People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
University of 8 Mai 1945Guelma

Institute of Telecommunications.
Department of Telecommunications Systems.
Telecommunications Laboratory (LT).

Thesis

Submitted in Candidacy for the Degree of *Doctorate in Third Cycle*

Field: Science and Technology. Stream: Telecommunications.
Specialty: Telecommunications.

Presented by:
ELHACHI Hana

Title

Study and Improvement of the Quality of Service of a Video Streaming by the HEVC Codec in Ad-hoc Networks.

Defended on : 08/01/2025 Before the jury composed of:

Full name	**Rank**	**University**	
Mr. BOUROUBA Hocine	**Professeur**	Univ. of 8 Mai 1945 Guelma	President
Mr. REDADAA Salah	**Professeur**	Univ. of 8 Mai 1945 Guelma	Supervisor
Mr. BOUMEHREZ Farouk	**MCA**	Univ. of Khenchela	Co-supervisor
Mr. MESSAOUDI Kamel	**Professeur**	Univ. of Université de Souk Ahras	Examiner
Mr. DOGHMANE Hakim	**MCA**	Univ. of 8 Mai 1945 Guelma	Examiner
Mr. LABIOD M.Aymen	**MC**	Univ. of Paris-Est Creteil (France)	Invited

Academic year: 2024/2025

بسم الله الرحمن الرحيم

To my parents, who have supported me during my whole life.

To my big and little family.

To my sisters.

To my little dear Ihsane and her daddy.

Acknowledgement

First and foremost, I thank Allah for granting me health, patience, knowledge, and everything necessary to complete this thesis.

I want to express my deepest gratitude to those who have supported and guided me throughout this thesis.

I would like to thank my thesis director, Professor REDADAA Salah, who suggested the subject and supervised me during these years of research.

I would also like to thank Dr. BOUMAHREZE Farouk for listening to me and his guidance, as well as for trusting me during these years in my research activities.

All my gratitude goes to Dr. LABIOD Mohamed Aymen for their continuous support, invaluable guidance, and insightful feedback. Their expertise and encouragement have been crucial in shaping this work and my development as a researcher.

I would like to thank my colleagues and fellow researchers at the Institute of Telecommunications of 8 Mai 1945 University, Guelma.

Finally, I am grateful to my family and friends for their unwavering love and encouragement, patience and understanding, and my friends for their support and belief in me.

Thank you.

الملخص

تطور أنظمة النقل الذكية أدى إلى تقدم كبير في شبكات الاتصال اللاسلكي بين المركبات، حيث يعدّ بث الفيديو الموثوق أمرًا بالغ الأهمية لمجموعة واسعة من التطبيقات، بما في ذلك السلامة والملاحة والترفيه داخل السيارة. ومع ذلك، فإن الطبيعة الديناميكية وغير المتوقعة غالبًا للبيئات المرورية تشكل تحديات كبيرة للحفاظ على بث فيديو عالي الجودة. إن قضايا مثل ظروف الشبكة المتغيرة، وفقدان الحزم، وزيادة التأخير يمكن أن تؤدي إلى تدهور كبير في جودة الخدمة وجودة التجربة للمستخدمين النهائيين. تقدم هذه الأطروحة نُهُوجٌ جديدة متعدد الطبقات يجمع بين حلول المصدر والقناة المشتركة لتحسين بث الفيديو في البيئات المرورية، مع مراعاة كل من معلمات البث وتشفير الفيديو. يعتمد الحل المقترح على الترميز متعدد الوصف والترميز القائم على منطقة الاهتمام و نسخة معدلة من بروتوكول متعدد المسارات كويك. يتيح هذا الاخير إرسال بيانات الفيديو عبر مسارات شبكة غير متجانسة، باستخدام آلية هجينة تجمع بين أوضاع الإرسال الموثوق وغير الموثوق. تسمح هذه الاستراتيجية المزدوجة المسار بمزيد من المرونة والقوة في التعامل مع ظروف الشبكة المتقلبة النموذجية لشبكات الاتصال اللاسلكي بين المركبات. يتم تقييم فعالية الحلول متعددة الطبقات المقترحة من خلال عمليات محاكاة واسعة النطاق في سيناريوهات مرورية واقعية، والتي تُظهر تحسينات كبيرة في جودة بث الفيديو مقارنة بالطرق التقليدية. تساهم هذه الأطروحة في البحث المستمر في شبكات الاتصال اللاسلكي بين المركبات من خلال تقديم إطار عمل قوي لتحسين أداء بث الفيديو، مما يمهد الطريق لأنظمة الاتصال المرورية الأكثر موثوقية وكفاءة.

Abstract

The evolution of intelligent transportation systems has spurred significant advancements in Vehicular Ad Hoc Networks (VANETs), where the reliable transmission of video is crucial for a wide range of applications, including safety, navigation, and in-vehicle entertainment. However, vehicular environments' dynamic and often unpredictable nature poses substantial challenges to maintaining high-quality video streaming. Issues such as varying network conditions, packet loss, and latency can severely degrade the Quality of Service (QoS) and Quality of Experience (QoE) for end-users. This thesis presents a novel cross-layer approach that combines joint source-channel solutions to enhance video streaming over vehicular environments, considering both transmission and video encoding parameters. The proposed solution leverages Multiple Description Coding (MDC), Rrgion Of Interest-based coding, and a modified Multipath QUIC (MPQUIC) protocol extension. It enables video data transmission through heterogeneous network paths, utilizing a hybrid mechanism that combines reliable and unreliable transmission modes. This dual-path strategy allows for greater flexibility and robustness in handling the fluctuating network conditions typical of VANETs. The effectiveness of the proposed cross-layer solutions are evaluated through extensive simulations in realistic vehicular scenarios, which demonstrate significant improvements in video streaming quality compared to conventional methods. This thesis contributes to the ongoing research in VANETs by offering a robust framework for improving video streaming performance, paving the way for more reliable and efficient vehicular communication systems.

Résumé

L'évolution des systèmes de transport intelligents a stimulé des avancées significatives dans les réseaux ad hoc véhiculaires (VANET), où la transmission fiable de la vidéo en streaming est essentielle pour un large éventail d'applications, notamment la sécurité, la navigation et le divertissement embarqué. Cependant, la nature dynamique et souvent imprévisible des environnements véhiculaires pose des défis considérables au maintien d'une diffusion vidéo de haute qualité. Des problèmes tels que les conditions de réseau variables, la perte de paquets et la latence peuvent dégrader considérablement la qualité de service (QoS) et la qualité d'expérience (QoE) pour les utilisateurs finaux. Cette thèse présente une nouvelle approche inter-couches combinant des solutions conjointes source-canal pour améliorer la diffusion vidéo dans les environnements véhiculaires, en considérant à la fois les paramètres de transmission et de codage vidéo. La solution proposée exploite le codage à descriptions multiples (MDC), le codage basé sur la région d'intérêt et une extension modifiée du protocole Multipath QUIC (MPQUIC). Elle permet la transmission de données vidéo via des chemins de réseau hétérogènes, en utilisant un mécanisme hybride combinant des modes de transmission fiables et non fiables. Cette stratégie à double chemin offre une plus grande flexibilité et robustesse pour gérer les conditions de réseau fluctuantes typiques des VANET. L'efficacité des solutions inter-couches proposées est évaluée par le biais de simulations approfondies dans des scénarios véhiculaires réalistes, qui démontrent des améliorations significatives de la qualité de diffusion vidéo par rapport aux méthodes conventionnelles. Cette thèse contribue à la recherche en cours sur les VANET en offrant un cadre robuste pour améliorer les performances de la diffusion vidéo, ouvrant la voie à des systèmes de communication véhiculaires plus fiables et efficaces.

Contents

List of Figures

List of Tables

Abbreviations List

ITS	Intelligent Transport System
VANET	Vehicular Ad-hoc NETwork
V2V	Vehicle-to-Vehicle
V2I	Vehicle-to-Infrastructure
QUIC	Quick UDP Internet connection
JPEG	Joint Photographic Experts Group
MPEG	Moving Picture Experts Group
VLC	variable-length coding
MJPEG	Motion JPEG
DV	Digital Video
QCIF	Quarter Common Intermediate Format
MV	motion vector
FEC	forward error correction
TCP/IP	Transmission Control Protocol/Internet Protocol
ARQ	Automatic Repeat Request
ACK	acknowledgment
RTO	retransmission timeout
RTT	Round Trip Time
RWND	Receive Window
ssthresh	slow start threshold
TLS	Transport Layer Security
AES	Advanced Encryption Standard
AEAD	Authentication Encryption with Associated Data
PSK	pre-shared key
NTNs	Non-Terrestrial Networks
5G	5 Generation
MPQUIC	MultiPath QUIC
MDC	Multiple Description Coding
PSNR	Peak signal-to-noise ratio
SSIM	Structural SIMilarity
VMAF	Video Multimethod Assessment Fusion
HEVC	High-Efficiency Video Coding
CCHP	Center for Connected Health Policy
NCSL	National Conference of State Legislatures
IoT	Internet of Things
WHO	World Health Organization
QoS	Quality of Service

BGP	Border Gateway Protocol
RP	Resource Pooling
SCTP	Stream Control Transmission Protocol
NAT	Network Address Translation
OOO	out-of-order
DSS	Data Sequence Signal
LIA	Linked Increase Algorithm
OLIA	Opportunistic Linked Increase Algorithm
Balia	Balanced Link Adaptation
wVegas	Weighted Vegas
RR	Round-Robin
SUMO	Simulation of Urban MObility
NS-x	The Network Simulator
OMNET	Objective Modular Network Testbed
OSM	OpenStreetMap
fps	frames per second
SI	Signal Intensity
TI	Temporal Intensity
MSE	The Mean Squared Error
DASH	dynamic adaptive streaming over HTTP
OFO	out-of-order
OLS	OverLapped Scheduler
HERP	Hybrid Error Recovery Protocol
QPSK	Quadrature Phase Shift Keying
OFDMA	Orthogonal Frequency Division Multiple Access
EPC	Evolved Packet Core
AR	augmented reality
VR	virtual reality
MIMO	Multiple Input, Multiple Output
mmWave	Millimeter Wave
3GPP	3rd Generation Partnership Project
GEO	Geostationary orbit
MEO	Medium Earth orbit
HDTV	High Definition Television
OSI	Open Systems Interconnection
PDR	Packet Delivery Ratio
PLR	Packet Loss Ratio
E2E	end-to-end
MWOA	Moth Whale Optimization Algorithm
MS	Moth Search
WOA	Whale Optimization Algorithm
MAC	Media Access Control
ACs	access categories
OFO	Out-oF-Order
MPTCP	Multipath TCP
OLS	OverLapped Scheduler
SPFEC	Sub-Packet Forward Correction

ROI	region of interest
ETP	Echocardiogram Transmission Protocol
UDP	User Datagram Protocol
DR	Diabetic Retinopathy
CNN	Convolutional Neural Network
D2PAM	Deep Dual-Patch Attention Mechanism
DERs	Distributed Energy Resources
WSN	Wireless Sensor Network
NR	New Radio
WTS	Wireless Tele Surgery
KPI	Key Performance Indicator
TCP	Transmission Control Protocol
IETF	Internet Engineering Task Force
HOL	Head-Of-Line blocking
BLEST	Both Blocking Estimation-based MPTCP Scheduler
ECF	Earliest Completion First
CWND	congestion window
RTT	Round-Trip-Time
GADaM	Generic Adaptive Deep-learning-based Multipath Scheduler Selector
QP	Quantizer Parameter
AI	All-Intra
HM	HEVC test Model
JCT-VC	Joint Collaborative Team of ISO/IEC MPEG and ITU-T VCEG
LD	Low-Delay
RA	Random-Access
IDR	Instantaneous Decoder Refresh
VVC	Versatile Video Coding
AV1	Alliance for Open Media Video 1

General Introduction

Over the past few years, there has been a growing interest in intelligent transportation systems and vehicular communications among researchers and industry professionals. Vehicular Ad Hoc Networks (VANETs) have emerged as a promising development that can facilitate the deployment of safety, traffic management, comfort, and entertainment applications in the transportation sector. One of the key advancements that could be achieved through VANETs is the transmission of video content over vehicular networks. This development would be instrumental in enabling overtaking maneuvers, pedestrian assistance, support for public transportation, video surveillance, and entertainment video communication, among other applications. However, transmitting compressed video over vehicular networks presents significant challenges, including susceptibility to noise, jitter, and data loss. To meet the requirements of the targeted applications, transmission systems must provide high throughput, low latency, and sound transmission quality. This is particularly challenging in VANETs, characterized by challenging transmission conditions and an associated Packet Loss Rate (PLR) that provides no Quality of Service (QoS) guarantees. Therefore, researchers and industry professionals are keenly interested in developing innovative approaches to address these challenges and unlock the full potential of VANETs for video streaming applications.

Therefore, several technological solutions have been suggested to improve multimedia transmissions over vehicular networks. Among these, Multipath transport protocols have emerged as a significant innovation in data transmission. These protocols significantly change how data moves through the complex network of devices and networks, thereby providing a solution to the challenges of modern data transmission needs. They can potentially transform how we transmit data and offer a profound solution to the intricate web of interconnected devices and networks. Conversely, the H.265/High-Efficiency Video Coding (HEVC) video codec is an effective solution for the bandwidth constraints experienced by VANETs. It carries significant compression gains, which are highly beneficial. The HEVC codec is known for reducing the amount of data that needs to be transmitted, thus enabling the efficient use of the available bandwidth. Because of the limited bandwidth, using HEVC codec is an effective solution for multimedia transmissions over vehicular networks.

In the context of video transmission in VANETs, this thesis work aims to provide some insights into improving the quality of service of wireless video transmissions. With this objective, several solutions have been proposed to enhance video streaming quality at different transmission chain stages. A notable approach that has been proposed is cross-layer solutions, which have been shown to offer significant improvements in quality. This thesis proposes a cross-layer scheme to enhance the High-Efficiency Video Coding (HEVC) video streaming in VANETs that combines source coding techniques, such as Multiple Description Coding (MDC) and region of interest encoding methods, with a modified extension of the Multipath QUIC protocol (MPQUIC). The proposed cross-layer mechanisms have been evaluated through simulation in realistic vehicular environments to assess their effectiveness in enhancing video streaming quality in VANETs.

The manuscript is divided into four chapters. The first chapter provides an overview of the thesis's application context. It describes the approaches and techniques upon which the chosen solutions will rely. The second chapter presents a state-of-the-art review of multipath transport protocols. We also evaluated video transmission performance in a VANET scenario using single-path and multipath transport protocols. The third chapter presents a cross-layer solution to improve video steaming using the Multiple Description Coding technique with a novel extension of the MPQUIC protocol for low-latency streaming. In the fourth chapter, another solution based on the Region Of Interest coding technique was proposed. Finally, the general conclusion synthesizes the main findings of this study and proposes future directions for this work.

CHAPTER 1

Study Context

1 Introduction

In a short time, vehicular networks can expect to become as widespread as smartphones today. Vehicles will be equipped with various sensors, some level of computing power, and the ability to communicate with other vehicles and available infrastructures when needed. This will enable them to gather, process, and share information, paving the way for developing the Intelligent Transport System (ITS). This collection of connected vehicles is called a Vehicular Ad-hoc NETwork (VANET), a kind of Ad-hoc network considering moved vehicles as nodes connected through a wireless connection to enable efficient communication by V2V (Vehicle-to-Vehicle) or V2I (Vehicle-to-Infrastructure) communication. Vehicular Ad-hoc networks are emerging as a promising solution to revolutionize transportation systems, enhance road safety, and improve traffic management. One of the critical applications in the VANET environment is video streaming.

One of the key focus areas of ITS is VANETs, which are currently one of the most promising applications of self-organizing Ad-hoc wireless networks without predefined infrastructure. These vehicular networks allow vehicles to communicate with one another and with nearby road network infrastructures, ultimately leading to safer and more efficiently utilized roads through the timely exchange of information among drivers and relevant authorities such as the police and emergency services.

Video transmission in VANETs poses a significant challenge due to the potential impact of video content on user experience and environmental awareness. Streaming videos in VANETs is complicated due to the hostile nature of vehicular networks, which experience frequent packet losses and node mobility. Before streaming begins, compression technologies like High-Efficiency Video Coding (HEVC), Versatile Video Coding (VVC), Advanced Video Coding (AVC), etc., are commonly utilized to decrease the overall amount of data being transmitted. This study will focus on HEVC to compress data before sending it to the following endpoint. Additionally, video transmission requires high-quality service due to its sensitivity to disruptions on the transmission channel, especially because of the compression techniques used to reduce bandwidth usage.

In this chapter, we will delve into advanced concepts related to video compression, covering the fundamental principles of video compression, including the basics of the encoding and decoding process and the evolution of standards, starting with the first worldwide compression standard, JPEG, until the High-Efficiency Video Coding standard. Furthermore, the chapter will center on the Multiple Description Coding technique (MDC). Finally, the various transport protocols pertinent to this field of study are presented, such as the Transport Controle Protocol(TCP) and the Quick UDP Internet connection (QUIC).

2 Vehicular Ad-hoc NETwork overview

The concept of using radio communications to enhance road safety and communication among vehicles was introduced before the advent of the digital radio communications we are familiar with today. For instance, in 1922, a patent was submitted and later issued in 1925 for "Radio Warning Systems for Use on Vehicles," which involved peer-to-peer radio communication between two devices installed on different vehicles.

The global definition of VANET refers to the ability to establish wireless communication among vehicles, often in movement or between vehicles and infrastructure. High mobility may introduce challenges in establishing and maintaining stable video transmission links, lead to rapidly changing network topologies, and may cause some connectivity failures. One way to overcome this issue is using multi-hop relaying in VANET networks. This means that instead of only relying on direct communication between vehicles (V2V), other vehicles can act as relays to transmit video packets. This approach increases the range of communication, improves network coverage, and increases the possibility of successful video transmission, especially in intermittent connectivity. Moreover, this dynamicity in VANET networks also requires robust routing algorithms and protocols that can adapt to changing conditions and provide reliable video delivery.

In such environments, having limited bandwidth can negatively impact the user experience, particularly with high-definition or real-time video delivery. As a solution, various resource allocation techniques and video compression algorithms have been developed to optimize the utilization of limited bandwidth while ensuring low latency and seamless video delivery. This study will focus on the HEVC encoder as a compression technique to minimize latency. HEVC is a video codec created by a team of experts from ISO/IEC MPEG and ITU-T VCEG (JCT-VC) to enhance compression performance beyond current standards. With HEVC, high video resolution can be achieved with nearly 50% less bit-rate compared to other existing standards[15]. For more information, the next section (Section3) presents the basis of video compression techniques.

3 Video Compression

Video technology is advancing towards digital formats on all levels. A sequence of images can create a slight animation if they are displayed rapidly enough for the human eye to perceive it. The rate of animation is called the number of frames per second. The human eye needs at least ten fps to see a slight animation. However, the quality of the video depends not only on fps but also on image resolution. Resolution refers to the number of pixels that make up the image displayed on the screen. A higher resolution results in a better-quality video. These parameters are crucial for digital video and determine the amount of data to be transmitted and recorded for broadcasting. Often, there needs to be a balance between the optimal video quality and its limitations.

The early 1990s marked a significant era for digital images when the first international standards for compressing audiovisual data were introduced[16, 17]. These standards, Joint Photographic Experts Group (JPEG) [18] and Moving Picture Experts Group (MPEG) [19, 20], were able to establish themselves rapidly in the computer and television industries [19], which enabled the exchange and dissemination of images. A digital image is an image that is stored in binary format in a computer file. It can be created through optical sensors like a camera or scanner or through software such as Photoshop or Libre Office. Digital video is a representation of a visual scene spatially and temporally sampled. A scene is captured at a particular moment to produce an image or a field consisting of rows of spatial samples. This process is repeated at regular intervals (1/25 or 1/30 seconds) to create a moving video signal. Three sets of samples are typically needed to represent a color scene. ITU-R 601 is among the most commonly used digital formats for video representation [21].

3.1 Pixel color space

The first approach estimates all the spectral components by filtering the color signal with numerous narrow-band filters to represent the color. The intensity of each filtered output would estimate the wavelengths present in the spectrum. However, this approach requires too many sensors. Studies on psychovisual equalization have demonstrated that combining three stimuli of specific wavelengths can create nearly any color. The RGB color space is founded on three monochromatic colors: Red (700 nm), Green (546 nm), and Blue B (435.8 nm). Mixing these three colors can produce all other colors [22].

3.2 YUV Format

In addition to the RGB system, other color systems, such as the YUV or YCbCr systems, take advantage of how the brain processes trichromatic signals from the eyes. These signals are made up of a luminance component (Y) and two chrominance components (Cb and Cr) [23]. This principle is utilized in image and video compression. The YUV or YCbCr representation is obtained through a specific transformation:

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.229 & 0.587 & 0.114 \\ 0.596 & -0.274 & -0.322 \\ 0.212 & -0.523 & 0.110 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{1.1}$$

To better distinguish light intensity and color data, we suggest using a luminance and chrominance (YUV) space in which the primary colors are independent of each other.

3.3 First compression idea

To account for the fact that human eyes are less sensitive to color than light intensity, the color information is sub-sampled. This means that half of the information is taken in a 4:2:2 format or a quarter in a 4:2:0 format [24]:

- Format (4:4:4): All components are coded uniformly without any chrominance sub-sampling.

Definition	Scanning	Frequence	Bite rate
QCIF	176x144 progressive	10 to 30 Hz	380ko to 1.1Mo/s
CIF	352x288 progressive	10 to 30 Hz	1.5 to 4.6 Mo/s
SD(NTSC)	640x480 interleaved	60 Hz	13.8 Mo/s
SD(PAL, SECAM)	768x576 interleaved	50 Hz	16.6 Mo/s
SD (DVD)	720x576 interleaved	50 to 60 Hz	15.6 to 18.6 Mo/s
HD(720p)	1280x720 progressive	24 to 60 Hz	33.2 to 83 Mo/s
HD(1080i)	1920x1080 interleaved	50 to 60 Hz	78 to 93 Mo/s
HD(1080p)	1920x1080 progressive	24 to 30 Hz	75 to 93 Mo/s

Table 1.1: The parameters of standard digital video formats.

- Format (4:2:2): The two chrominances have the same number of rows as the luminance but half as many columns.
- Format (4:2:0): The number of rows and columns in chrominances is half that of luminance.

Each video signal component can be coded on eight to ten bits. Eight bits are sufficient to represent a component since the eye cannot distinguish two neighboring levels. In general, the video signal components are most often coded on eight bits with a 4:2:0 format. We, therefore, have an average of 1.5 bytes per pixel [25].

3.4 The reason behind compression

Table 1.1 summarizes the parameters of standard digital video formats [26]. The bit rate required for transmitting uncompressed data is calculated for a 4:2:0 format with 8 bits per component. These formats represent vast amounts of data while the maximum throughput, in Mbps (Megabits per second), offered by access providers is estimated at [27]:

- Broadband networks: 10 Mbps;
- Satellite/internet: 1-2 Mbps;
- ADSL connection: 30 Mbps;
- Video conferences: 128-1000 kbps;
- PDA/ 3G mobile phone: 160-320 kbps

The speeds are thus much too high compared to the bandwidths available for their distribution. Therefore, the video data must be compressed to allow their distribution with realistic bit rates. Video coding standards also need to be defined so that anyone viewing can decompress this compressed data.

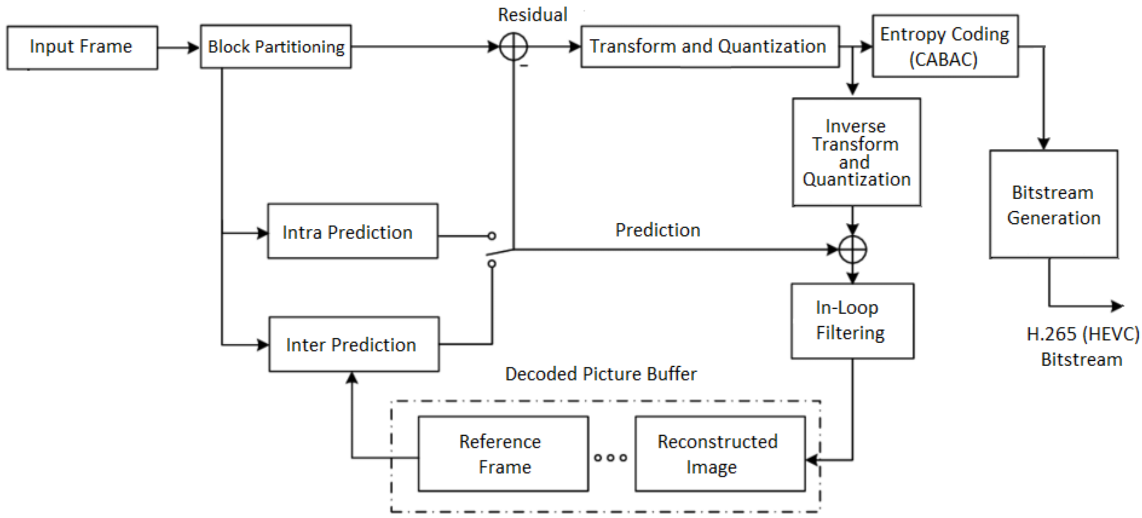

Figure 1.1: Video encoding and decoding process in the HEVC encoder [1]

3.5 Video compression

The goal of video compression, or video coding, is to minimize the amount of spatial and temporal redundancy within video data. This makes it possible to transmit and store the videos on narrow frequency bands and limited-capacity media. The process involves utilizing an algorithm (several to choose from) to process the video source and create a compressed file ready for transmission and/or storage. An inverse algorithm is applied to play the compressed file, resulting in a video containing almost identical content to the original source video (as shown in Figure 1.1). A pair of algorithms that work together is called a video codec, which includes both a coder and a decoder.

3.6 Video Encoding Process

A video codec compresses a source image or video sequence and then decompresses it to create a copy or approximation of the original sequence. The compression is lossless if the decompressed sequence is exactly the same as the original. It is lossy if there are differences between the decompressed sequence and the original.

In video encoding, the CODEC creates a model of the original sequence. The model should use the least amount of bits while maintaining the highest level of accuracy. However, achieving both goals can be challenging because reducing the number of compressed bits may lead to lower picture quality during decoding. A video encoder has three primary functional components: a prediction model, a spatial model, and an entropic encoder, as shown in Figure1.2 [20]. A video encoder processes data in a sequence, including prediction, transformation, and coding, to compress it into a smaller bitstream. The primary treatments are designed to be reversible and result in less data than the original bitstream. For more information, see tables1.2 and 1.3.

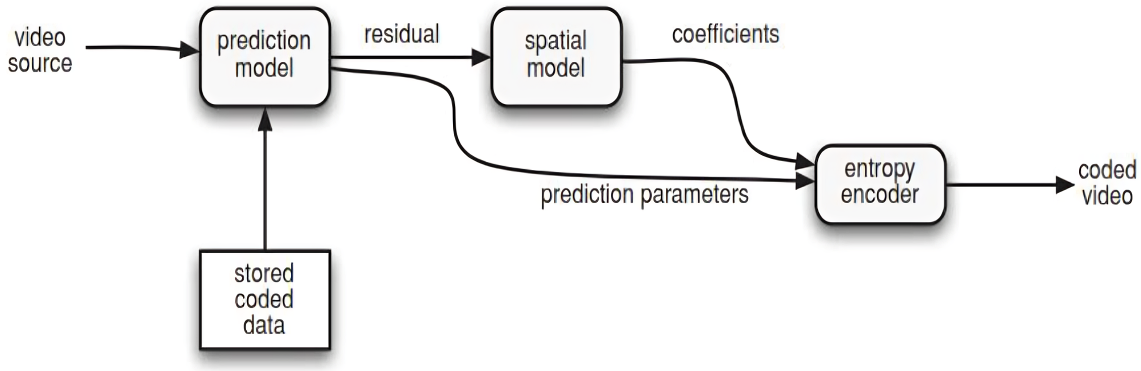

Figure 1.2: Block diagram of a video encoder[2].

