, Wireshark permite al usuario poner en modo promiscuo a los controladores de interfaz de red que lo soportan [59]. Otra ventaja es la capacidad de escanear cualquier tipo de red, ya sea Ethernet, Wi-Fi, modo monitor o incluso Bluetooth. Esta es la razón principal por la que Wireshark se utiliza globalmente en los gigantes de la industria no solo como un *sniffer*, sino como un sistema de detección de intrusiones Intrusion Detection System (IDS).

Wireshark puede llegar hasta la profundidad de los bits de los paquetes [60]. Cabe destacar que este analizador de redes utiliza "libpcap" para capturar paquetes de la red, por lo que solo funciona en redes que soportan libpcap, pero se permite también leer los datos de un paquete ya capturado como entrada para el análisis [61]. Las principales características de Wireshark incluyen:

- Soporte para Ethernet, IEEE 802.11, PPP y *loopback*.
- Captura en vivo y análisis fuera de línea.
- Interfaz gráfica de usuario interactiva y versión de línea de comandos.
- Permite crear plug-ins para analizar nuevos protocolos.
- Proporciona un rico análisis de VoIP.
- Los resultados se pueden exportar a varios formatos de archivo, como XML, CSV, texto plano y PostScript, texto plano y PostScript.

2.4.3. Tcpstat

Las redes informáticas no son fiables y un host, o un enlace, puede caer en cualquier momento, afectando a la accesibilidad y a los servicios prestados por el sistema. Las herramientas de monitorización de redes son generalmente una colección de herramientas simples y comandos del sistema operativo que ayudan a monitorizar eficazmente el rendimiento, la calidad del servicio, el retardo y el ancho de banda de una red. Las herramientas de monitorización proporcionan una vista general de toda la red o de un segmento de la misma. Tcpstat permite obtener estadísticas relacionadas con TCP leyendo un archivo tcpdump o monitorizando una interfaz. Específicamente, muestra estadísticas como el ancho de banda, el número de paquetes, el tamaño promedio de los paquetes, la carga de la interfaz, la desviación estándar del tamaño de los paquetes, etc. Tcpstat es capaz de manejar un gran número de paquetes por segundo y tiene una interfaz compacta [61].

2.4.4. Iwlist

Iwlist se utiliza para mostrar información adicional de una interfaz de red inalámbrica que el comando iwconfig no presenta. El argumento principal se utiliza para seleccionar una categoría de información, iwlist muestra en forma detallada toda la información relacionada con esta categoría, incluyendo la información ya mostrada por iwconfig.

2.4.4.a. Scanning

Al utilizar este parámetro, se proporciona la lista de puntos de acceso y celdas ad hoc en el rango, y opcionalmente una gran cantidad de información relacionada (Extended Service Set Identification (ESSID), Calidad, Frecuencia, Modo, etc). La activación del escaneo es una operación privilegiada (solo para el *root*) y los usuarios convencionales solo pueden leer los resultados del escaneo. Por defecto, la forma de escanear depende de la tarjeta y de la configuración de la misma [62].

2.4.5. Netcat

Netcat es un programa de red diseñado para leer y escribir datos a través de conexiones TCP y UDP utilizando el conjunto de protocolos TCP/Internet Protocol (IP). Algunos de los usos de Netcat incluyen el escaneo de puertos, la transferencia de archivos, la captura de *banners*, la escucha y redirección de puertos [63].

2.5. Transmisión de flujos multimedia

En esta sección se presenta una breve introducción al protocolo Real Time Transport Protocol (RTP), el cual es utilizado para la tranmisión de audio y video en tiempo real.

2.5.1. RTP

RTP proporciona funciones de transporte de red de extremo a extremo adecuadas para aplicaciones que transmiten datos en tiempo real, como audio, video o datos de simulación, a través de servicios de red multicast o unicast. RTP no se ocupa de la reserva de recursos y no garantiza la calidad del servicio para los servicios en tiempo real. El transporte de datos se complementa con un protocolo de control Real Time Transport Control Protocol (RTCP) para permitir la supervisión de la entrega de datos de una manera escalable a grandes redes de multidifusión, y para proporcionar una funcionalidad mínima de control e identificación. RTP y RTCP están diseñados para ser independientes de las capas de transporte y red subyacentes [64].

2.5.2. RTCP

RTCP se utiliza para supervisar la calidad del servicio y transmitir información sobre los participantes en una sesión en curso. Este protocolo se basa en la transmisión periódica de paquetes de control a todos los participantes en la sesión, utilizando el mismo mecanismo de distribución que los paquetes de datos. RTCP realiza cuatro funciones:

1. Proporcionar información sobre la calidad de la distribución de datos.

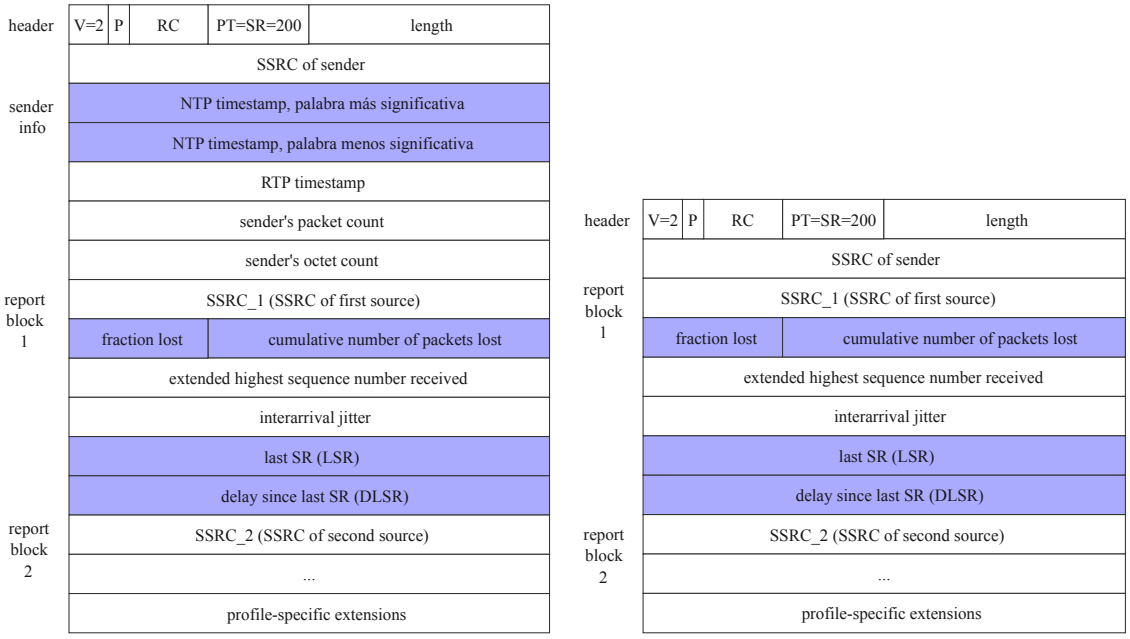
2. RTCP lleva un identificador a nivel de transporte para una fuente RTP denominado CNAME. Dado que el identificador SSRC puede cambiar si se descubre un conflicto o se reinicia un programa, los receptores requieren el CNAME para hacer un seguimiento de cada participante. Los receptores también pueden necesitar el CNAME para asociar múltiples flujos de datos de un determinado participante en un conjunto de sesiones RTP relacionadas, por ejemplo para sincronizar audio y video. La sincronización entre medios también requiere las marcas de tiempo Network Time Protocol (NTP) y RTP incluidas en los paquetes RTCP por los emisores de datos.
3. Cada participante envía paquetes de control a todos los demás, por lo que cada uno puede observar independientemente el número de participantes. Este número se utiliza para calcular la velocidad de envío de los paquetes.
4. Transmitir una mínima información de control de la sesión, por ejemplo, la identificación de los participantes que se mostrará en la interfaz de usuario. Es muy probable que esto sea útil en sesiones "poco controladas" en las que los participantes entran y salen sin control de pertenencia o negociación de parámetros.

2.5.2.a. Formato del paquete RTCP

Se definen varios tipos de paquetes RTCP para llevar una información de control, entre los cuales se tiene:

- Sender Report (SR) o informe del emisor, se utiliza para las estadísticas de transmisión y recepción de los participantes que son emisores activos.
- Receiver Report (RR) o informe del receptor, para las estadísticas de recepción de los participantes que no son remitentes activos y en combinación con SR para los remitentes activos que informan sobre más de 31 fuentes.
- Source Description (SDES) o elementos de descripción de la fuente, incluido el CNAME.
- BYE: Indica el fin de la participación.
- APP: Funciones específicas de la aplicación.

Los RR proporcionan información sobre la calidad de la recepción utilizando paquetes de informe RTCP que pueden adoptar una de las dos formas dependiendo de si el receptor es también un emisor o no. La única diferencia entre las formas de SR e informe del receptor RR, además del código del tipo de paquete, es que el informe del remitente incluye una sección de información del remitente de 20 bytes para que la utilicen los remitentes activos. El SR se emite si un sitio ha enviado algún paquete de datos durante el intervalo transcurrido desde la emisión del último informe o del anterior, en caso contrario se emite el RR. En la Figuras 2.14a y 2.14b se presenta la estructura de un paquete SR y RR respectivamente. Donde se hace énfasis en los campos: NTP *timestamp*, *fraction lost*, *cumulative number of packets lost*, Last Sender Report (LSR) *timestamp* y Delay since Last Sender Report (DLSR).



(a) Sender report (b) Receiver Report

Figura 2.14: Estructura de los paquetes de reporte.

En este contexto, el campo NTP *timestamp* indica la hora a la que se envió este informe, de modo que puede utilizarse en combinación con las marcas de tiempo devueltas en los RR de otros receptores para medir la propagación de ida y vuelta a esos receptores. Así mismo, el campo *fraction lost* es la fracción de paquetes de datos RTP perdidos desde que se envió el anterior paquete SR o RR. Esta fracción se define como el número de paquetes perdidos dividido por el número de paquetes esperados.

Por otro lado, el campo *cumulative number of packets lost* proporciona el número total de paquetes de datos RTP procedentes del origen que se han perdido desde el inicio de la recepción. Este número se define como el número de paquetes esperados menos el número de paquetes realmente recibidos, donde el número de paquetes recibidos incluye cualquiera que sea tardío o duplicado. Así, los paquetes que llegan tarde no se cuentan como perdidos, y la pérdida puede ser negativa si hay duplicados.

Por otra parte, el campo LSR cuenta con los 32 bits centrales de los 64 de la marca de tiempo NTP recibida como parte del paquete SR más reciente de la fuente. Si todavía no se ha recibido ningún SR, el campo se pone a cero. Y finalmente, el campo DLSR es el retraso, expresado en unidades de 1/65536 segundos, entre la recepción del último paquete SR desde la fuente y el envío de este bloque RR. Si aún no se ha recibido ningún paquete SR desde la fuente, el campo DLSR se pone en cero.

2.6. Frameworks multimedia

En esta sección se presenta una descripción del *framework* multimedia GStreamer, sus características y su modo de operación.

2.6.1. GStreamer

GStreamer es un *framework* multiplataforma para crear aplicaciones de *streaming* de medios. El *framework* de GStreamer está diseñado para facilitar la escritura de aplicaciones que manejen audio, video o ambos. Además, no está restringido a audio y video, es decir, puede procesar cualquier tipo de flujo de datos. El diseño de la *pipeline* (tubería) está hecho para tener poca sobrecarga por encima de los filtros aplicados. Esto hace que GStreamer sea un buen *framework* para diseñar incluso aplicaciones de audio de alta gama, es decir con latencias mínimas.

Uno de los usos más obvios de GStreamer es para construir un reproductor multimedia. Sus principales ventajas son que los componentes se pueden mezclar y combinar en *pipelines* arbitrarios de modo que es posible escribir una aplicación de edición de video o audio completa. El *framework* se basa en *plugins* que proporcionan los distintos códecs y otras funcionalidades. Los *plugins* se pueden enlazar y organizar en un *pipeline*. Los *pipelines* también pueden ser editados con un editor GUI y guardados como XML para que las bibliotecas de *pipelines* puedan ser construidas con un mínimo de esfuerzo. La función principal de GStreamer es proporcionar un marco para los *plugins*, el flujo de datos y el manejo/negociación del tipo de medio. También proporciona una API para escribir aplicaciones utilizando los diversos *plugins* [65]. En la Figura 2.15 se observa la arquitectura de GStreamer.

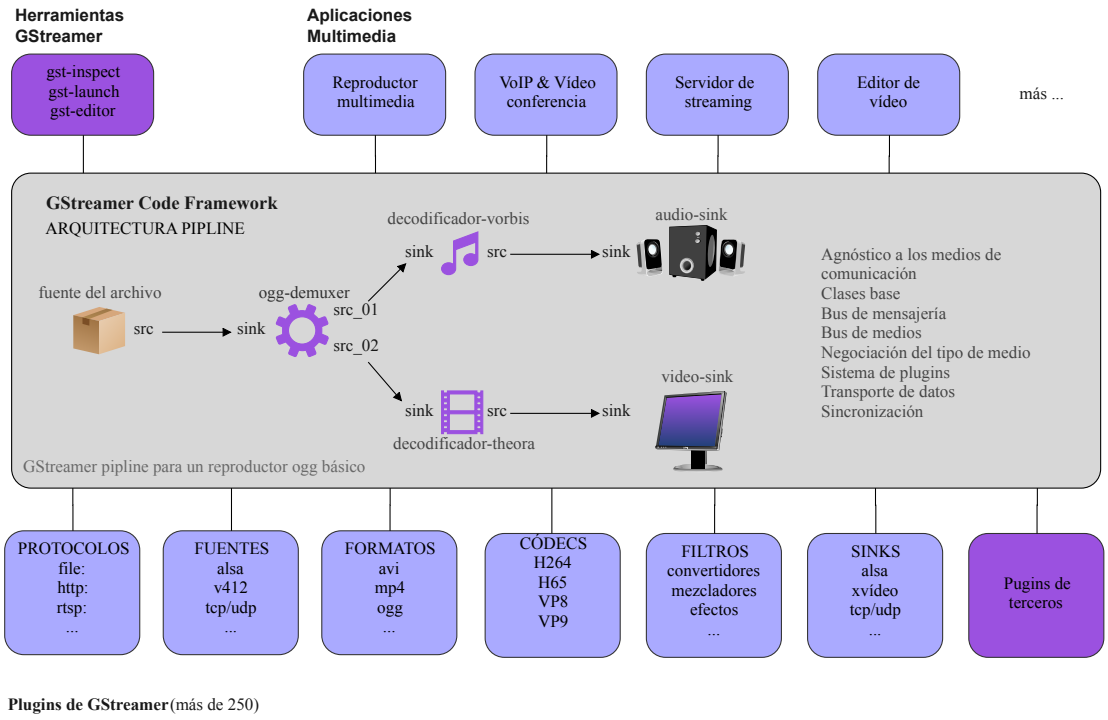


Figura 2.15: Herramientas y plugins de GStreamer.

Los *plugins* de GStreamer se pueden clasificar en:

- Manejo de protocolos
- Fuentes para audio y video (implica *plugins* de protocolo)
- Formatos: parsers, formateadores, muxers, demuxers, metadatos, subtítulos

- Códecs: codificadores y decodificadores
- Filtros: convertidores, mezcladores, efectos
- *Sinks* para el audio y el video (implica *plugins* de protocolo)

2.6.1.a. Conceptos básicos de Gstreamer

En esta sección, se presentan los conceptos básicos de GStreamer y los objetos más utilizados, como elementos, *pads* y *bins*. Se utilizará una representación visual de estos objetos para poder observar los *pipelines* más complejos, toda la información se encuentra disponible en el manual de aplicación de GStreamer [65].

2.6.1.b. Elementos

Un elemento es el bloque de construcción básico para un *pipeline* de medios. El objeto más importante en GStreamer para el programador de aplicaciones es el objeto GstElement. Todos los diferentes componentes de alto nivel que se utilizarán se derivan de GstElement. Cada decodificador, codificador, demuxer, salida de video o audio es de hecho un GstElement. Los elementos se visualizan mejor como cajas negras. En un extremo se puede incluir algo, el elemento interactúa con él y sale algo más en el otro lado. En el caso de un elemento decodificador, por ejemplo, se introducen datos codificados y el elemento emite datos decodificados.

- Fuentes (*Sources*)
  Los elementos fuente generan datos para ser utilizados por un *pipeline*, por ejemplo, al leer del disco o de una tarjeta de sonido. La visualización de un elemento fuente muestra cómo se ve un elemento fuente. Siempre se representa un *pad* de la fuente a la derecha del elemento, más adelante en la Sección 2.6.1.e se detalla la funcionalidad de los *pads*. Los elementos fuente no aceptan datos, solo los generan. Esto se puede ver en la Figura 2.16 porque solo tiene un *pad* de origen (a la derecha). Un *pad* de origen solo puede generar datos.

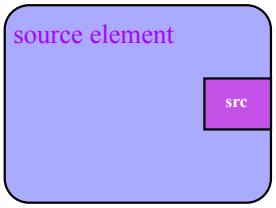
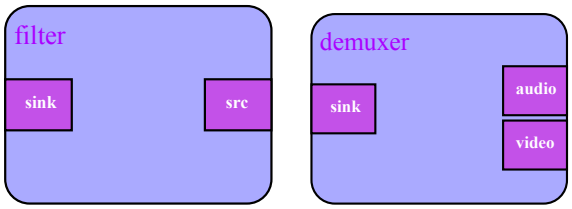


Figura 2.16: Elemento fuente.

- Filtros, conversores, demuxers, muxers y codecs
  Los filtros y elementos similares a los filtros tienen *pads* de entrada y de salida. Operan con los datos que reciben en sus *pads* de entrada (*sink*), y proporcionarán datos en sus *pads* de salida (*source*). Ejemplos de estos elementos son un filtro, un escalador de video (convertidor), un demuxer o un decodificador. Los filtros pueden tener cualquier número de *pads* de origen o de destino. Un demuxer de video, por ejemplo, tendría un *sink pad* y varios (1-N) *source pads*, uno por cada flujo elemental contenido en el formato contenedor. Los decodificadores, en cambio, solo tendrán un *pad* de origen y otro de destino.

  La visualización de un elemento de filtro se presenta en la Figura 2.17a. Este elemento específico tiene un *pad* de origen y un *pad* de destino. Los *sink pads*, que reciben datos de entrada se representan a la izquierda del elemento; los *pads* de origen siguen estando a la derecha. En la Figura 2.17b se presenta

la visualización de un elemento de filtro con más de un *pad* de salida muestra otro elemento similar a un filtro, este con más de un *pad* de salida (fuente). Un ejemplo de un elemento de este tipo podría ser, por ejemplo, un demuxer Ogg para un flujo Ogg que contenga tanto audio como video. Un *pad* de origen contendrá el flujo de video elemental, otro contendrá el flujo de audio elemental. Los demuxers generalmente disparan señales cuando se crea un nuevo *pad*.



(a) Filtro (b) Demuxer

Figura 2.17: Tipos de elementos.

- Sinks
  Los elementos Sink son los puntos finales de una cadena de medios. Los mismos aceptan datos pero no producen nada. La escritura en el disco, la reproducción en la tarjeta de sonido y la salida de video serían implementadas por elementos sumideros. En la Figura 2.18 se presenta la visualización de un elemento *sink*.

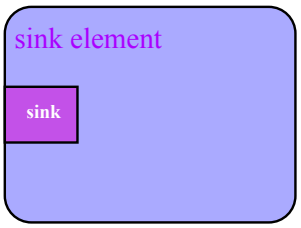


Figura 2.18: Elemento *sink*.

2.6.1.c. Bin o contenedor

Un bin es un elemento contenedor donde se pueden añadir varios elementos. Dado que un contenedor es un elemento por sí mismo, un contenedor puede ser manejado de la misma manera que cualquier otro elemento. Por lo tanto, toda la sección 2.6.1.b (Elementos) se aplica también a los contenedores.

Los contenedores permiten combinar un grupo de elementos vinculados en un elemento lógico, como se visualiza en la Figura 2.19. Como se puede observar, no se trata de los elementos individuales, sino de un solo elemento, el contenedor. Esto es de gran utilidad para construir *pipelines* complejas, ya que permite dividir el *pipeline* en trozos más pequeños. El contenedor también gestionará los elementos que contiene, realizará cambios de estado en los elementos así como recogerá y reenviará los mensajes del bus.

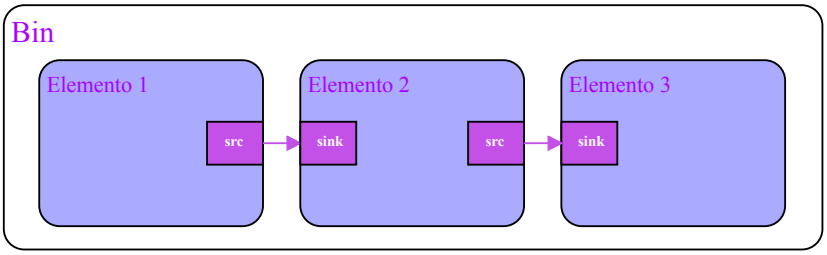


Figura 2.19: Bin o contenedor.

2.6.1.d. Bus

Un bus es un sistema simple que se encarga de reenviar los mensajes de los hilos de *streaming* a una aplicación en su propio contexto de hilo. La ventaja de un bus es que una aplicación no necesita ser consciente de los hilos para utilizar GStreamer, aunque GStreamer en sí mismo está fuertemente hilado.

Cada *pipeline* contiene un bus por defecto, por lo que las aplicaciones no necesitan crear un bus ni nada parecido. Lo único que deben hacer las aplicaciones es establecer un manejador de mensajes en un bus, que es similar a un manejador de señales para un objeto. Cuando el bucle principal se está ejecutando, el bus comprobará periódicamente si hay nuevos mensajes, y se llamará al *callback* cuando haya algún mensaje disponible.

2.6.1.e. Pads

Los *pads* son la interfaz del elemento con el mundo exterior. Los datos fluyen desde el *pad* de origen de un elemento hasta el *pad* de destino de otro elemento. El tipo específico de medio que el elemento puede manejar será expuesto por las capacidades (caps) del *pad*. Un tipo de *pad* se define por dos propiedades: su dirección y su disponibilidad. Como se mencionó anteriormente, GStreamer establece dos direcciones de *pads*: *pads* de origen y *pads* de destino. Esta terminología se define desde el punto de vista interior del elemento: los elementos reciben datos en sus *pads sink* y generan datos en sus *pads source*. Esquemáticamente, los *sink pads* se dibujan en el lado izquierdo de un elemento, mientras que los *pads* de origen se dibujan en el lado derecho de un elemento. En estos gráficos, los datos fluyen de izquierda a derecha.