Stage	Encoding Process (HEVC Encoder)
Partitioning	- The input frame is divided into **Coding Tree Units (CTUs)** of sizes 64×64, 32×32, or 16×16 pixels. - CTUs are recursively partitioned into **Coding Units (CUs)** to adapt to content complexity. - CUs split into **Prediction Units (PUs)** for prediction and **Transform Units (TUs)** for transform and quantization. - Hierarchical partitioning allows flexible adaptation to complex regions and smooth areas.
Prediction	- **Intra-prediction**: Predicts a block using spatial correlation with adjacent blocks in the same frame. - Supports 33 directional prediction modes for precision. - **Inter-prediction**: Utilizes motion estimation to predict blocks using past and future reference frames. - Supports **Bidirectional Prediction (B-slices)** for high efficiency. - Advanced motion prediction includes **Merge Mode** and **Skip Mode** to reduce signaling overhead.
Motion Estimation and Compensation	- Performs block matching in reference frames using **Sum of Absolute Differences (SAD)** or **Sum of Squared Differences (SSD)**. - Generates **motion vectors** with **quarter-pixel** or **half-pixel** precision for finer motion representation. - Applies **Motion Compensation** to create predicted blocks, reducing temporal redundancy.
Transformation	- Calculates the **residual** (difference between the original and predicted blocks). - Applies a **4-point** or **8-point Integer Discrete Cosine Transform (DCT)** or **Discrete Sine Transform (DST)** to decorrelate data. - Supports variable transform block sizes (**4×4** to **32×32**). - Enables better adaptation to texture complexity.

Continued on next page

Table 1.2 – continued from previous page

Stage	Encoding Process (HEVC Encoder)
Quantization	- Transformed coefficients are quantized to reduce precision, introducing controlled distortion. - The **Quantization Parameter (QP)** governs compression strength. - Custom **Quantization Matrices** can be applied for perceptual optimization.
Entropy Coding	- Uses **Context-Adaptive Binary Arithmetic Coding (CABAC)** for compressing syntax elements, quantized coefficients, and motion vectors. - CABAC adapts context models based on neighboring data for better compression. - Bypasses less significant bits using **bypass coding** to reduce complexity.
Reconstruction	- Applies **inverse quantization** and **inverse transformation** to reconstruct the residual. - The reconstructed residual is added to the prediction block to create the encoded frame. - This frame is used as a reference for subsequent predictions.
In-loop Filtering	- **Deblocking Filter (DF)** smooths block boundaries to eliminate artifacts. - **Sample Adaptive Offset (SAO)** adjusts pixel values to reduce ringing artifacts. - Optionally, the **Adaptive Loop Filter (ALF)** can enhance video quality by reducing distortion.
Bitstream Generation	- All encoded data, motion vectors, prediction modes, and filter parameters are packed into a compliant **HEVC bitstream**. - Bitstream structure includes **Network Abstraction Layer (NAL)** units for error resilience.
Output	- A compressed video stream optimized for storage and transmission is generated, maintaining high visual quality at reduced bitrates.

Table 1.2: Video Encoding Steps.

Stage	Decoding Process (HEVC Decoder)
Bitstream Parsing	- Reads the **Network Abstraction Layer (NAL)** units to extract compressed data, syntax elements, and metadata. - Syntax includes motion vectors, quantization parameters, and block partitioning information.
Entropy Decoding	- Applies **CABAC decoding** to recover quantized transform coefficients, motion vectors, and syntax data.

Continued on next page

Table 1.3 – continued from previous page

Stage	Decoding Process (HEVC Decoder)
Inverse Quantization	- Reconstructs the transform coefficients by applying inverse quantization using the same **Quantization Parameter (QP)**.
Inverse Transformation	- Applies the **Inverse DCT (IDCT)** or **Inverse DST (IDST)** to convert frequency-domain data back into spatial residuals.
Motion Compensation	- Reconstructs predicted blocks using motion vectors and reference frames for inter-predicted content.
Prediction Reconstruction	- Uses decoded intra- or inter-prediction modes to rebuild predicted blocks.
Reconstruction	- Combines the residual with the predicted block to reconstruct the decoded frame.
In-loop Filtering	- Applies **Deblocking Filter** and **Sample Adaptive Offset (SAO)** to enhance video quality.
Output	- The fully reconstructed video frames are displayed or stored for playback.

Table 1.3: Video Decoding Steps.

Data compression is achieved by removing redundancy, i.e., components that are not necessary for the faithful reproduction of data. Many data types contain statistical redundancy and can be efficiently compressed using lossless compression. Thus, the reconstructed data at the decoder output constitutes a perfect copy of the original data. Unfortunately, lossless compression of image and video information only gives moderate compression. Lossy compression is needed to achieve higher compression. In a lossy compression system, the decompressed data is not identical to the source data, and much higher compression ratios can be achieved over time, detriment of a loss of visual quality. Lossy video compression systems work by eliminating subjective redundancy - elements of the image or video that can be removed without substantially impacting the viewer's perception of visual quality.

- Spatial redundancies (intra-images): A single image contains many redundancies, especially if it has large uniform areas where neighboring pixels have strong correlations. In such cases, coding each of these points is unnecessary.

- temporal redundancies (inter-images): in a video sequence, the successive images are practically identical, except for the points in motion. Only these movements will be coded.

- subjective redundancies (vision of fine details): compressing means weighing what is more or less visible and not coding what is not visible.

- Statistical redundancy: This redundancy applies to the codes themselves. Some are used more than others, so we will reserve the shortest words for them. This operation is called entropy coding and does not involve any loss.

3.7 Types of video encoders

The hybrid type video coding scheme is the most common since it is used in all video coding standards (MPEG-x, H.26x). It is called hybrid because it is a closed-loop scheme that uses the already encoded/decoded information to perform the encoding of the current part using different modules. The hybrid scheme is always of the same form. Only the modules that compose it (prediction, transformation, and binary coding) differ from one coder to another [18]. Other than the hybrid scheme, there are other types of video coding schemes based on wavelet approaches, namely:

1. Wavelet-based (open-loop) encoders typically use motion-compensated temporal filtering coupled with a 2D wavelet transform. They are called "t+2D" schemes since they first perform a temporal decomposition before performing the classical 2D spatial decomposition.

2. Coders by analysis-synthesis often use wavelet transformations. In this scheme, the analysis step corresponds to estimating the motion of the objects in the video sequence using deformable 2D meshes and decorating the motion and texture information. The textures are straightened according to the estimated motion.

These wavelet schemes are just as efficient and exciting as the hybrid-type schemes, but they have never been retained for normalized solutions.

3.8 Standard

It is worth noting that standards and codes are two different things. The standard refers to a specific format, whereas the codec is either hardware or software that can encode and decode (compress and decompress) a data stream. Thus, for example, it is inaccurate to speak of an H.265 codec because several codecs exist, such as HM[28], x265[29], or libde265[30]. However, in the absence of an alternative, particular codecs directly borrow the name of the corresponding standard.

Video compression standards provided by:

- ISO International Organization for Standardization.
- ITU International Telecommunications Union.
- JVT Joint Video Team (H.264/MPEG-4-AVC in 2003).
- JCT-VC Joint Collaborative Team on Video Coding: for HEVC codec development.

3.8.1 Joint Photographic Experts Group (JPEG)

The JPEG standard for still image compression was introduced in 1989. Initially developed for IT purposes, it compresses images by taking advantage of spatial redundancies. However, this compression results in the loss of some information. According to this standard, a video sequence is a series of still images. However, each manufacturer has created their own format based on JPEG. The same method used to compile still images in JPEG is used for animated sequences, resulting in Motion JPEG (MJPEG). It is used by most video capture cards and in Digital Video (DV) technology. Temporal compression utilizes a distinct approach by detecting minimal variations between successive frames. MPEG implements this technique, which is about four times more space-efficient than MJPEG while maintaining the same level of visual quality. Both JPEG and M-JPEG coding are based on intra-frame compression.

3.8.2 MPEG1

MPEG-1 is a standard for lossy video and audio compression for multimedia applications. It compresses moving pictures and audio data for digital storage media at approximately 1.5 Mbit/s, enabling video CDs, digital cable, satellite television, and digital audio broadcasting (DAB)[31]. Similar to JPEG, MPEG-1 eliminates spatial redundancies within an image and exploits temporal redundancies between images (inter-image coding). Nowadays, MPEG-1 is widely used in various products and technologies. The most popular part of this standard is the MP3 audio format. The MPEG-1 standard is published as ISO/IEC 11172 [32].

3.8.3 MPEG-2

First introduced in 1994, MPEG-2 uses the same interframe coding as MPEG-1 but offers superior image quality ranging from 3 to 10 Mbps for standard video to 300 Mbps for high definition. All digital television operators use the MPEG-2 standard, and both MPEG-1 and MPEG-2 process their data in GOPs.The MPEG-2 standard covers image and sound broadcasting via satellite, cable, terrestrial, and DVDs. It was developed for broadcast production, post-production, and editing. Compared to the MPEG-1 standard, MPEG-2 has the following differences:

- Support for higher quality image formats (up to HD).
- Interlaced scan (50 frames/s) and progressive scan (50 frames/s) processing.
- Compatibility with MPEG-1, i.e., an MPEG-2 decoder can decode an MPEG-1 signal.

3.8.4 H.261

The ITU-T initially introduced the H.261 video compression standard in 1990. It belongs to the H.26x family of video coding standards developed by the VCEG domain of ITU-T. This standard is considered the first. The H.261 model is a practical and valuable video coding standard. It was created for transmission over ISDN lines, with data rates multiples of 64 kbit/s. The encoding algorithm is designed to work with video bitrates ranging from 40 kbit/s to 2 Mbit/s. This standard offers support for two video frame sizes: CIF and QCIF. Images can be I or P encoded, but not B [20].

3.8.5 H.263

H.263 is a video compression standard initially designed as a low-bitrate compressed format for video conferencing. VCEG developed it as part of the ITU-T domain's H.26x family of video coding standards. In 1998 and 2000, additional enhanced functions were added, followed by minor additions in 1997 and 2001. A unified edition of the standard was created in 2005. This standard supports Sub-QCIF and QCIF and optionally CIF, 4CIF, and 16CIF. The use of a B-frame is possible. In addition to this base configuration, six coding options have been added to increase coding performance and functionality.

H.263 (2000) has been updated to include three new options, now known as H.263++. This version offers better quality and compression rates while also considering the challenges of real-time video transmission over networks with unreliable quality of service, such as packet loss.

3.8.6 H.264

MPEG-4 is a method of compressing digital audio and video data. It was introduced in late 1998 and designated a standard for audio and video coding formats and related technology agreed upon by the ISO/IEC Moving Picture Experts Group (ISO/IEC JTC1/ SC29 / WG11). ISO/IEC 14496 [20]. The MPEG-4 format includes compression of audiovisual data for webcasting, CD distribution, voice, and television broadcasting. The standard allows coding a wide variety of video formats (size, resolution, video frequency) and objects of arbitrary shape, still images, and 3D synthetic objects. MPEG-4 encompasses all-new multimedia applications such as Internet downloading and streaming, mobile phone multimedia, digital radio, video games, and HD television.

3.8.7 High-Efficiency Video Coding (HEVC)

HEVC is a new video coding standard that improves coding efficiency and network transmission. It reduces storage by 50% compared to its predecessor, H264/AVC, while maintaining high image quality [33]. However, it does require extra computational power, which can be 2 to 10 times greater. The new standard promises bandwidth savings of 20 to 60% compared to its predecessor.

Like H.264 and MPEG-2 encoding, HEVC encoding uses three types of I-B and P frames in a set of frames, combining compression elements between one frame and another. The HEVC video coding layer has several processing blocks, as depicted in Figure 1.3 below. The raw video frame is divided into blocks, and each block region is passed through various encoder elements. The first frame is coded using intra-frame prediction and is called I-Frame. The remaining frames in the same group can be encoded using inter-frame prediction, where the prediction can be made (P-Frames) or bidirectional (B-Frames). The inter-frame prediction uses reference frames and selected motion vectors (MVs) that indicate the motion of various blocks [34], [41].

Performance comparison between HEVC and H.264/AVC:

- HEVC encoding incorporates various advancements, such as frame block encoding. H.264 encoding employs small clusters with a maximum size of 16×16, whereas HEVC encoding uses frame block or CTB encoding with a maximum size of 64×64 pixels. Larger block sizes are more effective when encoding large images, like 4K.

- The H.264 encoding technique relies on nine internal prediction trends, whereas HEVC encoding can utilize more than 35 directions. The HEVC technique can achieve higher internal frame compression efficiency by incorporating more potential reference pixel blocks. However, this comes at the expense of longer coding time required to search for additional trends.

In general, H.265 provides several advantages over H.264, including better compression, delicate image, and bandwidth saving.

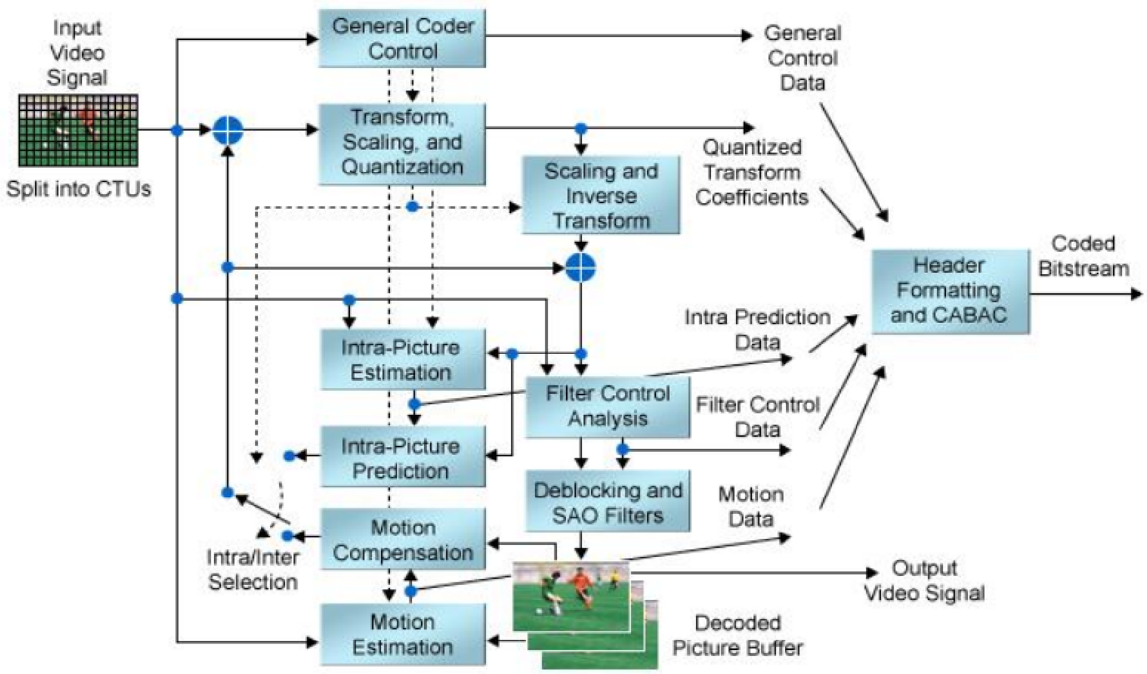

Figure 1.3: Block diagram of the HEVC encoder (with decoder modeling elements shaded in light gray) [3].

4 Multiple Description Coding.

Multiple description coding was initially used for speech communication over a circuit-switched network in the 1970s. Traditionally, to avoid communication interruptions, an additional transmission link was on hold and would be activated in failure of the main link. However, this approach was not cost-effective, hence the idea of splitting the information into two or more channels.

In 1979, MDC was introduced at the IEEE Information Theory Workshop. The MD problem was presented by researchers including Gersho, Witsenhausen, Wolf, Wyner, Ziv, and Ozarow [35]. The scenario is based on a source that can be described using two distinct codes with bit rates R1 and R2, respectively. Each code can be decoded separately with D1 and D2 distortion, while decoding both codes results in D0 distortion. The MD problem aims to define the possible combinations of D0, D1, D2, R1, R2 that can be achieved.

Multiple Description Coding (MDC) is a technique used in multimedia data compression to provide robustness and error resilience in situations where the transmitted or stored data may be subject to packet losses or other transmission errors. It is beneficial in applications that involve real-time communication, such as video streaming or teleconferencing, where maintaining the quality of the media is crucial.

The basic idea behind Multiple Description Coding [4] is to create multiple independent representations or descriptions of the source data. Each description contains a portion of the information needed to reconstruct the original data. These descriptions are then transmitted or stored using separate communication channels or storage media. If some descriptions are lost or corrupted during transmission or storage, the

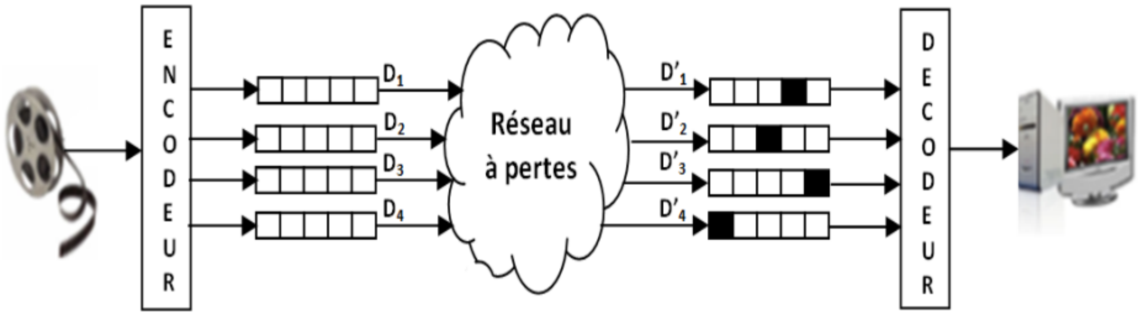

Figure 1.4: The main structure of the MDC technique of a video sequence (black boxes indicate lost information) [4].

receiver can still reconstruct a reasonably good approximation of the original data using the available descriptions.

Here are the key concepts and techniques commonly associated with MDC:

1. Redundancy: MDC introduces redundancy by encoding the source data to allow error resilience. By creating multiple descriptions, redundant information is included in each description, ensuring that the loss of one or more descriptions does not lead to significant degradation in the reconstructed data quality.

2. Error Resilience: The redundant information in multiple descriptions helps in mitigating errors. When some descriptions are lost, the receiver can use the remaining descriptions to reconstruct the data by combining them or selecting the most reliable descriptions.

3. Encoding: The source data is encoded into multiple descriptions using techniques like transform coding (e.g., Discrete Cosine Transform), predictive coding (e.g., motion compensation), or other compression methods. Each description typically represents a different portion or perspective of the source data.

4. Transmission or Storage: The multiple descriptions are transmitted or stored using independent channels or media. This means that if some descriptions are lost or corrupted during transmission or storage, the others can still be utilized to reconstruct the data.

5. Decoding: At the receiver's end, the received descriptions are decoded to reconstruct the original data. The decoding process may involve combining the received descriptions, selecting the most reliable descriptions, or using other techniques to minimize the impact of lost or corrupted descriptions.

6. Quality and Rate-Distortion Tradeoff: MDC allows for a tradeoff between the quality of the reconstructed data and the amount of redundancy or bitrate required. By adjusting the number of descriptions and the amount of redundancy in each description, it is possible to achieve different levels of error resilience and data quality.

MDC has found applications in various multimedia systems and networks, including video streaming over unreliable networks, video conferencing, peer-to-peer communication, and multimedia content delivery over the Internet. It offers advantages in scenarios where the communication channels or storage media are prone to errors, packet losses, or network congestion, providing a more robust and reliable transmission or storage of multimedia data.

4.1 Challenges and solutions

Multiple Description Coding (MDC) is a powerful video transmission technique to improve data delivery resilience, particularly in unreliable network environments. It achieves this by generating multiple descriptions of the same video content, each encoded with varying levels of redundancy and independence. These multiple descriptions are then sent through different transmission paths to the receiver, providing robustness against packet losses, network congestion, and other transmission impairments. While MDC offers significant advantages, it also comes with its own challenges and requires thoughtful solutions for efficient video transmission. Below is a detailed explanation of some of the key challenges and corresponding solutions in Multiple Description Coding for video transmission:

1. Redundancy vs. Efficiency Tradeoff: One of the primary challenges in MDC is finding the right balance between redundancy and efficiency. Creating multiple descriptions involves additional data overhead, which increases the bandwidth requirement and may reduce the overall efficiency of video transmission. Solution: Researchers focus on developing adaptive MDC techniques that dynamically adjust the redundancy levels based on the network conditions. This adaptive approach optimizes the tradeoff between robustness and efficiency, ensuring efficient utilization of the available bandwidth while maintaining reliable video delivery.

2. Scalability and Network Diversity: Video transmission often involves diverse network environments, such as wired, wireless, or hybrid networks. Ensuring effective MDC performance across these different scenarios is challenging. Solution: It is crucial to design scalable MDC algorithms that can adapt to varying network conditions. These algorithms should consider the characteristics of the specific network, such as packet loss rates, bandwidth constraints, and latency, and adapt the multiple descriptions accordingly to achieve optimal video quality and delivery.

3. Synchronization and Reconstruction: Multiple descriptions are transmitted over different paths so that they may arrive at the receiver at different times. Ensuring synchronization and compelling reconstruction of the original video from these descriptions is a significant challenge. Solution: Researchers work on advanced synchronization and reconstruction algorithms that consider the different arrival times of descriptions. They can reconstruct the video with minimal distortion and latency by aligning and combining the received descriptions.

4. Overhead and Resource Utilization: Generating multiple descriptions incurs additional computational overhead, which can be a concern, especially in resource-constrained devices or real-time applications. Solution: Optimizing the encoding and decoding processes is essential to reduce computational complexity while maintaining high video quality. Efficient coding techniques, such as subband-based methods or spatial-temporal segmentation, are explored to minimize overhead and improve resource utilization.

5. Robustness to Packet Loss: The primary motivation behind MDC is to enhance video transmission in the presence of packet losses. However, efficiently handling

packet losses in a distributed manner poses its own challenges. Solution: Advanced error concealment and recovery techniques are employed to manage packet losses effectively. This includes methods like forward error correction (FEC), interleaving, and retransmission strategies to recover missing data and improve video quality.

6. Decoder Complexity and Latency: The receiver must process and combine multiple descriptions to reconstruct the video, which can increase decoder complexity and latency. Solution: Designing efficient decoding algorithms and hardware implementations is crucial to minimizing complexity and reducing latency. Tradeoffs between complexity and video quality are carefully considered to ensure practical, real-time video transmission.

7. Transmission Path Selection: Determining the most suitable transmission paths for the multiple descriptions is challenging, as network conditions may change dynamically. Solution: Implementing intelligent path selection mechanisms that consider real-time network conditions, available bandwidth, and packet loss rates can significantly improve MDC's success in delivering reliable video content.

5 Transport ProtocolS

Before the 1960s, networks were primarily based on the telephone network. This network was created to connect telephones during a call. A call was initiated by establishing a connection between the two parties. This connection was made by creating an electrical circuit between the two telephones for the duration of the call. The connection was cleared once the call was completed, and the circuit became available for other users. Users were billed based on the call's duration and the identification of the connection endpoints. Each user was given a certain amount of bandwidth or capacity to transmit information, typically voice sounds. The telephone network evolved from its analog origins to digital, significantly improving its reliability and performance. [36]

Since its inception in the late 1980s, the Web has evolved significantly. Modern web pages have grown in size and complexity, often comprising hundreds of individual pieces that must be downloaded for the web browser to load them fully. However, the mechanisms behind the Web have not kept pace with its rapid evolution. Web transport protocols that have been traditionally used are becoming less effective in providing a satisfactory user experience. This is due to the growing complexity of modern web pages and the inefficiencies of the underlying transport protocols, resulting in prolonged load times. As a result, the load time of a web page is vital for a positive user experience and can potentially impact the revenue it generates.

Studies conducted by Google have shown that users have little patience for slow-loading mobile web pages, which significantly affects their overall performance [37]. Studies have revealed that the average time for a mobile web page to load completely is 15 seconds. In 70% of cases, it takes more than five seconds before the web browser starts rendering and seven seconds to load all visible elements before scrolling. Furthermore, research shows that over 50% of users will leave a page if it takes more than 3 seconds to load. The probability of abandonment rises as loading times increase. For instance, as the load time increases from 1 second to 10 seconds, the probability of abandonment increases by 123%. They have found that mobile sites that load in 5 seconds have significantly better ad viewability, longer average session times, and earn up to twice as much from advertisements compared to sites that take 19 seconds to load [38].

5.1 Computer Networking

Computer networking is the practice of connecting multiple computers and devices via links to share resources and exchange data. It enables the seamless transfer of information between devices, regardless of their physical location. Networks can be established over various physical mediums, such as wired (e.g., Ethernet cables) or wireless (e.g., Wi-Fi), and they follow specific protocols and standards to ensure data integrity, security, and efficient transmission.

Critical components of computer networking include *Nodes:* These devices connected to the network, such as computers, servers, routers, switches, and other devices capable of sending or receiving data; *Links* refer to the physical or wireless connections that allow data transmission between nodes. These links can be wired (like Ethernet cables) or wireless (like Wi-Fi); *Protocols:* Networking protocols define the rules and conventions for communication between devices on the network. Examples include TCP/IP (Transmission Control Protocol/Internet Protocol) for Internet communication and HTTP (Hypertext Transfer Protocol) for web browsing; *Routers and Switches:* These network devices are responsible for directing data traffic between networks or devices. Routers connect multiple networks, while switches facilitate communication within a local network; *Network Security:* This involves implementing measures to protect the network from unauthorized access, data breaches, and other cybersecurity threats.

The Internet operates as a packet-switched network, unlike circuit-switched telephone networks. In a packet-switched network, nodes do not have a dedicated channel for communication. Instead, messages are split into packets and forwarded between nodes until they reach the intended recipient. These packets contain headers that help nodes route them to the intended receiver. The message itself is contained in the payload. To describe the Internet communication process, various layers are used as an abstraction, each with its own responsibility. This is commonly referred to as the TCP/IP protocol stack [39] (Figure1.5).

To make the TCP/IP stack easier to understand, let's compare a scenario to the information in Table 1.4. Imagine the following scenario:

L5: Alice intends to communicate with Bob through a written letter. For Bob to comprehend the letter's contents, Alice must craft it using a language that Bob is familiar with.

L4: Bob may share his home with other individuals. Alice writes Bob's name on the envelope to guarantee the letter reaches the intended recipient.

L3: Alice cannot deliver a letter directly to Bob's home as they live in different cities. Therefore, she must rely on the postal service to deliver the letter. Alice writes Bob's address on the letter to ensure it reaches his home. Since the two cities are not adjacent, the letter must go through various distribution centers before reaching Bob's address. For efficiency, the letter should be transported through the shortest path.

L2: When a postal service worker is transporting a letter between stops, they do not need to worry about how to reach Bob's address. Their only concern is delivering the letter to the next stop.

L1: Finally, Alice must put the letter in a letter box so the postal service can pick it up.

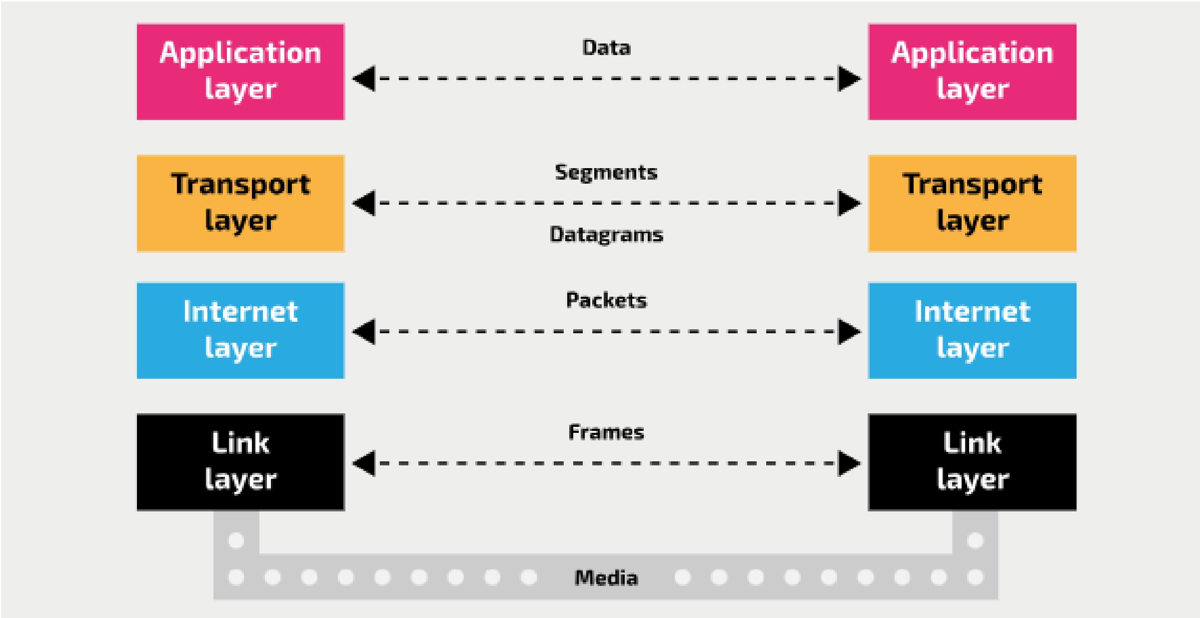

Figure 1.5: The TCP/IP Protocol Stack [5].

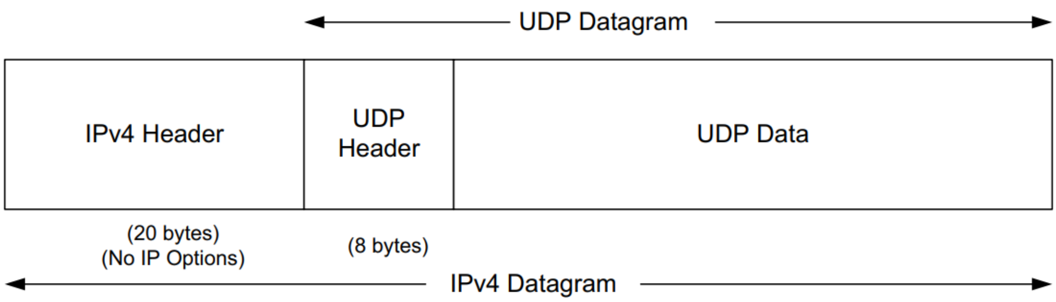

Figure 1.6: Encapsulation of a UDP datagram in a single IPv4 datagram (the typical case with no IPv4 options) [6]

5.2 User Datagram Protocol (UDP)

The article [40] is the official specification of UDP, which has remained a standard for more than three decades without any significant revisions. One of the primary characteristics of UDP is that it does not provide any error correction mechanism. In other words, it sends the data packets that the application writes to the IP layer, but there is no guarantee that they will arrive at their intended destination. Additionally, UDP has no protocol mechanism to prevent high-rate UDP traffic from negatively impacting other network users. At first glance, these limitations might lead one to think that there are no benefits to using UDP. However, despite its lack of reliability and protection, UDP still has its advantages. Its connectionless nature results in less overhead than other transport protocols, making it a popular choice for applications that require low delay and are less sensitive to data loss. Figure 1.6 shows the encapsulation of a UDP datagram as a single IPv4 datagram. The IPv4 Protocol field has the value 17 to indicate UDP.

As an individual interested in application development, you may have questioned

layer	name	responsibility
L1	physical	Deals with the physical data transmission over the local network medium, such as Ethernet or Wi-Fi. It is responsible for packaging data into frames and managing the interaction between the networking hardware and the physical network medium.
L2	link	Handles communication between adjacent nodes
L3	network	Routing data packets across different networks, regardless of their physical paths. The core protocol in this layer is the Internet Protocol (IP), which assigns unique IP addresses to devices and allows them to communicate with each other over the Internet.
L4	transport	It is responsible for end-to-end communication between devices on the network. It ensures the reliable and ordered delivery of data.
L5	application	It includes various protocols that enable user applications to communicate over the network. This layer deals with high-level communication tasks and provides interfaces for applications to access network services.

Table 1.4: overview of the different layers.

the rationale behind constructing an application over UDP instead of TCP. After all, TCP is known to offer a dependable data transfer service, whereas UDP does not. However, it is essential to note that certain applications are better suited for UDP. This is due to the following reasons [41]:

- **Finer application-level control over what data is sent and when**. When an application process sends data via the UDP, the data is immediately packaged into a UDP segment and transmitted to the network layer. This means that there is no congestion control mechanism in place for UDP, unlike TCP. TCP has a congestion control mechanism that slows down the transmission rate when one or more links between the source and destination hosts are congested. Additionally, TCP will continue to resend a segment until it is acknowledged by the destination host, regardless of how long it takes to achieve reliable delivery. However, the service model TCP provides is not always well-suited to the needs of real-time applications. These applications typically require a minimum sending rate and cannot tolerate excessive delays in segment transmission while also being able to handle some data loss. In contrast, UDP's no-frills segment-delivery service can provide the necessary speed and flexibility that real-time applications require. Nonetheless, real-time applications can still utilize UDP and implement any additional functionality they may need as part of their application.

- **No connection establishment**. Later on, we will delve into the technicalities of the TCP and how it uses a three-way handshake before it begins the actual transfer of data. On the other hand, UDP does not have any formalities and simply transmits data without any preliminaries. This characteristic is advantageous since there is no delay in establishing a connection. This is why Domain Name

System (DNS) runs over UDP instead of TCP, as it would be significantly slower if it ran over TCP. Conversely, Hypertext Transfer Protocol (HTTP) uses TCP as its transport protocol because reliability is crucial in transmitting text-heavy web pages. However, the TCP connection-establishment delay in HTTP is a major contributor to the delays associated with downloading web documents. It is worth mentioning that Google's Chrome browser uses the QUIC protocol, which utilizes UDP as its underlying transport protocol and implements reliability in an application-layer protocol on top of UDP (See Section5.4 for more information.).

- **No connection state**. TCP is designed to maintain a connection state in the end systems. This connection state is comprised of several critical parameters, such as receive and send buffers, congestion-control parameters, and sequence and acknowledgement number parameters. The purpose of this connection state is to enable the implementation of TCP's reliable data transfer service and to provide effective congestion control. In contrast, UDP does not maintain a connection state and, therefore, does not track any of these parameters. This means that a server devoted to a particular application can typically support many more active clients when running over UDP instead of TCP. The lack of a connection state in UDP can offer benefits in terms of increased flexibility and speed, but it also comes with certain trade-offs regarding reliability and consistency.

- **Small packet header overhead**. It is important to note that there is a significant difference in the amount of header overhead between TCP and UDP protocols. Specifically, every TCP segment contains 20 bytes of overhead, while UDP only requires 8 bytes. This disparity in overhead can have a notable impact on network performance and efficiency, making it essential for network administrators to consider which protocol is best suited for their needs carefully.TCP segment has 20 bytes of header overhead in every segment, whereas UDP has only 8 bytes of overhead.

5.2.1 UDP Segment Structure

Figure 1.7 shows a UDP datagram, including the payload and UDP header (which is always 8 bytes in size). Port numbers are crucial in facilitating communication between different processes in a network. Essentially, these numbers act as virtual mailboxes, enabling a protocol implementation to identify the sending and receiving processes accurately. It's important to note that port numbers are entirely abstract and don't correspond to any physical entity on a host. Regarding UDP, port numbers are represented as positive 16-bit numbers. Additionally, the source port number is optional and can be set to 0 if the datagram's sender doesn't require a reply. As for transport protocols such as TCP, UDP, and SCTP, they utilize the destination port number to help demultiplex incoming data from IP [36].

One interesting aspect of port numbers is that they can be made independent of different transport protocols. This is because IP demultiplexes incoming IP datagrams to a specific transport protocol based on the value of the Protocol field in the IPv4 header. As a result, this separation allows TCP port numbers to be distinct from UDP port numbers and so on. In other words, different servers can use the same port number and IP address using different transport protocols. This has significant implications for network design and can help increase efficiency and flexibility [36].

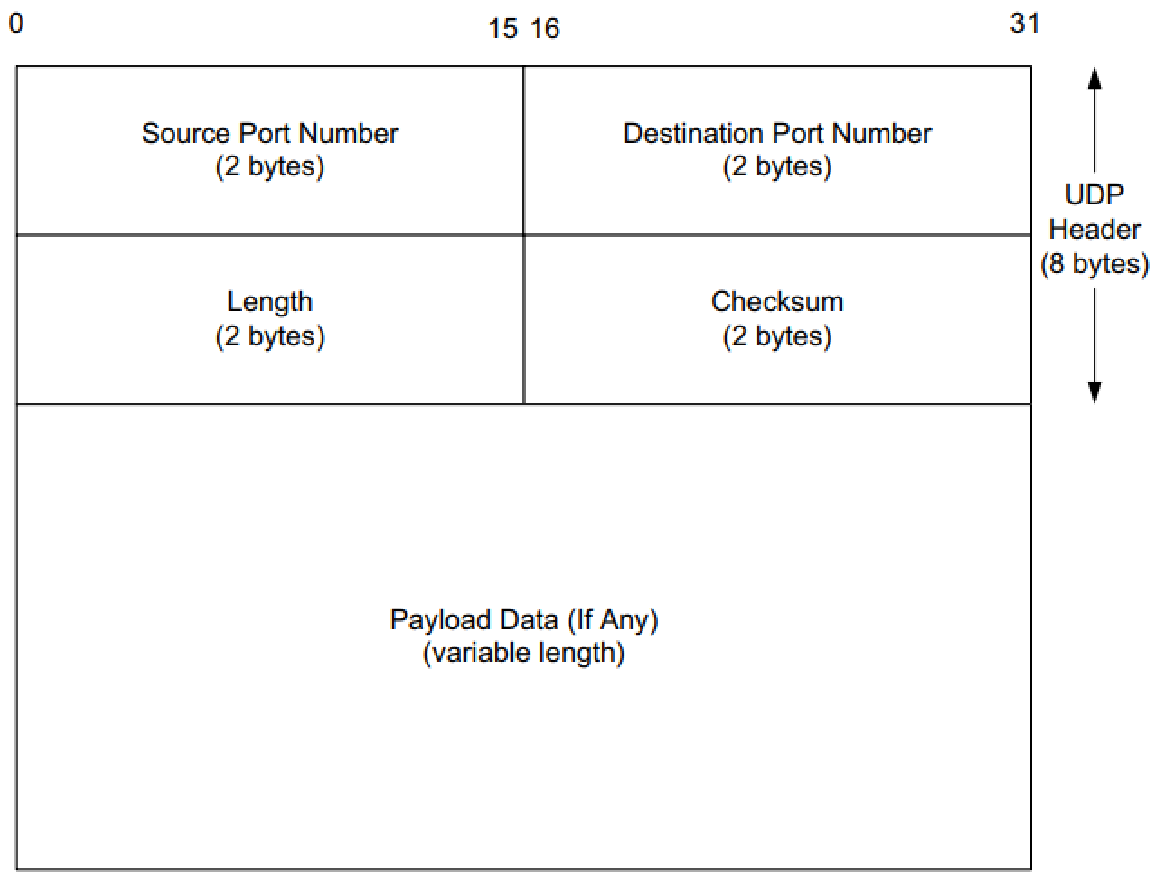

Figure 1.7: The UDP header and payload (data) area [6].

If we take a look at Figure 1.7, we can see that the UDP Length field is actually the length of the UDP header and the UDP data, all measured in bytes. The minimum value that we can assign to this field is 8. It's important to note that it's acceptable to send a UDP datagram that contains 0 bytes of data, although this is a relatively rare occurrence. Additionally, we should acknowledge that the UDP Length field is somewhat redundant, as the IPv4 header already contains all the information about the datagram's total length. Therefore, the length of a UDP/IPv4 datagram is simply the total length of the IPv4 datagram minus the length of the IPv4 header. It's crucial that the UDP Length field matches the length that we computed from the IP-layer information, regardless of the circumstances.

5.2.2 UDP Checksum

The UDP employs a checksum to detect errors that may occur during transmission. This verifies whether the bits in the UDP segment have been modified as they travel from the source to the destination. At the sender's end, UDP performs a 1s complement of the sum of all the 16-bit words in the segment, with any overflow encountered during the sum being wrapped around. The final result is then placed in the checksum field of the UDP segment. This process ensures that the data transmitted is free from errors and is received accurately at the destination.

The checksum process plays a crucial role in ensuring that the data transmitted from the source to the destination is free from errors and is received accurately. It guarantees that the data is not modified while in transit. It is worth noting that the checksum in the IPv4 header only covers the header and not any data in the IP packet. Moreover, it is recomputed at each IP hop to maintain data integrity. Transport protocols such as TCP and UDP use checksums to cover their headers and data. With UDP, the checksum is optional, although it is strongly recommended to use it. On the other hand, with TCP and other transport layer protocols, the checksum is mandatory. Without the checksum, the data transmitted can be susceptible to errors and may contain inaccuracies. For applications to receive error-free data, the transport layer protocol, such as UDP, must always compute a checksum or use some other error detection mechanism before delivering the data to the receiving application. This ensures that the data received is accurate and error-free, thus providing reliable data transmission.

5.3 Transmission Control Protocol (TCP)

Up until now, we have discussed protocols that lack their own reliable mechanisms for delivering data. While they can detect erroneous data through mathematical functions like checksum or CRC, they do not prioritize fixing these errors. With UDP, no error repair is done, while Ethernet and its related protocols provide some retries before giving up if unsuccessful. Researchers use information theory to encode data in ways resistant to communication channel errors to address errors. Error-correcting codes, which add redundant bits to recover information despite damaged bits, are a critical method for handling errors. Another approach is Automatic Repeat Request (ARQ), which involves trying to send the information again until it is received. Many communication protocols, including TCP, rely on ARQ.

The Transmission Control Protocol (TCP) [42] is the dominant transport layer protocol on the Internet. As shown in Table 1.5 and Figures 1.8 and 1.9, TCP is considerably more intricate than the simple alternative of UDP. TCP offers a broader range of features compared to UDP, which addresses the shortcomings of underlying layers. It establishes a reliable, two-way communication channel between the sender and receiver before data exchange. It ensures that all data packets arrive in order and without errors by retransmitting lost packets. TCP provides reliable data transmission. It uses acknowledgments, sequence numbers, and retransmissions to ensure data packets reach the destination correctly and in order. If any packet is lost or corrupted, TCP will retransmit it until successful delivery. TCP employs congestion control mechanisms to avoid network congestion and prevent data loss. It dynamically adjusts its sending rate based on the perceived network conditions. These mechanisms are covered in the following sections.

Features	UDP	TCP
Connection establishment	No	Yes
Reliable data transfer	No	Yes
Error detection	Yes	Yes
Flow control	No	Yes
Congestion control	No	Yes

Table 1.5: UDP and TCP features.

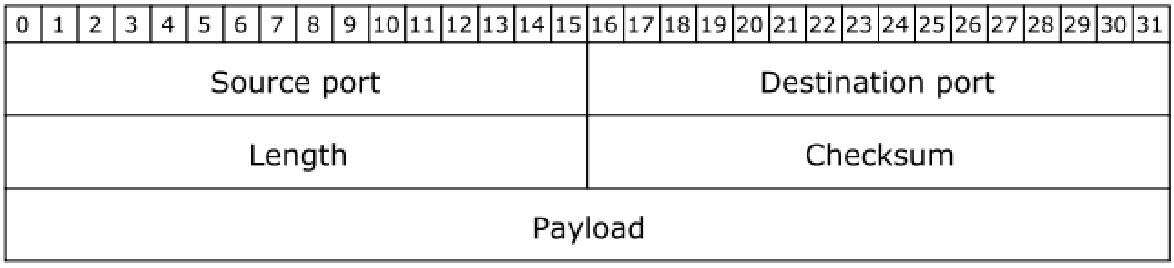

Figure 1.8: UDP header format [6].

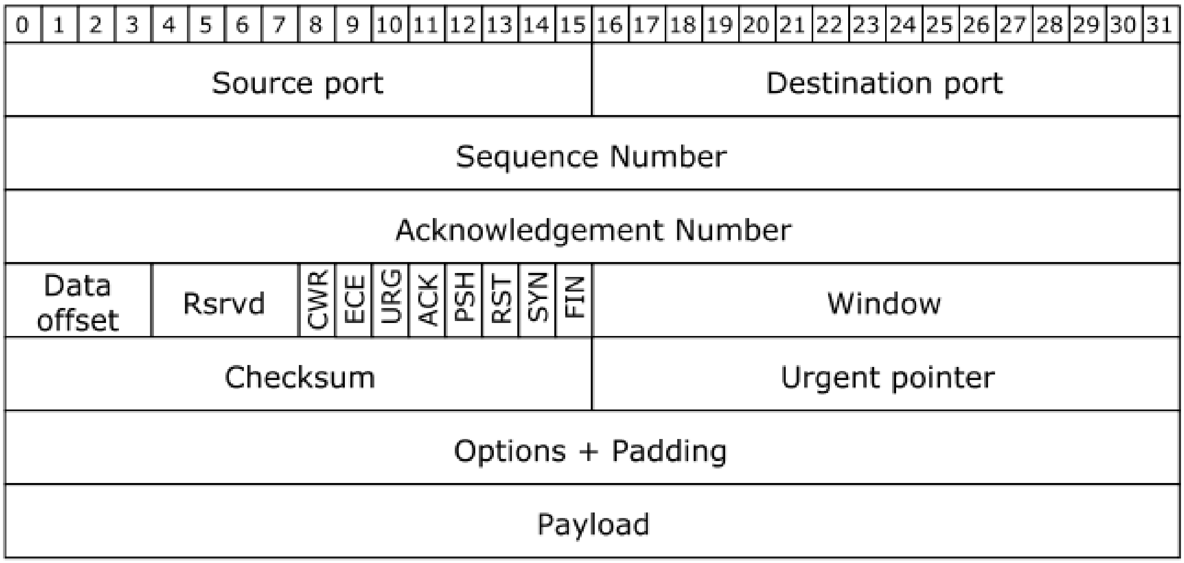

Figure 1.9: TCP header format [6].

5.3.1 Three-Way Handshake

Before application data can be transmitted over TCP, the endpoints must first establish a connection. This is necessary because TCP relies on certain features, such as reliability and congestion control mechanisms, which require both endpoints to maintain information about each data stream. A three-way handshake is initiated when establishing a TCP connection (as depicted in Figure 1.14). This ensures that both parties synchronize and agree on various connection-specific variables, including the sequence numbers of the starting packets. For security purposes, these sequence numbers are randomly selected from both sides. [43]

- *SYN*

  The client randomly selects a sequence number, x, and then sends a SYN packet. This packet may contain extra TCP flags and options.

- *SYN ACK*

  The server increases the value of x by one, generates a random sequence number y, adds its own set of flags and options, and then sends out the response.

- *ACK*

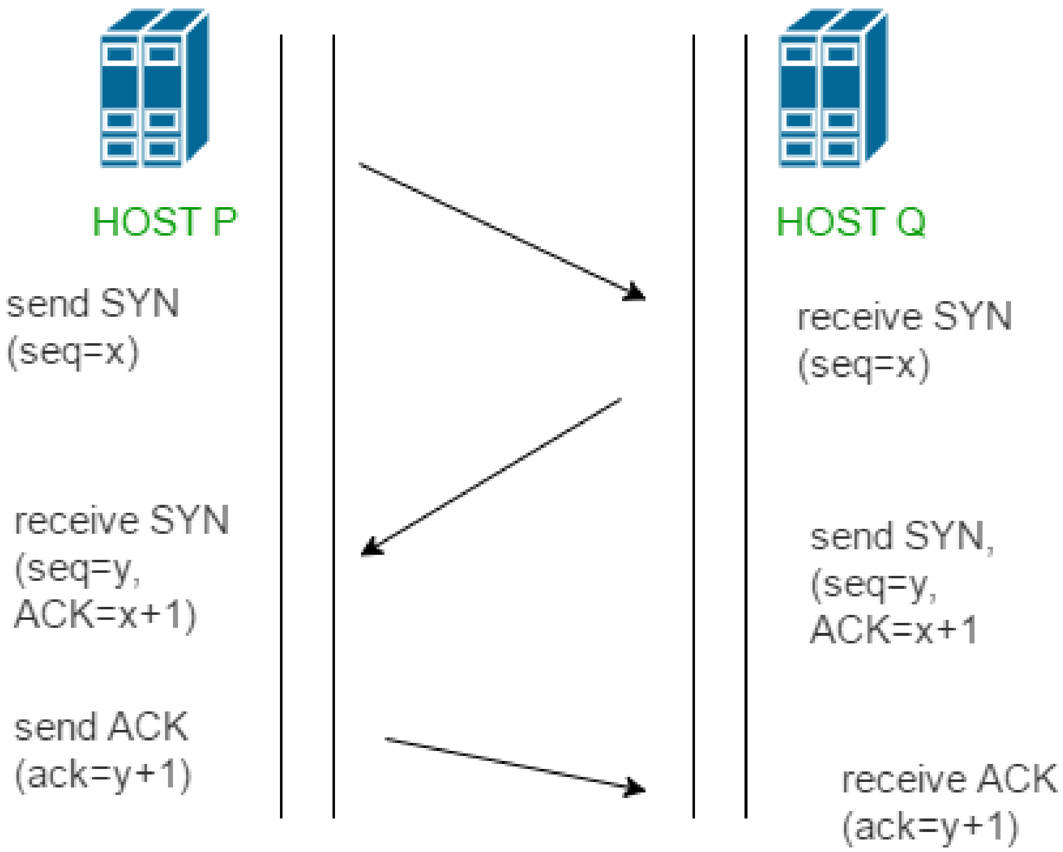

Figure 1.10: Three-way handshake [7]

The client increases the values of both x and y by one and then finalizes the handshake by sending the last ACK packet in the handshake process.

After the three-way handshake is finished, application data can be exchanged between the client and server. The client can send a data packet right after the ACK packet, whereas the server has to wait for the ACK to arrive before transmitting any data. This startup process is valid for all TCP connections and significantly impacts the performance of TCP network applications. Each new connection will experience a round-trip of latency before any application data can be sent [36].

5.3.2 TCP Sequence Number

In TCP communication, sequence numbers ensure reliable and ordered data delivery. The sequence number is a 32-bit field in the TCP header that identifies the position of a data segment within the stream of data being sent from one device to another.

In TCP, the sequence number for a segment refers to the byte-stream number of the first byte in that segment. To illustrate, let's consider a scenario where a process in Host A needs to send a data stream to a process in Host B via a TCP connection. Host A's TCP will assign a unique number to each byte in the data stream. For instance, if the data stream comprises a 500,000-byte file, and the MSS is 1,000 bytes, the first byte of the data stream will be numbered as 0. As shown in Figure 1.11, The TCP breaks down the data stream into 500 segments. The first segment is given the sequence number 0; the second is assigned 1,000; the third is assigned 2,000; and so on. The corresponding TCP segment header is updated with the respective sequence number in the sequence number field [41].

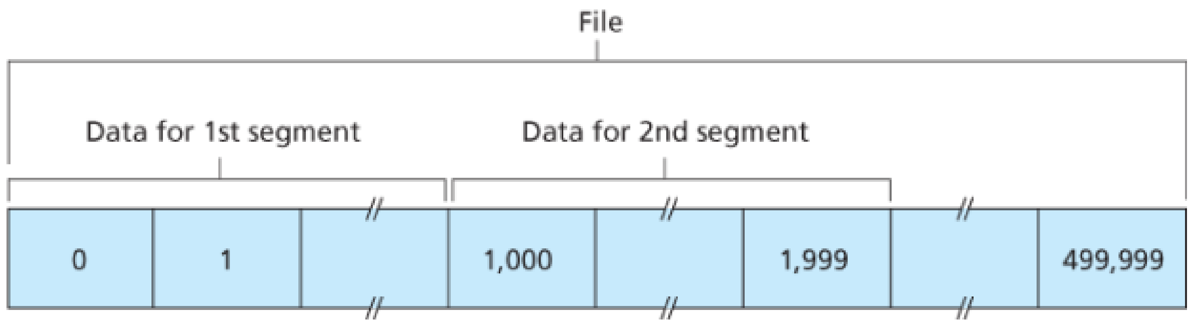

Figure 1.11: Dividing file data into TCP segments [8]

When an endpoint receives a packet, it must acknowledge that it has received and processed it correctly. This is done by sending a packet with the acknowledgment number in the header, which is the next byte expected to be received. This packet also sets the ACK control bit, which is commonly referred to as an ACK packet. The ACK packet confirms that all bytes with lower sequence numbers have been received correctly, as it carries the sequence number of the next expected byte.

A sender can determine if any packets have been lost using sequence numbers. It is considered lost if the sender doesn't receive an ACK for a packet within a reasonable time. The round-trip time (RTT) determines the time frame and sets off a retransmission timeout (RTO). In this case, lost packets can be retransmitted, and the retransmitted packets will have the same sequence numbers as the originals. This setup lets the receiver recognize whether packets have arrived out of order. Other methods, such as SACKs or delayed ACKs, are available, but we won't cover them here to keep things brief [44].

5.3.3 Congestion and Flow Control

To address the issue of a slow receiver relative to a sender, [45] introduces flow control, which slows down the sender when the receiver cannot keep up. There are two ways to handle this problem. One way, called rate-based flow control, allocates a specific data rate to the sender and ensures that data is never sent at a rate that exceeds the allocation. This type of flow control is most appropriate for streaming applications and can be used with broadcast and multicast delivery. The most popular form of flow control is called window-based flow control, used when sliding windows are employed. In this approach, the window size is not fixed but can change over time. To achieve flow control, the receiver must signal the sender about the window size by sending a window advertisement or an update. The sender then adjusts the window size accordingly.

If the window size at the sender is changed, it becomes clear how flow control is achieved. The sender can inject W packets into the network before it receives an ACK for any of them. The transfer rate is proportional to (SW/R) bits/s if the sender and receiver are sufficiently fast, the network loses no packets, and has an infinite capacity, where W is the window size, S is the packet size in bits, and R is the RTT. When an advertisement from the receiver window clamps the value of W at the sender, it limits the sender's overall rate to avoid overwhelming the receiver. This approach protects the receiver, but what about the network in between? Routers with limited memory between the sender and the receiver may contend with slow network links, causing the sender's rate to exceed the router's ability to keep up and leading to packet loss. To address this, we use a particular form of flow control called congestion control [36].

When packets arrive faster than a bottleneck link can handle, it causes network congestion. TCP uses a congestion control mechanism to avoid congestion and maximize available bandwidth. TCP's standard congestion control [46] includes four algorithms. Congestion Window (CWND) determines how much data a sender can send per Round Trip Time (RTT). In addition to congestion control, TCP features a flow control mechanism that ensures a sender does not send more data than the receiver can handle. Receive Window (RWND) limits the amount of data a sender can transmit.

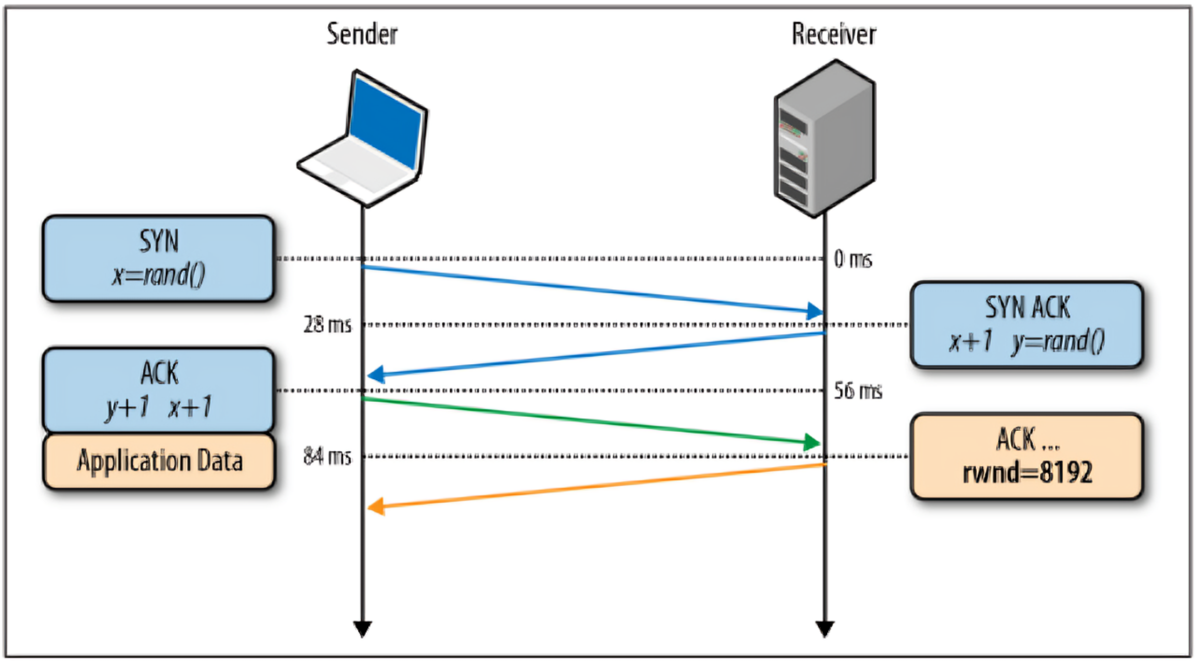

Figure 1.12: Receive window (RWND) size advertisement [9].