Las direcciones de los *pads* son muy simples comparadas con la disponibilidad de los *pads*. Un *pad* puede tener cualquiera de tres disponibilidades: siempre, a veces y a petición. El significado de esos tres tipos es exactamente el que dice: los *pads* siempre existen, los *pads* a veces existen solo en ciertos casos (y pueden desaparecer aleatoriamente), y los *pads* bajo petición solo aparecen si son solicitados explícitamente por las aplicaciones.

- Capacidades de un Pad
  Dado que los *pads* juegan un papel muy importante en la forma en que el elemento es visto por el mundo exterior, se implementa un mecanismo para describir los datos que pueden fluir o que actualmente fluyen a través del mismo mediante el uso de capacidades. Las capacidades se adjuntan a las plantillas de *pads* y a los *pads*. Para las plantillas, describirá los tipos de medios que pueden transmitirse sobre un *pad* creado a partir de esta plantilla. Para los *pads*, puede ser una lista de posibles “caps”, en cuyo caso el *pad* aún no está negociado, o es el tipo de medio que actualmente se transmite por este, donde ya ha sido negociado. Las capacidades describen el tipo de datos que se transmiten entre dos *pads*, o que soporta un *pad* (plantilla). Esto las hace muy útiles para varios propósitos:

- **Autoplugging:** encontrar automáticamente elementos para vincular a un *pad* en base a sus capacidades. Todos los *autopluggers* utilizan este método.
- **Detección de compatibilidad:** cuando dos *pads* son enlazados, GStreamer puede verificar si los dos *pads* están hablando del mismo tipo de medio. El proceso de enlazar dos *pads* y comprobar si son compatibles se llama "negociación de capacidades".
- **Metadatos:** mediante la lectura de las capacidades de un *pad*, las aplicaciones pueden proporcionar información sobre el tipo de medio que se está transmitiendo a través del *pad*, que es información sobre el flujo que se está reproduciendo actualmente.
- **Filtrado:** una aplicación puede utilizar las capacidades para limitar los posibles tipos de medios que pueden transmitirse entre dos *pads* a un subconjunto específico de sus tipos de flujo soportados. Una aplicación puede, por ejemplo, utilizar "caps filtradas" para establecer un tamaño de video específico (fijo o no) que debe transmitirse entre dos *pads*. Un ejemplo de "caps" filtradas consiste específicamente en añadir o eliminar datos manualmente de/en una canalización. Los filtros de "caps" se colocan a menudo después de elementos de conversión como *audioconvert*, *audioresample*, *videoconvert* o *videoscale* para forzar a esos convertidores a convertir los datos a un formato de salida específico en un punto determinado del flujo.

2.6.2. RTPsession

El gestor de sesiones RTP modela a los participantes con un único Synchronization source (SSRC) en una sesión RTP. Esta sesión puede utilizarse para enviar y recibir paquetes RTP y RTCP. El gestor de sesiones actualmente implementa el RFC 3550 donde incluye:

- Validación de paquetes RTP basada en números de secuencia consecutivos.
- Mantenimiento de la base de datos de participantes del SSRC.

Cuando hay múltiples flujos multimedia enviados a través de una sesión RTP, se utiliza el identificador de fuente SSRC para distinguir entre las fuentes.

- Para utilizar GstRtpSession como receptor RTP, se solicita un *pad recv_rtp_sink*, que creará automáticamente el *pad recv_rtp_src*. Los datos recibidos en el *pad recv_rtp_sink* serán procesados en la sesión y después de ser validados serán reenviados al *pad recv_rtp_src*.
- Para utilizar también GstRtpSession como receptor RTCP, se solicita un *pad recv_rtcp_sink*, que creará automáticamente un *pad sync_src*. Los paquetes recibidos en el *pad* RTCP serán utilizados por el gestor de sesiones para actualizar las estadísticas y la base de datos de los otros participantes. Los paquetes SR serán reenviados en el *pad sync_src* para que puedan ser utilizados para realizar la sincronización entre flujos cuando sea necesario [66].

2.6.3. Autovideosink

Autovideosink es un *sink* de video que detecta automáticamente un formato de video apropiado a utilizar. Para ello, busca en el registro todos los elementos que tienen "*Sink*" y "Video" en el campo de clase de su información de elemento [67].

Capítulo 3

Trabajos relacionados

En el presente capítulo se hace una revisión exhaustiva de investigaciones que hablan sobre redes de sensores, redes ad hoc, comunicaciones resilientes, tráfico multimedia y códecs de video. Con el fin de poner en evidencia la actualidad de estos temas, se realiza una introducción a las redes de sensores y después se revisan aplicaciones de redes ad hoc multisalto con sus características generales. Posteriormente, se detallan los estudios existentes acerca de redes resilientes aplicadas en escenarios de emergencia. Después, se analizan estudios referentes al tráfico multimedia sobre redes tipo ad hoc. Finalmente, se repasa la actualidad de los códecs de video H.264, VP8, H.265 y VP9.

3.1. Redes de sensores

Los avances en la tecnología de los sensores y las redes informáticas han permitido que las redes de sensores evolucionen desde comunicaciones por cable hasta comunicaciones inalámbricas y desde una topología de red estática hasta una topología dinámica [68].

En [69] se afirma que las redes de sensores cableadas están muy extendidas, pero el coste de su instalación, pruebas, mantenimiento y apagado es bastante elevado. En muchos casos, una red de sensores inalámbricos (WSN) es una alternativa más atractiva porque no requiere ninguna infraestructura fija y puede aplicarse en zonas distribuidas donde el cableado es costoso. Sin embargo, en las WSN los nodos sensores dependen únicamente de su propia batería, por lo que sus recursos energéticos, su potencia de cálculo y sus recursos de almacenamiento son limitados.

Una red de sensores inalámbricos (WSN) está formada por sensores autónomos distribuidos espacialmente para monitorizar condiciones físicas o ambientales [70] que permiten procesar y transmitir la información. Las WSN son cada vez más populares con la llegada del Internet de las cosas (IoT). En tal contexto, diversas aplicaciones del mundo real de las WSN, como las redes inteligentes, la agricultura inteligente y la salud inteligente, necesitarían un despliegue potencial de miles o quizá cientos de miles de nodos/actuadores sensores. Para gestionar estas redes, en [71] se propone el uso de las redes definidas por software (SDN), ya que permiten la separación de la lógica de control de los nodos/actuadores sensores.

3.2. Redes ad hoc

Como se describió en el Capítulo 2, las redes ad hoc son redes descentralizadas y sin infraestructura, donde los nodos pueden generar datos y también actuar como routers para el reenvío de la información hacia nodos que estén dentro de su radio de cobertura [72]. Partiendo de este concepto, en [73] se propone el uso de redes ad hoc para reducir el *delay* de procesamiento utilizando recursos en la nube. Para obtener el resultado optimizado, se plantea un algoritmo de optimización de colonias de hormigas basado en el retardo (DoACO).

Por otro lado, en [74] se propone interconectar diferentes grupos WiFi Direct y crear redes ad hoc multisalto, permitiendo la comunicación entre dispositivos con sistema operativo Android. Los resultados experimentales demuestran la superioridad de las técnicas que aprovechan la capacidad del dispositivo para mantener conexiones físicas simultáneas con varios grupos, lo que permite crear redes ad hoc multisalto con poca sobrecarga.

De la misma forma, en [75] se muestra la construcción de una MANET sobre una gran variedad de teléfonos inteligentes y se demuestra la transmisión de video en tiempo real, el intercambio de archivos y los juegos móviles en la MANET multisalto. Con tal objetivo, en dicho estudio se emplea un nuevo método que utiliza el Protocol Bridging Mechanism (PBM), de esta manera se logra que una MANET basada en clústeres admita la comunicación multisalto y multiclúster entre los smartphones comerciales basados en Android.

En el caso de VANETs, la transferencia de datos se convierte en un reto debido a características inherentes como la velocidad excesiva, topologías geográficamente limitadas, enlaces de comunicación inestables, o la diversidad en la capacidad del canal. En [76] se plantea un protocolo para la distribución de mensajes de seguridad denominado Mobility-aware Multi-hop Clustering-based MAC (MAMC-MAC) con el propósito de lograr una sobrecarga mínima, una alta fiabilidad y la entrega de mensajes de seguridad en tiempo real. El protocolo se ha desarrollado especialmente para los contornos de las autopistas con el fin de lograr una mejora de la red y un uso eficiente del canal, así como garantizar la integridad de los vehículos.

Para mejorar la distribución de datos en redes VANET urbanas, en [77] se presenta dos nuevos protocolos de difusión de enrutamiento: el protocolo de difusión mejorado basado en el contador en VANET urbanas (ECUV) y el protocolo de difusión mejorado basado en la distancia (EDUV). Los dos utilizan un enfoque basado en la topología de la red de carreteras para seleccionar un conjunto de nodos de retransmisión con capacidades de cobertura mejoradas durante la entrega de datos en escenarios urbanos de vehículo a vehículo (V2V). También mejoran el rendimiento de los protocolos basados en el receptor al aliviar el efecto negativo de su comportamiento estocástico.

Con respecto a las redes ad hoc, algunos parámetros como el *delay* y el *throughput* tienen comportamientos especiales. En [78] se realiza una evaluación del rendimiento del *streaming* de video escalable en redes ad hoc móviles, y se demuestra que a medida que aumenta el número de nodos, el *throughput* disminuye en el receptor. Este comportamiento se debe a los diferentes niveles de congestión en los nodos según su ubicación en la topología. En particular, la contención por el acceso al canal es más intensa en los nodos finales, lo cual provoca la reducción de la tasa de datos obtenida en el nodo Rx.

Además, en [79] se establece que el *throughput* se verá afectado dependiendo del uso de otros dispositivos

que ocupen el mismo canal inalámbrico. También, en [80] se recalca que debido a la naturaleza dinámica del entorno del tráfico y la alta movilidad en redes MANET, existen problemas de pérdida de enlaces lo que disminuye el *throughput*. Por otro lado, en [81], se demuestra que el retardo medio de propagación (*delay*) de un enlace multisalto depende del número medio de repetidores y de la distancia media entre repetidores adyacentes.

3.3. Comunicación resiliente

La resiliencia de una red es la capacidad de proporcionar y mantener un nivel de servicio aceptable ante fallas que supongan un reto para las operaciones normales. Por esto, en [82] se indica que una de las alternativas frente a desastres son las redes ad hoc, ya que la infraestructura de telecomunicaciones puede destruirse parcial o totalmente, con lo que el despliegue de una red ad hoc multisalto puede hacer frente a la falta de comunicación. Es decir, permite anticipar las posibles situaciones disruptivas y predefinir soluciones listas para usar. Por ejemplo, un terremoto es una catástrofe natural común, donde el rescate es más difícil que el de otras catástrofes debido a que destruye tanto el suelo como las instalaciones de comunicación que hay en él. En [83], se propone una red ad hoc resiliente basada en drones, que se utiliza no solo para restablecer la comunicación en esa zona, sino también para la búsqueda de supervivientes en zonas de catástrofe en Pakistán.

De la misma manera, en [84] se propone un sistema FANET Emergency Application (FEA) que utiliza el protocolo de enrutamiento MP-OLSR para la detección de incendios a partir de drones que permite la recopilación y el análisis de datos y, a través de un sistema de gestión central, proporciona instrucciones de seguridad a las personas en peligro. Así mismo, los vehículos aéreos no tripulados (UAV) son una buena opción para la búsqueda de supervivientes, con las redes ad hoc como medio de comunicación. En este sentido, en [85] se propone un algoritmo dinámico de selección del clúster head basado en la energía, la movilidad, la distancia y el grado de correlación de los nodos. El algoritmo calcula los pesos de los UAV en función de estos cuatro parámetros, y selecciona el mejor *cluster head*.

Por otro lado, como consecuencia del movimiento de las placas tectónicas, en las profundidades de los océanos se generan tsunamis que afectan a las zonas costeras. La implementación adecuada de un sistema de detección de tsunamis ayuda a detectarlos y a prevenir pérdidas de vidas humanas y de infraestructura. Por lo tanto, en [86] se propone el despliegue de una WSN para detectar y responder a la detección del movimiento de las placas tectónicas. La implementación de una red WSN en la zona de la placa permite notificar directamente a las estaciones base para tomar medidas de precaución.

3.4. Tráfico multimedia sobre redes ad hoc

La carencia de infraestructura de las MANET hace que la transmisión de video sea una tarea difícil para los proveedores de servicios de video. La alta probabilidad de pérdida de paquetes en las MANETs puede crear una notable distorsión en la calidad del video recibido. En [87] se señala que al usar una estrategia adaptativa de video se reduce la congestión de la red y la pérdida de paquetes al reducir la tasa de transmisión en función del cambio en el ancho de banda disponible.

Para la vigilancia del tráfico vial en ciudades inteligentes, en [88] se propone una técnica de *streaming* en

directo de video adaptativo para transportar los flujos de video de las cámaras a los servidores externos. A partir de los resultados, tanto en escenarios simulados como reales, se demuestra la viabilidad de la propuesta en estos escenarios específicos, validada con varias métricas de rendimiento en términos de calidad de servicio y calidad de video.

Además en [89], se diseña un esquema de transmisión de video para utilizar eficazmente el ancho de banda disponible en las MANET, minimizar las distorsiones de la red y mejorar la calidad del servicio (QoS). El esquema de *streaming* de video propuesto supera a los protocolos de enrutamiento más avanzados diseñados para MANETs (v.g, DSDV y OLSR).

En cambio para las redes de sensores, la entrega de contenido multimedia requiere un alto rendimiento y un retardo reducido de un extremo a otro debido a su ancho de banda y energía limitada. Es por ello que en [90] se propone un protocolo de enrutamiento que proporciona mecanismos simples para construir rutas no correlacionadas hacia el receptor acelerando la transferencia de datos simultáneos.

Por otro lado, en [91] se presenta un método para evaluar el impacto de los datos multimedia, como la voz y el video, en el rendimiento de la red inalámbrica ad hoc multisalto, utilizando el servicio Wi-Fi MultiMedia (WMM). Los resultados indican que, sin utilizar la función WMM, los dispositivos de la red no tendrán una distinción entre los diferentes tipos de datos. Además, los parámetros de rendimiento y *jitter* de los tres tipos: voz, video y background son casi los mismos, sin embargo, a medida que la transmisión de datos pasa a través de nodos intermedios para reenviar los datos, el rendimiento de la red se degrada.

3.5. Códecs de video

Los dispositivos modernos han impulsado un rápido crecimiento del consumo de contenidos de alta resolución y alta calidad, siendo consumidores predominantes de ancho de banda, por tanto, el uso de códecs es de vital importancia ya que permiten codificar y comprimir datos, y de esta manera se reduce el almacenamiento requerido y se aumenta la velocidad de trasferencia. Entre los estándares más usados se tiene: H264, H265, VP8 y VP9 [92].

El estándar H.264 logra una mayor eficiencia en términos de bit rate que otros códecs anteriores, según [93], por lo que es un excelente candidato para proporcionar la mejor calidad de video en redes de calidad de servicio limitada y tasas de bits particularmente bajas. Además, en [94] se demostró que en redes ad hoc el uso de la extensión H.264/SVC es mejor que el uso de CBR aunque se trasmitan menos datos.

Por otro lado, en [95] se ha demostrado que el estándar H.265/HEVC es superior en compresión a los estándares H.264/AVC y VP9 en un escenario VANET. Específicamente, con una eficiencia del 27 % sobre VP9 y un 49 % sobre AVC, también se destaca que H.265/HEVC tiene mayor complejidad computacional. En cuanto a velocidad de codificación, en [96] se señala que H264/AVC es más rápido que H.265/HEVC y VP9 pero H.264/AVC tiene la mitad de eficiencia de codificación que estos dos. Además, en [97] se recalca que VP9 tiene un tiempo de codificación de hasta 100 veces mayor que H.264.

También, en [98] se analiza el estándar de compresión de video H.265 frente a H.264 y VP9. El estudio

demuestra que los nuevos códecs 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 [99], se establece que el rendimiento real de los códecs de video depende en gran medida de muchos factores como: los codificadores, la limitación de la duración de la codificación (tiempo real/offline), el tipo de contenido (videos naturales/de animación/video sintético), la complejidad del video, la resolución (UHD/FHD/HD/SD), ajustes de codificación (tamaño GoP, fps, VBR/CBR) y métricas de evaluación de la calidad (objetivo/subjetivo).

Por último, según [100], VP9 y H.265 necesitan realizar más procesamiento para lograr una mayor tasa de compresión, lo que significa que tardarán más en codificar el video, y por lo tanto, se retrasará la emisión del video, aumentando la latencia. VP9 y H.265 son un 50 % mejor que H.264, pero también son de 10 a 20 veces más lentos. En cuanto al consumo de CPU, tanto VP9 como H.265 tienen que pasar por más algoritmos de compresión que H.264, lo que aumentará su uso de la CPU. Sin embargo, se puede utilizar codificación basada en hardware para disminuir el consumo o aumentar el número de CPUs, del estudio descrito en [101], donde se demuestra que al aumentar el número de CPUs el tiempo de codificación disminuye considerablemente, aunque como consecuencia se tiene una disminución del PSNR.

Capítulo 4

Metodología

El presente capítulo se enfoca en describir de forma clara la metodología para la evaluación y comparación de códecs actuales para el desarrollo de los experimentos a ser realizados en el Capítulo 5. Por una parte, en la Sección 4.1 se describe el procedimiento empleado para la caracterización de la red ad hoc sobre el escenario de estudio (edificio matriz de la Empresa Eléctrica Regional Centro Sur), para tal objetivo se utilizó la herramienta iperf para determinar el *throughput* en cada piso y la herramienta ping para determinar el *delay* y el *packet loss*. Por otro lado, en la Sección 4.2 se detalla la metodología para la evaluación subjetiva y objetiva de la calidad de video, que va desde la elección de los videos hasta la selección del mejor códec. Posteriormente, en la Sección 4.3 se muestra el diseño y desarrollo de la herramienta siguiendo la metodología del punto anterior, la herramienta esta diseñada en Python, la cual permite observar y graficar las métricas propuestas, así como una comparación entre los diferentes códecs. Finalmente, en la Sección 4.4 se detalla la herramienta para la transmisión adaptativa de video en tiempo real.

4.1. Caracterización de la red ad hoc

En esta sección se describe la metodología empleada para caracterizar la red ad hoc multisalto utilizada en el presente trabajo de tesis. La red se encuentra implementada en el edificio matriz de la Empresa Eléctrica Regional Centro Sur. La misma tiene instalada una red de sensores acelerómetros para la monitorización de la salud estructural del edificio. Además, dispone de un sistema de detección automática de eventos sísmicos [102]. El sistema consta de un total de 10 nodos y un nodo *gateway*, de los cuales únicamente se utilizaron los ubicados en los pisos 1, 3, 5, 7 y en el sótano. Es necesario recalcar que la ubicación de los nodos es fija y cuentan con direcciones IP estáticas.

Para determinar la caracterización de la red ad hoc se realizaron cuatro experimentos. En la Figura 4.1 se muestra el diagrama de la metodología empleada. En cada experimento se ubicó un ordenador fijo en el sótano y otro ordenador en alguno de los pisos uno, tres, cinco o siete, tal como se observa en la Figura 4.2. El experimento desarrollado en el piso 1 genera tres saltos, el experimento en el piso 3 genera cuatro saltos, el experimento en el piso 5 genera cinco saltos y finalmente el experimento en el piso 7 genera seis saltos. Cabe mencionar que el ordenador que recorre los pisos se ubicó siempre a la misma distancia del nodo sensor ubicado

en tal piso. En la Figura 4.3 se observa el posicionamiento de los ordenadores dentro del Edificio (Centrosur). Por otro lado, cada experimento consiste en generar tráfico constante y transmitirlo desde un ordenador hacia el otro, aumentando el *bitrate* desde los 100Kbps hasta los 10Mpbs para encontrar el umbral del *throughput* en cada piso, además en cada piso con la herramienta iwlist se mide el número de redes Wi-Fi que pueden interferir con la red ad hoc desplegada. Para generar el tráfico se empleó la herramienta `iperf`. Además se utilizó la herramienta ping para determinar el retardo (*delay*) y los paquetes perdidos (*packet loss*) que existe en cada piso.