The sender maintains the congestion window, while the receiver maintains the receive window, which is then advertised to the sender. TCP uses the lowest value between the two windows to determine how much data can be sent. In addition, TCP employs a slow start to ramp up the sending rate gradually. It begins by sending a small number of packets and then doubles the number of packets for each acknowledgment received. This approach helps avoid abrupt congestion by allowing the network to adjust to the increasing load. A slow start threshold (ssthresh) state variable is used to determine which algorithm is used for the congestion control [47]:

- When the value of CWND is lower than ssthresh, the congestion control employs the Slow Start algorithm.
- When the value of CWND exceeds ssthresh, the Congestion Avoidance algorithm is employed.

- If CWND equals ssthresh, a slow start or congestion avoidance is possible.

According to [46], The Slow Start algorithm is used at the beginning of a TCP connection. During this time, TCP is not aware of the network's characteristics. The congestion window size is set to a conservative value in the slow start phase to avoid congestion. Each ACK received during the slow start phase increases the congestion window by one packet, doubling the congestion window per RTT. This enables TCP to ramp up the sending rate appropriately and quickly. The slow start phase continues until a loss is detected, indicating congestion or ssthresh is reached. During the Congestion Avoidance phase, the congestion window increases by one packet for every Round Trip Time (RTT) until congestion is detected. If packet loss is detected due to Retransmission TimeOut (RTO) during the Slow Start and Congestion Avoidance phases, ssthresh is set to max(lightSize/2, 2). FlightSize is the number of outstanding network packets that have not been acknowledged. The congestion window is also reduced to one packet in response to congestion.

The Fast Retransmit algorithm attempts to predict packet losses before an RTO occurs. The receiver responds with duplicate ACKs when receiving packets out-of-order, either due to re-ordering or packet loss. Out-of-order packets will have sequence numbers higher than that which the receiver expects. When replying to such packets, the receiver reuses the acknowledgment number from the previously sent ACK. This is because the next expected byte has not yet been received. Such ACKs are thus known as duplicate ACKs. Upon receiving the missing packet, the receiver may immediately ACK all received bytes until the next expected byte in sequence. The fast retransmit algorithm uses incoming duplicate ACKsm. After receiving three duplicate ACKs, the sender assumes the packet has been lost and retransmitted immediately. At this point, ssthresh is set to max(FlightSize/2, 2), and CWND = ssthresh + 3. The congestion control also switches to the Fast Recovery algorithm. The congestion window is incremented by one for each additional duplicate ACK received. Once the next ACK, which acknowledges previously unacknowledged data, has been received, the congestion window will be set to ssthresh, and the congestion avoidance algorithm can take over.

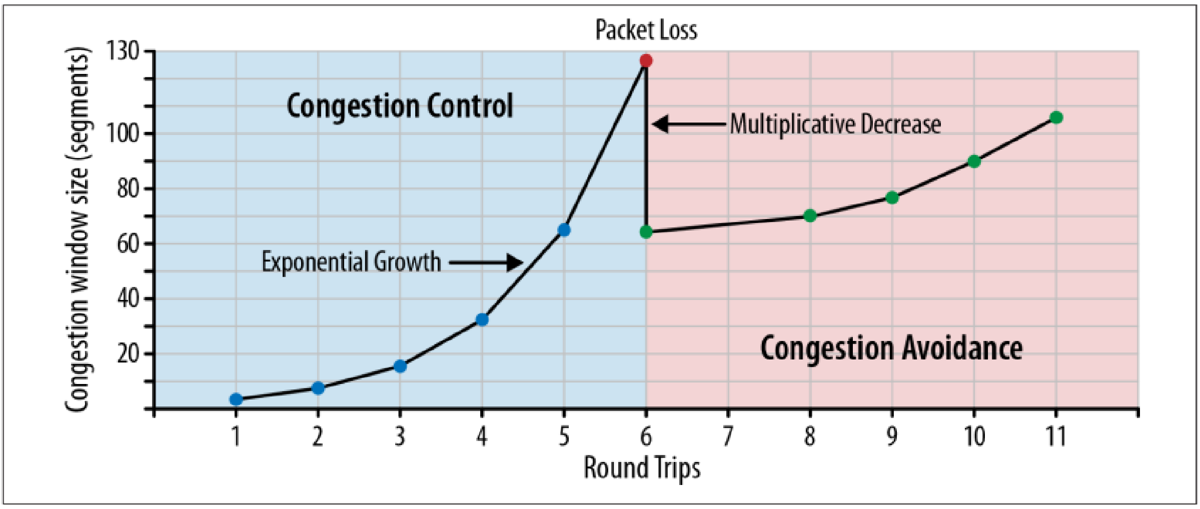

Figure 1.13: Congestion control and congestion avoidance [7]

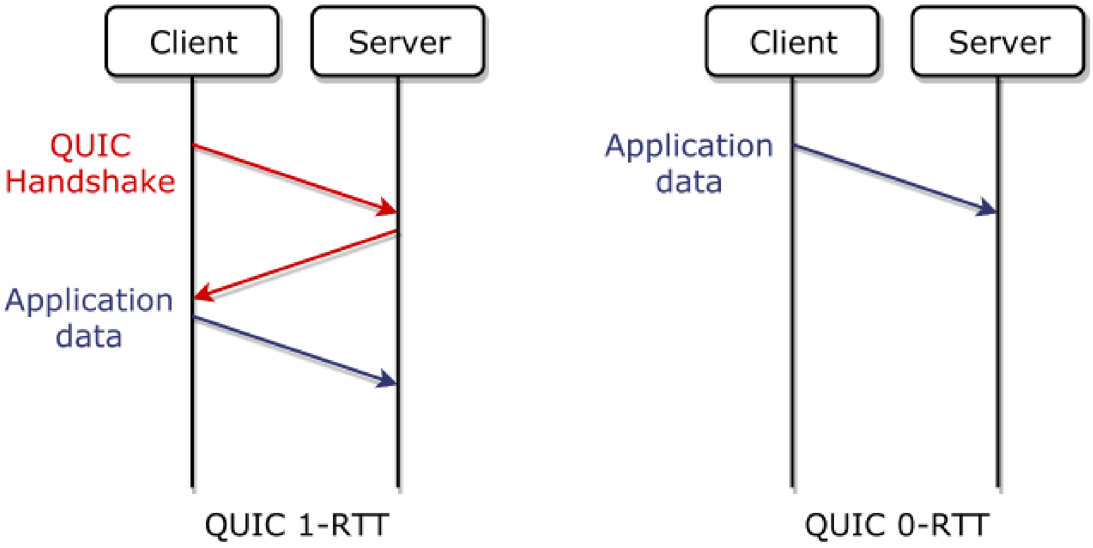

Figure 1.14: QUIC 1-RTT and 0-RTT handshakes [8].

5.4 QUIC

QUIC [10], short for Quick UDP Internet Connections, is a transport protocol that Google has proposed. Its primary purpose is to enhance the overall performance of the web. This is done by offering faster connection establishment than the traditional TCP protocol, allowing for the simultaneous transmission of multiple independent data streams, thereby preventing head-of-line blocking. Another significant advantage of QUIC is its built-in security features, designed to protect against potential security threats. QUIC is tunnelled over UDP, which allows for greater deployability. This means that QUIC packets can pass through various middleboxes, such as firewalls or network address translators, without the risk of being blocked. Overall, QUIC has the potential to significantly enhance the performance and security of the web, making it a promising option for the future. Additionally, The QUIC protocol runs without requiring changes to the endpoints' operating systems [10]. QUIC is currently undergoing standardization by the IETF [48].

5.4.1 QUIC handshakes

QUIC handshakes range from 1-RTT for first-time connection establishment to 0-RTT for connections to known peers. QUIC uses the TLS 1.3 [49] handshake for low-latency connection establishment. The transport and TLS handshakes are combined and performed using STREAM frames over a dedicated stream. QUIC provides reliability and ordered delivery for TLS handshake messages. In return, the TLS handshake carries desired QUIC transport parameters [10].

The QUIC handshake begins with a Client Hello packet, where the client selects its Connection IDs and QUIC version to start the connection. The Initial packet payload contains data used by TLS to negotiate the cipher suite and other secrets, encrypted using QUIC version one's default encryption algorithm AES-128-QCM [50]. Except for Version negotiation and Stateless reset, all packets are encrypted by an Authentication

Encryption with Associated Data [51] (AEAD) algorithm [52].

Next, when the server receives the Client Hello packet, it checks the version field in the header. The version field is one of the headers in QUIC that are present in every version of QUIC. If the given version is not supported, the server creates a version negotiation packet and sends it to the client. If the version is supported, the server answers with a Handshake packet containing the Server Hello, and thus, the handshake is finished. However, the client must complete the handshake by receiving the Server Hello packet. Once received, the handshake is complete.

If a client connects to a server with no prior connection, the handshake can be completed within 1-RTT. By caching information from this connection, such as a pre-shared key (PSK), subsequent handshakes can be reduced to 0-RTT, allowing the client to send application data immediately after the handshake packet. TLS offers a handshake variation that initiates a new key exchange, which QUIC can use to validate a client's claimed ownership [49, 10].

5.4.2 Connection Megration

In the QUIC protocol, connections are identified using a 64-bit Connection ID in the QUIC header. This Connection ID enables endpoints to switch from one network path to another without needing to close the connection if the quality of the current path deteriorates. When an endpoint receives packets from a new source address but with a familiar Connection ID, it can promptly send packets via the new path. However, before fully committing to the switch, the endpoint must verify that it can receive packets on the new path. As a result, the endpoint must restrict the rate at which it sends data until the new source address is validated. To validate a new path, QUIC uses a PING frame containing a validation token sent to the peer via the new remote address. Upon receiving a PING frame, the peer is required to respond with a PONG frame containing the same token. If the validation fails, the endpoint terminates the connection. On the other hand, if the validation is successful, the endpoint can gradually increase the rate at which it sends data. Additionally, the endpoint may refresh any validation tokens it has issued to its peer. Moreover, an endpoint can periodically use empty PING frames, which do not contain a validation token, to check its peer's reachability. These empty PING frames are acknowledged as usual without transmitting a corresponding PONG frame. Since connection migration can result in packet reordering due to irregular path conditions, a host must be designed to receive packets from multiple source addresses simultaneously. Consequently, the path utilized to transmit data is determined based on which path has received the highest packet number.

5.4.3 Stream Multiplexing

It is common for applications to utilize TCP's single-byte-stream abstraction to multiplex data units. However, this approach can result in head-of-line blocking as TCP delivers data sequentially. To address this issue, QUIC supports multiple streams within a connection. In the event of a lost UDP packet, only the streams whose data was carried in that packet are impacted, while subsequent data received on other streams can continue to be reassembled and delivered to the application. This ensures that the overall performance and reliability of the application are maintained, even in

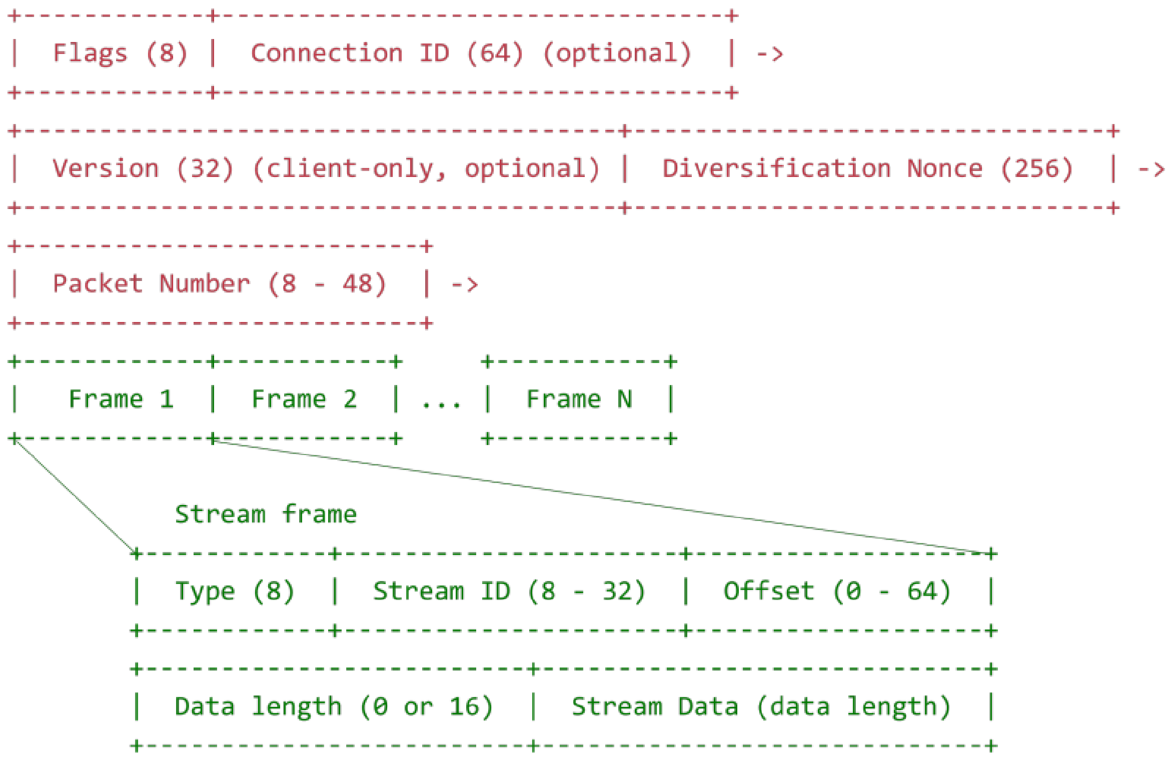

Figure 1.15: Structure of a QUIC packet, as of version 35 of Google's QUIC implementation [10].

the face of network issues [10].

The QUIC protocol can achieve multiplexing by utilizing STREAM frames to transport application data. These frames contain a Stream ID field, which is critical in determining the specific stream to which the frame belongs. It's worth noting that a single QUIC packet can carry various frames, including STREAM frames from different streams. A QUIC packet should contain as many frames as possible to optimize efficiency. However, it's important to remember that all streams included in a packet will be blocked due to a loss. It's worth mentioning that QUIC doesn't handle losses on a packet level. This means that lost packets won't be retransmitted. Instead, lost frames can be retransmitted by bundling them into new outgoing packets. This approach implies that frames are independent of the packets carrying them [53].

A QUIC packet consists of a common header followed by one or more frames, as depicted in Figure 1.15 where red is the authenticated but unencrypted public header and green indicates the encrypted body. This packet structure is evolving as QUIC gets standardized at the IETF. Stream data is encapsulated in one or more stream frames, and a single QUIC packet can carry stream frames from multiple streams. This approach reduces the number of packets required to transmit the same amount of data, thus reducing latency and improving the overall speed of data transfer [10].

The prioritization of streams can significantly affect the performance of an application. In the case of QUIC, the application is responsible for indicating the preferred order in which stream frames should be transmitted. The first stream, Stream 0, is always given priority over other streams until a secure connection has been established. It is worth noting that this prioritization approach helps to ensure that the connection is established correctly before other streams are transmitted. As such, it plays a crucial role in optimizing the performance of applications that rely on QUIC [53].

5.4.4 Reliability

The QUIC protocol's loss detection and congestion control mechanisms resemble TCP's but with some notable distinctions. Like TCP, each packet in QUIC contains a sequence number in the header. The receiver sends ACK frames to confirm the receipt and processing of data. If the receiver takes too long to respond with an ACK frame, a RTO event occurs. An ACK frame can cumulatively acknowledge multiple ranges of packets. It includes a Largest Acknowledged field, indicating the highest packet number received, regardless of any gaps. Additionally, QUIC ACK frames feature an ACK Block Section that can encompass multiple blocks, representing acknowledged packet ranges. QUIC does not retransmit lost packets. When a loss is detected, the frames from the lost packet are bundled into a new packet with a fresh sequence number. Therefore, sequence numbers do not repeat within a connection. This also means that the packet's sequence number does not determine the order of the data within the packet. The order of data in a stream is determined by the Offset and Length fields within the STREAM frames [10].

5.4.5 Congestion control

As is the case in TCP and other transport protocols, QUIC incorporates congestion control to detect when a network device cannot handle all of the incoming traffic. When this occurs, the device will begin to drop packets on the path between the two endpoints. By detecting congestion, the sender can adjust the pace at which it sends data. Similar to TCP, QUIC utilizes a congestion window, which refers to the amount of data currently in flight and has yet to be acknowledged. This feature helps to ensure that data is transmitted efficiently and effectively between the endpoints while avoiding network congestion.

The QUIC protocol does not rely on a specific congestion control algorithm. The GO implementation of QUIC [54] uses the CUBIC congestion control algorithm by default. CUBIC is the TCP's congestion control, specifically designed for high-speed networks and used by the TCP implementation in Linux as a default congestion control mechanism. CUBIC aims to provide a fair and balanced approach to standard TCP connections. It achieves this by modifying the congestion window growth function to be more scalable than the standard congestion control used in TCP. This approach ensures that the congestion control mechanism remains adequate and more efficient, providing a better experience for users. By using CUBIC in the quic-go implementation of QUIC, the protocol can maintain high-speed connections while ensuring the fairness of the congestion control mechanism [55, 56].

5.4.6 Flow Control

Flow control is a crucial aspect of network protocol design, and QUIC takes it seriously by implementing it at multiple levels. In particular, QUIC implements flow control at the connection, stream, and stream ID levels. This approach has several advantages over TCP, which only provides flow control at the connection level. With connection flow control, TCP ensures that the receiving side of the connection does not become overwhelmed by the sender. However, when dealing with multiple streams, this approach can have drawbacks. To mitigate this issue, QUIC incorporates flow control at both the stream and connection levels. Doing so ensures that no single stream can monopolize the connection's resources. This means that senders cannot send more data on a stream than the stream and connection level flow controls allow. It's worth noting that flow control only applies to stream frames and not control frames like Max Data and Ack frames. Overall, the multi-level flow control implemented by QUIC helps ensure that data is transmitted efficiently and fairly between different streams on the same connection.

To further elaborate on the concept, let us consider an instance where a student is viewing a lecture video and simultaneously downloading the relevant resources. In this scenario, the video and the resources are transmitted through the same QUIC connection. As the video needs to be streamed in real time, the data stream carrying the video data is prioritized over the transmission of resources. However, suppose the student pauses the video to wait for the resources to finish downloading. In that case, the client may choose to throttle the data stream being used for video transmission to prevent a large amount of data buffering. Once the resources have been successfully downloaded, the student can then resume the video lecture, which results in the client sending a MAX_STREAM_DATA frame to continue receiving data for the video transmission.

Furthermore, QUIC also implements flow control on the level of stream ID to prevent either endpoint from opening an excessive amount of streams without checking if the other endpoint can handle it. This is imperative to avoid situations where the recipient opens overwhelming streams. For instance, if a client initiates 5000 streams to request different resources, and the maximum amount of data allowed on the connection is 50,000 bytes. The server can only transmit 10 bytes on each stream to distribute the available memory evenly. In addition, the server must maintain a state for each stream, which requires memory that could be used for other purposes. Hence, the recipient can set the maximum number of streams they want to keep open to prevent such scenarios. It is noteworthy that there are several types of stream IDs. When an endpoint receives a MAX_STREAM_ID frame, the first step is to verify whether the given stream ID is a unidirectional stream ID or a bidirectional frame. Additionally, the endpoint must check if the given stream ID is larger than the current maximum stream ID. If it is not, the frame must be ignored. It is important to note that neither the connection nor the stream and stream ID flow control can decrease its maximum given value by a new frame. Once these checks are completed, the maximum stream ID the frame recipient can use is adjusted accordingly [57].

5.4.7 Close Connection

There are three distinct ways in which a QUIC connection can be terminated. Firstly, it may occur as a result of a timeout, which refers to an instance where no data is exchanged between the endpoints within the allotted time frame. Alternatively, a connection may be immediately closed, which happens when one of the endpoints sends a signal to terminate the connection. Finally, a stateless reset can also terminate a QUIC connection, which occurs when a packet is received with an unknown connection

ID.

In the QUIC handshake process, the endpoints negotiate an appropriate idle timeout value. This value specifies the duration of time that can elapse without any data transmission or reception. If the idle timeout value is exceeded, the connection between the endpoints is automatically terminated.

An immediate close can be initiated by sending a closing frame to terminate a connection gracefully. Once the closing frame is sent, the sender enters the closing state. During this state, the sender responds to all incoming packets with another closing frame. If an end-point receives a closing frame, it may choose to respond with a closing frame of its own, but it is not required. If an idle timeout occurs or an end-point receives a closing frame, it enters the draining state. While in the draining state, an end-point cannot transmit any packets. If an end-point in the closing state is notified that its peer is in the draining state, it may transition to the draining state as well. The closing and draining states last for three times the RTO values. This ensures that the termination process is thorough and gives both end-points enough time to handle any remaining data before the connection is completely closed.

Another method for terminating a connection immediately is the stateless reset, which is similar to TCP's RST. To execute a stateless reset, one of the endpoints involved in the connection will send a packet containing a stateless reset token derived from the QUIC handshake.

6 Conclusion

In conclusion, the rapid advancement of vehicular networks, driven by the proliferation of connected vehicles and intelligent transportation systems, is poised to transform how we manage and experience road transportation. As VANETs become increasingly prevalent, the challenges associated with video streaming, such as ensuring reliable transmission and maintaining high-quality service, become critical focus areas. By exploring advanced video compression techniques like HEVC and transport protocols like QUIC, this study aims to address these challenges, paving the way for more efficient and robust video communication in vehicular environments. The insights gained from this exploration will contribute to the ongoing development of VANETs, enhancing their potential to improve road safety, optimize traffic management, and, ultimately, revolutionize the transportation landscape.

CHAPTER 2

Comparative Analysis of different QUIC extensions for Video Streaming in VANETs.

1 An Overview of the Multipath Transport Protocols

1.1 Introduction

In the rapidly changing realm of vehicular networks, one main challenge is ensuring that video streaming is efficient and reliable. This is particularly difficult due to the unpredictable nature of these networks, characterized by frequent changes in connectivity, high mobility, and varying network conditions. One crucial factor that directly affects the quality of video streaming in VANETs is the choice of transport protocol. Transport protocols are vital in determining how data packets are transmitted across the network, impacting aspects such as latency, packet loss, and overall QoS.

This section delves into a comparison between multipath and single-path transport protocols within the context of streaming video in vehicular networks. Single-path protocols, commonly used in various network settings, send data packets along a predetermined route. While straightforward, this method may struggle to adapt to the rapidly changing conditions of VANETs, potentially resulting in interruptions in video transmission. On the other hand, multipath transport protocols provide a more robust solution by leveraging multiple paths simultaneously or dynamically switching between paths based on real-time network conditions. This strategy can enhance fault tolerance, optimize bandwidth utilization, and reduce latency, all of which are critical for maintaining high-quality video streaming in the demanding environment of vehicular networks.

This section will explore the core principles underlying multipath transport protocols, exploring their strengths and limitations. Through a comprehensive comparative analysis, we aim to pinpoint the scenarios where each protocol excels and how they can be fine-tuned for video streaming in VANETs. This analysis aims to offer valuable insights for researchers and engineers working to enhance the performance and reliability of video transmission in next-generation vehicular networks.

In the dynamic landscape of modern digital communication, the quest for efficiency, reliability, and speed has led to the development of groundbreaking technologies. One innovation that stands at the forefront of this endeavor is multipath transport protocols. These protocols represent a transformative shift in the way data traverses the intricate web of interconnected devices and networks, offering a profound solution to the challenges of today's data transmission needs. At its core, multipath transport protocols are communication techniques that enable the simultaneous use of multiple network paths to transmit data packets between two endpoints. In essence, they embody the philosophy that the whole is greater than the sum of its parts, leveraging the collective capacity of diverse network routes to deliver data with enhanced robustness, resilience, and performance. To fully appreciate the significance of multipath transport, one must delve into the complexities of modern data transmission.

In the digital age, our reliance on connectivity has reached unprecedented heights. Whether streaming high-definition video, conducting critical financial transactions, or remotely controlling autonomous vehicles, the uninterrupted data flow is paramount. However, the digital ecosystem is riddled with potential obstacles, from network congestion and outages to latency and packet loss. Multipath transport protocols emerge as a beacon of hope in this landscape, offering a multifaceted solution to address these challenges. Imagine a scenario where data is a river, and the traditional single-path transmission is a narrow channel susceptible to blockages and turbulence. In contrast, multipath transport protocols are like a delta, where numerous interconnected streams flow freely, adapting dynamically to the changing terrain. This metaphor underscores the inherent resilience and adaptability of multipath systems. These protocols are not confined to theoretical concepts; they have been harnessed and refined in the real world. They are integral to the backbone of the Internet and have found applications in various domains, such as cloud computing, content delivery networks, and mobile communication. Multipath transport has the potential to revolutionize how we experience the digital realm, enhancing the speed and reliability of our interactions while ensuring our connectivity remains unbroken in the face of adversity.

1.2 Multipath communications: Incentives

Initially, the Internet was designed to have two network connections, which inherently supported multipath transmission. However, in the early stages, computers with multiple network interfaces were not a priority in the design process, and only routers were equipped with several physical network interfaces. Over time, the Internet has undergone significant development and transformation. Nowadays, most servers are equipped with more than one network interface, which has led to the proliferation of network resources in the server domain. This, in turn, has driven the adoption of multipath transmission in data center networks. The availability of multiple network interfaces in servers has enabled them to use multiple paths to transmit data, thereby improving network performance, reliability, and resilience. With the advancement of technology, the use of mobile devices equipped with cellular and WiFi interfaces has increased significantly in the consumer electronics domain. These devices, such as smartphones, have brought many multi-homed hosts onto the Internet. In fact, even desktop computers nowadays often have both WiFi and wired connectivity options, as depicted in Figure 2.1. For instance, a laptop can use its interfaces A1 and A2 to exchange data over several paths ("subflows") with the server. This has led to a more complex network architecture with multiple paths for data transmission, thereby improving the reliability and performance of Internet connections.

As per Agarwal, Chuah, and Katz's research [58], it was observed that there is a significant mismatch between single-path transport and the vast multitude of network paths that are available. According to their findings, at least 60% of the stub domains are multihomed to two or more providers, which makes it challenging to choose the

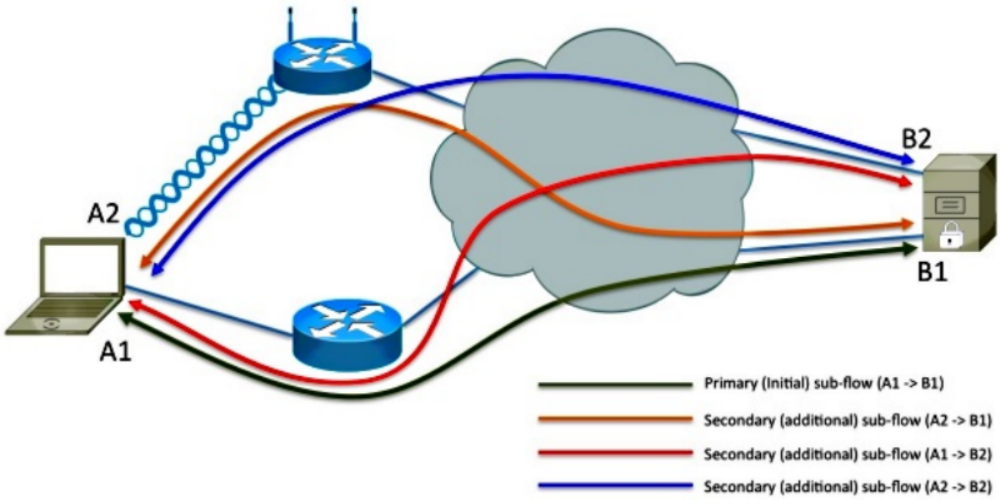

Figure 2.1: An example of server and client multihoming (Source: Cisco documentation) [11].

most efficient network path. Even when Border Gateway Protocol (BGP) reduces the advertised path diversity by selecting the best route towards the destination, there is still a possibility that flows may follow different paths, as mentioned in [59]. This is where multipath capability comes into the picture to enhance end-to-end communication performance and resilience of multi-interface devices. Technological advancement has brought about benefits from multipath transmission. Below are some of the major benefits [60]:

- Reliability Multipath transmission is a technique that can significantly improve the reliability of data transfer. This is because when multiple paths are available, the data can be transmitted through all of them simultaneously, thereby increasing the chances of successful delivery. In the event of a low path performance or partial path failure, the additional paths can ensure that the connection remains alive. One of the most notable examples of multipath transmission is MPTCP (Multi-Path Transmission Control Protocol), which supports a "break before making" scenario. In other words, if all paths lose connectivity, MPTCP can buffer the packets from the application while reestablishing connectivity on another path and then resume the connection. This can be particularly helpful in situations where the user cannot predict connectivity loss, such as when they are on the move. Overall, the use of multipath transmission is an effective method to enhance the reliability of data transfer, and MPTCP's break before making the scenario is a prime example of how this technique can be used to ensure the connection remains alive even in the face of partial path failure.

- Bandwidth aggregation

  One of the major advantages of multipath transmission is the ability to leverage multiple flows initiated from different TCP/IP layers to aggregate bandwidth. By doing so, the overall bandwidth can potentially increase by a factor equal to the number of available paths. This results in a significant improvement in the throughput experienced by the multi-homed device. If efficient bandwidth aggregation can be achieved, the device can benefit from a much better network performance, which can be especially useful in scenarios where high bandwidth is critical, such as in multimedia streaming or real-time applications.

- Confidentiality When a data flow is split and forwarded on different paths, it becomes more challenging for a malicious attacker to reconstruct it. This is because the attacker must intercept several probes to capture all traffic or access a shared hop on all paths. By splitting the data flow, it creates a layer of security that makes it harder for attackers to gain access to sensitive information. This technique is especially useful when there is a high risk of cyber attacks or data breaches. By taking these security measures, organizations can better protect their data and safeguard against potential threats.
- An alternative vision to Resource Pooling

  Engineers have designed several mechanisms, such as load balancing, statistical multiplexing, and failure resilience, over the years in order to increase reliability, flexibility, and efficiency. These approaches can be considered Resource Pooling (RP) approaches as they try to make a collection of resources behave like a single pool of resources. These different mechanisms are not perfect: as an example, it has been demonstrated that injecting more specific prefixes to support multihoming does not scale [61]. Also, they generally require a few seconds to recover from failures (e.g., hard handover, routing convergence). Instead of handling per path resource independently, a revised multipath vision of the RP principle advocates harnessing the responsiveness of multipath-capable end systems, making improved use of multiple path resources by allowing separate paths to act as if they were a single large resource. This approach solves most of the previously mentioned problems. It is a significant step towards a practical multipath-aware end system, especially when joint congestion control algorithms were introduced to offer both RP and TCP-friendliness features.

In short, the demand for network resources in the server and mobile device domain and the drive for technological advancements have made multipath transmission a hot topic in recent years.

1.3 Challenges

Multipath communications offer various benefits, such as increased network reliability and reduced latency. However, despite these advantages, their deployment can be challenging due to several factors outlined below. Additionally, some features that are presented as incentives may also pose obstacles to their adoption, depending on one's perspective. For instance, the ability to evade surveillance from a single point, which we previously referred to as an enhancement of confidentiality, may be perceived as a threat from a company's standpoint. This highlights the need for a balanced approach that considers the various implications of multipath communications for all stakeholders involved [62, 63].

- Deployment concerns Deploying new protocols in the current Internet is a complex task that poses challenges across many areas, including multipath communications. Despite these challenges, there have been notable efforts to design and implement protocols such as MPTCP and MPQUIC to address these issues. However, the difficulty of deploying these protocols has been a significant barrier, and it has influenced their design to a great extent. One of the challenges of deploying new protocols is the lack of incentives for upgrading software and hardware. For instance, it took a long time to deploy Internet Protocol version 6 (IPv6) due to the low incentives and the perceived risks involved in upgrading. Even protocols that should be easier to deploy, such as Stream Control Transmission Protocol (SCTP) because of their end-to-end design, face similar challenges. This is because middleboxes such as Network Address Translation (NAT) and firewalls are prevalent and can modify or drop suspicious packets that do not conform to standard protocols such as UDP or TCP. In summary, deploying new protocols in the current Internet is a complex process that requires a multifaceted approach, including addressing the challenges of upgrading software and hardware and overcoming the hurdles of middleboxes.

- Heterogeneous Networks

  When we talk about heterogeneous networks, we refer to networks that consist of paths with varying round-trip times (RTTs) or loss rates. In certain network protocols that are designed to deliver data sequentially, sending successive data segments on different paths can lead to out-of-order (OOO) arrivals. This can be problematic for window-based protocols, as the system needs to buffer these out-of-order packets. Depending on the size of the buffer, the system may experience Head-of-Line blocking (HoL), which can significantly decrease throughput. In such scenarios, using multipath protocols may not always be helpful and may instead lead to decreased throughput. Therefore, in situations where the network paths are heterogeneous, it is important to carefully consider the use of multipath protocols and their potential impact on the system's throughput.

- Pareto-optimality Multipath protocols aim to provide users with enhanced transport options. For such protocols to be effective, they need to be Pareto-optimal [64], meaning that the benefits gained from using multiple paths should not come at the cost of a worse experience for any single user. One commonly used metric for measuring the quality of communication is throughput. However, in the case of transport multipath protocols, the concept of fairness plays a critical role in ensuring Pareto optimality. The current notion of fairness dictates that each TCP flow should receive an equal share of the available bandwidth. However, in the case of multipath protocols, simply utilizing multiple flows can result in an "unfair" bandwidth distribution at the bottleneck. For example, if there are n TCP flows, they may receive approximately n times the throughput of a competing Single Path TCP (SPTCP) flow. To ensure compliance with TCP fairness, multipath transport protocols should consume the same amount of router buffer as an SPTCP flow when passing through bottlenecks. Unfortunately, detecting shared bottlenecks is a challenging problem, and as a result, the current approach taken by MPTCP is to be conservative and assume that all paths share a bottleneck. While this conservative approach helps to maintain fairness, it also makes achieving the goal of aggregation more difficult. As a result, it is crucial to strike a balance between achieving high throughput and ensuring fairness for all users when designing multipath transport protocols.

- Resource Consumption When multiple paths are utilized, it can result in higher resource consumption since there are more states to keep track of. Therefore, to compensate for this increase in resource usage, it's essential to consider another metric such as throughput. However, it's crucial to note that enabling multipath for short communications can be counterproductive regarding energy efficiency, as shown in a study conducted by Nikravesh et al. [65]. In a mobile configuration, when a smartphone powers on the energy-hungry cellular interface to enable multipath right before the end of the connection, it increases energy consumption without any significant gain in throughput. Hence, it's essential to monitor buffer size carefully, as it's directly linked to the number of paths one can use.

1.4 Multipath Transmission Control Protocol (MPTCP)

Multipath Transmission Control Protocol (MPTCP) [66] extends the traditional TCP protocol, allowing concurrent multipath transfer functionality. This means that MPTCP can open and use multiple paths to transfer data concurrently. This is achieved by allowing TCP to open subflows, essentially additional connections that operate in parallel with the primary connection. The benefit of this approach is that it allows MPTCP to take advantage of multiple available paths, leading to improved network performance, better reliability, and faster transfer speeds.

Each open subflow operates similarly to a traditional TCP connection, but many subflows can be aggregated over a single MPTCP connection. MPTCP can utilize all available network resources to optimize data transfer. To accomplish this, MPTCP uses what is known as 'faux-headers,' which are implemented using TCP options. These headers enable each subflow to carry its own data and metadata, which can then be used to optimize the overall data transfer across multiple paths. Overall, MPTCP is a powerful and flexible protocol that can improve the performance and reliability of applications that rely on TCP for data transfer. Figure 2.2 illustrates a comparison between TCP and MPTCP.

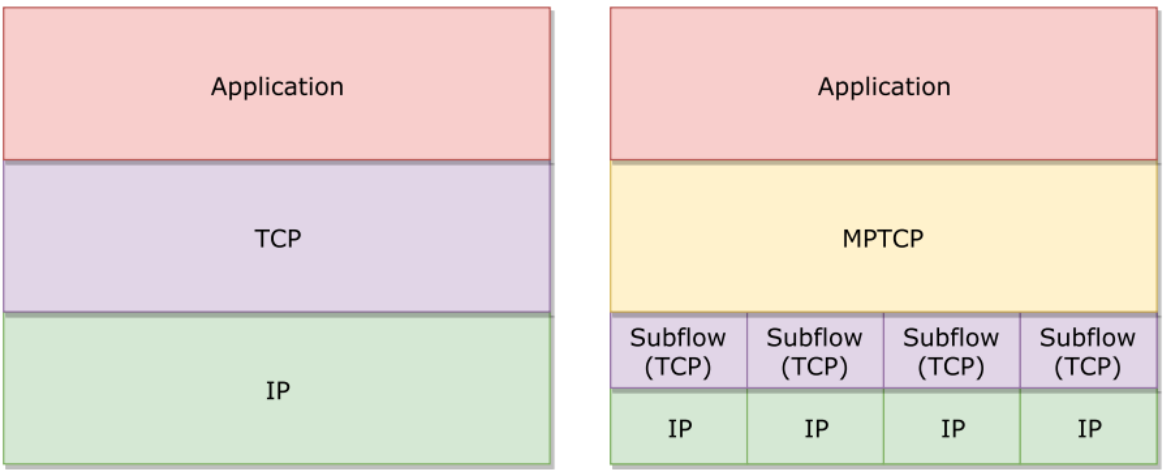

Figure 2.2: TCP (left) and MPTCP (right) protocol stacks. MPTCP aggregates multiple TCP subflows over different paths [8].

When establishing an MPTCP connection, the process is initiated on a single subflow using a three-way handshake that is similar to TCP. However, unlike TCP, the packets used for the MPTCP handshake carry an additional option called MP-CAPABLE. This option serves the purpose of verifying that both endpoints have the capability to support MPTCP. In addition to verifying compatibility, the MP-CAPABLE option is also utilized to exchange tokens, which can be used to authenticate additional sub-flows. This authentication process ensures that only authorized sub-flows are able to join the MPTCP connection, maintaining the overall security of the communication. By exchanging tokens, each endpoint can confirm that the other endpoint is legitimate, preventing any unauthorized parties from accessing the connection [67, 66].

Path Management

After successfully establishing an MPTCP connection, the endpoints may discover new paths and open up additional subflows. There are two ways for a host to signal that it owns an additional IP address: either by establishing a subflow directly over that path or by learning about the additional addresses of its peer through address signaling. If a host wants to advertise additional addresses, it can do so by transmitting packets with the ADD_ADDR option over an already established subflow. Conversely, previously advertised addresses can be removed using the REMOVE_ADDR option. This feature allows for efficient and dynamic allocation of network resources, enabling MPTCP to achieve higher throughput and better reliability than traditional TCP.

When establishing a subflow, the process starts by sending a SYN packet that contains the MP_JOIN option. The MP_JOIN option holds the token that was established during the initial subflow handshake. To ensure security and prevent replay attacks, the token is hashed and sent with a Nonce. The Nonce is a random number that can only be used once, ensuring that the token cannot be used again in subsequent subflows. This process helps maintain the integrity and security of the subflow communication.

The ADD_ADDR, REMOVE_ADDR, and MP_JOIN options in the context of Multipath TCP carry an Address ID. This unique identifier is used to identify the source address of the packet. It is particularly useful when a middlebox, such as a NAT, changes the original source address in the IP header. In this situation, the Address ID can be used for address removal without requiring knowledge of the true address. Additionally, the Address ID can be used to prevent end-points from setting up duplicate subflows. By using the Address ID, each subflow can be identified and distinguished from one another, ensuring that the connection remains stable and reliable even in complex network topologies.

Reliability Ensuring MPTCP's reliability is a crucial aspect of its functionality. One simple approach to achieve this is using TCP sequence numbers to indicate packet sequence, irrespective of the subflow through which the packet is sent. However, this method has certain vulnerabilities. Middleboxes may tamper with sequence numbers or block traffic with gaps in the sequence space, thereby posing a threat to the reliability of MPTCP. As a result, it is essential to employ additional mechanisms to guarantee reliability and mitigate the impact of such vulnerabilities.

Thus, in MPTCP, every subflow has its own sequence number space. This means that each subflow operates independently, and the packets sent on a specific subflow are numbered using a unique sequence number. Additionally, each packet also includes a data sequence mapping that maps the payload to the sequence of bytes for the entire MPTCP connection. Since the ACKs on a subflow can only acknowledge packets on that specific subflow, MPTCP uses Data Acknowledgements (Data ACK) to cumulatively ACK packets on a connection level. This means that the Data Sequence Signal (DSS) option is used to carry out data sequence mapping and Data ACKs. This way, the receiver can reconstruct the original data stream by using the data sequence mapping and acknowledge the packets on a connection level through Data ACKs.

Multipath congestion control in TCP/IP

Multipath congestion control algorithms are used in MPTCP to enhance throughput and facilitate quick adaptation to changing network conditions. One such algorithm is the Uncoupled algorithm, which creates subflows with individual congestion windows and independent window management. Although this design provides the expected benefits, it can be overly aggressive towards single path connections, raising serious TCP-frienemphasizing concerns and improving fairness. To address this issue, EWTCP was developed. EWTCP [68] splits traffic evenly among subflows to cumulatively grasp the same share of resources as a regular TCP connection. However, this approach often results in the underutilization of available network resources because the proportional management of the subflows ignores the unique properties of the dissemination paths. Overall, while multipath congestion control algorithms can boost throughput and improve network flexibility, it is crucial to consider the impact on singlepath connections and ensure fair resource allocation across all connections.

The Coupled algorithm [69] is a notable design that focuses on being TCP-friendly and effectively distributing traffic across available paths, emphasizing balancing congestion levels and increasing resource utilization. This algorithm treats the available paths as a pool of resources and seeks to balance the congestion level across all available paths, ultimately directing traffic towards the least congested path. However, while it offers some benefits, pushing all traffic towards the least congested path can lead to performance degradation in cases where the chosen path is not well-matched to the traffic, and it can also suffer from poor responsiveness to network changes. Despite these limitations, the Coupled algorithm remains an important and influential contribution to the field of network traffic management.

The Linked Increase Algorithm (LIA) [70] is a traffic management technique that was designed to address two important issues - TCP-friendliness and responsiveness. Its main objective is to enhance load balancing, similar to the Coupled approach, by directing traffic to the least congested path. However, LIA also introduces an aggressiveness parameter to keep a moderate amount of traffic flowing through the more congested paths, ensuring that the system remains responsive. This parameter is based on two main equilibrium conditions: firstly, LIA balances the congestion window increases and decreases at a steady state, ensuring system stability. Secondly, it equalizes the resource shares of MPTCP and TCP in the bottleneck link, making it TCP-friendly. In summary, LIA is an efficient and dynamic traffic management technique that ensures both load balancing and responsiveness while prioritizing stability and TCP-friendliness. While favoring the least congested path, LIA does not push traffic exclusively there, thus penalizing the overall network resource utilization under certain conditions.

The Opportunistic Linked Increase Algorithm (OLIA) [64] was designed as an extension of the LIA algorithm to improve resource pooling while still maintaining high responsiveness. The algorithm achieves this by increasing the congestion window of subflows that have high transfer rates but relatively small windows. Additionally, OLIA introduces minimal probing traffic over the worst paths to ensure sufficient responsiveness. However, [71] indicates that OLIA may not respond well in case of abrupt load changes, which can lead to performance issues. To address this limitation, the authors of a research paper propose the Balanced Link Adaptation (Balia) algorithm.

Balia is a generalization of existing algorithms that effectively balances friendliness, responsiveness, and window oscillation, making it a more suitable alternative to OLIA.

Weighted Vegas (wVegas) [72] is a congestion control algorithm that is inspired by TCP Vegas. Unlike other congestion control proposals that use time-out timers, wVegas uses queuing packet delay to detect congestion. This delay-based approach allows wVegas to quickly shift traffic in response to changes in network load, making it a valuable contribution to the field of congestion control. However, there are some challenges associated with using wVegas. One of these challenges is tuning the algorithm's sensitivity, which can be difficult. Additionally, the investigation of wVegas's behavior is not yet complete, and there are still some questions regarding how it handles RTT variation in the case of rerouting. Despite these challenges, wVegas is a promising congestion control algorithm that has the potential to improve network performance in a variety of settings.

Flow Control

When it comes to flow control, MPTCP operates in a similar way to TCP, utilizing a shared receive window that is used by all subflows. This shared receive window ensures that the subflows are not constrained by their own individual receive windows, which enables MPTCP to make full use of its available paths. By using a single receive window for the entire connection, MPTCP can use its subflows in any way it desires as long as the receive window limit for the whole connection is not exceeded. While it is possible to use unique receive windows for each subflow, this approach would not provide any actual benefits and would only serve to throttle the faster paths unnecessarily. Therefore, MPTCP's use of a shared receive window allows it to maximize its potential and achieve optimal performance.

Packet Scheduling When multiple paths are being used for data transmission, it becomes necessary to have a packet scheduler in place that can efficiently determine which path a packet should be sent through. In order to cater to this requirement, the MPTCP implementation in the Linux kernel [73] offers a modular scheduler infrastructure that provides users with a range of different packet schedulers to choose from. This section focuses on the standard schedulers that come pre-installed with the Linux implementation and provides a detailed analysis of each of them. Additionally, the thesis will also cover the usage and testing of several other packet schedulers, such as Peekaboo [74], ECF [75], and Blest [76], in order to provide a comprehensive comparison of their performance and effectiveness.

One of the simplest ways to schedule data transmission across multiple subflows is to use the Round-Robin (RR) scheduler. This scheduler works by allowing each subflow to take turns transmitting data in a cyclical fashion. However, it should be noted that the RR scheduler is very basic and does not consider the specific characteristics of each subflow's path, which may result in poor performance if the paths are not symmetrical. If the sender is able to fill the congestion windows of all subflows, the scheduling becomes ack-clocked [77], which means that new packets will be scheduled on whichever subflow has space available in its congestion window as it empties. This approach ensures that data is transmitted continuously across all subflows, but it may not be the most efficient scheduling method if the subflows have different characteristics or if there are significant differences in the congestion levels of the various paths.

The default scheduler of MPTCP is the Lowest-RTT-First (LowRTT) scheduler. LowRTT scheduler selects the path with the lowest estimated RTT as the first choice for data transmission. This ensures that the data is transmitted through the path with the lowest latency. Once the congestion window of the first path is filled, LowRTT moves on to the path with the second lowest RTT, and so on. This process ensures that all the available paths are utilized and the most efficient path is used first. Similar to the RR scheduler, LowRTT also becomes ack-clocked if the congestion windows of all subflows are filled. This means that LowRTT will wait for an ACK from the receiver before transmitting more data to prevent congestion and ensure smooth data transmission [77].

The Linux implementation of MPTCP has a redundant scheduler feature that is designed to optimize network performance. The redundant scheduler duplicates network traffic on all available paths, which helps to achieve low latency and improve the reliability of data transmission. However, this comes at a cost of reduced bandwidth as more traffic is being transmitted over multiple paths. Despite this trade-off, the redundant scheduler is a valuable tool for ensuring data is transmitted quickly and efficiently, particularly in environments with critical low latency.

BLEST [78] is an algorithm that was developed to tackle the challenges of improving application performance in complex network scenarios. One of the main issues that BLEST aims to address is the problem of HoL blocking, which occurs when data packets are delayed or lost in a network, leading to retransmissions and, ultimately, degraded application performance. To address this issue, BLEST introduces a novel approach that relies on a Blocking Estimation-based MPTCP Scheduler. This approach involves estimating the likelihood of HoL blocking and implementing a "Wait" mechanism to mitigate its impact. Specifically, if the network path with the highest RTT is the only option available, BLEST can decide to wait for the lowest RTT path to become available again. This prediction is based on the understanding that sending data on the path with the highest RTT may block the receiver, causing further delay and reduced performance. By introducing such an estimation and wait mechanism, BLEST effectively reduces spurious retransmissions and boosts application performance in heterogeneous network scenarios. Overall, BLEST is a valuable algorithm that helps to ensure that data is transmitted efficiently and with minimal delay, making it a critical tool for improving network performance.

The Earliest Completion First (ECF) algorithm [79] is a scheduling policy that aims to minimize the average response time of network traffic flows. Like some other algorithms, ECF also applies a wait mechanism to ensure that each flow gets its fair share of the network resources. However, ECF uses a slightly different approach to estimate the time a flow should wait before being scheduled. ECF estimates the waiting time by decreasing the idle time of the lowest RTT path among the flows. In other words, ECF prioritizes the flow with the shortest idle time on its lowest RTT path. This approach ensures that flows waiting for a long time get a chance to be scheduled first while still giving preference to flows with shorter RTT paths. By doing so, ECF can achieve better average response times and fairness among network flows.

Peekaboo [74] makes use of a reinforcement learning approach based on a Multi-armed bandit strategy to decide whether to use the slow subflow or not. The Peekaboo approach was proposed as a solution to the challenging research problem of designing a multipath scheduler that has the ability to learn and adapt to diverse paths with dynamically changing channel conditions. This approach aims to provide a more comprehensive solution by considering the heterogeneity of paths in a network and the dynamic variations in channel conditions that can occur over time. With Peekaboo, the scheduler can intelligently adjust to the changing conditions and select the most suitable path based on the available information, ultimately leading to better network performance. This novel learning-based multipath scheduler continually monitors the dynamicity level of each available path and selects the most suitable scheduling strategy accordingly. This ensures that optimal scheduling decisions are made based on the current network conditions. To achieve this, Peekaboo employs an online adaptive learning mechanism that first selects a deterministic strategy to deal with different levels of dynamicity. This strategy involves choosing a particular path or waiting for better conditions to transmit a packet. However, Peekaboo also applies a stochastic adjustment strategy on top of the deterministic decision. This allows the system to counteract better the dynamicity experienced over the available paths. By combining these two strategies, Peekaboo is able to manage the network resources and ensure optimal performance effectively.

Connection Teardown

In a regular TCP connection, both endpoints use a 'FIN handshake' to terminate the connection, indicating that they have no more data to transmit. However, this type of handshake only closes the specific subflow used to send the FIN packets. This poses a challenge for MPTCP, as it needs to be able to close individual subflows that it no longer wishes to use while keeping the overall connection intact. MPTCP can close specific subflows as needed to address this. Once all subflows have been closed, the MPTCP connection remains active until a timeout period has elapsed, at which point the connection is officially declared dead. This ensures that MPTCP connections can maintain their flexibility and efficiency while also ensuring that the connection is properly terminated when necessary.

To gracefully close all active subflows in a MPTCP connection simultaneously, MPTCP leverages the DATA FIN mechanism. To signal that it has no more data to send, an end-point may set the F flag in the DSS option. This action initiates a 'DATA FIN handshake,' equivalent to a 'FIN handshake' for the entire MPTCP connection. Once both end-points acknowledge each other's DATA FINs, the connection is terminated. Following this, each subflow is terminated using standard 'FIN handshakes'. This approach ensures that all data is transmitted and acknowledged before terminating the connection and helps prevent the loss of unsent or unacknowledged data.

1.5 Multipath Quick UDP Internet Connections (MPQUIC)

As for MPTCP, there are two main motivations for adding multipath capabilities to a transport protocol like QUIC. The first is to pool resources of different network paths to carry the data over a single connection [80]. Such pooling is important for multi-homed devices when transferring large files, but it can also help dual-stacked hosts automatically select the best network when the quality of the IPv4 and IPv6 paths differ. Another motivation is the resilience to connectivity failures. On dual-homed devices with wireless interfaces, such as smartphones, one network can fail at any time, and users expect that their applications will immediately switch to the other one without any visible impact [81, 82]. These use cases are covered by Multipath TCP [83, 82].

MPTCP is a promising solution for carrying data over multiple network paths. This has led to integrating multipath capabilities into transport protocols like QUIC. The key motivation behind this integration is to pool the resources of diverse network paths,

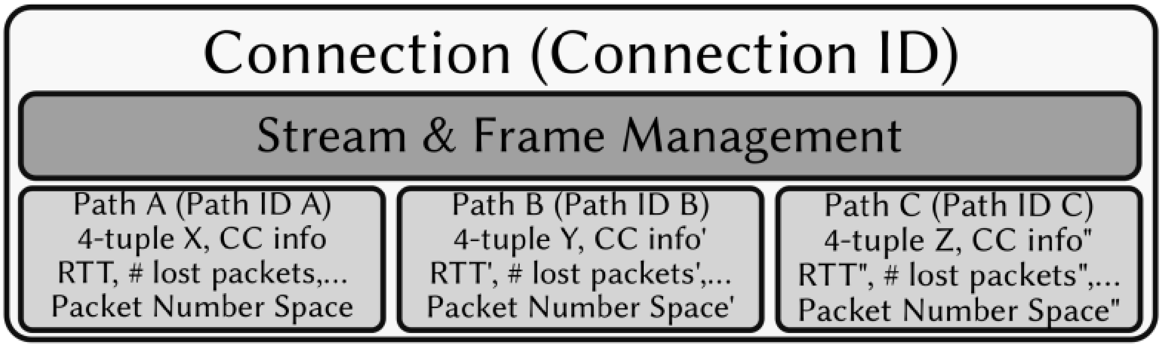

Figure 2.3: The high-level architecture of MPQUIC illustrates a connection with three paths [8].

enabling the data to be transferred over a single connection [80]. Multi-homed devices can benefit significantly from such pooling when transferring large files. Additionally, dual-stacked hosts can leverage this pooling to automatically select the best network path when the quality of the IPv4 and IPv6 paths differ.

Another significant motivation is its ability to ensure resilience to connectivity failures. For instance, on dual-homed devices with wireless interfaces, such as smartphones, one of the networks can fail at any time. In such cases, users expect their applications to immediately switch to the other network without any visible impact [81, 82]. Multipath TCP has been designed to address such use cases and provides a reliable solution for ensuring seamless connectivity even during network failures [83, 82].

MPQUIC [84] is an extension of the QUIC transport protocol that allows for the aggregation of multiple network paths, similar to how MPTCP enables multipath capabilities for TCP. With MPQUIC, congestion control mechanisms are natively supported, and they are inspired by the fair mechanisms used by MPTCP. This ensures that multiple network paths are used in an equitable manner, providing a better user experience. Compared to MPTCP, MPQUIC has a simpler and more efficient design that is less susceptible to middlebox tampering. This means that there are fewer compromises when it comes to security, and the protocol is more reliable in a wider range of scenarios. Thanks to the stream multiplexing feature, MPQUIC is more robust against head-of-line blocking and is capable of using stream-based packet scheduling techniques, which further enhances its performance and efficiency. All in all, MPQUIC is a highly advanced transport protocol that is capable of providing a faster, more reliable, and more secure network experience for users.