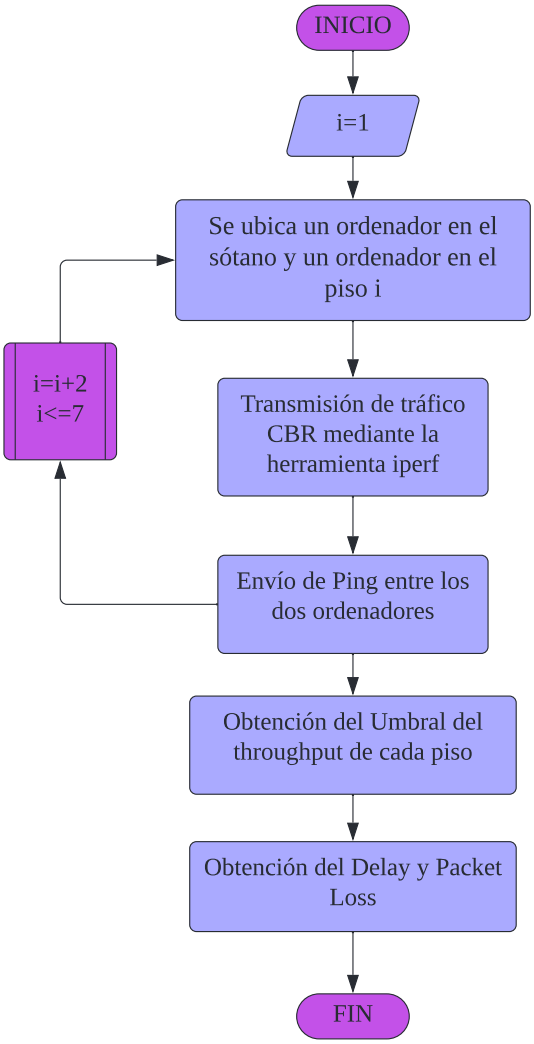
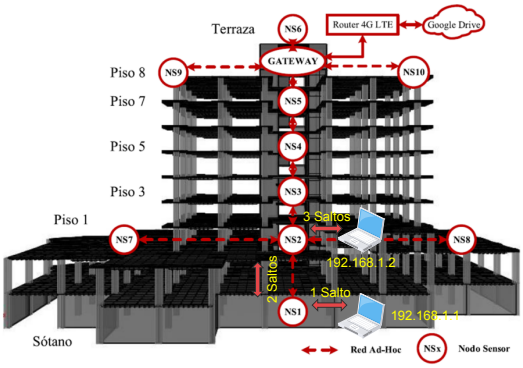
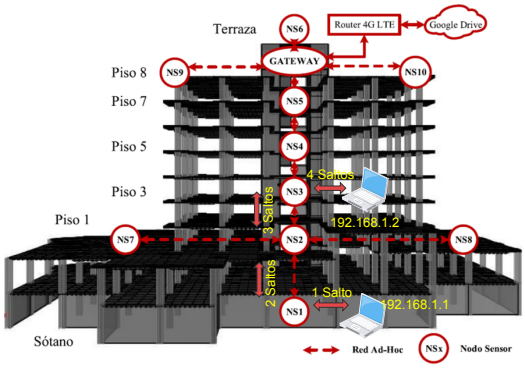


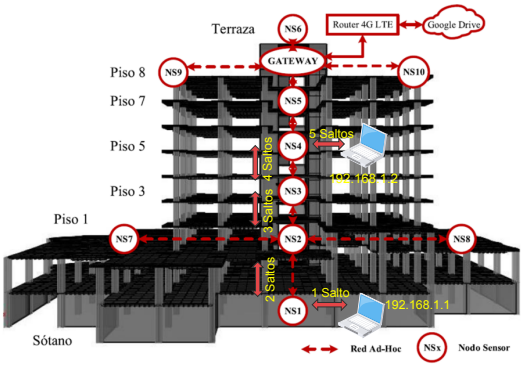
Figura 4.1: Metodología implementada para la caracterización de la red ad hoc.



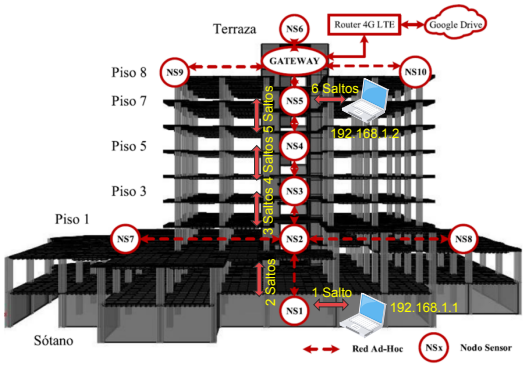
(a) Experimento 1: 3 saltos



(b) Experimento 2: 4 saltos



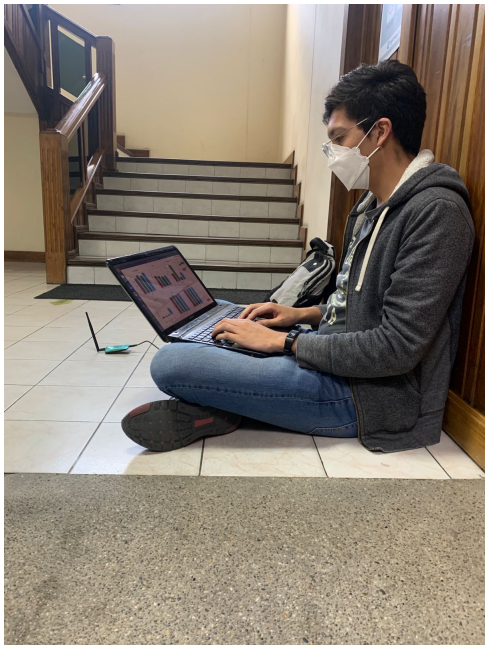
(c) Experimento 3: 5 saltos



(d) Experimento 4 : 6 saltos

Figura 4.2: Experimentos por piso.

(a) Ubicación del PC en uno de los pisos.



(b) Ubicación del PC en el sótano.

Figura 4.3: Ubicación de los ordenadores en el edificio (Centrosur).

4.2. 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 la Figura 4.4 se presenta un resumen de la metodología diseñada para definir el mejor códec de video en el escenario de estudio. Como se puede apreciar en la Figura 4.4, el primer paso es escoger algunos videos en formato CIF similares a un contexto de videoconferencia. En tal sentido, se escogieron y descargaron desde la página web *YUV Video Sequences* [103] tres videos que presentan diferentes condiciones como por ejemplo, cambios rápidos y lentos, diferente cantidad de objetos presentes y estabilidad o no de la cámara. En la Tabla 4.1 se presenta un resumen de las principales características de los videos seleccionados.

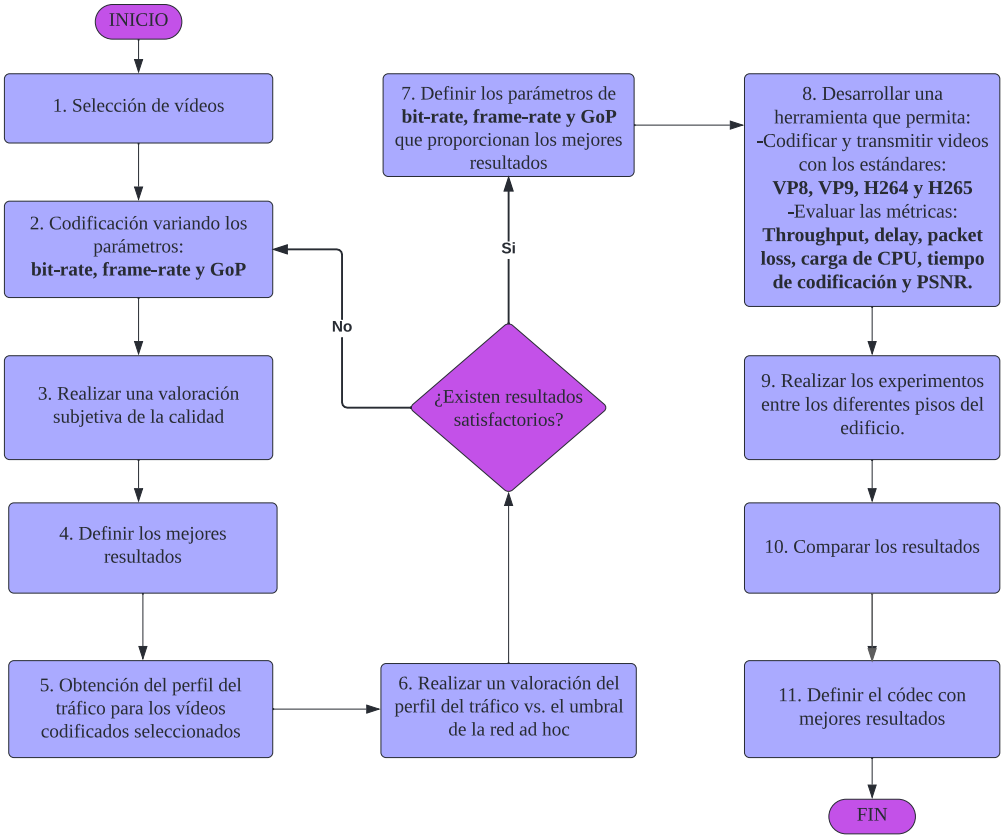


Figura 4.4: Metodología para realizar una evaluación objetiva de la calidad de video.

A continuación, con el uso de la herramienta FFmpeg que consiste en una biblioteca de codificadores [104], se procedió a codificar los videos empleando como parámetro principal el *bitrate* y como parámetros adicionales el *GoP* y el *framerate*. En este punto se consideró el umbral determinado en la caracterización de la red ad hoc, el mismo que se determina aplicando la metodología descrita en la Sección 4.1. El tercer paso consiste en realizar una valoración subjetiva de la calidad del video con los diferentes parámetros de codificación. Para esta evaluación se aplicó el método DSIS, descrito en la Sección 2.3.5.b, a tres personas y se empleó como referencia el estándar H.264. El cuarto paso fue definir de forma subjetiva la mejor opción, realizando una media entre las valoraciones de las tres personas.

Tabla 4.1: Características principales de cada video.

Video	Forman	Mother-daughter	News
Duración	12 seg	12 seg	12 seg
Frames	300	300	300
Formato	CIF	CIF	CIF
Descripción	•Existe movimiento de la cámara. •Existe un primer plano de un rostro gesticulando. •Video llamada típica.	•Existen dos personas, una en constante movimiento mientras que la segunda permanece estática. • Fondo fijo o estático.	•En la primera capa existen dos personas gesticulando. •En una segunda capa (fondo), se muestran dos personas bailando (bastante movimiento).

A continuación, utilizando la herramienta GStreamer y las métricas escogidas a partir del análisis subjetivo, se codificaron y transmitieron en tiempo real los videos sobre el propio ordenador, donde se capturanron todos los paquetes 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 ajustaron nuevamente los valores del *bitrate* y demás parámetros de codificación volviendo al paso 2. El siguiente bloque tiene como objetivo definir los parámetros de *bitrate*, GoP y *framerate* que proporcionen los mejores resultados buscando un equilibrio entre calidad y compresión.

En el paso número 8, se desarrolló una herramienta en Python con interfaz gráfica y con capacidad de codificar y transmitir en tiempo real con los códecs: H.264, H.265, VP8 y VP9. Además de evaluar las métricas de *throughput*, *delay*, *packet loss*, carga de CPU, tiempo de codificación y PSNR. Esta herramienta fue aplicada para las pruebas de transmisión de video sobre la red ad hoc. Una vez lista la herramienta, se realizaron los experimentos entre los diferentes pisos del edificio con los distintos videos escogidos y se compararon los resultados. Finalmente se definió el códec con mejores resultados, el cual se emplea posteriormente en el desarrollo de una herramienta de video conferencia como se describe a continuación en la Sección 4.3.

4.3. Diseño de una herramienta para la evaluación objetiva de la calidad de video

Para poder evaluar la calidad del video se desarrolló una herramienta software con interfaz gráfica en Python, tanto para el transmisor como para el receptor. En la Figura 4.5, se muestra un diagrama secuencial del funcionamiento de la herramienta y la comunicación entre el transmisor y el receptor, donde los números identifican al botón dentro de las interfaces gráficas, siendo los de color rojo para el transmisor y los morados para el receptor. Esta herramienta permite transmitir cualquier video en formato YUV hacia otro ordenador dentro de la misma red y calcular las métricas de *delay*, thoughput, *packet loss*, consumo de CPU, tiempo de codificación y PSNR.

En este contexto, en la Figura 4.6 se muestra la interfaz del transmisor y en la Figura 4.7 se observa la interfaz del receptor. Las herramientas se diseñaron mediante scripts de Linux para cumplir con las distintas funcionalidades y los experimentos se efectuaron en ordenadores con sistema operativo Ubuntu 20.04. Para

efectuar los diferentes cálculos también se implementaron scripts en Python 3.7. Cabe resaltar que la herramienta desarrollada en este trabajo de tesis ha sido liberada para su uso y se puede descargar en [105] y [106] para el transmisor y receptor respectivamente.

Con respecto al funcionamiento de la herramienta, primero se cargan los datos mediante el botón *Cargar*. Para el transmisor se selecciona el video deseado en formato YUV, las IPs de origen y destino, el códec, el puerto de destino, el *framework* multimedia, el piso del ordenador receptor y finalmente el número de transmisión. Del lado del receptor se cargan las IPs de origen y de destino, el códec y el puerto de recepción. Seguidamente, por medio del botón *Sincronizar*, se sincroniza con el servidor NTP ubicado en uno de los ordenadores.

Posteriormente, en el lado del receptor se da clic al botón *Recibir*, para poder admitir flujos de video. A continuación, se envía el video al presionar el botón *Transmitir* y se reproduce automáticamente en el lado del receptor. Al terminar de transmitir y reproducir el video, en cada lado se da clic al botón *Stop* para finalizar la captura de paquetes y convertirlos a Comma Separated Values (CSV). Luego, del lado del transmisor se oprime el botón *Recibir PCAPs*, para abrir un puerto determinado para aceptar los archivos con la herramienta *netcat*.

A continuación, desde el usuario receptor se envían al usuario transmisor los archivos Packet Capture (PCAP) capturados convertidos al formato CSV. Finalmente se da clic en el botón *Calcular* ubicado en la herramienta del usuario transmisor para medir el *throughput*, el porcentaje de uso de la CPU, el *delay* promedio, el porcentaje de *packet loss* y finalmente el tiempo de transmisión.

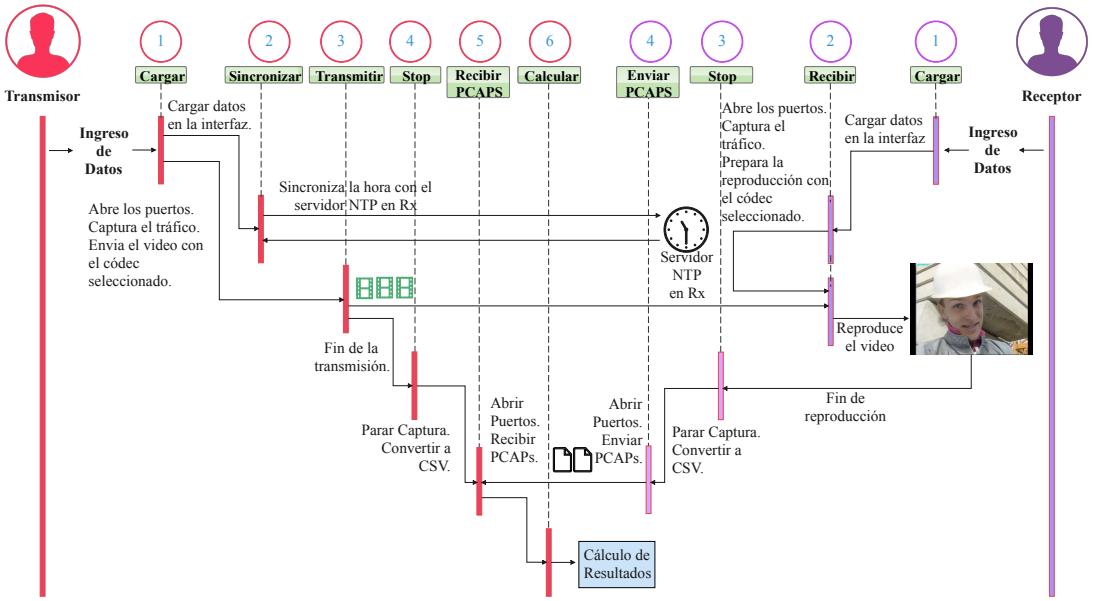


Figura 4.5: Diagrama secuencial de las herramientas en Tx y Rx.

Además, se incluyeron los botones: *Codificar Video*, *Play Video*, *Generar Gráficas* y *GRÁFICAS*. De los cuales, el primero se utiliza para codificar y almacenar un video con los parámetros seleccionados pero no se transmite. El botón *Play Video* sirve para reproducir el video previamente codificado en el transmisor.

Figura 4.6: Interfaz gráfica de la herramienta para el transmisor.

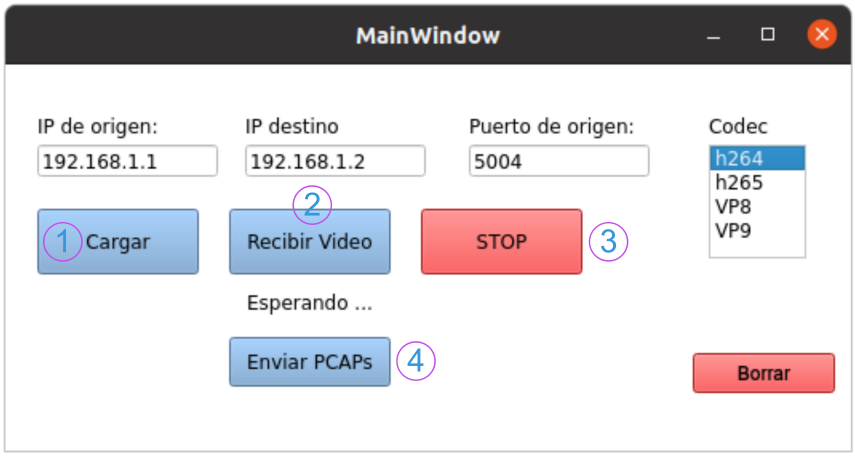


Figura 4.7: Interfaz gráfica de la herramienta para receptor.

Por último, los dos botones *Generar Gráficas* y *GRÁFICAS* abren dos diferentes interfaces, con el primer botón se abre la interfaz que se observa en la Figura 4.8, donde se realiza una comparación entre los códecs por cada transmisión de los cuatro diferentes códecs. En cambio, el segundo botón abre otra interfaz que se observa en la Figura 4.9 donde se realiza un promedio de las cinco diferentes transmisiones de cada métrica (*delay*, *throughput*, *packet loss*, PSNR, consumo de CPU y tiempo de codificación) y se determinan los intervalos de

confianza del 95 % de fiablidad. Al final de esta interfaz se encuentra el botón *Grabar*, el cual mueve todas las figuras generadas a una carpeta específica donde existe una mayor organización para documentar los resultados. Cabe recalcar que en las dos interfaces se puede seleccionar el tipo de herramienta y el video a ser analizado.

Figura 4.8: Interfaz gráfica para resultados comparativos.

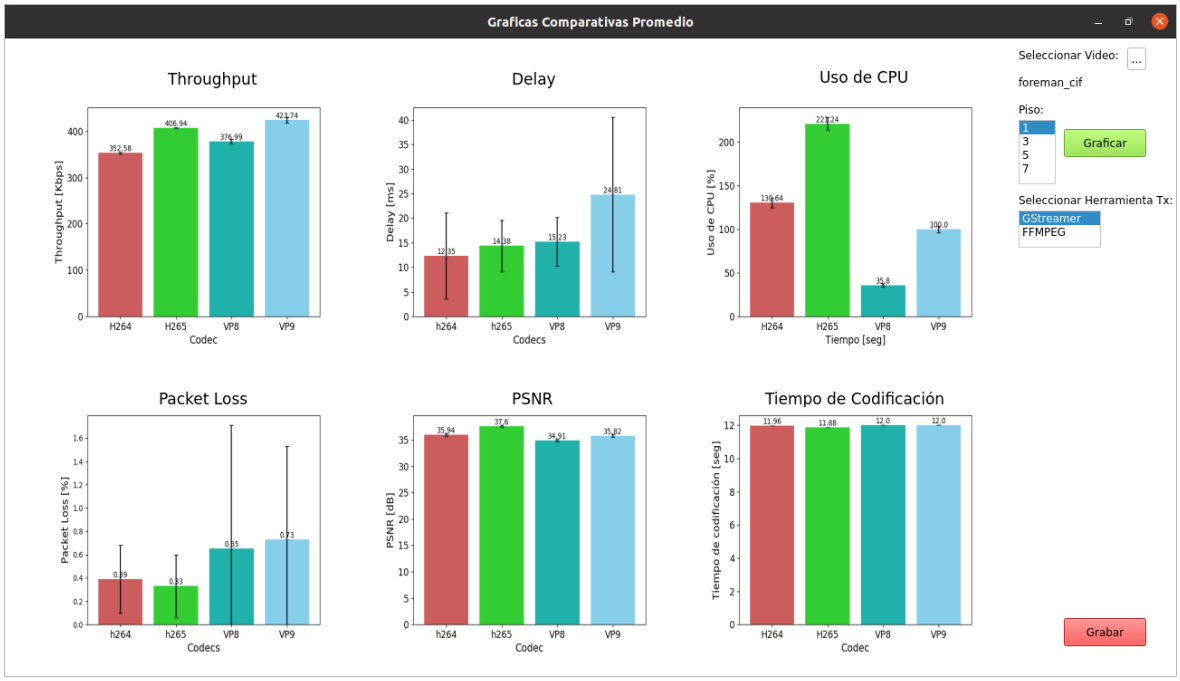


Figura 4.9: Interfaz gráfica para resultados totales con un intervalo de confianza del 95 %.

4.4. Herramienta para la transmisión de audio y video en tiempo real

Siguiendo un esquema parecido a aplicaciones como Zoom o Google Meet, se desarrolló una herramienta para la transmisión de audio y video en tiempo real con una sencilla interfaz y facilidad de uso. La misma que, reúne las características de reuniones individuales y la capacidad de video adaptativo. La herramienta funciona en conjunto con un servidor, el cual recolecta información de los usuarios (nombre, apellido e IP) y crea una base de datos al que cualquier usuario puede acceder al ingresar. Cabe resaltar que la herramienta desarrollada en este trabajo de tesis ha sido liberada para su uso y se puede descargar del repositorio de **Github** [107].

En la Figura 4.10 se presenta un diagrama secuencial del funcionamiento de la herramienta para la transmisión de audio y video en tiempo real. Los números identifican los botones dentro de las interfaces gráficas que se presentan en la Figura 4.11. Para el manejo de esta herramienta de video conferencia, primero se proporciona un nombre y un apellido y se da clic en el botón *Ingresar*, como resultado se envían los datos junto con la dirección IP a un servidor local para formar una base de datos con los usuarios registrados con su respectiva dirección IP.

Posteriormente, se abre otra interfaz (Figura 4.11b) donde se visualizan todos los usuarios que están en la base de datos del servidor. Cabe mencionar que al ingresar a esta interfaz la aplicación siempre está a la escucha de un iperf para poder establecer una llamada, al momento que un usuario llama a un contacto este se convierte en un cliente iperf enviando paquetes User Datagram Protocol (UDP) a un puerto específico (8001). Entonces, el usuario que recibe la llamada ejerce como servidor. De esta manera al detectar mensajes UDP entrantes en este puerto se despliega una interfaz para aceptar o declinar la llamada (Figura 4.11c).

En este contexto el usuario que recibe la llamada escucha un tono de llamada entrante. Además, con los resultados obtenidos del iperf se mide el ancho de banda de la red y así dependiendo de los umbrales del *throughput* obtenidos se codifica con diferentes valores de bitrate. De esta manera se consigue una opción de vídeo adaptativo, el cual es transparente al usuario. Al finalizar la llamada con el botón *Colgar*, los dos usuarios regresan a la interfaz principal y se ponen a la escucha de un iperf nuevamente.

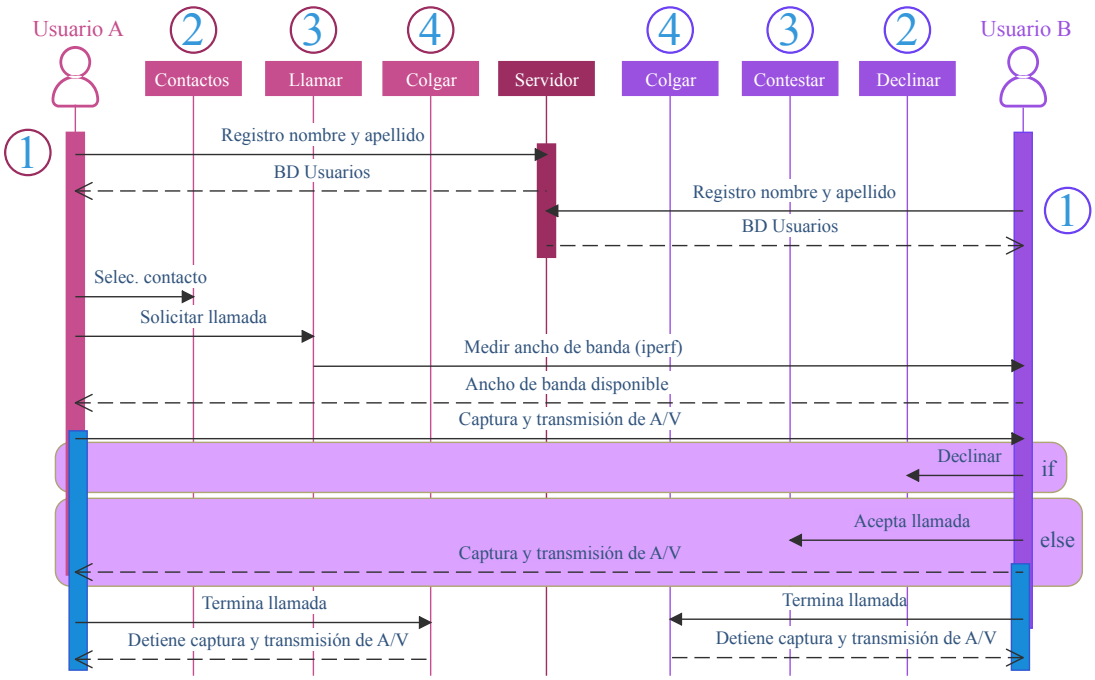
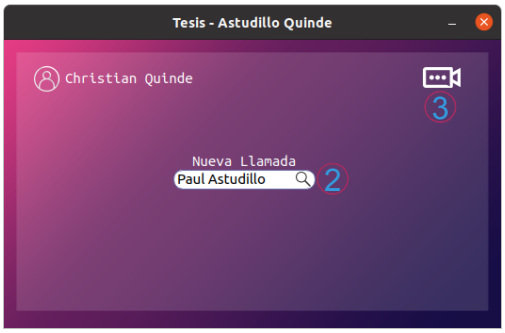


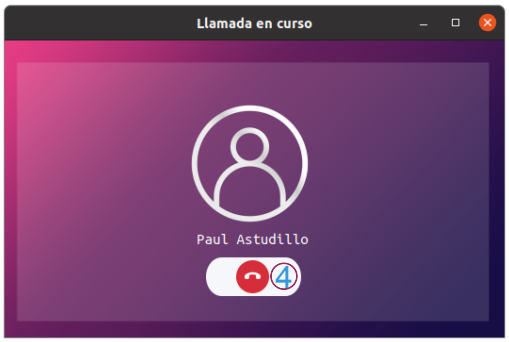
Figura 4.10: Diagrama de secuencia para la herramienta de audio y video en tiempo real.

(a) Ingreso.



(b) Pantalla principal.

(c) Llamada entrante.



(d) Llamada en curso.

Figura 4.11: Interfaces de la herramienta de audio y video en tiempo real.

Al ejecutar la herramienta se despliega un icono el cual representa la aplicación de audio y video, el icono se muestra en la Figura 4.12.

Figura 4.12: Icono representativo de la herramienta de audio y video en tiempo real.

4.5. Herramienta para el cálculo de métricas durante la transmisión de audio y video en tiempo real

En esta sección se presenta la arquitectura y el funcionamiento de la herramienta desarrollada para la obtención del *delay end-to-end*, el *throughput* y el porcentaje de *packet loss* tanto para audio como video. Cabe resaltar que la herramienta desarrollada en este trabajo ha sido liberada para su uso y se puede descargar en [108] y [109] para la herramienta de medición y el servidor respectivamente. El software desarrollado se ejecuta mientras se realiza una video conferencia utilizando la aplicación descrita en la Sección 4.4, para ello se da clic en el botón *Capturar*, lo que permite guardar todo el tráfico de la interfaz utilizada. Transcurridos dos minutos se detiene la captura y se procede a dar clic en el botón *Stop*, de modo que con el uso de tshark se filtran los paquetes por puertos para diferenciar los flujos de audio y video, y se convierte la información a archivos de texto plano. Finalmente, se da clic en el botón *Calcular* para determinar las métricas propuestas.

Para el cálculo del *delay* y de la pérdida de paquetes se hace uso de los paquetes RTCP, Sender Report (SR) y Receiver Report (RR), los mismos que se intercambian entre los dos terminales que realizan la video llamada. El *delay* se calcula utilizando la ecuación 4.1, 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, esto puede utilizarse como una medida aproximada del retardo entre los dos extremos.

$$delay = (A - LSR - DLSR)/2 \tag{4.1}$$

Para obtener el porcentaje de *packet loss* se utiliza el campo *fraction Loss* de los paquetes RR recibidos y el campo *cumulative number of packets lost* del último RR recibido. El primer campo representa la fracción de paquetes de datos RTP perdidos desde que se envió el anterior paquete SR, esta fracción se define como el número de paquetes perdidos dividido por el número de paquetes esperados (256) el cual esta definido por 8 bits dentro de campo *fraction lost* del paquete RTCP. El segundo campo utilizado contiene el número total de paquetes de datos RTP de la fuente que se han perdido desde el inicio de la recepción. Con estos valores y con el número de paquetes RR capturados se obtiene el *packet loss* mediante la Ecuación 4.2.

$$PL[\,\%] = \frac{\text{total de paquetes perdidos acumulados} \cdot 100\,\%}{total\_RR \cdot 256} \tag{4.2}$$

Por otro lado, para obtener el *throughput* de la transmisión, ya sea audio o video, se hace uso de 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. Finalmente, en la Figura

4.13 se presenta un diagrama del funcionamiento de la herramienta desarrollada en Python, donde los números identifican los botones dentro de la interfaz gráfica mostrada en la Figura 4.14. Cabe mencionar que el botón *Promediar*, realiza un promedio de 10 transmisiones y hace una comparación de cada métrica por piso y el botón *Cargar* permite ver los resultados de cualquier transmisión realizada.

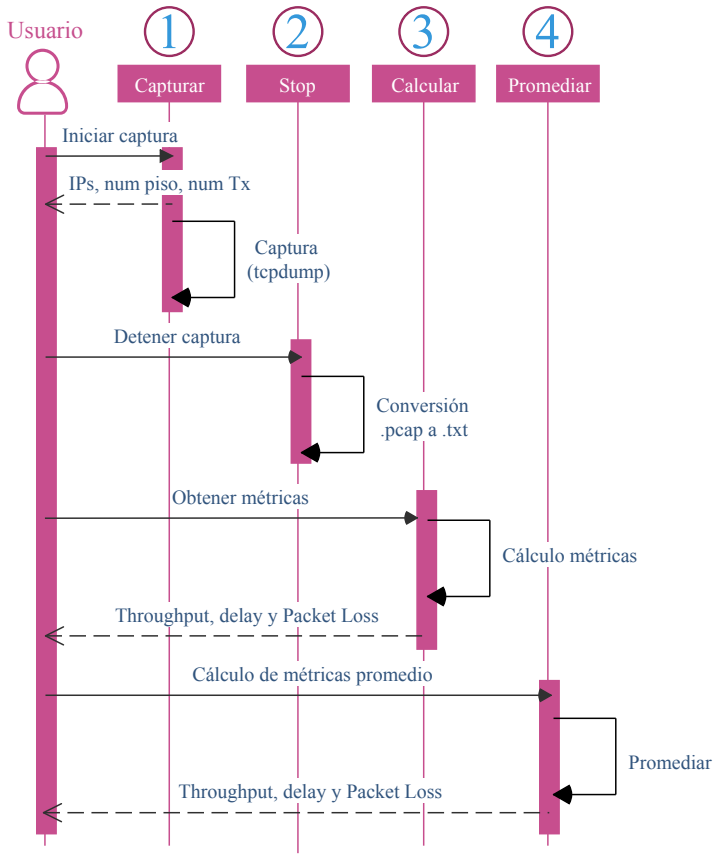


Figura 4.13: Diagrama secuencial de la herramienta de medición de métricas.

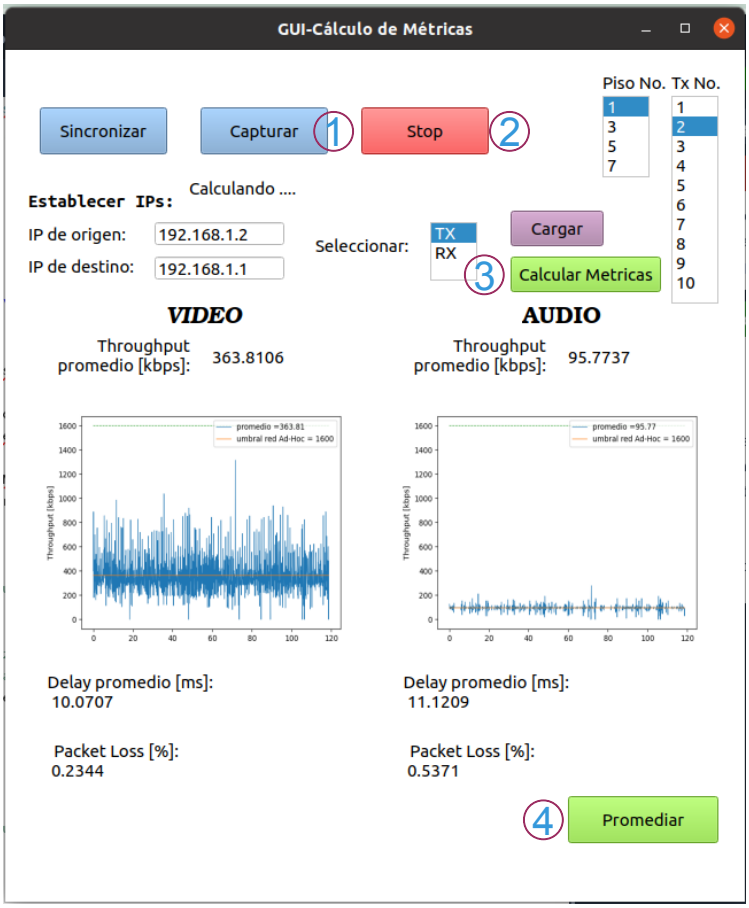


Figura 4.14: Interfaz de la herramienta para el cálculo de métricas.

Capítulo 5

Evaluación Experimental y Resultados

En este capítulo se presentan los resultados de los experimentos realizados en el edificio de la Centrosur, basándose en la metodología planteada en el Capítulo 4. Además, se realiza un análisis de cada métrica para escoger el códec más adecuado para implementar la herramienta de video conferencia.

Este capítulo se compone de las siguientes secciones, en la Sección 5.1 se presentan los resultados de la caracterización de la red ad hoc desplegada en el edificio de la Centrosur, centrándose en las métricas de *throughput*, *delay* y porcentaje de *packet loss*. Seguidamente, en la Sección 5.2, se detalla una evaluación objetiva de la calidad del video, donde se realiza un análisis de desempeño de cada códec de video en los diferentes pisos del edificio. De este modo, en la Sección 5.3 se escoge el códec con los mejores resultados a partir de las métricas evaluadas. Finalmente, en la Sección 5.4 se evalúa la herramienta para la transmisión de audio y video en tiempo real, con el códec definido previamente.

5.1. Caracterización de la red ad hoc

De acuerdo a lo descrito en el Capítulo 4, 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 de los resultados.

5.1.1. Análisis del throughput

Los resultados obtenidos demuestran que el *throughput* tiene un umbral distinto en cada piso del edificio, esto debido al número de saltos generados y al número de interferencias en ese instante. Se evidencia que mientras se incrementa el número de saltos disminuye el *throughput* de la red, tal y como se observa en la Figura 5.1. Cabe considerar que la red tiene un comportamiento diferente dependiendo de la hora del día, por lo que se decidió realizar los experimentos entre las 14h30 y las 17h30, donde la red tienen el menor número de interferencias y el mayor ancho de banda. De los resultados obtenidos, se puede observar que el máximo *throughput* alcanzado es de 3.5 Mbps y el mínimo de 1.6 Mbps para los pisos 1 (3 saltos) y 7 (6 saltos) respectivamente. Al comparar los resultados con los obtenidos en [102], se observa que los valores de *throughput* disminuyen debido a que en

este este experimento no existe línea de vista entre los nodos extremos y los nodos de la red preexistente, sin embargo, la tendencia se mantiene y se establece un umbral diferente dependiendo del número de saltos. En la Tabla 5.1, se presenta el resumen del máximo *throughput* promedio dependiendo del número de saltos.

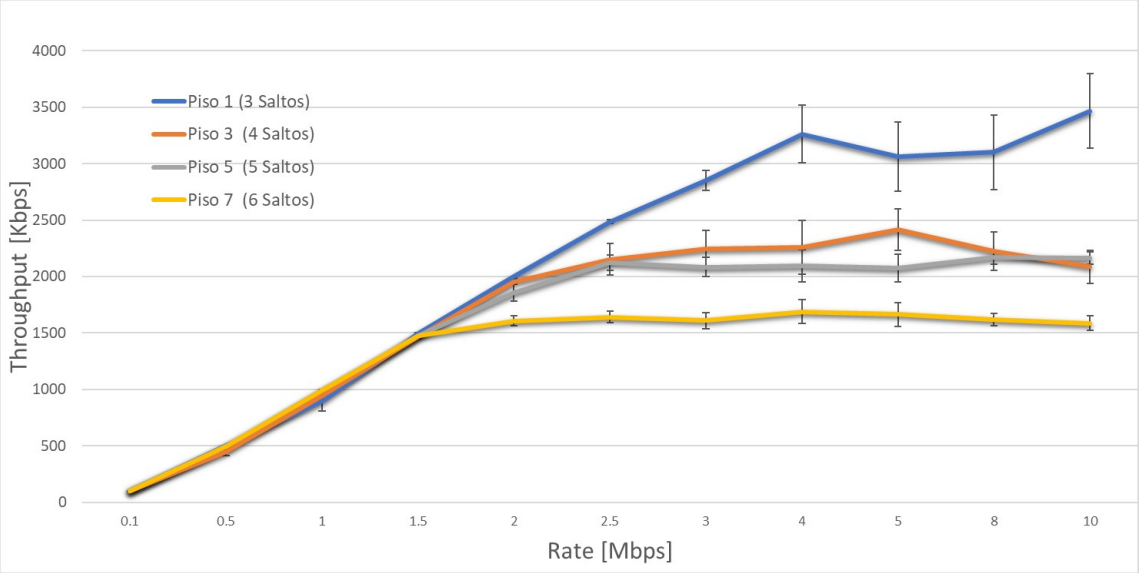
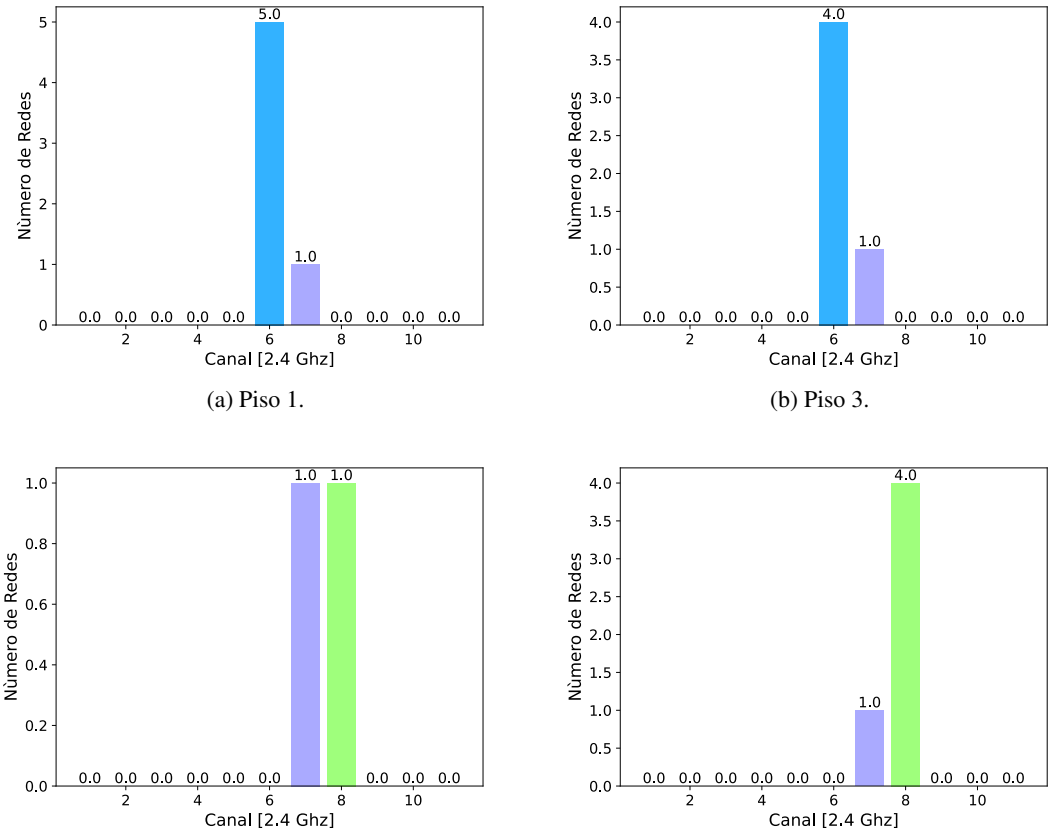


Figura 5.1: Throughput [Kbps] vs data rate deseado [Mbps].

Tabla 5.1: Throughput promedio.

Saltos	Throughput Promedio [Mbps]
3	3.5
4	2.3
5	2
6	1.6

Siguiendo con el análisis, cabe mencionar que los mecanismos de reconocimiento de mensajes (ACK), los rangos de transmisión e interferencias y la contención por el medio deriva una reducción del *throughput*. Por lo cual, se realizó un escaneo del medio inalámbrico para comprobar el número de redes que existen utilizando el mismo canal Wi-Fi. En la Figura 5.2 se presenta un resumen del número de redes existentes en cada piso del edificio. Cabe señalar que estas mediciones se realizaron a partir de las 14h30 y adicionalmente que la red se encuentra configurada en el canal número 7. Considerando que estas mediciones varían dependiendo de la hora en la que se realizó la medición y del número de piso, no se cambió el canal configurado previamente en la red ad hoc. Por otro lado, en la Figura 5.3 se presentan los resultados de un escaneo realizado en la mañana, a las 11h00, donde es evidente que el número de redes aumenta, y por lo tanto, aumentan las interferencias. En este caso el *throughput* de la red no superaba los 2 Mbps en el piso 1 (3 saltos).



(a) Piso 1.

(b) Piso 3.

(c) Piso 5.

(d) Piso 7.

Figura 5.2: Redes Wi-Fi existentes en cada canal.

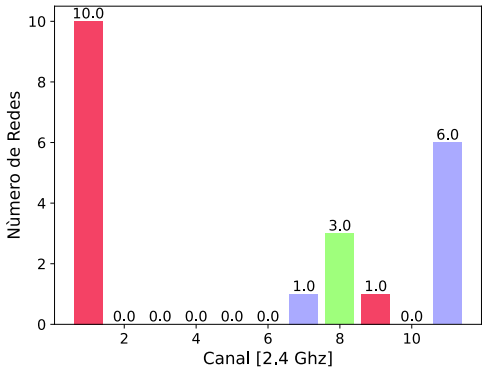


Figura 5.3: Redes Wi-Fi existentes por canal en el piso 1 (11h00).

5.1.2. Análisis del delay y packet loss

Para obtener el retardo de la red entre los ordenadores (es decir los nodos extremo), se utilizó la herramienta ping, tal y como lo describe la Figura 4.1. En particular, se envió un total de 30 mensajes Internet Control Message Protocol (ICMP) de los cuales se obtuvo el número de paquetes perdidos y el tiempo requerido para cada uno.

Además, mediante un script de Python, se extrajo la media de estos tiempos y se calcularon los intervalos de confianza del 95 % de fiabilidad. Los resultados para cada piso se muestran en la Figura 5.4. Estos resultados muestran nuevamente que aumenta el retardo conforme se incrementa el número de saltos. Adicionalmente, de estos datos se observa que el mayor retardo es de 14.02 ms y el mínimo de 5.47 ms, para el piso 7 y 1, respectivamente. Por otro lado, para obtener el porcentaje de *packet loss* se utilizaron los mismos resultados obtenidos con la herramienta ping, donde se realizó un promedio de las 30 repeticiones. En la Figura 5.5, se presentan los valores obtenidos para cada piso, donde se observa que existe una baja pérdida de paquetes; sin embargo, esta aumenta para los pisos 5 y 7 en comparación con los pisos 1 y 3.

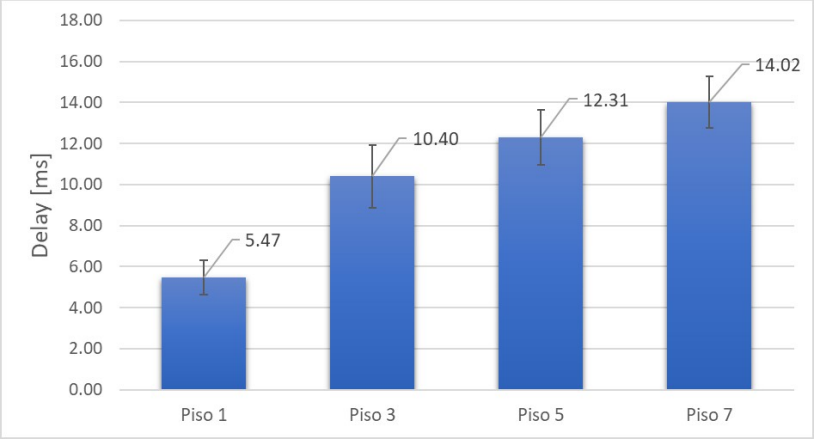


Figura 5.4: Retardo vs número de piso.