Path Identification:

In the context of using multiple paths for communication, it is crucial for hosts to agree on a method to distinguish and identify different paths. One possible approach is to use a range of packet numbers sent over a specific path, implicitly indicating the route used. However, it is essential to keep in mind that various paths can have vastly different characteristics, such as delays, which can cause problems. One potential issue is that the packet number in the clear-text header of QUIC is not encrypted, which means that middleboxes in the network can see it. These middleboxes may

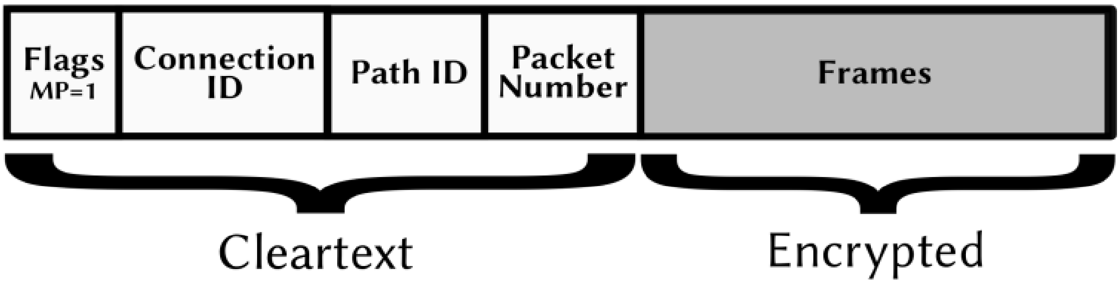

Figure 2.4: The format of an MPQUIC packet [8].

decide to drop packets with smaller packet numbers than the highest number seen on the connection. If a device like this is placed in front of a server that uses multiple paths, it could potentially disrupt the slowest path, leading to degraded performance or even connection failure. Therefore, it is crucial to consider the limitations of different paths and middleboxes when designing a communication system that uses multiple paths.

Figure 2.4 illustrates that MPQUIC has a unique approach by including the Path ID in the clear-text header of the QUIC packets. This is achieved by utilizing a previously unused bit in the Flags field of QUIC packets. The explicit inclusion of the Path ID in the header not only helps identify the path on which a packet was sent but also enables Multipath QUIC to collect several metrics for each path, such as round-trip-time and lost packets. Additionally, including the Path ID in the header allows hosts to detect if there has been a remote address change over a specific path, which could be due to NAT rebinding. If such events occur, hosts can update the 4-tuple associated with the path accordingly. This feature of Multipath QUIC ensures that the communication between the hosts remains smooth and uninterrupted even if there are changes in the network infrastructure.

Reliable Data Transmission:

To send data, QUIC uses encrypted STREAM frames. These contain a stream identifier and an absolute byte offset. This information is sufficient to enable a receiver to reorder the data contained in STREAM frames that it receives over different paths. However, the QUIC acknowledgment is per-packet based, and reordering could affect packets sent on different paths due to network heterogeneity. With a single packet number space, this could lead to huge ACK frames containing many ACK blocks. To cope with this, each path maintains its own packet number space, as emphasized in Figure2.4. By combining the Path ID and the Packet Number in the public header, MPQUIC exposes the paths to middleboxes. Because packet numbers are now relative to paths, MPQUIC also adds a Path ID field in the ACK frame. This enables a receiver to acknowledge QUIC packets that have been received on different paths.

Path Management

When establishing a connection using QUIC protocol, a secure handshake is initiated to ensure communication security. Similarly, MPQUIC also utilizes the cryptographic handshake over the initial path to maintain secure data transmission. However, unlike QUIC, MPQUIC utilizes a path manager to control the creation and deletion of paths. This path manager manages the different paths and selects the best path for data transmission.

Furthermore, MPQUIC implementation embeds a full-mesh path manager, which enables it to create multiple paths between the sender and receiver. Once the handshake process is completed, the client is requested to open one path over each interface. This approach ensures that data is transmitted through multiple paths simultaneously, providing higher resilience to network failures and congestion. As a result, MPQUIC offers a more reliable and efficient communication protocol than traditional TCP/IP protocols.

There is an important difference between the two protocols regarding how they handle the initiation of new paths. With MPTCP, a three-way handshake is required before the protocol can use any new path. This means that there is a delay in establishing the new path while the handshake takes place.

On the other hand, MPQUIC can directly use a new path by placing data in the first packet. This means that there is no delay in establishing the new path, and data can be sent over the new path immediately.

However, since MPQUIC allows both hosts to create bidirectional paths, it is important to ensure that the chosen identifiers do not clash. To achieve this, bidirectional paths created by the client have an even Path ID, while those created by the server have an odd Path ID. This ensures that there is no overlap between the identifiers used by the two hosts and that each path can be uniquely identified.

Congestion Control

In a multipath setting, where a single connection is split across multiple paths, using a single-path congestion control scheme can lead to unfairness against other single-path protocols. This is because the single-path protocol may not have access to the same amount of network resources as the multipath protocol, leading to a skewed bandwidth distribution. The current MPQUIC implementation has adopted OLIA Routing-based Congestion Control for QUIC to address this issue as the standard congestion control scheme. OLIA is designed to be fair to both single-path and multipath protocols by dynamically allocating bandwidth based on network conditions and congestion signals. This helps to ensure that all protocols have equal access to network resources, regardless of whether they are using a single path or multiple paths. For more description, turn to Section 1.4.

Packet Scheduling

The current version of MPQUIC implements a packet scheduler that utilizes the LowestRTT-first algorithm, also referred to as the default scheduler, used by the MPTCP implementation in the Linux kernel [9]. The Lowest-RTT-First scheduler is designed to prioritize packets with the lowest RTT in order to reduce latency and increase network throughput. For a more detailed understanding of how the Lowest-RTT-First scheduler operates, refer to Section 1.4.

The MPQUIC protocol used the Lowest-RTT-First scheduler algorithm with some key differences. As mentioned in Section 1.5, MPQUIC's scheduler is designed to be more flexible than MPTCP's default scheduler. In contrast to MPTCP, MPQUIC is not limited to retransmitting frames on the same path. This means that MPQUIC can switch to a different path if the current path is experiencing congestion or packet loss.

On the other hand, MPTCP's scheduler is designed to avoid being blocked by middleboxes. Since middleboxes can interfere with packet transmission, MPTCP is required to retransmit packets on the same path to prevent congestion. However, since a new path's RTT is unknown, MPTCP duplicates all traffic of another path until the RTT of the new path becomes known. This ensures the path is reliable and can be used for further communication.

Overall, both MPQUIC and MPTCP provide unique benefits and drawbacks. Understanding the differences in their scheduling mechanisms can help network administrators make informed decisions when choosing between these protocols for their specific use cases.

2 Comparative Analysis of QUIC, Unreliable QUIC, and MPQUIC for Video Streaming in VANETs

Numerous research studies have been conducted to understand how different aspects of the vehicular environment impact the transmission of information. These factors include node density, vehicular topology, vehicle mobility, propagation medium, and others. In this section, we focused on analyzing the influence of various transmission techniques on the visual quality of the video received. To achieve this, we conducted a thorough survey of the effect of different single-path transport protocols, such as QUIC and unreliable QUIC, and multipath protocols, like MPQUIC, in a realistic urban VANET network setting. The study provided us with a better understanding of the strengths and weaknesses of each protocol, thus allowing us to make informed decisions when selecting the appropriate protocol to improve video transmission via a vehicular network.

The following subsections contain comprehensive information about the methodology employed to compare and assess the performance of single-path and multipath transport protocols in vehicular environments, namely QUIC, Unreliable QUIC, and MPQUIC. Start by outlining the VANET simulation platform. Followed by providing detailed descriptions of the three videos selected for transmission over the network. The three videos were chosen based on their varied characteristics and complexity, representing different types of video content. Lastly, the section presents the objective video quality assessment techniques that were utilized to compare and evaluate the performance of the transport protocols. These techniques were chosen based on their reliability and effectiveness in measuring video transmission quality over networks.

2.1 Related Work

Delivering high-quality video over vehicular communication networks remains a significant challenge due to the unstable nature of VANET communication channels. One approach to address this is the LEQRV (Link Efficiency and Quality of Experience Aware Routing Protocol), developed by [85], which enhances video streaming in urban VANETs by creating and maintaining reliable routes through an improved greedy forwarding method. Additionally, [86] presents a new real-time traffic shaping mechanism for multimedia content in 5G-VANETs, utilizing distributed reinforcement learning. This method optimizes traffic rates and adjusts coding parameters dynamically to ensure smooth video streaming in high-mobility settings. In [13], a hybrid model combining terrestrial and non-terrestrial networks (NTNs) is proposed to provide reliable video streaming in vehicular environments. The model uses channels such as satellite, 5G, and LTE, with a mobile telemedicine unit that applies a modified MPQUIC extension and multiple description coding (MDC) to boost video reliability, aiming to enhance healthcare services in remote regions. Meanwhile, [87] examines the hurdles of multimedia transmission in VANETs, particularly in the context of audio-visual content via IEEE 802.11p, addressing technical difficulties like bandwidth limitations and QoS issues, while exploring AI-driven solutions to these challenges.

Video streaming in VANETs encounters obstacles such as dynamic network topologies and frequent connection interruptions. Advances in routing protocols, including methods for selecting the next-hop relay vehicle, have made strides in improving the Quality of Experience (QoE) for video streaming. For instance, [88] suggests a QoE-aware geographic protocol that correlates QoE and QoS parameters to improve video delivery in VANETs. QoE-focused strategies are critical in such settings. The AntArmour mechanism, described in [89], employs ant colony optimization to dynamically manage redundancy based on real-time network conditions and video coding details, reducing network overhead while improving video quality. Another protocol, the Relay Suitability-based Routing Protocol (RESP), introduced in [90], assesses link stability and geographic advancement to select forwarding nodes, aiming to enhance video streaming in dense vehicular networks. It incorporates metrics such as packet delay, collision drop rate, and Expected Transmission Count (ETX) to ensure reliable video delivery. To address latency issues, [91] proposes a hybrid Whale-Dragonfly Optimization edge computing model that minimizes delay while employing 5G-based Multi-Access Edge Computing (MEC) servers to optimize resource usage and meet QoS requirements. In another study, [92] analyzes the performance of UDP and TCP protocols for video streaming in VANETs, measuring them against throughput, packet delivery ratio, end-to-end delay, and PSNR to identify the most suitable transport layer protocol.

2.2 VANET simulation framework

To accurately assess the effectiveness of the suggested solutions, we conducted a thorough experimentation of a practical scenario of video transmission in a vehicular setting. Our comprehensive framework comprises three key components: an automotive traffic simulator, a network simulator, and a video encoder/decoder. The automotive traffic simulator models the vehicular traffic flow to replicate a real-world scenario. The network simulator evaluates the network traffic and latency by emulating the wireless communication channel between the vehicles. And finally, the video encoder/decoder is responsible for compressing and decompressing the video data. These components work harmoniously to offer a realistic simulation of video transmission in a vehicular environment, as represented in Figure 2.5.

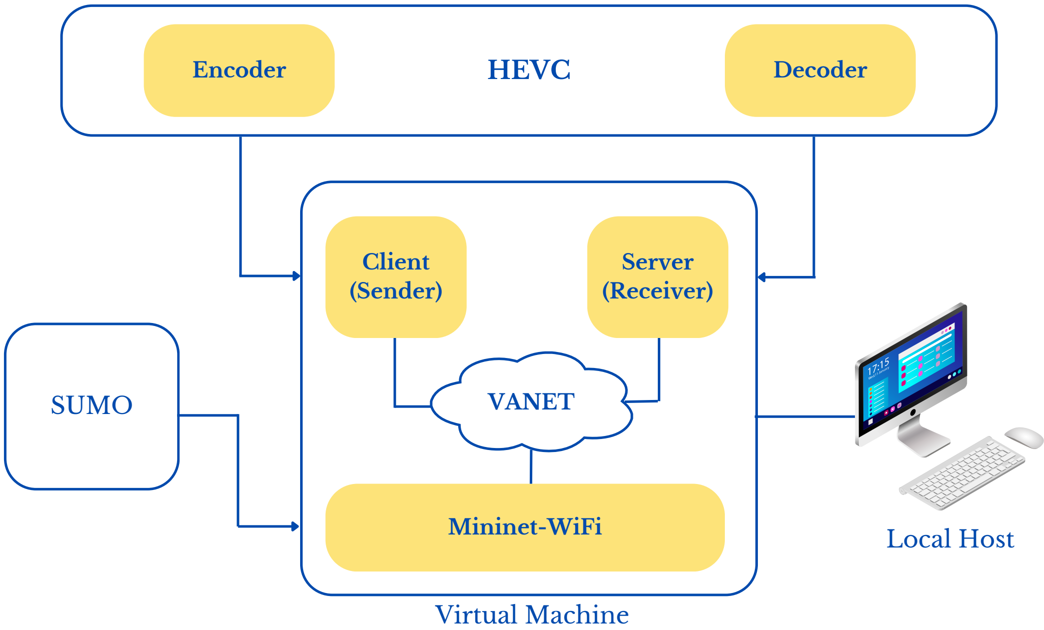

Figure 2.5: Representation of our VANET modeling.

2.2.1 Simulation of Urban MObility (SUMO)

SUMO, short for Simulation of Urban MObility [93], is a highly sophisticated open-source traffic simulator software implemented in C++. It offers a microscopic view of traffic and facilitates the modeling of not only automobiles but also public transportation and pedestrians. SUMO is based on real maps, which makes it an excellent tool for realistic traffic simulation. With SUMO, users can vary a wide range of parameters, including the number of vehicles, their speed, routes, etc., allowing for a highly customized simulation experience. Its advanced features and flexibility make SUMO a popular choice among researchers, urban planners, and transportation engineers.

SUMO is a popular tool used in various domains, including V2X applications. One of the main advantages of SUMO is that it can produce highly realistic vehicle traces, which makes it possible to interact with network simulators like NS-2, NS-3, Omnet++, and more. While there are several traffic generators available, including VanetMobiSim, C4R, and FreeSim, research conducted by M. Fogue et al. [94] has shown that SUMO offers the most comprehensive features for the desired applications. Therefore, SUMO continues to be a top choice for researchers and engineers working in the field of V2X applications.

2.2.2 Mininet Wi-Fi

Mininet-WiFi [95] is an emulator that allows users to simulate wireless networks for research and experimentation. It was extended from Mininet [96], which is a popular network emulation tool that creates a realistic virtual network, running real kernel, switch, and application code on a single machine. One of its most significant advantages is the built-in support for WiFi. This feature allows users to easily create virtual WiFi stations and access points using standard Linux wireless drivers and the 80211_hwsim wireless simulation driver. To make this possible, Mininet-WiFi has

Parameters	Value
Number of vehicles	21
Simulation area	New York city
Simulation time (s)	1000
Propagation model	logDistance
MAC protocol	802.11p
Unreliable QUIC Packet size (bytes)	1268
MPQUIC Packet size (bytes)	1392

Table 2.1: Simulation Parameters.

	Bandwidth (Mbps)	Delay (ms)	Loss Rate (%)
LTE	50	50	0.5
5G	1000	1	0.1

Table 2.2: Different paths characteristics.

introduced new classes that support the addition of wireless devices to a network scenario. These classes also enable users to emulate the attributes of a mobile station, such as position and movement relative to the access points. With Mininet-WiFi, users can simulate complex network scenarios that include wireless devices, providing a more realistic testing environment for their projects.

Additionally, it can simulate other wireless networking technologies, such as VANETs, which makes it a versatile tool for various experiments. By using Mininet-WiFi, users can also leverage existing Mininet nodes, such as hosts, switches, and OpenFlow controllers, to create complex and realistic network topologies. Overall, Mininet-WiFi is an excellent tool for researchers and network engineers who need to test and evaluate wireless network configurations without expensive hardware equipment.

The study comprehensively evaluated various protocols for streaming video in vehicular environments using a simulation framework. The framework utilized the Mininet-WiFi network emulator [95], an extension of Mininet specifically designed for wireless networks [96]. Mininet-WiFi offers native support for WiFi and can simulate other wireless technologies, including vehicular networks. The SUMO tool was integrated with the Mininet-WiFi emulator, which allows for the modeling of vehicles' behavior from urban traffic maps imported from OpenStreetMap (OSM). The simulation framework allowed for sophisticated experiments that could be used to analyze and optimize network performance in vehicular environments. The simulation parameters are summarized in Table 2.1. To emulate real-world networks' properties in a Mininet-Wifi environment and test the utilized protocols, the study used Netem as the tool of choice [97]. Netem provides one or more network impairments, such as delay, loss, duplication, and packet corruption. Each network's path characteristics were outlined in Table 3.1, according to [98, 99, 100, 101]. As mentioned earlier, the study used the HM test model (v16.24), accompanied by the FFMPEG (libx265), for video encoding. This approach enabled a detailed and comprehensive evaluation of the protocols' performance, providing valuable insights for future research and development.

2.3 Video coding information

2.3.1 Video Description

Throughout the study, three different videos were utilized, each of which was meticulously chosen based on their unique characteristics. The selection process involved thoroughly analyzing various factors, such as the video's duration, content type, complexity level, and visual quality. Detailed information regarding the characteristics of each video is available in Table 2.3, which provides a comprehensive overview of the video's attributes. Additionally, the different frames extracted from each video are illustrated in Figure 2.6, which visually represents the specific segments of the videos used in the study.

More specifically, the first video, named *"Eyes Examination"*, provides a frontal view of the eyes, presumably captured from a remote patient and transmitted to a professional ophthalmologist for examination. The video has a resolution of 848×464 at 24 frames per second (fps), a duration of 16 seconds, and yuv420p raw format. These characteristics are further elaborated in Table 2.3. The first video is considered to have the lowest temporal information among all the videos, indicating that the complexity of this video is relatively low. The second video is called Ultrasound and has been created using ultrasound technology to provide a visual representation of a developing fetus within its mother's womb. The video has a resolution of 1272×720 and can run at 30 fps. It has a total duration of 16 seconds and has been recorded using the yuv420p raw format. According to the characteristics mentioned in Table 2.3, the second video has a higher level of TI than the first video. This indicates that it contains more significant levels of texture information, which can help identify and track the movements of the fetus. Overall, the second video is an important resource for medical professionals, researchers, and anyone interested in the study of fetal development. The third one is called Amoeba. This particular video showcases the amazing natural phenomenon of a paramecium devouring an amoeba, which is also known as phagocytosis. This video has the highest SI (Signal Intensity) and TI (Temporal Intensity) levels compared to the other videos in the collection. It has a high resolution of 1920×1080 at 30 fps, which ensures that the details are captured with high clarity. The video has a duration of 13 seconds, allowing for ample time to observe the entire process of phagocytosis. It is also saved in yuv420p raw format, which has some extra descriptive characteristics, which you can find in Table 2.3. Figure 2.7 illustrates the corresponding SI and TI values for each video.

2.3.2 Video codec setup

Specific encoding parameters were carefully chosen to replicate a practical scenario and improve the data rate. The HEVC standard provides a configuration mode customized for particular applications. This approach considers various factors such as coding efficiency, computational complexity, processing time, and error resilience techniques, all aimed at optimizing performance [33]. It is worth noting that for temporal prediction, three different structures have been proposed: All-Intra (AI), Low-Delay (LD), and Random-Access (RA).

In this work, we have used the All-Intra structure to ensure that each frame is independent and can be decoded separately. To achieve this, frames are encoded as Instantaneous Decoder Refresh (IDR) pictures, allowing for more playback flexibility.

Video Sequence	Eyes Examination	Ultrasound	Amoeba
Resolution	848 x 464	1272 x 720	1920 x 1080
Duration	16	16	13
Length	500	500	400
Frame Rate	24	30	30
Data Size	281 Mo	655 Mo	1150 Mo
Max TI	6.91012	11.8123	30.295
Max SI	40.4119	24.9537	116.755

Table 2.3: Video Characteristics

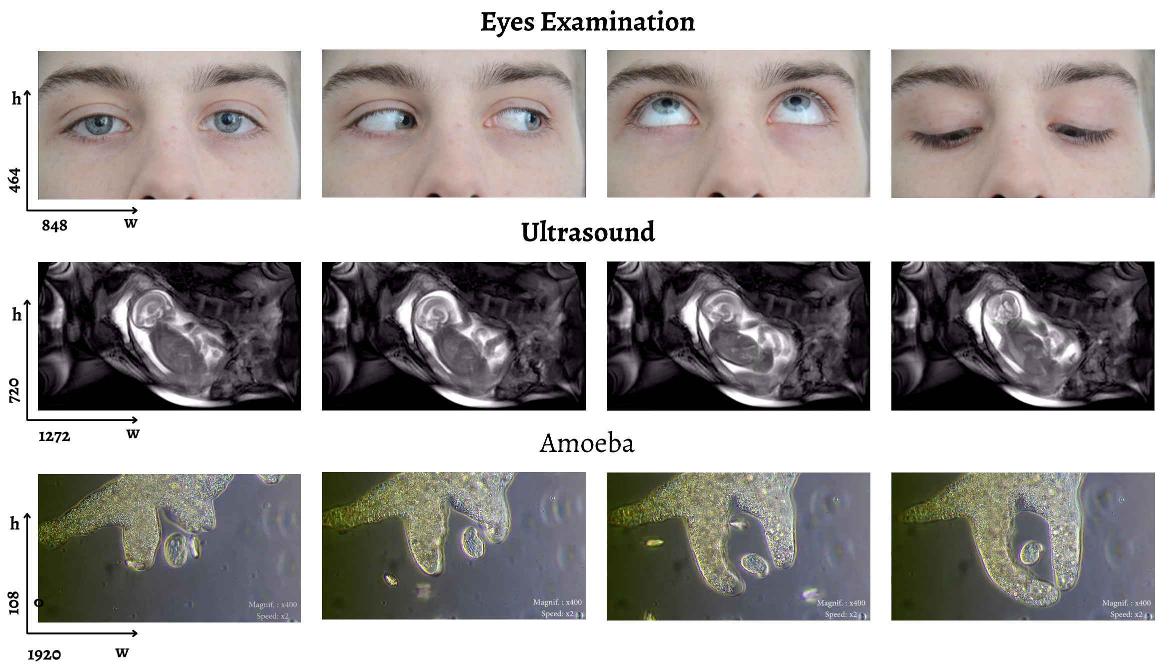

Figure 2.6: Extracted frames from the utilized videos [12].

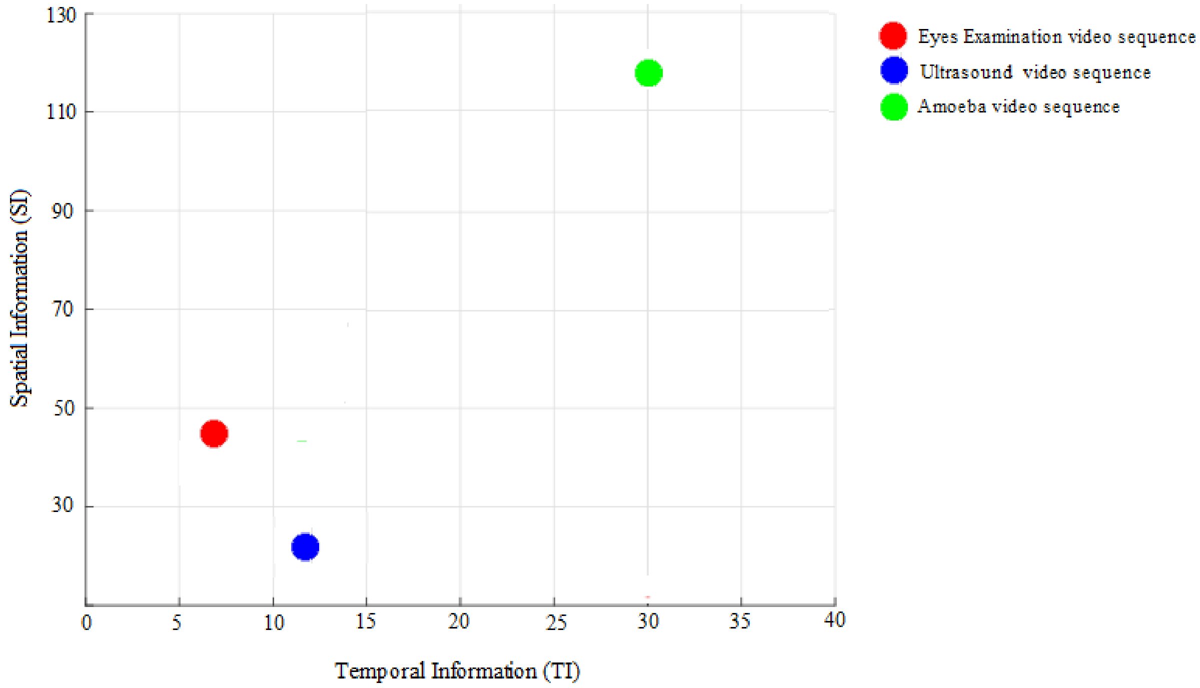

Figure 2.7: Maximum SI and TI values for each video.

Parameter	HEVC test Model v16.24
QP	22, 27, 32
Profil	All-Intra
GOP size	1
ItraPeriod	1

Table 2.4: Video codec experimental setup.

However, the All-Intra structure has a drawback: it can lead to a higher data rate, making it unsuitable for networks with limited bandwidth. We used selected quantization parameter values for constant quality encoding to ensure the video quality is acceptable. Specifically, a value of 27 was chosen for each video sequence. Additionally, we set the default preset parameters available in every video coded to All-Intra for HM. We used the latest reference software versions available for the HEVC codec HM 16.24 and the FFmpeg library Libx265 [102] to carry out these encoding tasks. These software versions gave us the tools and features to encode and decode the video files correctly. Finally, we want to note that the experimental setup used in this series of experiments is detailed in Table 2.4. This setup was designed to ensure that we could accurately measure the performance of our encoding and decoding methods under various conditions.

2.4 Evaluation Metrics

When computer networks were initially deployed, there were no specific access requirements and no way to differentiate between different types of traffic. Consequently, the only way to ensure that all applications functioned properly was to oversize the network as needed. However, as networks began to evolve, it became clear that a more sophisticated approach was required to meet the diverse requirements of various applications. This led to the development of QoS protocols that could prioritize different types of traffic and ensure they received the necessary resources to function effectively.

Today, QoS is a critical aspect of modern networking, enabling applications such as video conferencing, file transfer, web access, and instant messaging to function smoothly and without interruption. However, unlike the initial flows that networks had to manage, these new applications require latency, bandwidth, and jitter guarantees to ensure proper operation. Network administrators must carefully configure their networks to ensure these resources are available when needed.

To evaluate QoS at the application and network levels, several parameters must be considered. These include the maximum delay that can be tolerated by different types of traffic, the amount of available bandwidth, and the degree of packet loss that can be tolerated. By carefully monitoring these parameters, network administrators can ensure that all applications receive the resources to function effectively.

In addition to QoS, it is also important to consider QoE when evaluating network performance. This involves looking at factors such as user satisfaction, ease of use, and overall performance when evaluating a network's effectiveness. By taking a holistic approach to network evaluation, administrators can ensure that their networks provide the best possible experience for all users.

2.4.1 Quality of service evaluation criterion.

The main parameters influencing user perception, according to the ITU-T, are as follows:

- The information loss is due to two factors: Information encoding at the source. Transmission (mainly bit or packet losses).
- The packet transfer delay (the end-to-end delay) is due to propagation and transmission delays, waiting times in lines, and encoding and decoding delays in the case of a real-time application.
- Jitter, which is the variation in delay, represents the difference linked to the arrival time of the information packets.

Performance parameters can measure the quality of user experience for various applications. These parameters vary depending on the type of application, which can be broadly classified into three media categories: audio, video, and data. These categories encompass applications such as online streaming, video conferencing, online gaming, and data transfer. Each category's performance parameters include data transfer rate, latency, audio and video quality, and overall user satisfaction. Understanding these parameters is crucial for ensuring optimal performance and delivering a high-quality user experience.

2.4.2 Visual Quality Assessment Criterion

When it comes to evaluating the visual video quality, it is crucial to differentiate between two types of measurements: objective and subjective. Objective methods use mathematical techniques to generate numerical measures that quantify the quality of a service provided to a user. These measurements are obtained by analyzing various video properties, such as resolution, bit rate, and frame rate, and comparing them to a reference video. In contrast, subjective methods are based on user tests that evaluate different configurations of a service. Participants are asked to rate the quality of the video on different scales or answer specific questions related to their experience. These tests aim to capture the user's perceptual experience, taking into account factors such as visual quality, audio quality, and overall satisfaction. While objective methods provide precise and repeatable measures, subjective tests are essential to understand how users perceive the video quality and identify potential issues that objective methods might not capture.