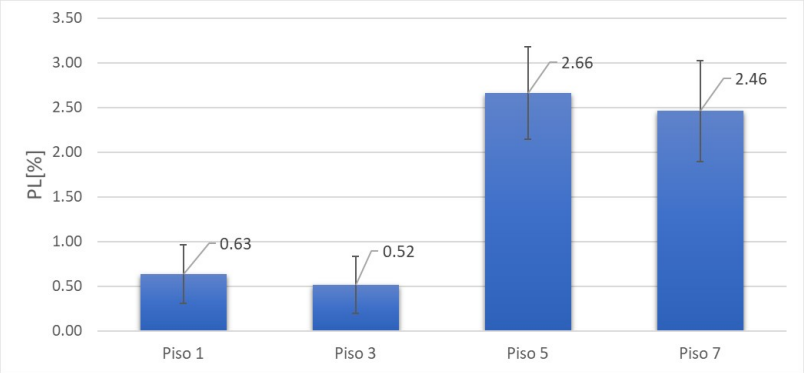


Figura 5.5: Perdida de paquetes vs número de piso.

5.2. Evaluación subjetiva y objetiva de la calidad de video

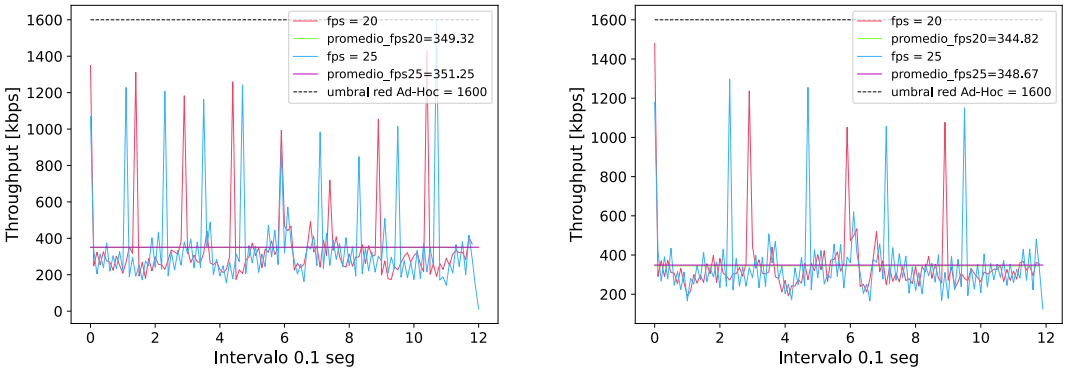
Siguiendo el esquema de la Figura 4.4 se obtuvo la puntuación media de opinión (MOS) de la calidad subjetiva por parte de las tres personas. Para determinar el resultado final, se promediaron las tres puntuaciones. Después se realizó un filtrado para obtener los mejores resultados. Para los videos de Foreman y Mother-daughter se usó un MOS mayor a 4, en cambio para el video News se usó un MOS mayor a 3, en total se obtuvo 42 videos. Al analizar estos resultados se observó que no tuvo mayor efecto el *bitrate* en el análisis subjetivo, con lo que se realizó un nuevo filtro con un *bitrate* de 350 kbps. De esta manera se generaron 4 videos para Foreman,

News, y Mother-daughter con dos diferentes valores de GoP y fps, tal como se muestran en la Tabla 5.2.

Tabla 5.2: Mejores resultados del análisis subjetivo con filtro de bit-rate = 350 kbps

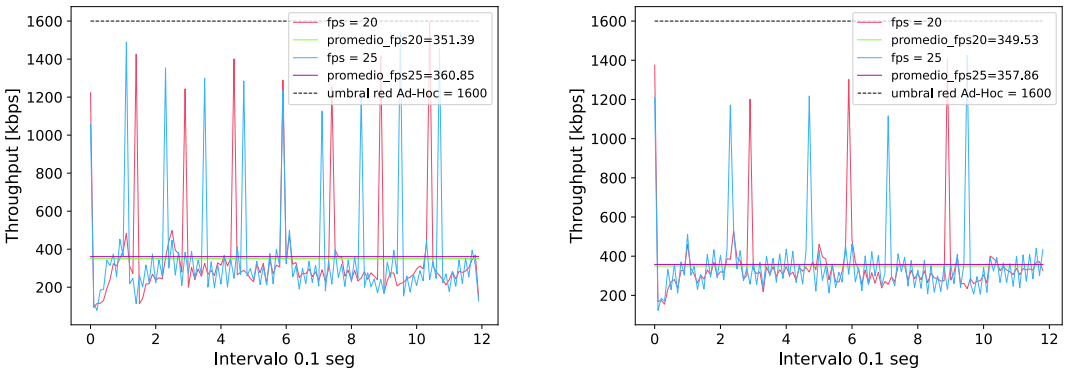
Videos	GOP	FPS
Foreman	30	20
Mother-daughter		25
News		
Foreman	60	20
Mother-daughter		25
News		

Continuando con el análisis de resultados, para cada video que se indica en la Tabla 5.2, se graficaron los perfiles de tráfico como se muestra en las figuras 5.6, 5.7, 5.8. Específicamente, se muestran las gráficas para cada video, agrupadas por GoP y con un intervalo de 0.1 segundos.



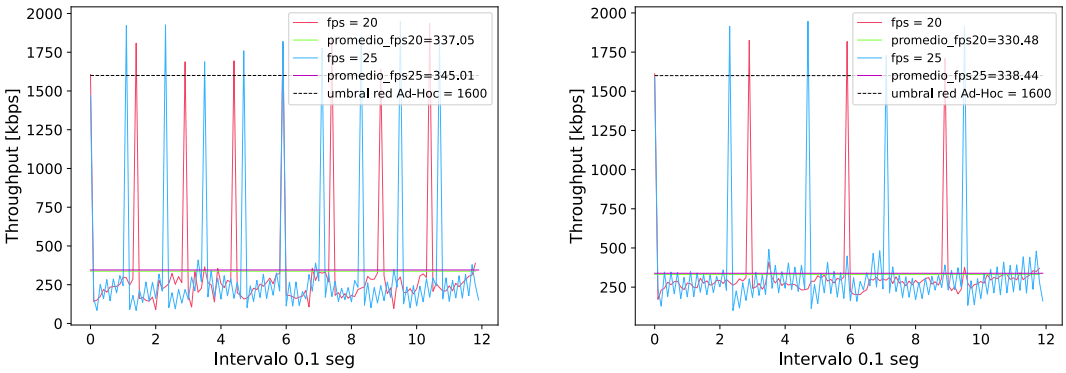
(a) GoP = 30 (b) GoP = 60

Figura 5.6: Perfil de tráfico para el video “Foreman” con bit rate de 350 Kbps.



(a) GoP = 30 (b) GoP = 60

Figura 5.7: Perfil del tráfico para el video “Mother-daughter” con bit rate de 350 Kbps.



(a) GoP = 30 (b) GoP = 60

Figura 5.8: Perfil de tráfico para el video “News” con bit rate de 350 Kbps.

Como se observa en las gráficas anteriores, para los videos “Foreman” y “Mother-daughter” en ningún caso se supera el umbral de la red ad hoc establecido. Por dicho motivo, se eligieron las métricas que aseguran una mayor calidad y fluidez, es decir un tamaño de GoP igual a 30 y un valor de 25 fps. En cambio, para el video “News” donde existe mayor movimiento, se nota que se supera el umbral en cualquiera de los dos casos en términos de GoP. De esta manera, se escogen los mismos valores que los video anteriores. Por último, las mejores métricas se resumen en la Tabla 5.3.

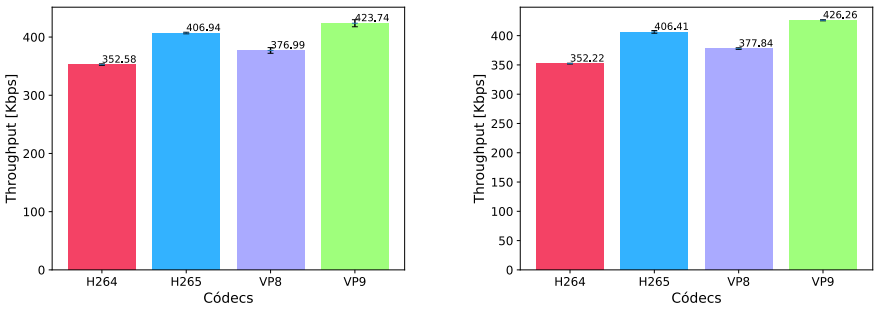
Tabla 5.3: Métricas que proporcionan mejores resultados.

Bit-rate	Frame-rate	GoP
350 kbps	25 fps	30

5.2.1. Evaluación del throughput

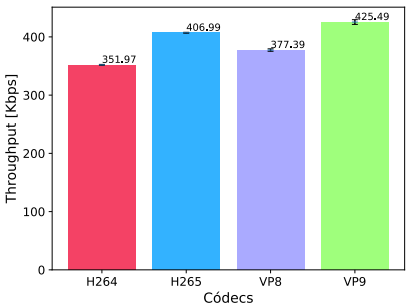
Con el objetivo de analizar el *throughput*, se utilizó la herramienta descrita en la Sección 4.3, se realizaron cinco transmisiones con cada códec de video en todos los pisos. A continuación, en las Figuras 5.9, 5.10 y 5.11 se presentan los resultados obtenidos de *throughput* para los tres videos, para cada códec y en cada piso. Las figuras muestran una comparación entre los cuatro códecs empleados y se observa que VP9 requiere mayor ancho de banda, para la transmisión en todos los pisos y para cada video. 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.

También se notó que los videos con mayor movilidad como “News” y “Foreman” requieren en promedio un mayor ancho de banda que el video “Mother-daughter”, el cual es más estático. Por otro lado, se tiene que el valor del *throughput* está alrededor de los 350kbps que es la tasa fijada en cada uno de los codificadores, sin embargo, cabe señalar que este número es una guía para el codificador. El grado de rigor con el que el codificador trata de ajustarse al valor establecido, ya sea fotograma a fotograma o promediado a lo largo de la duración del clip, se controla mediante parámetros adicionales como se indica en [110].

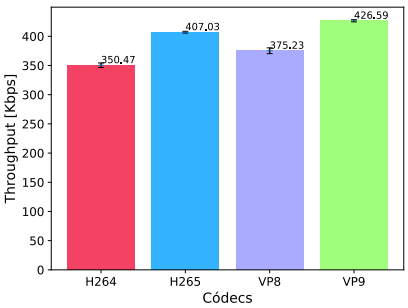


(a) Piso 1

(b) Piso 3

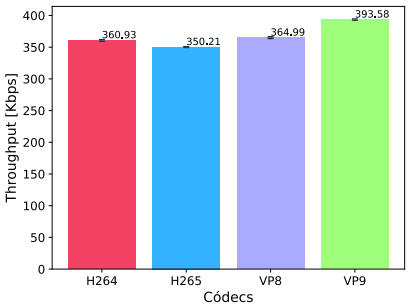


(c) Piso 5

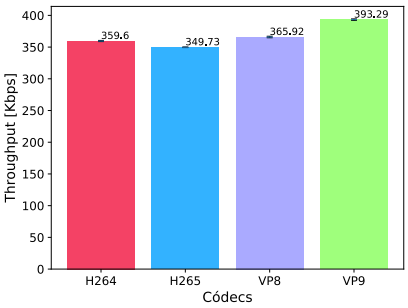


(d) Piso 7

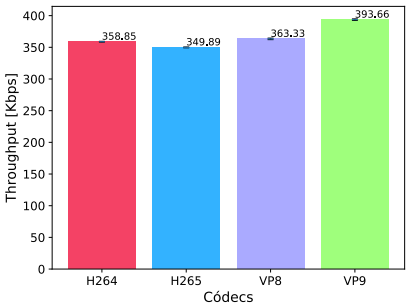
Figura 5.9: Throughput promedio para el video “Foreman” en cada piso.



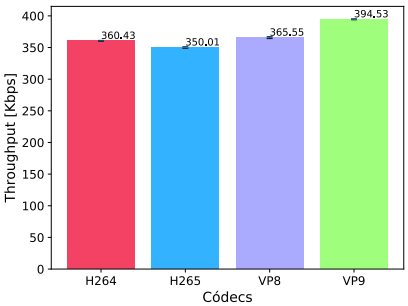
(a) Piso 1



(b) Piso 3

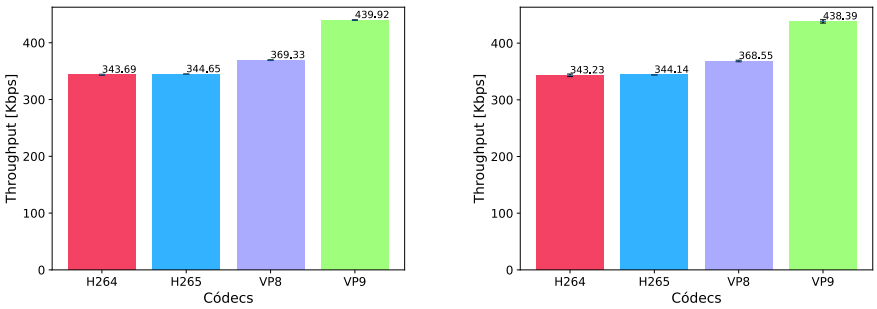


(c) Piso 5



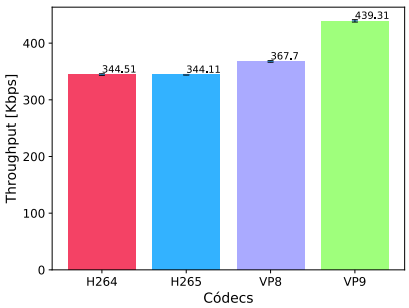
(d) Piso 7

Figura 5.10: Throughput promedio para el video “Mother-daughter” en cada piso.

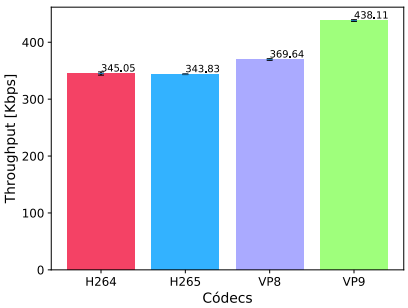


(a) Piso 1

(b) Piso 3



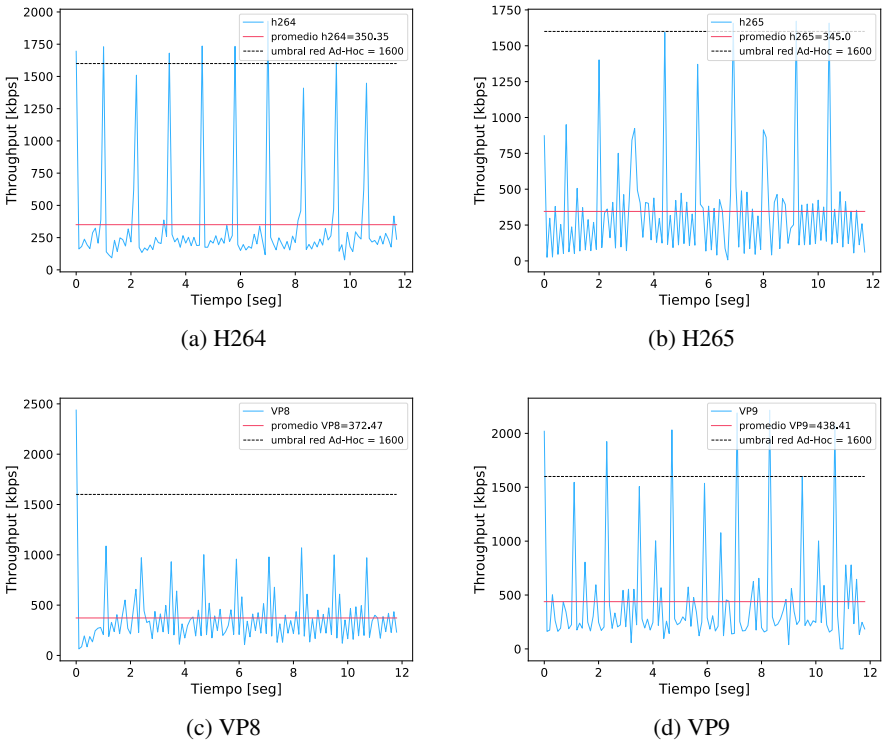
(c) Piso 5



(d) Piso 7

Figura 5.11: Throughput promedio para el video “News” en cada piso.

Además, en la Figura 5.12, se presenta los perfiles del tráfico de una tranmisión del video “News” con cada códec en el piso 7 (6 saltos). Como era de esperarse, el perfil del tráfico es variable y presenta ciertos picos. Estos picos al superar el umbral de la red pueden generar pérdida de paquetes, por lo que los mejores resultados se obtuvieron con los códecs VP8 y H265, seguidos de H264 y por último VP9.



(a) H264 (b) H265

(c) VP8 (d) VP9

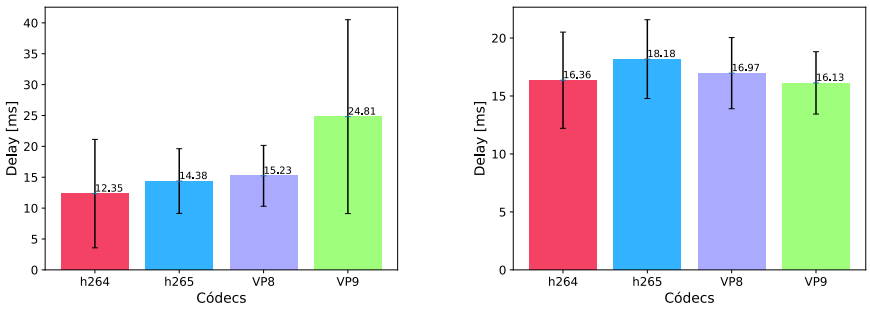
Figura 5.12: Throughput para el video “News” en el piso 7.

5.2.2. Evaluación del retardo

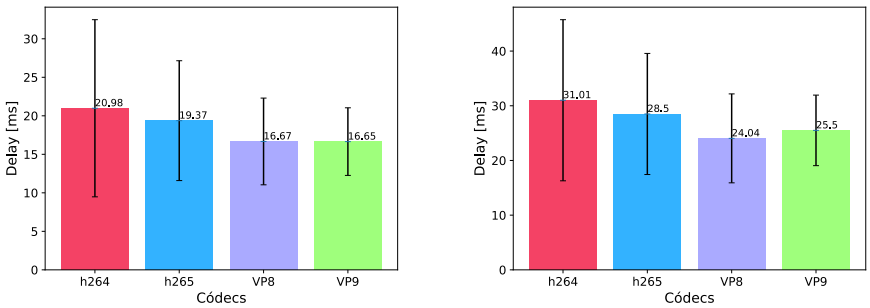
En las Figuras 5.13, 5.14 y 5.15 se presentan los resultados promedios del *delay* de las cinco trasmisiones correspondientes a cada video y se realiza una comparación entre cada piso. Para hallar el valor del *delay* se restan los tiempos entre el transmisor y receptor con ayuda del protocolo RTCP y un servidor NTP que sincroniza a las dos terminales.

Se puede notar que al aumentar de piso (es decir el número de saltos) el *delay* incrementa, guardando relación con la teoría y la caracterización de la red ad hoc detallada en la Figura 5.4. 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 el caso más extremo, que es el piso 7 con un número de saltos de seis, se nota que en términos de *delay*, fácilmente la aplicación cumple con los requisitos de transmisión en tiempo real para una videoconferencia establecida por la recomendación UIT-T G.114 [111], la cual recomienda un retardo menor a 150ms.

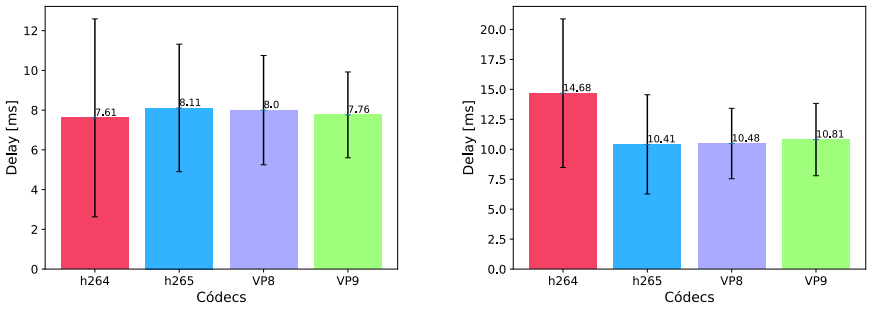


(a) Piso 1 (b) Piso 3

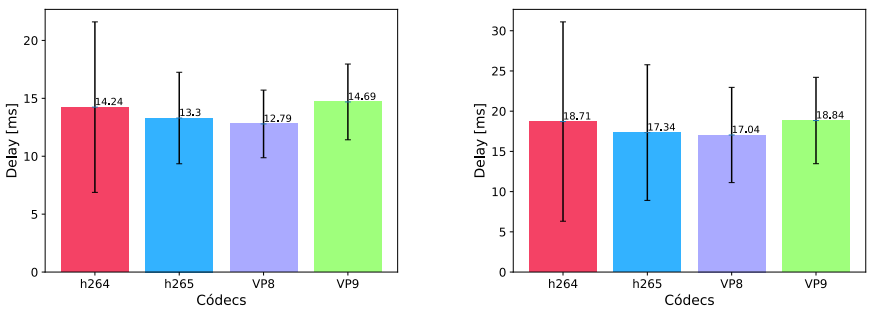


(c) Piso 5 (d) Piso 7

Figura 5.13: Delay promedio obtenido para el video “Foreman” en cada piso.

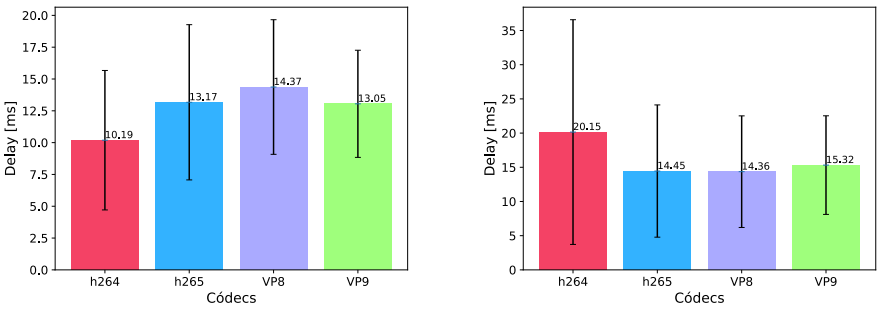


(a) Piso 1 (b) Piso 3



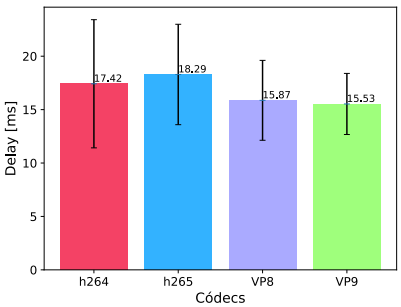
(c) Piso 5 (d) Piso 7

Figura 5.14: Delay promedio obtenido para el video “Mother-daughter” en cada piso.

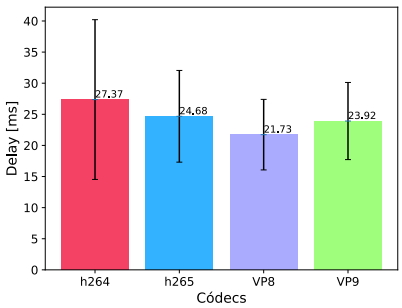


(a) Piso 1

(b) Piso 3



(c) Piso 5



(d) Piso 7

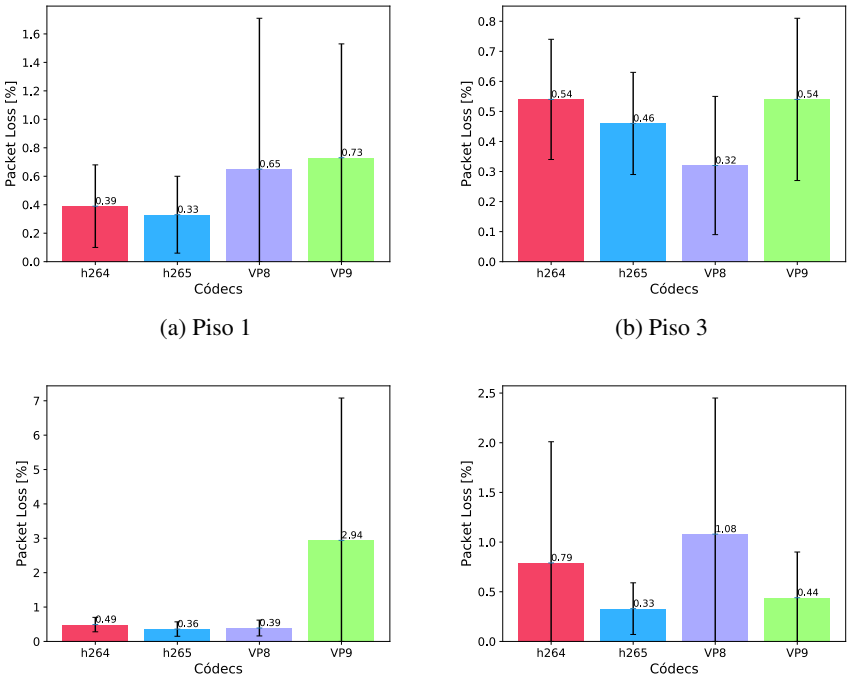
Figura 5.15: Delay promedio obtenido para el video “News” en cada piso.

5.2.3. 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 recibidos en el receptor utilizando la Fórmula 5.1. 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). En las Figuras 5.16, 5.17 y 5.18 se presentan los resultados obtenidos para cada video en cada piso. De estas gráficas se nota, que a partir de la gran amplitud de las barras de error 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 %.

$$\text{Packet Loss \%} = (\text{Paquetes enviados} - \text{Paquetes recibidos}) / \text{Paquetes enviados} \tag{5.1}$$

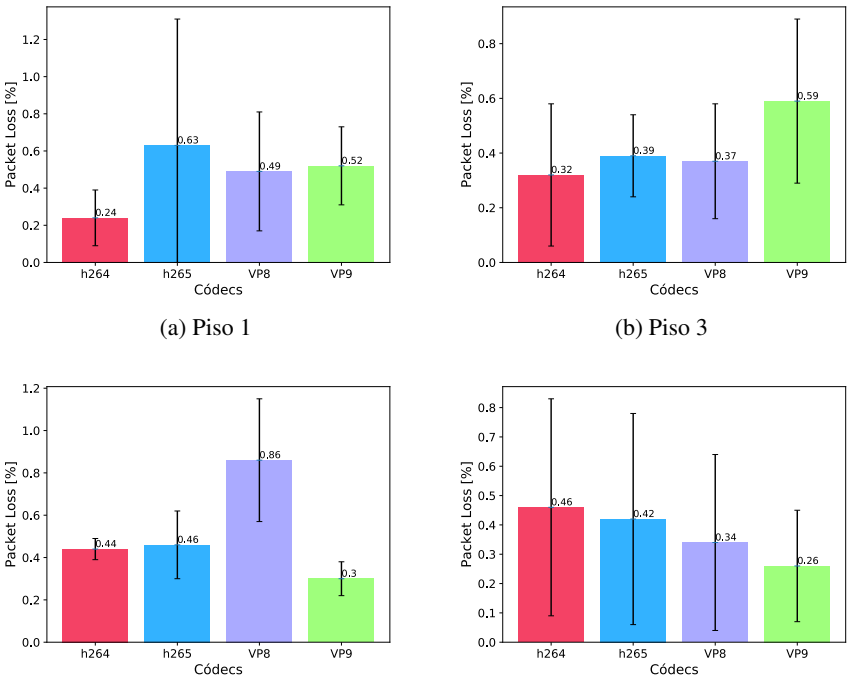
En cuanto al comportamiento de cada uno de los códecs, y basándose en los resultados obtenidos, no se puede determinar el mejor códec con respecto a esta métrica. Sin embargo, la pérdida de paquetes de cualquiera de los códecs evaluados está por debajo del umbral (3-5 %), en consecuencia se considera una pérdida de paquetes “aceptable” para una llamada de voz o video [112].



(a) Piso 1 (b) Piso 3

(c) Piso 5 (d) Piso 7

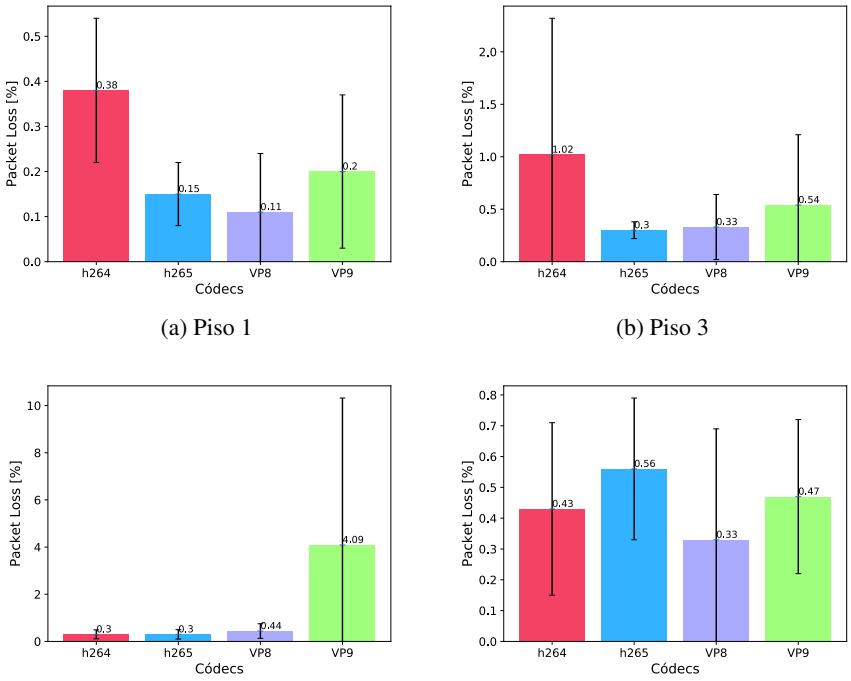
Figura 5.16: Porcentaje de packet loss para el video “Foreman” en cada piso.



(a) Piso 1 (b) Piso 3

(c) Piso 5 (d) Piso 7

Figura 5.17: Porcentaje de packet loss para el video “Mother-daughter” en cada piso.



(a) Piso 1

(b) Piso 3

(c) Piso 5

(d) Piso 7

Figura 5.18: Porcentaje de packet loss para el video “News” en cada piso.

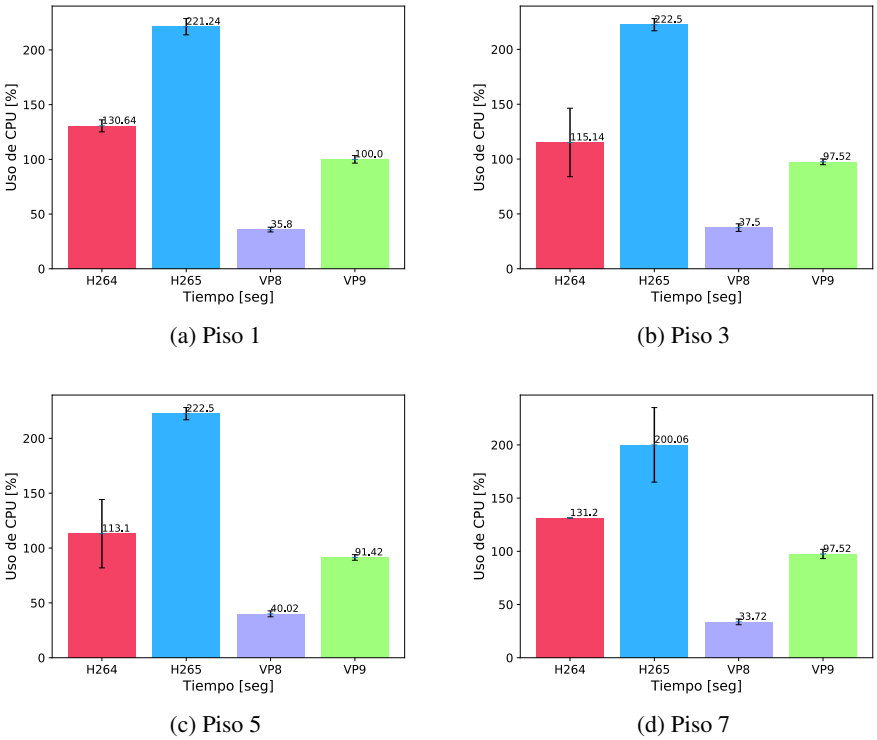
5.2.4. Evaluación del consumo de CPU

Para obtener el consumo de CPU en cada transmisión se usa el comando top de Linux, este comando permite monitorizar y administrar los procesos y tareas en cualquier sistema operativo UNIX, como por ejemplo GNU Linux. Al ejecutar top se observa una lista de los procesos y tareas que están ejecutándose en el equipo [113]. Los procesos se pueden clasificar por consumo de CPU y además filtrar por nombre del comando ejecutado. De esta manera se ejecuta el comando a la par que se realiza la transmisión y se almacenan los resultados en un archivo de texto. Además, mediante un *script* de Python se busca el valor de consumo de CPU más alto registrado durante la transmisión.

Cabe mencionar que el ordenador utilizado para la transmisión cuenta con un procesador Intel(R) Core(TM) i7-3612QM CPU @ 2.10GHz con 8 núcleos, además se fijó el uso de CPU de VP8 y VP9 en 2 y 4 núcleos respectivamente para poder realizar una tranmisión en tiempo real y ajustarse al tiempo de video. Los códecs H264 y H265 utilizan el preset *speed-preset* en su valor por defecto (*Medium*) que mantiene un equilibrio entre calidad y velocidad de codificación. En las Figuras 5.19, 5.20 y 5.21 se observan los resultados obtenidos para cada video en los diferentes pisos del edificio.

Los resultados obtenidos muestran la misma tendencia en todos los escenarios, siendo el códec VP8 el más eficiente, ya que cuenta con el menor consumo de CPU. Los códecs VP9 y H264 tienen un consumo parejo 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.



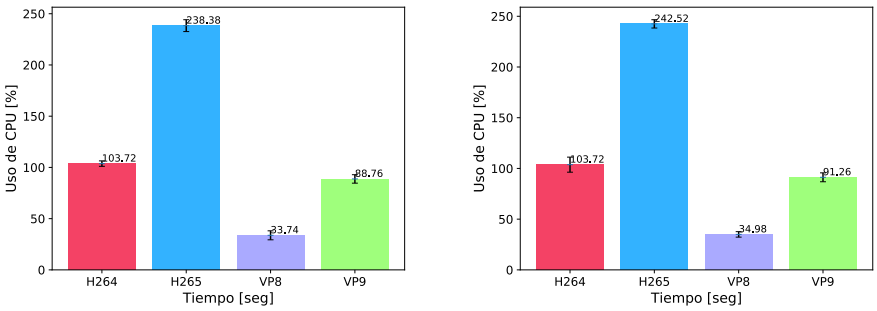
(a) Piso 1

(b) Piso 3

(c) Piso 5

(d) Piso 7

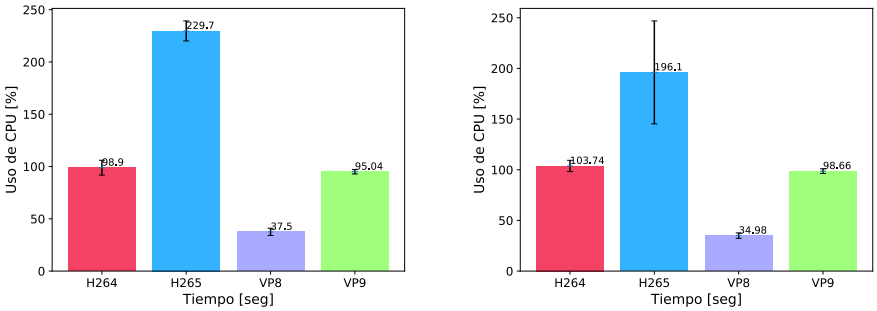
Figura 5.19: Consumo de CPU promedio para el video “Foreman” en cada piso.



(a) Piso 1 (b) Piso 3

(c) Piso 5 (d) Piso 7

Figura 5.20: Consumo de CPU promedio para el video “Mother-daughter” en cada piso.



(a) Piso 1 (b) Piso 3

(c) Piso 5 (d) Piso 7

Figura 5.21: Consumo de CPU promedio para el video “News” en cada piso.

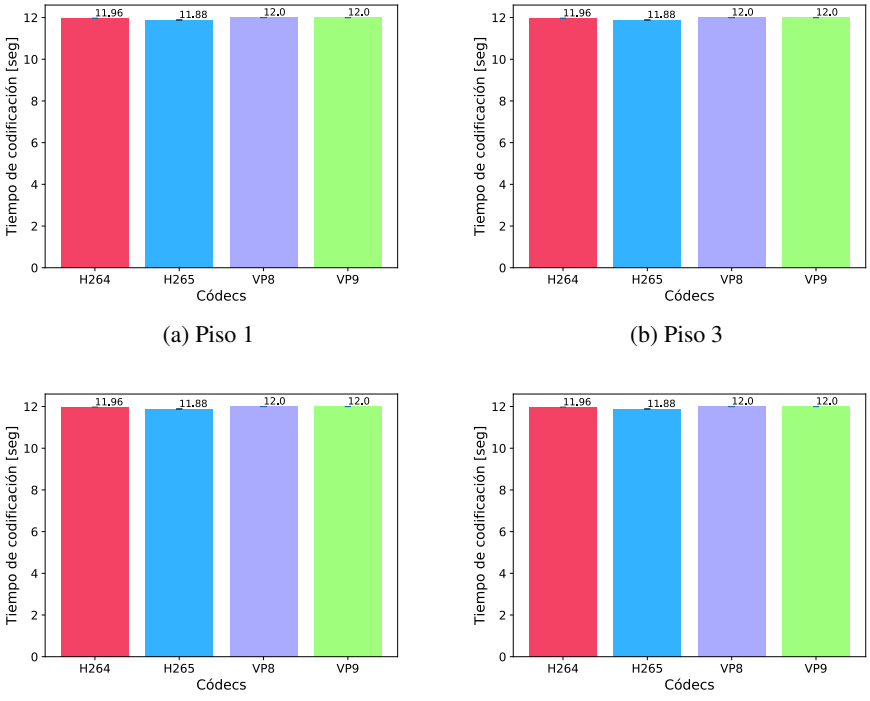
5.2.5. Evaluación del tiempo de codificación y transmisión

En cuanto a la métrica del tiempo de codificación y transmisión, se debe señalar que para los códecs H264 y H265 se estableció un *preset* de seis que corresponde a un valor de *medium* y para los códecs VP8 y VP9 se estableció un valor de CPU de dos y cuatro respectivamente, siguiendo la recomendación en [101], la misma que establece valores semejantes para disminuir el tiempo de codificación y permitir una transmisión en tiempo real.

Para encontrar el valor de la métrica a evaluar, primero se confirmó que el tiempo de transmisión y codificación sean iguales a la duración del video, después se guardó el tiempo que se presenta en la terminal y se escogió el último valor.

En la Figura 5.22 se observa que el valor para la métrica a evaluar es de 12 segundos, el cual es el tiempo que dura el video. Existen pequeñas diferencias para los códecs H265 y H264 debido a la forma de codificación que recortan unos milisegundos el tiempo de video. Se presenta una sola gráfica ya que el tiempo de duración de los otros videos es igual a la del video Foreman, manteniendo resultados iguales.

Es decir que para H264 y H265 no se puede escoger un mejor códec ya que el tiempo de codificación dependerá del valor del *preset* y para este caso es igual. Para los otros dos códecs se escoge a VP8 como el mejor ya que se define un menor número de CPUs, aunque el tiempo de codificación y transmisión es el mismo.



(a) Piso 1

(b) Piso 3

(c) Piso 5

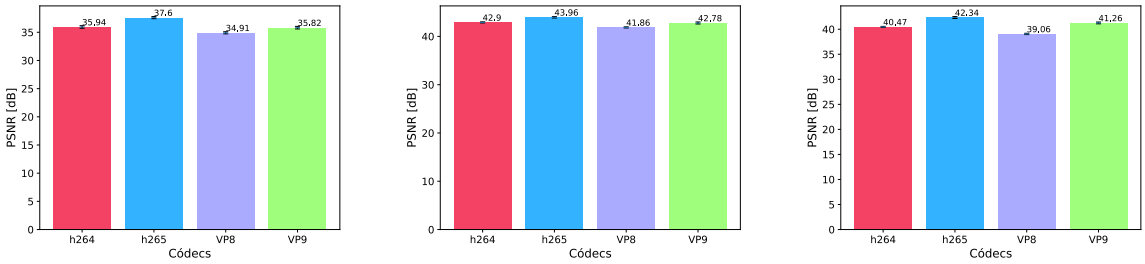
(d) Piso 7

Figura 5.22: Tiempo de codificación para el video “Foreman” en cada Piso.

5.2.6. Análisis del PSNR

Para la métrica del PSNR, previamente se codificaron los videos con los mismos parámetros utilizados en la trasmisión del video. Luego, se comparó el video codificado con el video en formato YUV. Para obtener el

PSNR se utilizó la herramienta FFmpeg, donde la primera entrada se considera el video codificado y la segunda entrada se usa como video de "referencia". En la Figura 5.23, se muestran los resultados de la métrica PSNR para cada video, dependiendo del tipo de códec. Cabe señalar que se muestra una sola figura ya que el procesamiento no incluye a un receptor. Respecto a los resultados se declara a H265 como el códec con mejor calidad debido a que tiene un mayor valor de PSNR, a este le sigue H264 y VP9 manteniendo resultados similares y por último estaél códec VP8.



(a) Video Foreman (b) Video Mother-daughter (c) Video News

Figura 5.23: Valor de PSNR de los video.

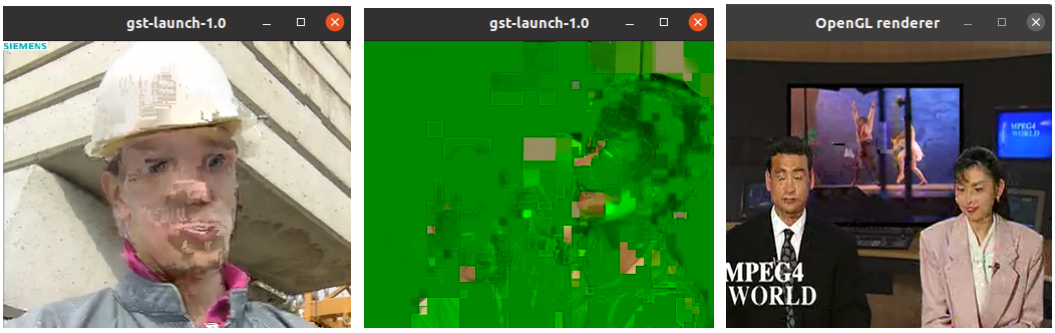
5.2.7. Resultados en el receptor

Para evaluar los resultados en el receptor, se consideró el piso 7 que es el más crítico, ya que es donde se genera más saltos "seis" y es el que determina el umbral para escoger las métricas de codificación. En este contexto, se realizaron de forma satisfactoria la reproducción de los videos en este piso. En las Figuras 5.24 se muestran capturas de la reproducción para cada video seleccionado.

(a) Video Foreman (b) Video Mother-daughter (c) Video News

Figura 5.24: Reproducción satisfactoria de video en el piso 7.

Cabe mencionar que en algunos videos se evidenció pérdida de paquetes, ya que en algunos instantes se perdía la imagen o el video no era fluido y habían saltos de escenas. En la Figura 5.25 se presentan algunas capturas de una reproducción con pérdida de paquetes. Además, la herramienta GStreamer notifica si existe gran pérdida de paquetes.



(a) Video Foreman (b) Video Mother-daughter (c) Video News

Figura 5.25: Reproducción de video con pérdida de paquetes en el piso 7.