Our research on video coding has been conducted using an objective approach, a commonly used method in this field. The main advantage of this approach is that it allows for a more standardized and less demanding set of experimental conditions. To evaluate the quality of the reconstructed video, we have selected specific tools designed to measure the visual quality of the output. These tools have been carefully chosen based on their ability to provide accurate and reliable results. As for the work in this thesis, we have chosen two methods:

1. Signal-to-Noise Ratio or Peak Signal-to-Noise Ratio (PSNR).
2. Structural Similarity or Structural SIMilarity (SSIM).
3. Video Multi-Method Assessment Fusion (VMAF).

Peak Signal to Noise Ratio (PSNR) The Mean Squared Error (MSE) is a commonly used metric to assess the quality of reconstructed images. It measures the average squared difference between the original and reconstructed images. This error metric enables us to evaluate the Signal to Noise Ratio of the reconstructed image, which is denoted as PSNR and expressed in decibels (dB). The PSNR value measures the quality of the reconstructed image compared to the original image, with a higher value indicating a higher-quality reconstruction. Therefore, the MSE is a helpful tool for evaluating the effectiveness of image processing algorithms and techniques. The expression of PSNR for a frame-by-frame calculation is given by the following equation [103] for the k^{th} image, where k = 0,1,···,K − 1, and is defined as follows, based on the MSE:

$$PSNR_k = 10 \times log_{10}\left(\frac{d^2}{MSE_k}\right) \tag{2.1}$$

In digital imaging, images are typically encoded using 8 bits, which allows for a maximum value of 255 for the variable "d". This means that each pixel in an 8-bit image can have a value between 0 and 255, with 0 representing the darkest possible shade and 255 representing the lightest possible shade. 8-bit encoding is a common digital imaging standard and is widely supported by software and hardware. The Mean Squared Error (MSE) of two images, Ir_r and I_o, with dimensions M × N, is defined as follows:

$$MSE_k = \frac{1}{M \times N}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}(I_o(m,n,k) - I_r(m,n,k))^2 \tag{2.2}$$

The average video PSNR between the received reconstructed video and the original video can also be calculated. As demonstrated by [103], for cases of channel errors or packet loss, the average PSNR should be calculated as follows:

$$PSNR_{avg} = 10 \times log_{10}\left(\frac{d^2}{\mu_{MSE}}\right) \tag{2.3}$$

With μMSE being the average of the images' Mean Squared Error (MSE). It's important to understand that relying solely on PSNR as a measure of image quality may not accurately reflect how the human eye perceives image degradation. While it is true that higher PSNR values generally indicate better quality for decompressed or reconstructed images during transmission, it's essential to keep in mind that other factors can also impact image quality and how the human eye perceives it. Therefore, it's crucial to use PSNR in conjunction with different metrics and measures to capture a more comprehensive understanding of image quality.

Structural SIMilarity (SSIM)

When measuring a video's quality, a commonly used method is SSI) [104]. This technique employs the concept of structural distortion in the video to estimate the perceived visual distortion. The basic idea behind SSIM is that the human visual system (HVS) is adept at extracting structural information from the field of view. Hence, the perceived quality of a video can be approximated by comparing its structural characteristics with those of a reference video. By evaluating the similarity between the two videos based on their structural features, SSIM provides a more accurate measure of visual quality as perceived by humans [105].

The degree of perceived degradation in an image is directly related to perceived structural loss compared to perceived errors. This is measured using the SSIM, which generates a value between 0 and 1. The SSIM value indicates how similar two images are, with a value closer to 1 indicating a higher degree of similarity. Therefore, the SSIM value is an important tool for evaluating the quality of images and determining the extent of degradation that may have occurred. The SSIM value is given by the following equation:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1) + (2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{2.4}$$

With: μ_x and μ_y represent respectively the average of the original image and the reconstructed image.

σ_x, σ_y and σ_{xy} represent respectively the standard deviation of the original image, that of the reconstructed image and the covariance of the two images.

Video Multimethod Assessment Fusion (VMAF)

Video Multimethod Assessment Fusion (VMAF) [106] is a video quality assessment metric that has gained prominence in video compression and quality assessment. Developed by Netflix, VMAF is designed to evaluate the perceived quality of a video by considering multiple video quality assessment algorithms and fusing their results into a single score. It was created to improve the assessment of video quality and is particularly useful in the context of video streaming services like Netflix. It aims to provide a more accurate and perceptually relevant evaluation of video quality compared to traditional metrics like PSNR (Peak Signal-to-Noise Ratio) and SSIM (Structural Similarity Index).

VMAF combines the outputs of various existing video quality assessment algorithms, including but not limited to PSNR, SSIM, and MS-SSIM, to provide a composite quality score. The fusion process is designed to capture the strengths of each individual metric and mitigate their weaknesses. It also employs machine learning techniques to train its model. It uses a large dataset of videos that have been rated by human viewers for quality. It leverages this data to learn the relationship between objective metrics and subjective human judgments.

VMAF is available as open-source software, which means it can be used and implemented by researchers, video encoding professionals, and video streaming companies. The code and related resources are typically available on platforms like GitHub. It helps in optimizing video quality while maintaining efficient data transmission and storage.

2.5 Performance Evaluation

Several experiments were conducted to assess the efficacy of protocols in vehicular environments. The scenarios varied based on the utilized cellular network (4G or 5G) or the mechanism used to stream the video. This section evaluated three protocols: QUIC, unreliable QUIC, and MPQUIC. The QUIC protocol ensured data delivery to its destination through its reliable mechanism. If transmitted packets were lost, the protocol alarmed and transmitted them again. In contrast, the unreliable QUIC lacked a reliable mechanism to ensure the delivery of packets, thereby not guaranteeing their reliable delivery. The MPQUIC protocol was a reliable transport mechanism that ensured lossless packet delivery by using multiple paths simultaneously to stream data. The second variable pertained to the connected network sending data, which varied from 4G to 5G or both.

A range of objective and subjective metrics were measured to evaluate the system's performance effectively. These metrics were designed to assess various aspects of the system's performance and included measurements such as PSNR, SSIM, VMAF, completion time for video streaming, and the receiving data rate. The metrics PSNR, SSIM, and VMAF were used to compare the transmitted video quality across different protocols, with each metric providing a unique perspective on video quality. Completion time, on the other hand, was used to measure the total delay of the protocols in finishing video transmission, providing insight into the efficiency of the video streaming process. Finally, the quality of video streaming was determined by the receiving data rate, which was measured to compare the performance of each protocol. This metric provided a clear picture of how well each protocol delivered video content to the end user, considering factors such as buffering and network congestion. These metrics provided a comprehensive evaluation of the system's performance, allowing for a detailed analysis of its strengths and weaknesses.

2.5.1 Video quality assessment

The simulation results illustrated below pertain to the image quality evaluation of the three utilized videos - eye examination, Ultrasound, and Amoeba - using PSNR and

SSIM values. These values are illustrated in Table 2.6. Additionally, Table 2.5 illustrates the encoding video quality before transmission for each used video. Because of the acceptable QP value used in the compression process, the visual quality remains of high quality. The received video quality from streaming the videos in a vehicular environment is outlined in Table 2.6.

It is essential to note that when streaming the encoded medical videos using the QUIC or MPQUIC protocols, the received video quality remains perfect and maintains the same encoding visual quality. This is due to the solid data recovery mechanism offered by QUIC and MPQUIC protocols, which ensures lossless data transmission by detecting any lost packet after not receiving its corresponding ACK packet. Therefore, the received video quality was maintained at the same level as the encoded video quality.

However, like any non-oriented protocol, the unreliable QUIC protocol recorded the most destructive PSNR and SSIM values when transmitting the video, mainly due to losses linked to its ignorance about the transmitted packet status and not interest in retransmitting or detecting lost packets. This highlights the significance of using a reliable protocol for medical video transmission in vehicular environments. It is worth noting that the impact of the unreliable mechanism in streaming video applications is more significantly noticeable when streaming a full HD video, as represented in the Amoeba video. This is because of the challenging network conditions in vehicular environments, which makes the protocol suffer from a lot of connectivity failers and a high data loss rate when transmitting a large amount of information compared to when sending a small video resolution, as represented in Figures' 2.8, 2.9, 2.10 results, where Amoeba video (Full HD) records the most degraded visual quality.

Sequence	PSNR	SSIM	VMAF
Eyes Examination	44.77	0.997	89.33
Ultrasound	44.50	0.999	90.98
Amoeba	44.36	0.999	96.85

Table 2.5: Encoding video quality.

		Eyes Examination		Ultrasound		Amoeba	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
QUIC	4G	44.70	0.997	44.50	0.999	44.36	0.999
	5G	44.70	0.997	44.50	0.999	44.39	0.999
U_QUIC	4G	37.91	0.972	36.67	0.960	21.68	0.713
	5G	35.35	0.973	39.44	0.973	20.33	0.753
MPQUIC	4G/5G	44.70	0.997	44.50	0.999	44.39	0.999

Table 2.6: PSNR and SSIM values for each received video.

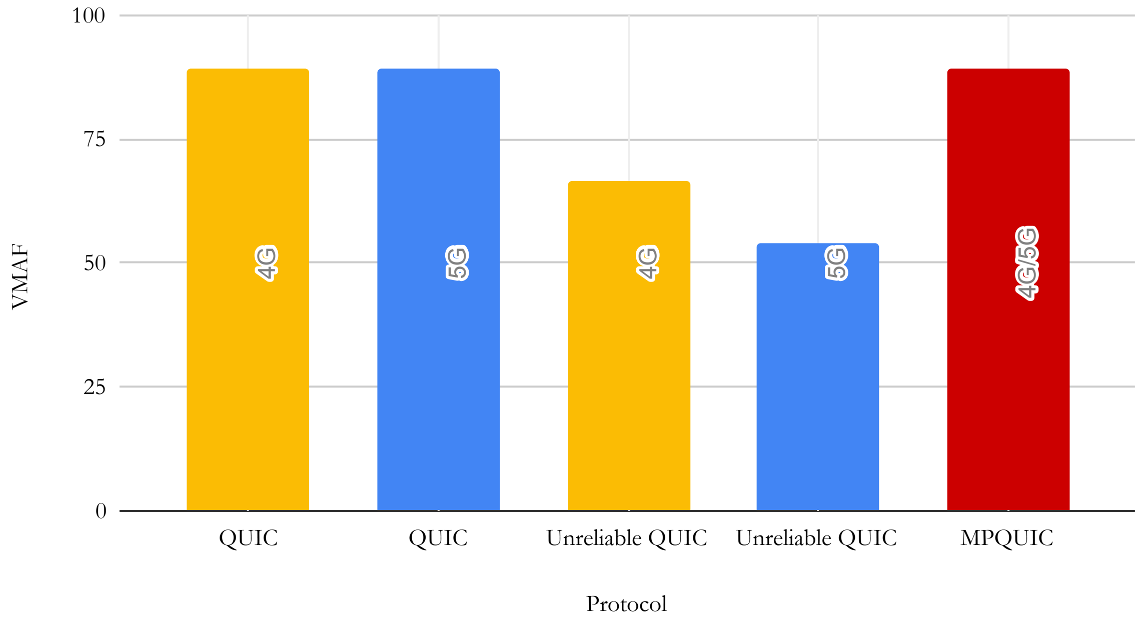

Figure 2.8: VMAF values for Eyes examination.

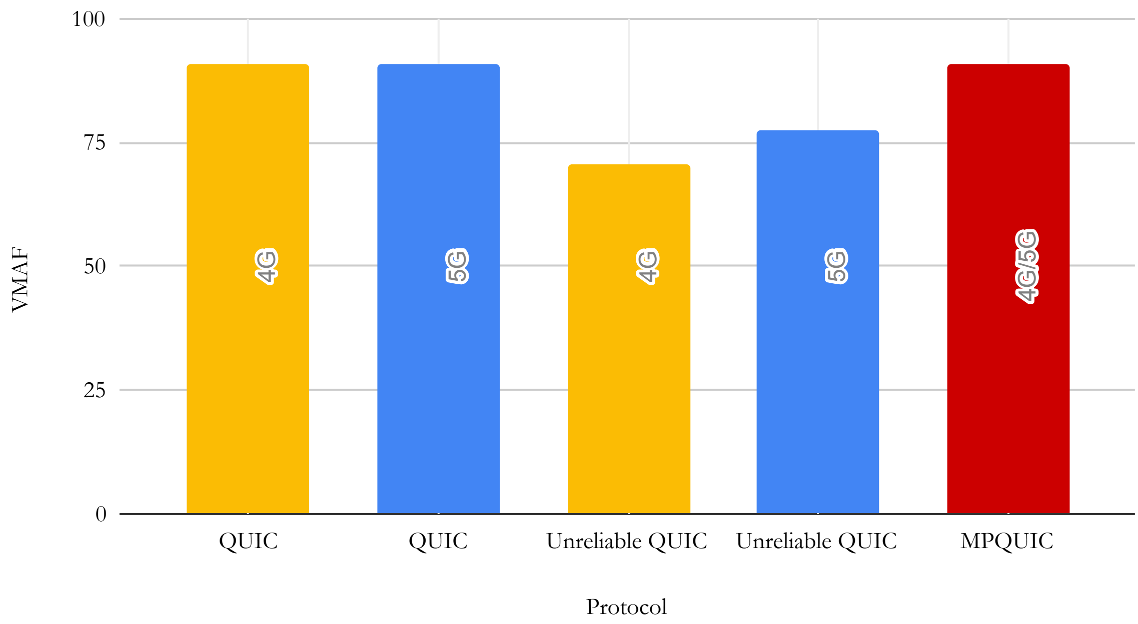

Figure 2.9: VMAF values for Ultrasound.

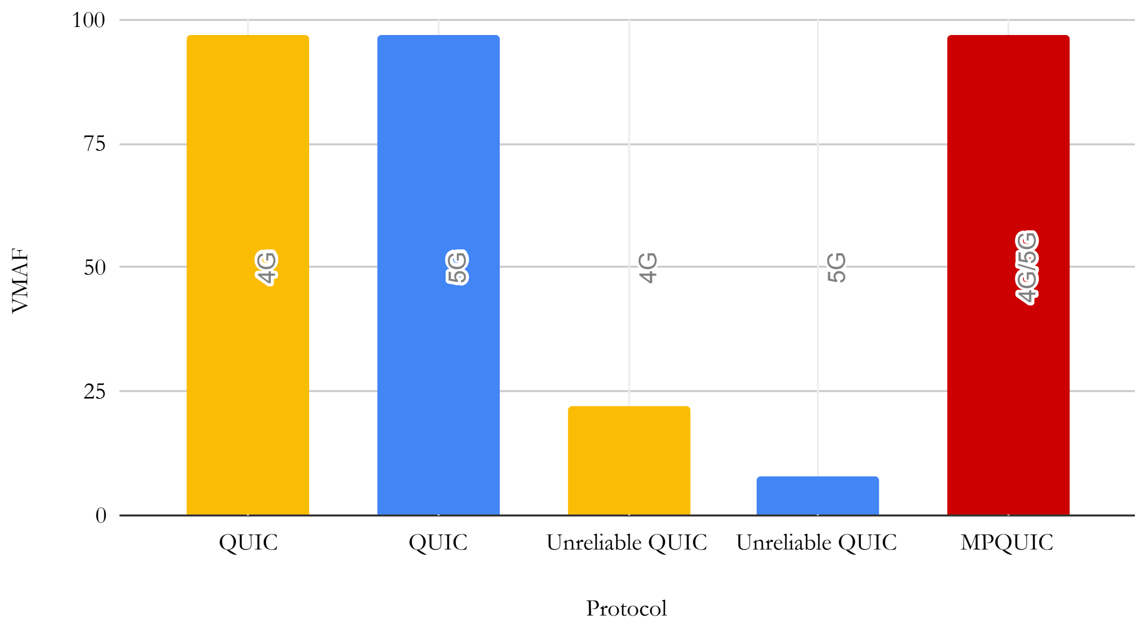

Figure 2.10: VMAF values for Amoeba.

2.5.2 Completion time assessment

Although the transmitted video quality is perfect using the QUIC and MPQUIC protocols, the transmission delay is very high, which is not optimal for real-time video streaming applications in wireless networks, including VANETs. Table 2.7 shows the total delay every protocol needs to finish transmitting the whole video. While streaming video through a 4G network, the QUIC protocol records the most considerable completion time. It must ensure the proper delivery of all transmitted packets, which can be difficult in vehicular environments because of its high mobility and high topology changes. Furthermore, 4G networks are characterized by their high data loss rate, which can reach 5%, and a latency of 50ms, which increases the retransmission rate and the total needed delays to finish the transmission.

On the other hand, because of the optimal 5G network characteristics, which it is characterized by a data rate of 1Gbps, a small data loss rate of 1%, and a latency of 1ms, it leads to recording a smaller completion time compared to the 4G network. In the case of streaming video using the MPQUIC protocol, the three videos record the most negligible delays to finish transmission, and this is because the total data has been streamed through multiple paths (links) simultaneously, which increases the data rate delivery even though the MPQUIC suffers from a slight degradation. After all, the packets that take the 4G link delayed the transmission.

In Table 2.7, the needed delay in sending the entire video using unreliable QUIC through the 5G network is the shortest compared to the sent using the 4G network or the other protocols. Thanks to the offered characteristics of the 5G network and its lack of any recovery mechanisms and transferring packets without any guarantee of this delivery. However, the quality of the received video is the worst due to its unreliability. Additionally, Table 2.7 also displays the data size of each transmitted and

		Eyes Examination		Ultrasound		Amoeba	
		4G	5G	4G	5G	4G	5G
QUIC	Delay	41	12	31	23	231	174
	T. Data	4015606	4015606	8042008	8042008	46206033	46206033
	R. Data	4015606	4015606	8042008	8042008	46206033	46206033
U_QUIC	Delay	9	6	22	10	133	36
	T. Data	4015606	4015606	8042008	8042008	46206033	46206033
	R. Data	3958006	3978406	7911208	7940008	45002433	45112833
MPQUIC	Delay	10		20		152	
	T. Data	4015606		8042008		46206033	
	R. Data	4015606		8042008		46206033	

Table 2.7: Completion time results, accompanied by the transmitted and received amount of data.

received video in bytes to represent the impact of the lossy transmission in vehicular environments for each protocol because of the dynamic topology and the vehicles' mobility.

2.5.3 Receiving data rate

The quality of streaming video is defined by the receiver's data rate. To compare the performance of each protocol, the receiving data rate of each one is measured and illustrated in Figures 2.11, 2.12 and 2.13. Our simulation results indicate that unreliable QUIC outperforms other protocols, QUIC and MPQUIC, by offering the best receiving data rate due to its low delay in finishing the entire transmission. However, it should be noted that most of the successfully received data are not in order, which results in the construction of corrupted frames and poor visual quality.

Conversely, QUIC records the lowest receiving data rate values in the three videos due to its high delays in delivering data through the 4G network characteristics. A slight improvement in receiving data rate values is observable when streaming through the 5G network, which offers better network characteristics (higher data rate, lower latency, and packet loss). The same observation can be deduced when video streaming using the MPQUIC protocol because it uses multiple paths to send data, minimizing total transmission delays. Figure 2.14 illustrates the data loss rate for each protocol, where QUIC and MPQUIC offer zero data loss and the unreliable QUIC records variant values that can change according to the network used to stream data.

3 Conclusion

This chapter presented the tools for modeling video transmission in a vehicular environment and the adopted evaluation metrics. We also addressed the evaluation of video transmission in VANET networks. This work also demonstrates the impact of the video sequence and its content in terms of resolution, spatial information, and temporal information on the quality of received data. This chapter also evaluates

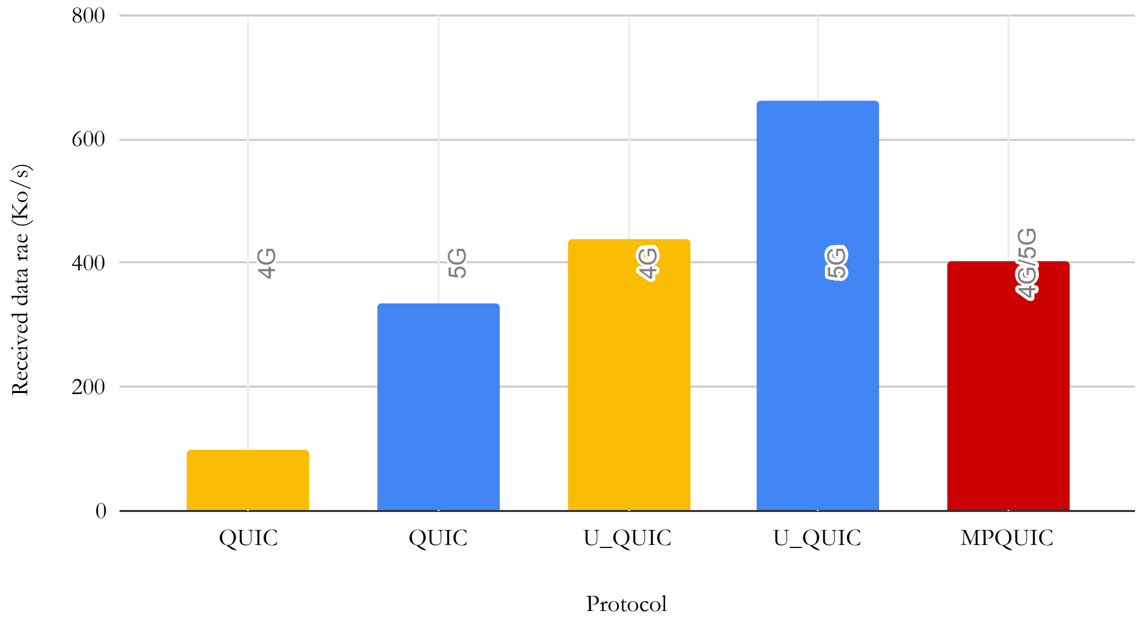

Figure 2.11: Received data rate for Eyes Examination sequence in Ko/s.

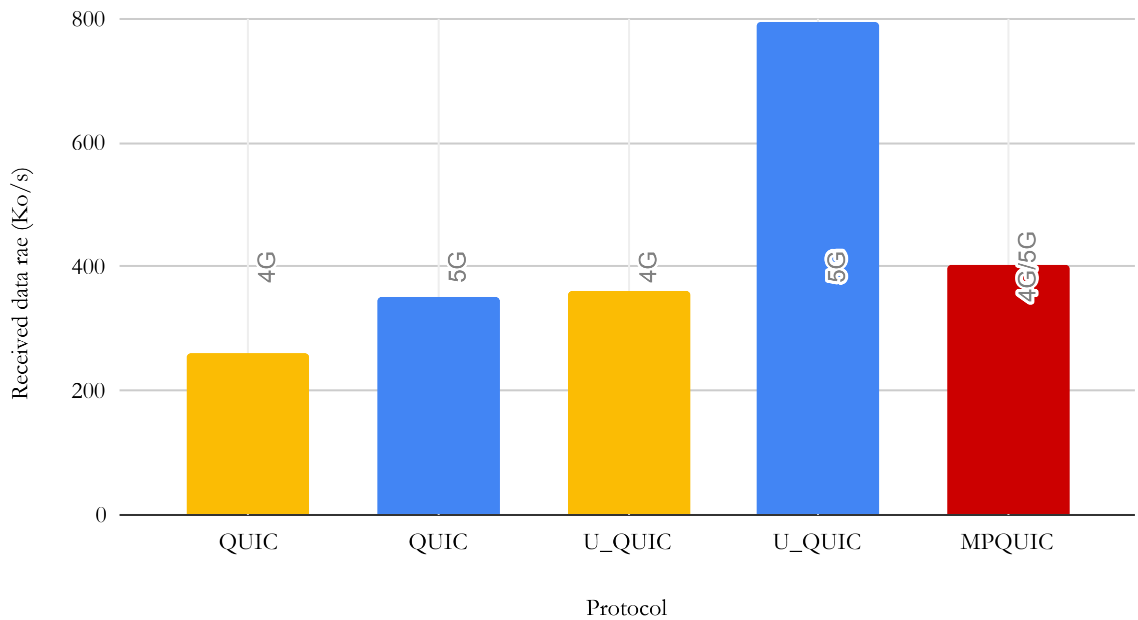

Figure 2.12: Received data rate for Ultrasound in Ko/s.

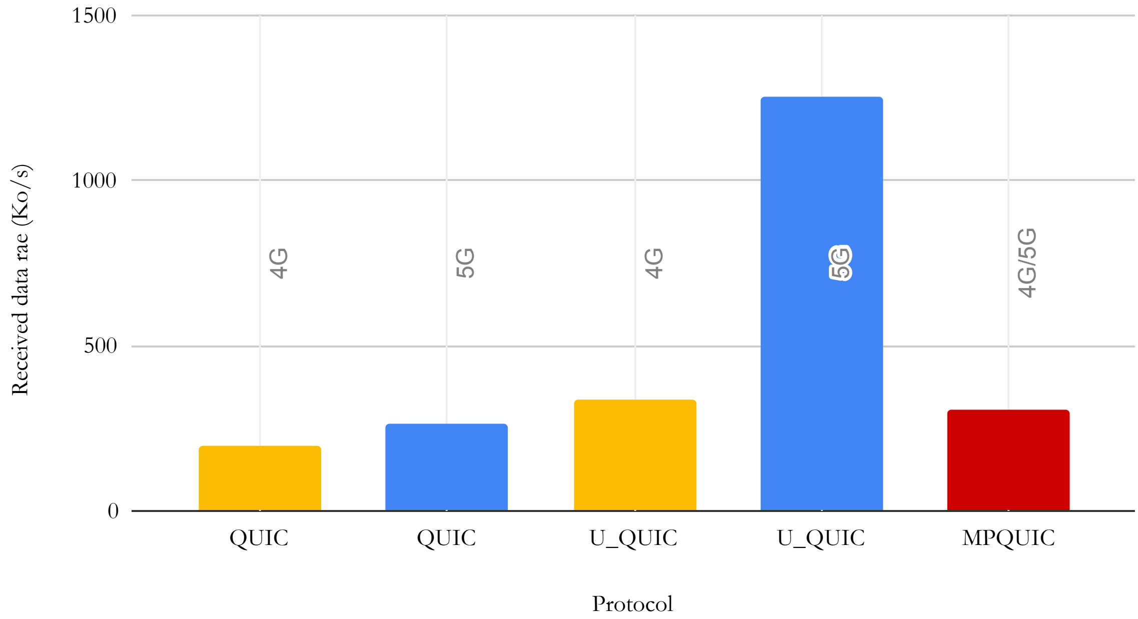

Figure 2.13: Received data rate for Amoeba in Ko/s.

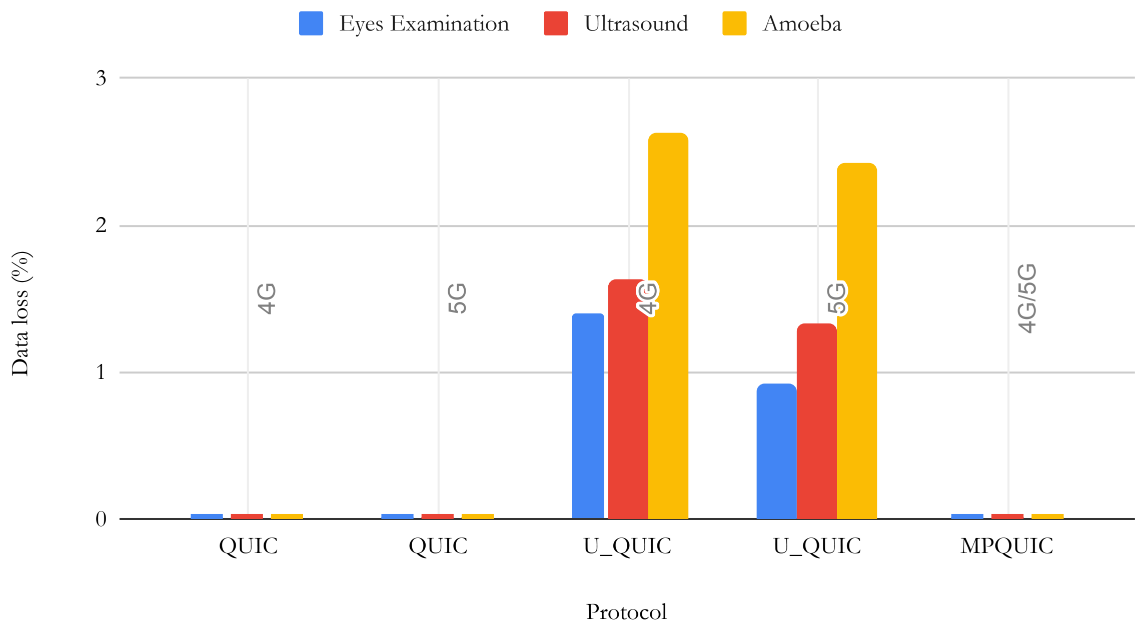

Figure 2.14: Data loss rate in %.

the quality of service of streaming video in low-latency applications over vehicular networks using single-path protocols, such as QUIC, unreliable QUIC, and multipath transport protocols, like MPQUIC.