5.3. Definición del códec más adecuado

Basándose en el análisis realizado en la sección anterior, en la Figura 5.26 se presenta un resumen de los mejores resultados para cada códec. De acuerdo a los resultados mostrados en la figura, se escoge a VP8 como el códec más adecuado para la aplicación requerida. Concretamente porque en términos de *throughput* es el códec que no tiene picos que superen el umbral establecido y esto permite que exista menos pérdida de paquetes. A su vez, es el códec que usa menos CPUs, permitiendo hacer uso de equipos con menos recursos de hardware. Finalmente, en términos de PSNR tiene resultados similares a los otros códecs.

Cabe recalcar que se definieron parámetros como el *preset* en H264 y H265, y el número de CPUs para VP8 y VP9 para que el tiempo de codificación sea igual al tiempo de video. De esta manera la métrica del tiempo de codificación y transmisión es similar para los 4 códecs. Además, las métricas del *delay* y *packet loss* es dependiente del estado de la red.

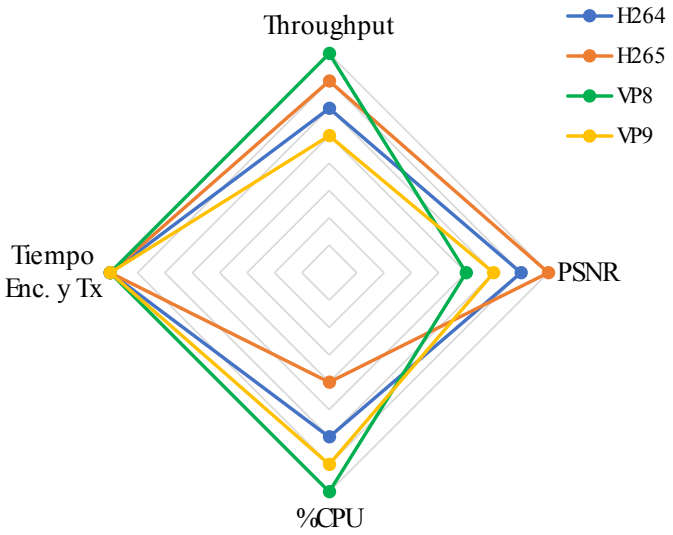


Figura 5.26: Rendimiento de cada códec por métrica.

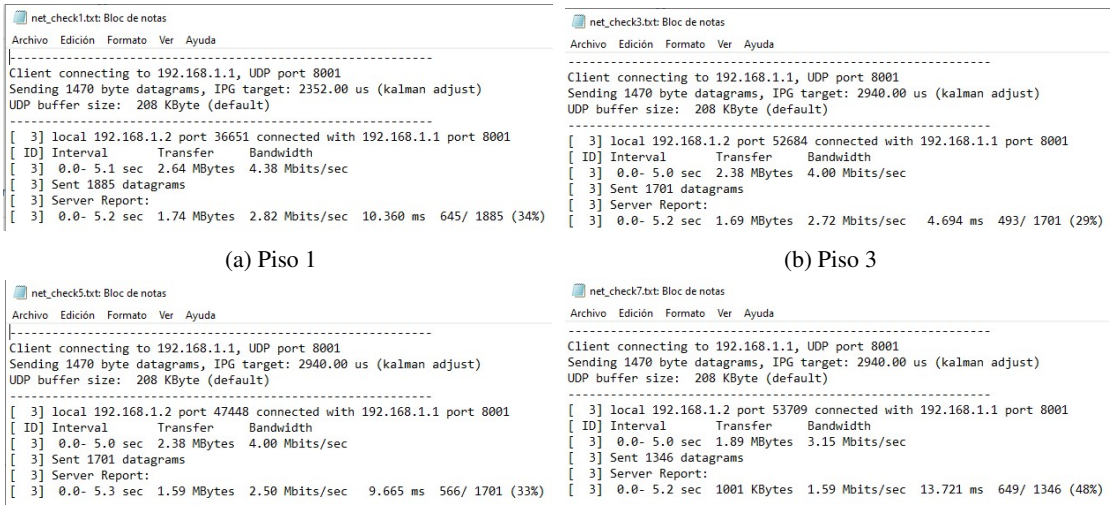
5.4. Evaluación de la herramienta de transmisión de audio y video en tiempo real

En esta sección se presentan los resultados obtenidos al transmitir audio y video entre los diferentes pisos del edificio (Centrosur). Para esto se usó las herramientas descritas en las Secciones 4.4 y 4.5. Como se explicó previamente en la Sección 4.4, el usuario que realiza la llamada se convierte en cliente iperf y el que recibe la llamada actúa como servidor, lo que permite medir el ancho de banda en el instante que se realiza la llamada. Con esto se establecieron umbrales por debajo de los 1.6 Mbps para realizar diferentes codificaciones con distintos *bitrates*, pero este umbral (1.6 Mbps) que se definió en la Sección 5.1, tenía en cuenta un solo flujo de video, por lo que para la herramienta de transmisión de audio y video en tiempo real se realizaron pruebas subjetivas en el piso 7 con diferentes *bitrates*. A partir de lo observado, se estableció que por encima del umbral de 1.6 Mbps el video se codificará con el *bitrate* seleccionado igual a 350 Kbps y por debajo de este umbral se codificará con 250 kbps.

En la Tabla 5.4, se presenta el ancho de banda medido y el *bitrate* seleccionado automáticamente para cada piso, basándose en los resultados del iperf, los cuales se muestran en la Figura 5.27. Donde se observa que el piso más crítico (piso 7) fue el único en codificar el video con 250 kbps. Cabe mencionar que las transmisiones se realizaron en las horas donde la red tiene el menor número de interferencias y el mayor ancho de banda, con lo cual si existe mayor sobrecarga en la red, la herramienta medirá el ancho de banda y de ser el caso seleccionará otro *bitrate* que sea el más adecuado para ese instante.

Tabla 5.4: Ancho de banda medido por piso, para la selección del bitrate adecuado.

Piso	**Saltos**	**Ancho de banda [Mbps]**	**Bitrate [Kbps]**
Piso 1	3	2.82	350
Piso 3	4	2.72	350
Piso 5	5	2.5	350
Piso 7	6	1.59	250



(a) Piso 1

(b) Piso 3

(c) Piso 5

(d) Piso 7

Figura 5.27: Resultados del iperf para cada piso.

Concretamente, con una llamada establecida, se ejecutó la herramienta para el cálculo de métricas que se explicó en la Sección 4.5, cabe detallar que se realizaron 10 capturas con una duración de dos minutos con el objetivo de efectuar un análisis estadístico de los resultados. Al terminar cada captura se puede revisar el cálculo de cada métrica realizando clic en el botón “Calcular métricas”, tal como se muestra en la Figura 5.28. Al finalizar las transmisiones en todos los pisos, se realizó un promedio de las tres métricas (*throughput*, *delay*, *packet loss*) tanto para audio como para video con el botón “Promediar”, en las Figuras 5.29, 5.30 y 5.31 se presentan los resultados de las tres métricas con respecto a los dos usuarios.

Cabe mencionar que la herramienta se desarrolló con un esquema general donde cualquier usuario puede medir las métricas de *throughput*, *delay* y *packet loss*, seleccionando si es el emisor o receptor de la llamada. Esto permite realizar el cálculo de las métricas con respecto a los flujos recibidos.

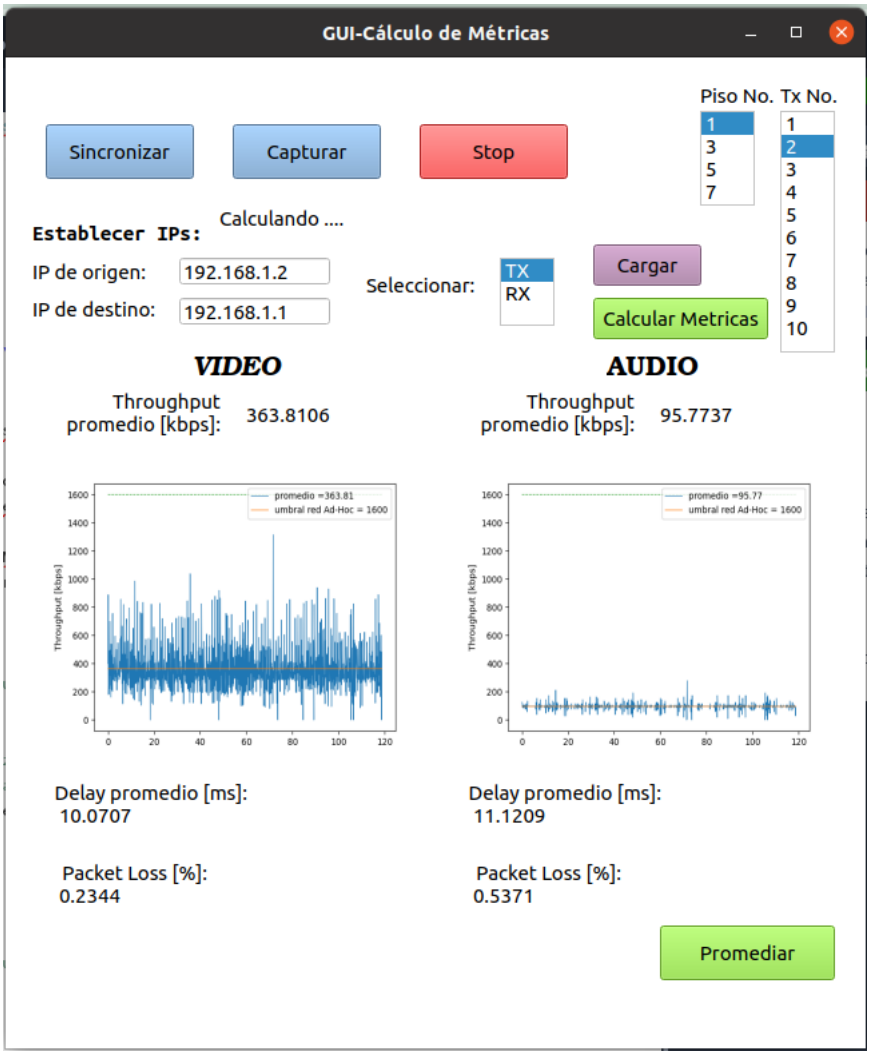
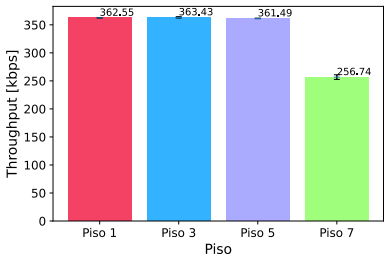
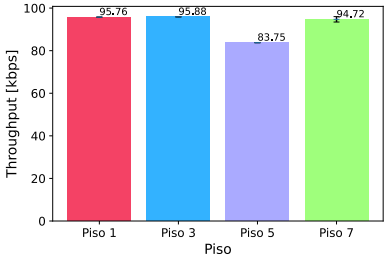


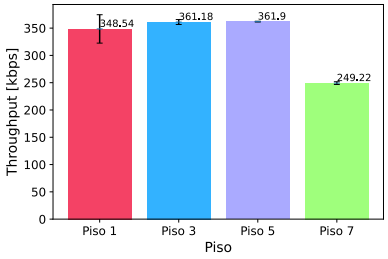
Figura 5.28: Herramienta para el cálculo de métricas.



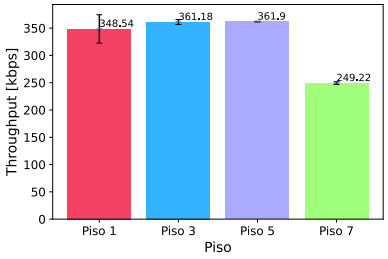
(a) Throughput de Video - Usuario A



(b) Throughput de Audio - Usuario A

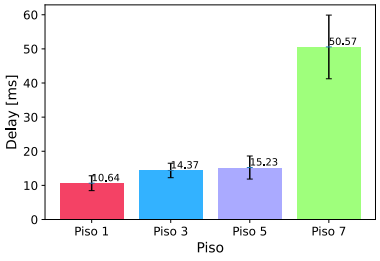


(c) Throughput de Video - Usuario B

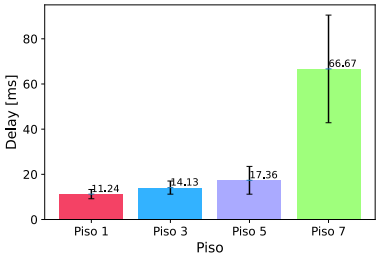


(d) Throughput de Audio - Usuario B

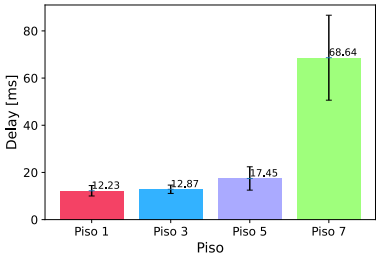
Figura 5.29: Resultados para la métrica throughput de audio y video del usuario A y B.



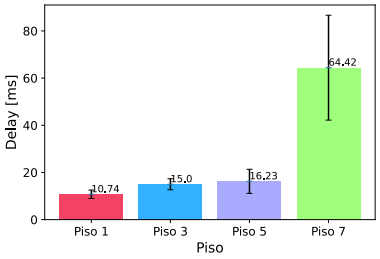
(a) Delay de Video - Usuario A



(b) Delay de Audio - Usuario A

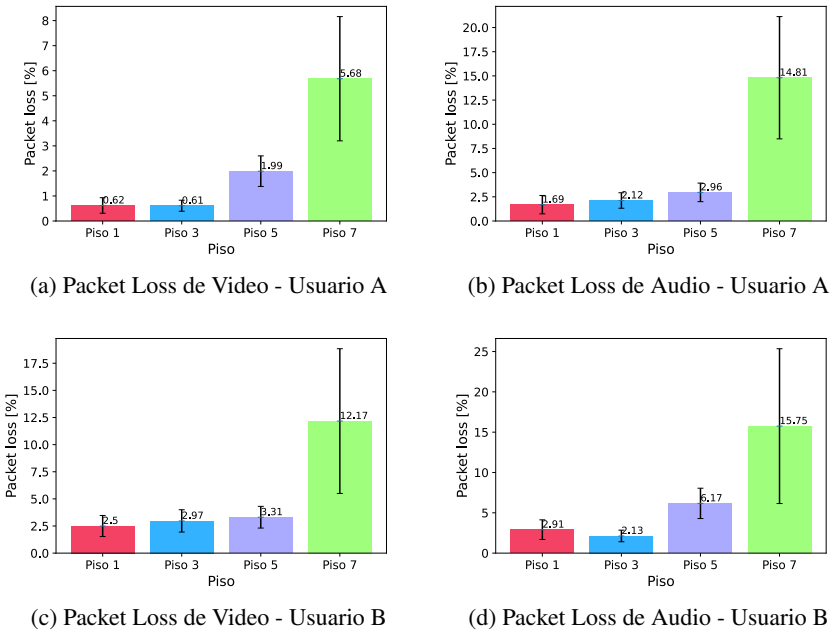


(c) Delay de Video - Usuario B



(d) Delay de Audio - Usuario B

Figura 5.30: Resultados para la métrica delay de audio y video del usuario A y B.



(a) Packet Loss de Video - Usuario A

(b) Packet Loss de Audio - Usuario A

(c) Packet Loss de Video - Usuario B

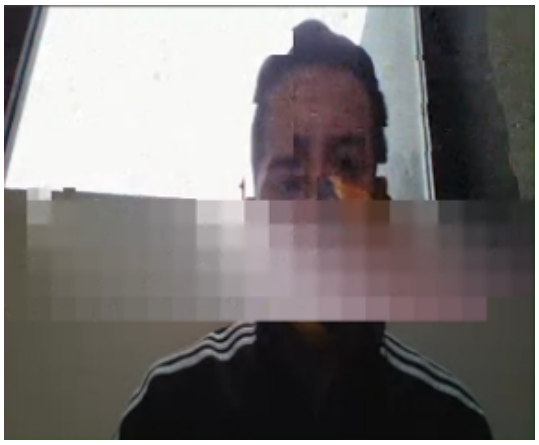
(d) Packet Loss de Audio - Usuario B

Figura 5.31: Resultados para la métrica packet loss de audio y video del usuario A y B.

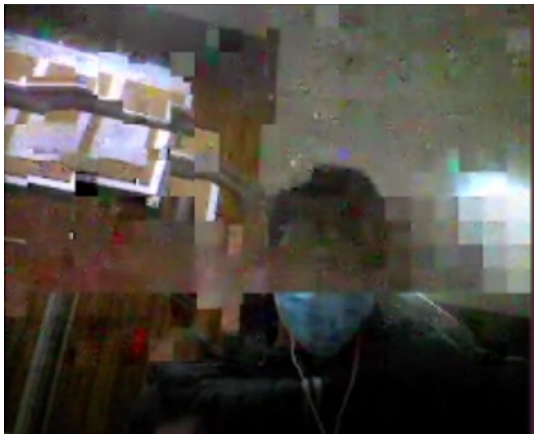
De esta manera, con los resultados mostrados en la Figura 5.29 correspondientes al *throughput*, se comprobó que al realizar la medición del ancho de banda en el piso 7, el video se codificó con un *bitrate* de 250 kbps, el cual coincide con el *throughput* promedio calculado. Para los piso más bajos, se puede notar que el ancho de banda superó el umbral, escogiendo un *bitrate* de 350 kbps y con lo que el *throughput* está alrededor de ese valor. Además, en la Figura 5.30 correspondiente al *delay*, se comprobó nuevamente que al incrementar el número de saltos el *delay* aumenta tanto para el audio como para el video. Finalmente, para la Figura 5.31 correspondiente al *packet loss* se comprueba 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. Estos resultados se ven reflejados en la videollamada, realizando un análisis subjetivo donde se nota una fluidez de la llamada y en otros casos algunos retrasos o pérdida de paquetes tal como se observa en la Figura 5.32.

(a) Fluidez de audio y video - Usuario A

(b) Fluidez de audio y video - Usuario B



(c) Pérdida de paquetes de audio y video - Usuario A



(d) Pérdida de paquetes de audio y video - Usuario B

Figura 5.32: Análisis subjetivo de la herramienta de transmisión de audio y video en tiempo real.

Capítulo 6

Conclusiones

La caracterización de la red ad hoc demostró que las condiciones de la misma son cambiantes y dependientes de varios factores externos. Concretamente, el número de personas dentro del edificio o la hora del día, ya que el número de interferencias incrementa y 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.

Por otro lado, el desarrollo de la herramienta para caracterizar la red, permitió visualizar los resultados conforme se realizaron los experimentos sin necesidad de realizar un procesamiento posterior de la información recopilada, posibilitó detectar errores o resultados inconsistentes que afecten al análisis estadístico efectuado, y finalmente, 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. En tal sentido, los valores promedio de las métricas obtenidos en la red estudiada facilitan la transmisión de contenido multimedia en tiempo real, y por lo tanto, se encontró el mejor códec para codificar el video y poder transmitirlo a través de la red.

Por consiguiente, en los experimentos realizados para obtener el códec óptimo, se comprobó que VP8 permitía la transmisión del mismo video ahorrando ancho de banda y usando como máximo dos CPUs. Esta fue la razón principal para descartar a los demás códecs estudiados. De esta forma, en caso de emergencia, cualquier persona puede hacer uso de la herramienta de videollamada desarrollada incluso con dispositivos limitados en hardware.

Por otro lado, se desarrolló una herramienta que cualquier usuario puede hacer uso para calcular el valor de las métricas en cuanto a *delay*, *throughput* y *packet loss* de los flujos multimedia entrantes con ayuda del protocolo RTP y RTCP y así establecer una comparación entre las medidas subjetivas y objetivas de cada videollamada.

En este sentido, la aplicación de videoconferencia desarrollada permite una comunicación bidireccional de audio y video entre dos usuarios dentro de la red ad hoc desplegada con la capacidad de adaptarse a las condiciones de la red para mantener un delay por debajo de los 150ms, valor recomendado para una videoconferencia fluida. Por otro lado, se creó un servidor que recopila la información de cada usuario, este permite conocer los

contactos que ingresaron a la herramienta y, en consecuencia, con los que se puede establecer una videollamada o comunicación.

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*, estableciendo 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 en el edificio de la Centrosur aprovechando la red ad hoc desplegada.

Bibliografía

[1] J. Frnda, M. Voznak, y L. Sevcik, "Impact of packet loss and delay variation on the quality of real-time video streaming," *Telecommunication Systems*, vol. 62, num. 2, pp. 265–275, 2016.

[2] D. Hutchison y J. P. Sterbenz, "Architecture and design for resilient networked systems," *Computer Communications*, vol. 131, pp. 13–21, 2018.

[3] A. Mauthe, D. Hutchison, E. K. Cetinkaya, I. Ganchev, J. Rak, J. P. Sterbenz, M. Gunkelk, P. Smith, y T. Gomes, "Disaster-resilient communication networks: Principles and best practices," in *2016 8th International Workshop on Resilient Networks Design and Modeling (RNDM)*. IEEE, 2016, pp. 1–10.

[4] T. Gomes, J. Tapolcai, C. Esposito, D. Hutchison, F. Kuipers, J. Rak, A. De Sousa, A. Iossifides, R. Travanca, J. André *y otros*, "A survey of strategies for communication networks to protect against large-scale natural disasters," in *2016 8th international workshop on resilient networks design and modeling (RNDM)*. IEEE, 2016, pp. 11–22.

[5] M. R. A. Kale, S. Gupta, y B. Prmit, "An overview of manet ad hoc network," *International journal of computer science and applications*, vol. 6, num. 2, pp. 257–264, 2013.

[6] S. Subbarayappa *y otros*, "Heterogeneous transcoding for next generation multimedia video codecs for efficient communication," Ph.D. dissertation, 2020.

[7] Y. Gosain y A. Gupta, "Xilinx advanced multimedia solutions with video codec/graphics engines," *Zynq UltraScale+ MPSoC. Xilinx, October*, vol. 23, 2017.

[8] Z. Pan, H. Qin, X. Yi, Y. Zheng, y A. Khan, "Low complexity versatile video coding for traffic surveillance system," *International Journal of Sensor Networks*, vol. 30, num. 2, pp. 116–125, 2019.

[9] M. Dipobagio, "An overview on ad hoc networks," *Institute of Computer Science (ICS), Freie Universität Berlin*, 2009.

[10] G. Khanna y S. K. Chaturvedi, "A comprehensive survey on multi-hop wireless networks: milestones, changing trends and concomitant challenges," *Wireless Personal Communications*, vol. 101, num. 2, pp. 677–722, 2018.

[11] S. Misra y S. Goswami, *Network routing: fundamentals, applications, and emerging technologies*. John Wiley & Sons, 2017.

[12] M. N. Sadiku *y otros*, *Elements of electromagnetics*. Oxford university press New York, 2001, vol. 428.

[13] C. E. Perkins, "Ad hoc networking: an introduction," *Ad hoc networking*, vol. 40, pp. 20–22, 2001.

[14] D. Papakostas, S. Eshghi, D. Katsaros, y L. Tassiulas, "Energy-aware backbone formation in military multilayer ad hoc networks," *Ad Hoc Networks*, vol. 81, pp. 17–44, 2018.

[15] S. Santhi, E. Udayakumar, y T. Gowthaman, "Sos emergency ad hoc wireless network," in *Computational Intelligence and Sustainable Systems*. Springer, 2019, pp. 227–234.

[16] D. Kandris, C. Nakas, D. Vomvas, y G. Koulouras, "Applications of wireless sensor networks: an up-to-date survey," *Applied System Innovation*, vol. 3, num. 1, p. 14, 2020.

[17] M. M. Hamdi, L. Audah, S. A. Rashid, A. H. Mohammed, S. Alani, y A. S. Mustafa, "A review of applications, characteristics and challenges in vehicular ad hoc networks (vanets)," in *2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*. IEEE, 2020, pp. 1–7.

[18] M. Lee y T. Atkison, "Vanet applications: Past, present, and future," *Vehicular Communications*, vol. 28, p. 100310, 2021.

[19] I. F. Akyildiz y X. Wang, "A survey on wireless mesh networks," *IEEE Communications magazine*, vol. 43, num. 9, pp. S23–S30, 2005.

[20] S. Y. Shahdad, A. Sabahath, y R. Parveez, "Architecture, issues and challenges of wireless mesh network," in *2016 International Conference on Communication and Signal Processing (ICCSP)*. IEEE, 2016, pp. 0557–0560.

[21] M. Eslami, O. Karimi, y T. Khodadadi, "A survey on wireless mesh networks: Architecture, specifications and challenges," in *2014 IEEE 5th Control and System Graduate Research Colloquium*. IEEE, 2014, pp. 219–222.

[22] A. y. M. R. Jadhav, SS y Kulkarni, "Red móvil ad-hoc (manet) para la gestión de desastres," in *2014 Undécima Conferencia Internacional sobre Redes de Comunicaciones Ópticas e Inalámbricas (WOCN)*.

[23] N. Raza, M. U. Aftab, M. Q. Akbar, O. Ashraf, y M. Irfan, "Mobile ad-hoc networks applications and its challenges," *Communications and Network*, vol. 8, num. 3, pp. 131–136, 2016.

[24] G. Obidike, C. Nwabueze, y V. Onuzulike, "Concept and characteristics of mobile ad-hoc network," *International Journal of Innovative Engineering, Technology and Science*, vol. 2, 2018.

[25] S. Srivastava, M. Singh, y S. Gupta, "Wireless sensor network: a survey," in *2018 International Conference on Automation and Computational Engineering (ICACE)*. IEEE, 2018, pp. 159–163.

[26] D. Brunelli, I. Minakov, R. Passerone, y M. Rossi, "Povomon: An ad-hoc wireless sensor network for indoor environmental monitoring," in *2014 IEEE Workshop on Environmental, Energy, and Structural Monitoring Systems Proceedings*. IEEE, 2014, pp. 1–6.

[27] A. Mehmood, J. L. Mauri, M. Noman, y H. Song, "Improvement of the wireless sensor network lifetime using leach with vice-cluster head." *Ad Hoc Sens. Wirel. Networks*, vol. 28, num. 1-2, pp. 1–17, 2015.

[28] R. Hooks, "The 9 styles of video (and when to use them)," Dec 2019. [En línea]. Disponible: https://99designs.com/blog/video-animation/styles-of-video/

[29] K. L. Dias, M. A. Pongelupe, W. M. Caminhas, y L. de Errico, "An innovative approach for real-time network traffic classification," *Computer Networks*, vol. 158, pp. 143–157, 2019.

[30] J. Kua, G. Armitage, y P. Branch, "A survey of rate adaptation techniques for dynamic adaptive streaming over http," *IEEE Communications Surveys & Tutorials*, vol. 19, num. 3, pp. 1842–1866, 2017.

[31] T. Stockhammer y M. G. Luby, "Dash in mobile networks and services," in *2012 Visual Communications and Image Processing*, 2012, pp. 1–6.

[32] "Http live streaming." [En línea]. Disponible: https://developer.apple.com/documentation/http_live_streaming

[33] A. N. Babatunde, R. G. Jimoh, O. C. Abikoye, y B. Isiaka, "Survey of video encryption algorithms," 2017.

[34] Y. Wu, Y. Zhao, y J. Li, "Brief analysis of the h. 264 coding standard," in *Third International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP 2007)*, vol. 2. IEEE, 2007, pp. 154–157.

[35] F. Mentzer, L. V. Gool, y M. Tschannen, "Learning better lossless compression using lossy compression," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6638–6647.

[36] I. E. Richardson, *The H. 264 advanced video compression standard*. John Wiley & Sons, 2011.

[37] Techopedia, "What is common intermediate format (cif)? - definition from techopedia," Oct 2013. [En línea]. Disponible: https://www.techopedia.com/definition/4737/common-intermediate-format-cif

[38] S. Akramullah, *Digital video concepts, methods, and metrics: quality, compression, performance, and power trade-off analysis*. Springer Nature, 2014.

[39] P. Yakaiah y C. Sheka, "Overview of h. 265," *Technology*, vol. 8, num. 8, pp. 48–54, 2017.

[40] S. P. Singh y G. Bhatnagar, "Chapter 1 - perceptual hashing-based novel security framework for medical images," in *Intelligent Data Security Solutions for e-Health Applications*, ser. Intelligent Data-Centric Systems, A. K. Singh y M. Elhoseny, Eds. Academic Press, 2020, pp. 1–20. [En línea]. Disponible: https://www.sciencedirect.com/science/article/pii/B9780128195116000017

[41] I. Richardson, "H.264/avc 4x4 transform and quantization." [En línea]. Disponible: https://www.vcodex.com/h264avc-4x4-transform-and-quantization/

[42] K. Sayood, "Chapter 19 - video compression," in *Introduction to Data Compression (Fifth Edition)*, fifth edition ed., ser. The Morgan Kaufmann Series in Multimedia Information and Systems, K. Sayood, Ed. Morgan Kaufmann, 2018, pp. 645–700. [En línea]. Disponible: https://www.sciencedirect.com/science/article/pii/B9780128094747000197

[43] R. Nabeel y M. H. Al-Jammas, "The gop inter prediction of h. 264 av\c," *Journal of King Saud University-Computer and Information Sciences*, 2019.

[44] J. Bankoski, P. Wilkins, y Y. Xu, "Technical overview of vp8, an open source video codec for the web," in *2011 IEEE International Conference on Multimedia and Expo*. IEEE, 2011, pp. 1–6.

[45] Y. O. Sharrab y N. J. Sarhan, "Detailed comparative analysis of vp8 and h. 264," in *2012 IEEE International Symposium on Multimedia*. IEEE, 2012, pp. 133–140.

[46] C. Feller, J. Wuenschmann, T. Roll, y A. Rothermel, "The vp8 video codec-overview and comparison to h. 264/avc," in *2011 IEEE International Conference on Consumer Electronics-Berlin (ICCE-Berlin)*. IEEE, 2011, pp. 57–61.

[47] S. Cassidy, "An analysis of vp8, a new video codec for the web," 2011.

[48] M. A. Layek, N. Q. Thai, M. A. Hossain, N. T. Thu, A. Talukder, T. Chung, E.-N. Huh *y otros*, "Performance analysis of h. 264, h. 265, vp9 and av1 video encoders," in *2017 19th Asia-Pacific Network Operations and Management Symposium (APNOMS)*. IEEE, 2017, pp. 322–325.

[49] T. Laude, Y. G. Adhisantoso, J. Voges, M. Munderloh, y J. Ostermann, "A comprehensive video codec comparison," *APSIPA Transactions on Signal and Information Processing*, vol. 8, 2019.

[50] G. J. Sullivan, J.-R. Ohm, W.-J. Han, y T. Wiegand, "Overview of the high efficiency video coding (hevc) standard," *IEEE Transactions on circuits and systems for video technology*, vol. 22, num. 12, pp. 1649–1668, 2012.

[51] D. Mukherjee, J. Han, J. Bankoski, R. Bultje, A. Grange, J. Koleszar, P. Wilkins, y Y. Xu, "A technical overview of vp9—the latest open-source video codec," *SMPTE Motion Imaging Journal*, vol. 124, num. 1, pp. 44–54, 2015.

[52] P. Akyazi y T. Ebrahimi, "Comparison of compression efficiency between hevc/h. 265, vp9 and av1 based on subjective quality assessments," in *2018 Tenth International Conference on Quality of Multimedia Experience (QoMEX)*. IEEE, 2018, pp. 1–6.

[53] Y. Yalman y İ. ERTÜRK, "A new color image quality measure based on yuv transformation and psnr for human vision system," *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 21, num. 2, pp. 603–612, 2013.

[54] P. Zhao, Y. Liu, J. Liu, A. Argyriou, y S. Ci, "Ssim-based error-resilient cross-layer optimization for wireless video streaming," *Signal Processing: Image Communication*, vol. 40, pp. 36–51, 2016.

[55] J. Xu, B. Zhou, C. Zhang, N. Ke, W. Jin, y S. Hao, "The impact of bitrate and gop pattern on the video quality of h. 265/hevc compression standard," in *2018 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC)*. IEEE, 2018, pp. 1–5.

[56] K. Kawashima, J. Okamoto, y T. Hayashi, "Verification on stability and reproducibility of dscqs method for assessing 4k ultra-hd video quality," in *2014 Sixth International Workshop on Quality of Multimedia Experience (QoMEX)*. IEEE, 2014, pp. 214–219.

[57] J. Filanová, "The process model of subjective quality assessment of videoconferencing in enterprise," in *SHS Web of Conferences*, vol. 83. EDP Sciences, 2020, p. 01015.

[58] A. Tirumala, L. Cottrell, y T. Dunigan, "Measuring end-to-end bandwidth with iperf using web100," in *In Web100, Proc. of Passive and Active Measurement Workshop*. Citeseer, 2003.

[59] G. Kaur y N. Bhatia, "Wireshark–packet capture tool."

[60] P. Goyal y A. Goyal, "Comparative study of two most popular packet sniffing tools-tcpdump and wireshark," in *2017 9th International Conference on Computational Intelligence and Communication Networks (CICN)*. IEEE, 2017, pp. 77–81.

[61] R. Joshi y E. S. Pilli, "Network forensic tools," in *Fundamentals of Network Forensics*. Springer, 2016, pp. 71–93.

[62] "iwlist(8) - linux man page." [En línea]. Disponible: https://linux.die.net/man/8/iwlist

[63] J. Kanclirz, *Netcat power tools*. Elsevier, 2008.

[64] H. Schulzrinne, S. Casner, R. Frederick, y V. Jacobson, "Rfc3550: Rtp: A transport protocol for real-time applications," 2003.

[65] GStreamer, "What is gstreamer?" [En línea]. Disponible: https://gstreamer.freedesktop.org/documentation/application-development/introduction/gstreamer.html?gi-language=c

[66] G.-D. V1.28, "rtpsession." [En línea]. Disponible: https://gstreamer.freedesktop.org/data/doc/gstreamer/head/gst-plugins-good/html/gst-plugins-good-plugins-rtpsession.html

[67] GStreamer, "autovideosink." [En línea]. Disponible: https://gstreamer.freedesktop.org/documentation/autodetect/autovideosink.html?gi-language=c

[68] B. B. Mishra, S. Dehuri, B. K. Panigrahi, A. K. Nayak, B. S. P. Mishra, y H. Das, *Computational intelligence in sensor networks*. Springer, 2019.

[69] Z. Zou y Y. Qian, "Wireless sensor network routing method based on improved ant colony algorithm," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, num. 3, pp. 991–998, 2019.

[70] G. Gautam y B. Sen, "Design and simulation of wireless sensor network in ns2," *International Journal of Computer Applications*, vol. 113, num. 16, 2015.

[71] M. Ndiaye, G. P. Hancke, y A. M. Abu-Mahfouz, "Software defined networking for improved wireless sensor network management: A survey," *Sensors*, vol. 17, num. 5, 2017. [En línea]. Disponible: https://www.mdpi.com/1424-8220/17/5/1031

[72] A. S. Mohammed, S. Basha, P. Asha, K. Venkatachalam *y otros*, "Fco—fuzzy constraints applied cluster optimization technique for wireless adhoc networks," *Computer Communications*, vol. 154, pp. 501–508, 2020.

[73] M. Wang, H. Jin, C. Zhao, y D. Liang, "Delay optimization of computation offloading in multi-hop ad hoc networks," in *2017 IEEE International Conference on Communications Workshops (ICC Workshops)*. IEEE, 2017, pp. 314–319.

[74] C. Funai, C. Tapparello, y W. Heinzelman, "Enabling multi-hop ad hoc networks through wifi direct multi-group networking," in *2017 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, 2017, pp. 491–497.

[75] K. Liu y Y. Cheng, "Enabling data sharing and multimedia applications over a multi-hop ad hoc network formed by un-rooted smartphones," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2018, pp. 1–2.

[76] R. Chiluveru, N. Gupta, y A. S. Teles, "Distribution of safety messages using mobility-aware multi-hop clustering in vehicular ad hoc network," *Future Internet*, vol. 13, num. 7, p. 169, 2021.

[77] L. Khamer, N. Labraoui, A. M. Gueroui, S. Zaidi, y A. A. A. Ari, "Road network layout based multi-hop broadcast protocols for urban vehicular ad-hoc networks," *Wireless Networks*, vol. 27, num. 2, pp. 1369–1388, 2021.

[78] S. González, W. Castellanos, P. Guzmán, P. Arce, y J. C. Guerri, "Simulation and experimental testbed for adaptive video streaming in ad hoc networks," *Ad Hoc Networks*, vol. 52, pp. 89–105, 2016.

[79] E. Flores, M. Coyotecatl, R. Torrealba, y R. Ambrosio, "Análisis del parámetro throughput en una red ad hoc y manet en el estándar 802.11 ac," *Revista de Aplicación Científica y Técnica*, vol. 3, num. 7, pp. 1–9, 2017.

[80] N. A. Tolentino Medrano *y otros*, "Diseño e implementación de un nodo vanet considerando un sistema de control disparado por eventos," Master's thesis, 2021.

[81] Z. Ahmed, S. Naz, y J. Ahmed, "Minimizing transmission delays in vehicular ad hoc networks by optimized placement of road-side unit," *Wireless Networks*, vol. 26, num. 4, pp. 2905–2914, 2020.

[82] D. Reina, M. Askalani, S. Toral, F. Barrero, E. Asimakopoulou, y N. Bessis, "A survey on multihop ad hoc networks for disaster response scenarios," *International Journal of Distributed Sensor Networks*, vol. 11, num. 10, p. 647037, 2015.

[83] H. Salam, S. Memon, L. Das, Z. HUSSAIN *y otros*, "Drone based resilient network architecture for survivals in earthquake zones in pakistan," *Sindh University Research Journal-SURJ (Science Series)*, vol. 50, num. 01, pp. 175–182, 2018.

[84] D. Radu, A. Cretu, B. Parrein, J. Yi, C. Avram, y A. Aştilean, "Flying ad hoc network for emergency applications connected to a fog system," in *International conference on emerging internetworking, data & web technologies*. Springer, 2018, pp. 675–686.

[85] L. Ye, Y. Zhang, Y. Li, y S. Han, "A dynamic cluster head selecting algorithm for uav ad hoc networks," in *2020 International Wireless Communications and Mobile Computing (IWCMC)*. IEEE, 2020, pp. 225–228.

[86] A. I. Husain y H. Bhardwaj, "A cluster based wsn for earthquake and tsunami: Detection and mitigation," in *2019 6th International Conference on Computing for Sustainable Global Development (INDIACom)*, 2019, pp. 847–849.

[87] W. Castellanos, P. Guzmán, P. Arce, y J. C. Guerri, "Mechanisms for improving the scalable video streaming in mobile ad hoc networks," in *Proceedings of the 12th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, & Ubiquitous Networks*, 2015, pp. 33–40.

[88] S. Felici-Castell, M. García-Pineda, J. Segura-Garcia, R. Fayos-Jordan, y J. Lopez-Ballester, "Adaptive live video streaming on low-cost wireless multihop networks for road traffic surveillance in smart cities," *Future Generation Computer Systems*, vol. 115, pp. 741–755, 2021.

[89] M. Usman, M. A. Jan, X. He, y M. Alam, "Performance evaluation of high definition video streaming over mobile ad hoc networks," *Signal Processing*, vol. 148, pp. 303–313, 2018.

[90] D. Ghrab, I. Jemili, A. Belghith, y M. Mosbah, "Correlation-free multipath routing for multimedia traffic in wireless sensor networks," in *International Conference on Ad-Hoc Networks and Wireless*. Springer, 2017, pp. 276–289.

[91] N. H. Anh y P. T. Giang, "Qos evaluation of multimedia data over multi-hop ad hoc networks for testbed system," in *The International Conference on Intelligent Systems & Networks*. Springer, 2021, pp. 150–156.

[92] J. Bienik, M. Uhrina, M. Kuba, y M. Vaculik, "Performance of h. 264, h. 265, vp8 and vp9 compression standards for high resolutions," in *2016 19th International Conference on Network-Based Information Systems (NBiS)*. IEEE, 2016, pp. 246–252.

[93] R. Pereira y E. Pereira, "Video streaming: H. 264 and the internet of things," in *2015 IEEE 29th International Conference on Advanced Information Networking and Applications Workshops*. IEEE, 2015, pp. 711–714.

[94] W. Castellanos, J. C. Guerri, y P. Arce, "Performance evaluation of scalable video streaming in mobile ad hoc networks," *IEEE Latin America Transactions*, vol. 14, num. 1, pp. 122–129, 2016.

[95] C. I. Paredes, A. M. Mezher, y M. A. Igartua, "Performance comparison of h. 265/hevc, h. 264/avc and vp9 encoders in video dissemination over vanets," in *International Conference on Smart Objects and Technologies for Social Good*. Springer, 2016, pp. 51–60.

[96] T. Zhang y S. Mao, "An overview of emerging video coding standards," *GetMobile: Mobile Computing and Communications*, vol. 22, num. 4, pp. 13–20, 2019.

[97] D. Grois, D. Marpe, A. Mulayoff, B. Itzhaky, y O. Hadar, "Performance comparison of h. 265/mpeg-hevc, vp9, and h. 264/mpeg-avc encoders," in *2013 Picture Coding Symposium (PCS)*. IEEE, 2013, pp. 394–397.

[98] J. Sharma, T. Choudhury, S. C. Satapathy, y A. S. Sabitha, "Study on h. 265/hevc against vp9 and h. 264: on space and time complexity for codecs," in *2018 International Conference on Communication, Computing and Internet of Things (IC3IoT)*. IEEE, 2018, pp. 106–110.

[99] N. Barman y M. G. Martini, "H.264/mpeg-avc, h.265/mpeg-hevc and vp9 codec comparison for live gaming video streaming," in *2017 Ninth International Conference on Quality of Multimedia Experience (QoMEX)*, 2017, pp. 1–6.

[100] R. S. Bultje, "Vp9 encoding/decoding performance vs. hevc/h.264," Sep 2015. [En línea]. Disponible: https://blogs.gnome.org/rbultje/2015/09/28/vp9-encodingdecoding-performance-vs-hevch-264/

[101] L. Mengzhe, J. Xiuhua, y L. Xiaohua, "Analysis of h. 265/hevc, h. 264 and vp9 coding efficiency based on video content complexity," in *2015 IEEE International Conference on Computer and Communications (ICCC)*. IEEE, 2015, pp. 420–424.

[102] I. Palacios, J. Placencia, M. Muñoz, V. Samaniego, S. González, y J. Jiménez, "Mqtt based event detection system for structural health monitoring of buildings," in *XV Multidisciplinary International Congress on Science and Technology*. Springer, 2021, pp. 56–70.

[103] "Yuv video sequences." [En línea]. Disponible: http://trace.eas.asu.edu/yuv/

[104] J. Newmarch, "Ffmpeg/libav," in *Linux Sound Programming*. Springer, 2017, pp. 227–234.

[105] C. Quinde y P. Astudillo, "Herramienta para la transmisión de video utilizando diferentes códecs de video," Feb 2022. [En línea]. Disponible: https://github.com/christianquinde/GUI

[106] C. Quinde2 y P. Astudillo2, "Herramienta para la recepción de video," Feb 2022. [En línea]. Disponible: https://github.com/christianquinde/GUI-rx

[107] C. Quinde y P. Astudillo, "Herramienta para transmitir y reproducir audio y video utilizando gstreamer." Feb 2022. [En línea]. Disponible: https://github.com/christianquinde/App-videoconferencia

[108] ——, "Herramienta medición," Feb 2022. [En línea]. Disponible: https://github.com/christianquinde/herramienta-medicion

[109] ——, "Servidor de contactos para la herramienta de videoconferencia," Feb 2022. [En línea]. Disponible: https://github.com/christianquinde/server

[110] T. W. Project, "Vp8 encode parameter guide." [En línea]. Disponible: https://www.webmproject.org/docs/encoder-parameters/

[111] Tsbmail, Nov 2003. [En línea]. Disponible: https://www.itu.int/rec/T-REC-G.114-200305-I/es

[112] Vyopta, "What's an acceptable amount of packet loss in 2019?" Feb 2019. [En línea]. Disponible: https://www.vyopta.com/blog/video-conferencing/understanding-packet-loss/#:~:text=Any%20significant%20packet%20loss%20on,could%20be%20considered%20%E2%80%9Cacceptable%E2%80%9D.

[113] J. Carles, "Sacar partido al monitor de recursos top en linux," Nov 2019. [En línea]. Disponible: https://geekland.eu/usar-entender-monitor-de-recursos-top/#:~:text=TOP%20es%20un%20programa%20inform%C3%A1tico,como%20por%20ejemplo%20GNU%20Linux.