CHAPITRE 3

Cross-layer solutions for streaming video over VANET networks: Real time mobile health use case.

1 Introduction

Mobile health, also known as M-health, is a term used to describe mobile and wireless technologies that deliver healthcare services. M-health has the potential to revolutionize how healthcare is delivered by enabling medical professionals to provide treatment more efficiently and effectively. Many countries worldwide have shown interest in developing a solid infrastructure for e-health technology. According to a 2020 survey, roughly 83% of the World Health Organization (WHO) nations offer professional training for delivering e-health services [107]. One of the most promising applications of M-health is the use of connected ambulances equipped with mobile telemedicine units. These units can transmit patient information and video footage in real-time to healthcare providers, enabling them to monitor patients and make decisions remotely. This technology has the potential to significantly improve healthcare delivery in isolated rural regions where access to medical care may be limited. Additionally, connected ambulances can be equipped with other advanced medical devices to provide critical patient care during transportation. Moreover, M-health can improve patient outcomes by providing timely and accurate information to healthcare providers. With the help of mobile devices, patients can track their vital signs, medication adherence, and other health-related data. This information can be transmitted in real-time to healthcare providers, who can use it to adjust treatment plans and provide more personalized care. Furthermore, M-health represents a significant opportunity to improve healthcare delivery and patient outcomes. By leveraging mobile and wireless technologies, medical professionals can provide better care, even in remote areas with limited access to medical services. The potential benefits of M-health are vast, and it is likely to play an increasingly important role in the future of healthcare.

Implementing connected ambulances is a complex endeavor that faces multiple challenges, including needing a reliable and efficient communication system [98, 99]. In isolated rural regions, satellite communication can be a viable solution for establishing a connection to an expert site. However, satellite communication can be unreliable at times, and it can also be expensive, limiting its practicality in some instances. Additionally, satellite communication may not be able to support real-time video transmission, which is a critical requirement for emergency medical situations. To overcome these challenges, cellular networks like 5G technology offer a potential solution. 5G networks are specifically designed to support high-speed and low-latency communication, making them well-suited for real-time applications such as mobile telemedicine. Combining satellite communication with cellular networks makes it possible to create a robust and efficient communication system for connected ambulances.

Other approaches can be utilized further to improve the reliability and efficiency of mobile health applications. One such approach is using heterogeneous networks, which combine different communication technologies to provide a more flexible and robust solution. If one communication technology fails or becomes unavailable, the system can quickly switch to an alternative technology, ensuring that vital communication channels remain open. Implementing a reliable and efficient communication system for connected ambulances is essential to ensure that emergency medical services can be delivered effectively and promptly, even in remote and isolated regions. Combining satellite communication, cellular networks, and other innovative approaches can help overcome the challenges and create a robust and efficient communication infrastructure for connected ambulances. Advanced transport protocols like QUIC and MPQUIC increase end-to-end efficiency and reliability in heterogeneous environments.

Our study proposes a modified MPQUIC extension to support real-time video transmission, specifically for mobile telemedicine units. With this extension, the sender can support multipath scheduling, allowing for the reliable transmission of priority data over 5G/LTE networks. In contrast, non-priority data can be transmitted unreliably through a satellite link. By doing so, we can improve overall mobile health performance. In addition to the proposed MPQUIC extension, we recommend using advanced video compression techniques, particularly multiple description coding (MDC). This technique divides the data into multiple streams, each containing only a partial description of the video data transmitted over different paths. This approach enables the data to be transmitted more efficiently, reducing the impact of packet loss and improving the overall transmission quality. Furthermore, MDC could be an excellent solution to compensate for missed frames by copying the corresponding ones from the other description. This ensures the transmitted video data is complete and accurate, even in packet loss or other transmission issues.

Overall, our approach offers a comprehensive solution to support real-time video transmission for mobile telemedicine units. We leverage advanced compression techniques and multipath scheduling to improve the quality and reliability of the transmission.

One crucial consideration is mobility. Moving the mobile unit and expert site can affect video transmission quality and cause connectivity failures when using network technologies with limited coverage areas. However, this issue can be solved using satellite communication, which provides high coverage. Figure 3.1 represents an overview of the proposed approach involving the connected ambulance communicating patient data with the medical department through heterogeneous networks simultaneously.

In this paper, we have proposed an intelligent approach to m-health in isolated rural regions, using a combination of satellite communication and cellular technologies. We have also discussed the use of heterogeneous networks, advanced protocols, and other techniques, such as multiple description coding, to improve the performance of mobile health applications. The rest of this paper is structured as follows: background knowledge and related works are introduced in Section 2. In contrast, Section 5 defines the proposal idea and the suggested architecture of the proposal. Section 6 discusses the experimental decisions and technical details. In Section 7, the simulation results are presented and discussed, and the final Section 8 sums up the entire work with a

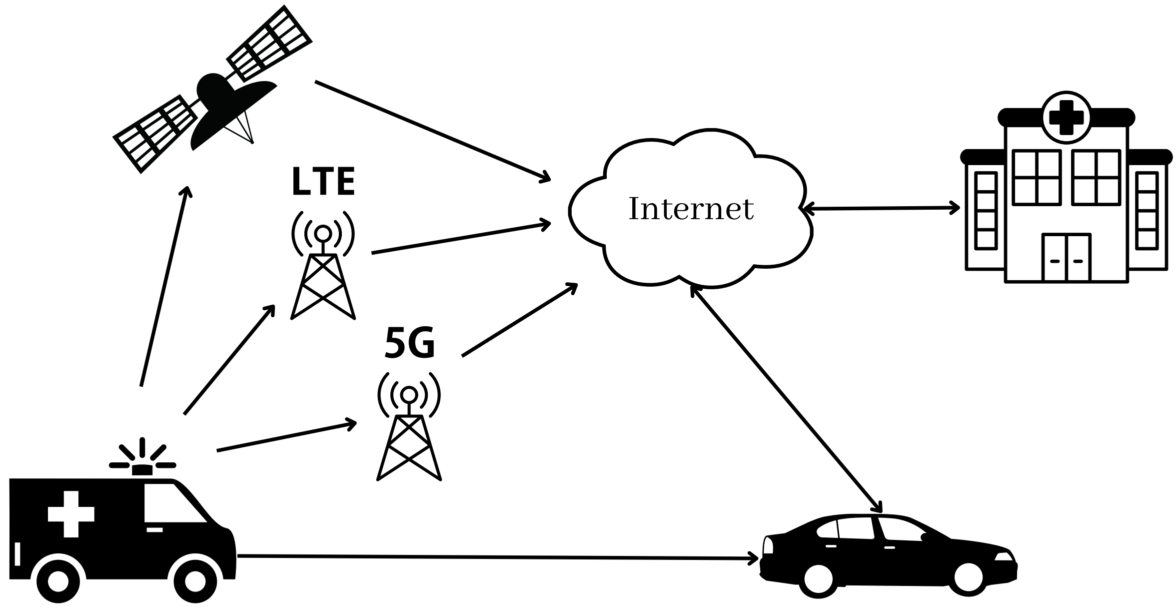

Figure 3.1: Overview of the proposed scheme [13]

conclusion.

2 Related works

Much academic literature has explored various methods of enhancing streaming video quality in vehicular settings. This research has delved into the benefits of implementing multipath scheduling and video compression techniques, such as multiple description coding (MDC), among other potential solutions. Another possible solution is using a cross-layer approach, which involves optimizing the interaction between different layers of the network protocol stack to improve the overall performance of video streaming. In a recent study by Zhenjie Deng et al. [108], a combined Cross-layer DASH-based multipath video streaming approach was proposed to optimize video streaming performance over 802.11ac and LTE technologies. The approach involved using dynamic adaptive streaming over HTTP (DASH) to adapt the streaming bitrate to the available network bandwidth and leveraging multipath routing to transmit the video data over multiple paths simultaneously. The study results showed that the proposed approach could significantly improve the QoE of video streaming over wireless networks compared to traditional approaches. Achieving high-quality video transmission in vehicular communications is a significant challenge due to the network's inherent mobility and dynamic nature. To address this issue, [109] proposes a novel cross-layer approach that leverages application layer information at the MAC layer, combined with MDC technologies. Specifically, the proposed approach involves classifying video packets into the most suitable available channels (ACs) based on their characteristics and requirements, such as priority, delay sensitivity, and bit rate. This way, the system can optimize the allocation of network resources and improve the overall QoS for video transmission. Overall, this approach shows promising results in enhancing the performance and reliability of video communication in vehicular networks, which can have critical applications in various domains, such as intelligent transportation systems, emergency response, and entertainment. To mitigate the issue of out-of-order (OFO) packets that can potentially overwhelm the receiver's buffer, a recent study conducted by Xing et al. [110] introduced a new scheduling mechanism for MPTCP called the OverLapped Scheduler (OLS). Specifically aimed at improving the quality of live video streaming in mobile networks, this novel scheduler operates by prioritizing packet arrival time and sending a limited number of duplicated packets to minimize the impact of network jitter on arrival-time estimates. By doing so, the OLS algorithm can maintain an optimal level of data throughput while significantly reducing the risk of buffer overflow and packet loss. To tackle the challenge of HTTP adaptive streaming in a 5G environment, [111] have suggested a novel multipath-based transmission scheme that leverages a collective segment request strategy. This approach aims to enhance both the accuracy of aggregated bandwidth estimation and the usage of available bandwidth. To achieve this goal, the proposed method combines block-based and segment-based bandwidth estimation values, thus improving the overall responsiveness and stability of the network. Overall, this innovative approach has the potential to significantly improve the performance of HTTP adaptive streaming in 5G contexts. The paper [112] introduces a novel approach to enhancing the QoE for video streaming applications. The proposed model integrates HEVC encoder adaptation schemes and MDC technologies. The authors conducted a comprehensive study on the impact of different QP values and packet loss on the QoE of transmitted video sequences. Their findings provide valuable insights into optimizing video streaming systems to improve QoE for users. In a study by Zaidi et al. [113], a Hybrid Error Recovery Protocol (HERP) was proposed for video streaming applications in VANETs. The HERP integrates two mechanisms, namely Sub-Packet Forward Correction and Retransmission, to address both uniform transmission errors and burst errors that may arise due to network congestion and route disconnection. To ensure that the network is not overloaded and to minimize latency, the HERP dynamically adjusts the redundancy rate, retransmission limit, and transmission rate based on network conditions and network load. This approach enables the HERP to provide a more efficient and reliable video streaming service in VANETs.

Regarding mobile health technologies, a study conducted by Zhai et al. [98] has put forward an innovative proposal for an intelligent ambulance service that takes advantage of 5G technology. The proposed architecture comprises three main components: a 5G communication network, remote video capabilities, and telemedicine medical data exchange. These components work together seamlessly to minimize the ambulance's time to reach the remote hospital and exchange crucial medical information in real-time. This groundbreaking approach could potentially revolutionize how emergency medical services are provided.

The article [114] presents a cutting-edge solution for diabetes management called the 5G-Smart diabetes system. This system integrates advanced technologies, such as machine learning, big data, and wearable 2.0, to automatically and comprehensively analyze diabetes patients' conditions. With the adoption of 5G communication as the infrastructure, this system enables high-quality and continuous monitoring, significantly improving the accuracy and efficiency of diabetes diagnosis and treatment. Overall, this proposed system offers a promising approach to revolutionize diabetes management and enhance patients' quality of life. The authors proposed an emotion-aware connected healthcare system to help caregivers detect patients' emotions remotely in a smart home scenario. The system utilizes several IoT devices to gather audio and visual signals from a patient and process them separately to generate a final score to decide on the emotion. In a study conducted by Hossain et al. [115], a novel emotion-aware connected healthcare system was proposed to aid caregivers in remotely monitoring the emotional state of patients in the context of a smart home environment. The system leverages multiple IoT devices to collect audio and visual signals from the patients, which are subsequently processed independently to produce a final score that accurately reflects the patient's emotional state. This innovative system has the potential to revolutionize the way caregivers provide healthcare services to patients by enabling them to detect and respond to changes in the patient's emotional state in a timely and effective manner. In [116], the focus was on exploring the potential of 5G mobile systems to support ultra-reliable low latency communications in the healthcare industry. Specifically, the study examined two use cases: Wireless Tele Surgery (WTS) and Wireless Service Robots (WSR). The authors provided a comprehensive analysis of the costs and revenues that operators and healthcare providers would encounter if they were to deploy the 5G system to support these services. The study aimed to provide a detailed understanding of the financial implications of using 5G technology in the healthcare industry. This could help inform decision-making processes for stakeholders in this field. In [117], the significance of small-cell heterogeneous networks for medical video streaming was investigated. The ambulance scenario was used as a use case for the study. The authors evaluated the effectiveness of mobile small cells in ultrasound video streaming applications by measuring Key Performance Indicators (KPIs) such as packet loss rate, delay, and throughput. The study results provide important insights into the potential benefits of using small-cell heterogeneous networks in medical video streaming applications. The migration of eHealth services to 5G networks has presented various challenges that must be addressed. To tackle these issues, Wang et al. [118] have proposed an innovative framework called SliceNet, built on advanced and customizable network slicing. This framework is designed to provide a solution for migrating eHealth services to 5G networks by addressing the challenges that arise during the process. The SliceNet framework offers a unique approach that enables the customization of network slicing, making it easier for eHealth services to migrate to 5G networks. With this new framework, the migration process will be more efficient and effective, ultimately ensuring that eHealth services can continue to be provided seamlessly on the latest network technology.

The academic papers that have been referenced provide insight into the various ways in which video streaming performance in heterogeneous networks can be improved. These methods include multipath transport protocols, MDC, and 5G technology. However, none of these papers have proposed a solution combining all these technologies into a unified approach. After careful consideration of the available research, a new approach was conceived. This approach combines a multipath transport protocol with MDC technology, enabling reliable and unreliable data transmission through heterogeneous networks.

3 Wireless communication technologies

3.1 4G/Long-Term Evolution (LTE)

4G/LTE (Long-Term Evolution) technology is a groundbreaking advancement in the history of wireless communication systems. It has revolutionized how people access and use mobile broadband networks by providing lightning-fast data transmission speeds, incredibly low latency, and unparalleled network capacity. This globally adopted standard is designed to offer a seamless and immersive user experience, allowing people to enjoy high-definition video streaming, online gaming, and other data-intensive applications on their mobile devices. Key Features of 4G/LTE:

- High-Speed Data Transmission:

  4G/LTE is a wireless communication technology that stands out for its high-speed data transmission capabilities. One of its most remarkable features is its ability to deliver peak download speeds of up to 100 megabits per second (Mbps) and peak upload speeds of 50 Mbps, a significant improvement over its predecessors. Thanks to these speeds, users can enjoy faster internet browsing, smoother video streaming, and quick file downloads. It is worth noting that real-world data rates may vary based on several factors, including network conditions and user demand. For example, if the network is congested, the data rates may be lower than the peak speeds advertised by the service provider. Additionally, the distance between the device and the cell tower, obstacles such as walls or trees, and the number of users sharing the network can also affect the speed and quality of the connection. Nonetheless, 4G/LTE technology has proven to be a game-changer in mobile communication, providing users with faster and more reliable internet connectivity.

- Low Latency:

  4G/LTE networks are cellular networks that provide high-speed data connectivity to mobile devices. These networks are designed to offer low latency, the time it takes for data to travel from one point to another. This feature is crucial for real-time applications such as video conferencing, online gaming, and autonomous vehicles. In LTE networks, the typical round-trip latency ranges from 30 to 50 milliseconds, which ensures responsive and seamless user experiences. This low latency is achieved by using advanced network technologies, such as small cell deployments, QoS mechanisms, and optimized packet routing algorithms. Overall, the low latency of LTE networks plays a critical role in enabling a wide range of real-time applications that require fast and reliable connectivity.

- Enhanced Network Capacity:

  LTE technology has been developed to manage many concurrent users and connected devices efficiently. This advanced network capacity is becoming increasingly crucial as the requirement for mobile data continues to rise, catering to a wide range of devices, including smartphones, tablets, IoT devices, and smart cities. With its high-speed and low-latency capabilities, LTE can handle the growing demand for mobile data and support a broad range of applications, from streaming videos and browsing the web to remote monitoring and controlling many IoT devices.

- Quality of Service (QoS):

  The fourth generation (4G) is equipped with advanced QoS mechanisms that prioritize traffic based on its type and nature. These QoS mechanisms enable the network to allocate the necessary bandwidth and low latency to high-priority traffic, such as voice calls and emergency services, ensuring they receive the required resources to maintain high service quality. This results in a smooth and seamless user experience, as critical applications are given priority over less important traffic, minimizing delays and packet loss.

- Seamless Handovers:

  LTE technology facilitates seamless handovers between different cells or base stations. This allows users to move from one location to another without experiencing any interruption in connectivity. This feature is of paramount importance to people who are often on the move, such as commuters on trains or people traveling in vehicles, as it ensures that they can remain connected to the network without any disruption. With LTE, users can enjoy uninterrupted connectivity, regardless of their location or mode of transportation, making it a critical technology for modern-day communication.

- Advanced Modulation Schemes:

  LTE uses advanced modulation schemes to encode and decode data. These modulation schemes include Quadrature Phase Shift Keying (QPSK), 16 Quadrature Amplitude Modulation (16-QAM), and 64 Quadrature Amplitude Modulation (64-QAM). QPSK is a digital modulation technique that employs four different phase states to represent data, while 16-QAM and 64-QAM use various combinations of amplitude and phase states to encode data. These modulation schemes improve spectral efficiency and data throughput and enable the transmission of more significant amounts of data over the airwaves.

- Spectrum Efficiency:

  LTE employs various techniques to optimize the use of available radio spectrum. One such technique is Orthogonal Frequency Division Multiple Access (OFDMA), which enables multiple users to share the same frequency band by allocating unique subcarriers to each user. This approach reduces interference and maximizes data transmission, enhancing spectral efficiency. By using OFDMA, LTE can allocate resources dynamically and adapt to changing network conditions, ensuring that users receive the best possible service. Overall, LTE's spectrum optimization techniques allow for faster and more reliable wireless communication, enabling users to stay connected wherever they go.

- Evolved Packet Core (EPC):

  The 4G/LTE architecture comprises various elements that work together to provide seamless connectivity to users. One of the critical components is the Evolved Packet Core (EPC), a highly efficient and streamlined core network that facilitates packet-switched data traffic. This architecture enhances network performance and scalability, enabling faster data transfer rates and improved user experience. The EPC is responsible for managing the flow of data packets between the user's device and the internet, making it a critical component of the 4G/LTE network architecture. With its robust and reliable design, the 4G/LTE architecture can provide high-quality connectivity to users, meeting their ever-growing demands for faster and more reliable data transfer.

3.2 5G Technology.

The Fifth Generation (5G) wireless communication technology is a remarkable advancement in telecommunications. It is a breakthrough that has the potential to revolutionize the way we connect, communicate, and interact with the digital world. The technology is designed to provide faster data speeds, lower latency, and higher network capacity than its predecessors, enabling new applications and services. With its ability to support massive machine-type communication, ultra-reliable and low-latency communication, and enhanced mobile broadband, 5G technology is expected to facilitate the development of innovative solutions across multiple industries, from autonomous vehicles and smart cities to remote healthcare and immersive entertainment. This comprehensive overview delves into the multifaceted aspects of 5G technology, including its technical and operational characteristics, the challenges of its implementation, and the opportunities it presents for businesses and consumers alike.

Key Features of 5G:

- Higher Data Rates:

  5G is the latest generation of wireless technology designed to offer remarkably high data rates, with peak speeds reaching up to multiple gigabits per second (Gbps). The immense bandwidth of 5G enables users to experience high-speed downloads, seamless 4K and 8K video streaming, and near-instantaneous access to digital content. With its low latency and higher capacity, 5G also offers improved network reliability and better connectivity, making it a game-changer in wireless communication.

- Ultra-Low Latency:

  5G technology boasts an impressive feature that sets it apart from its predecessors - ultra-low latency. With this technology, communication delays can be reduced to as low as one millisecond (ms), making it crucial for applications that require real-time response, such as augmented reality (AR), virtual reality (VR), remote surgery, and autonomous vehicles. This is because the low latency ensures the data is transmitted and received almost instantly, enabling seamless and uninterrupted interaction between the user and the application. This remarkable feature has made 5G a game-changer in the technology world, paving the way for a new era of innovative and transformative applications.

- Massive Device Connectivity:

  As the world becomes increasingly connected, the importance of network scalability has become more evident than ever before. The latest 5G networks are designed to accommodate an astounding number of connected devices, far surpassing the capabilities of previous 4G networks. This level of scalability is critical for the proliferation of the IoT, which is quickly becoming an essential part of modern life. With 5G networks in place, smart cities, smart homes, and industrial automation can all become a reality, making life more convenient, efficient, and productive. The new technology frontier is upon us, and 5G is at the forefront, leading the way.

- Network Slicing:

  One of the most significant advancements brought by 5G technology is the introduction of network slicing, which enables network operators to create customized virtual networks tailored to specific applications or industries. This innovative approach ensures that the network resources are allocated more efficiently and critical services receive dedicated resources that can help optimize the network performance. With network slicing, different virtual networks can be created within a single physical network infrastructure, each with its unique characteristics and functionalities. This means that network operators can allocate resources more effectively, providing the required service quality and performance level for different applications and use cases.

- Beamforming and Massive MIMO:

  5G networks utilize advanced antenna technologies such as beamforming and Massive Multiple Input, Multiple Output (MIMO) systems. These techniques improve the signal strength, coverage, and spectral efficiency of the network, resulting in a significant enhancement in overall network performance. With beamforming, 5G networks can focus the signal directly towards the device rather than broadcasting the signal in all directions. This leads to better signal quality, reduced interference, and improved battery life. Meanwhile, MIMO systems utilize multiple antennas to transmit and receive data, allowing faster data transfer rates and increased capacity. Overall, the combination of these advanced antenna technologies in 5G networks delivers a seamless, reliable, and high-speed mobile experience.

- Millimeter Wave (mmWave) Spectrum:

  The fifth-generation wireless technology, 5G, utilizes higher-frequency millimeter wave (mmWave) bands, which offer vast bandwidths and enable faster data rates. However, mmWave signals have shorter propagation ranges due to their high frequency, which makes them susceptible to obstructions such as buildings, trees, and even rain. This necessitates advanced infrastructure deployment, including installing more cell sites and using advanced antenna technologies such as beamforming and massive MIMO to overcome these propagation challenges.

- Security Enhancements:

  The fifth generation of cellular technology is a significant leap forward regarding security. It comes equipped with robust security measures that work together to provide enhanced encryption, authentication, and network-slicing isolation. These features offer a higher level of protection for user data and ensure the integrity of the network. 5G's enhanced encryption uses advanced algorithms to secure data transmission, making it more difficult for hackers to intercept and decipher the data flow. The authentication process ensures that only authorized users can access the network, while network slicing isolation adds another layer of protection by allowing virtual networks to be created within the main network infrastructure. These virtual networks are isolated, making it more difficult for an attacker to move from one part of the network to another. 5G's security features make it one of the safest and most secure cellular technologies available today.

- Energy Efficiency:

  The latest 5G networks have been designed to be highly energy-efficient, surpassing the capabilities of their predecessors. This is a crucial development in the world of technology as it not only helps to prolong the battery life of devices but also plays a significant role in reducing the carbon footprint of network operations. By consuming less power, 5G networks can operate at a lower cost, allowing for more sustainable and eco-friendly practices in the telecommunications industry. This innovation is a significant step forward in our efforts to reduce our impact on the environment while still enjoying the benefits of cutting-edge technology.

- Edge Computing:

  With the advent of 5G technology, edge computing has become possible. This means that data processing can occur closer to the data source, which has multiple advantages. One of the most enormous benefits is a significant reduction in latency, resulting in faster and more efficient data processing. Additionally, edge computing enhances the capabilities of various applications, including real-time analytics, IoT, and autonomous systems. Edge computing has revolutionized how we process and use data by enabling these applications to perform faster and more effectively.

- Global Standardization:

  The technology behind 5G is built on global standards established by renowned organizations such as the 3rd Generation Partnership Project (3GPP). These standards are designed to ensure seamless interoperability and consistent performance across a wide range of diverse network deployments worldwide. This means that regardless of where you are, you can expect to experience the same high-level performance and connectivity that 5G is known for. The standards also ensure that all devices that support 5G can communicate with each other, regardless of the manufacturer or service provider. This promotes a universal standard for 5G technology, making it easier for users to switch between providers and devices without any compatibility issues.

3.3 Satellite Communication:

Satellite communication is a sophisticated technology that enables transmitting data, voice, and video signals over vast distances via communication satellites orbiting the Earth. These artificial satellites are positioned in geostationary or low Earth orbits and serve as relays, receiving signals from ground-based stations, amplifying them, and retransmitting them to other ground stations or directly to satellite dishes, providing global connectivity. This technology is crucial in various sectors, including telecommunications, broadcasting, navigation, and scientific research. It offers numerous advantages, such as comprehensive coverage, resilience to natural disasters, and the ability to reach remote and inaccessible areas. Satellite communication has also revolutionized global connectivity, making it possible for people worldwide to communicate, access the internet, and receive real-time information, bridging geographical gaps and fostering worldwide communication and collaboration.

- Orbital Configurations:

  Satellite communication networks utilize different types of satellites positioned in various orbital configurations. Geostationary orbit (GEO) satellites are placed approximately 36,000 kilometers above the Earth's equator. Due to their high altitude, they can provide continuous coverage of fixed geographic areas. Medium Earth orbit (MEO) and low Earth orbit (LEO) satellites, on the other hand, are positioned at significantly lower altitudes and require more prominent constellations of satellites to function effectively. MEO satellites are located at around 2,000 to 35,000 kilometers, while LEO satellites operate at an altitude between 160 to 2,000 kilometers. Although MEO and LEO satellites require more satellites to provide global coverage, they offer lower latency and broader coverage. This makes them more suitable for applications such as internet connectivity and real-time monitoring, where high-speed and reliable connectivity are critical.

- Satellite Uplinks and Downlinks:

  Satellite communication is a complex and sophisticated process that relies on two primary links - the uplink and the downlink. The uplink link involves transmitting data from Earth-based ground stations to the satellite, while the downlink is responsible for relaying the data back to ground stations. These links rely on microwave frequencies and advanced modulation techniques that enable efficient and reliable data transmission. Satellite communication is essential in modern times as it allows for the transmission of vast amounts of data over long distances, making it possible to connect people and communities across the globe.

- Frequency Bands:

  Satellite communication operates in various frequency bands, including C-band, Ku-band, Ka-band, and V-band. The choice of frequency band depends on factors such as signal propagation characteristics, bandwidth requirements, and interference mitigation strategies.

- Signal Propagation: Understanding signal propagation through the ionosphere and troposphere is essential for optimizing satellite communication systems. Factors like rain fade, signal attenuation, and atmospheric interference must be considered to ensure reliable and efficient communication.

3.3.1 Advantages and Challenges

Satellite communication offers several critical advantages, including global coverage, resilience to terrestrial infrastructure failures, rapid deployment in remote and disaster-stricken areas, and multicast capabilities, making it ideal for broadcasting, emergency services, and military applications. Despite its merits, satellite communication faces challenges such as latency due to signal travel time, susceptibility to signal interference

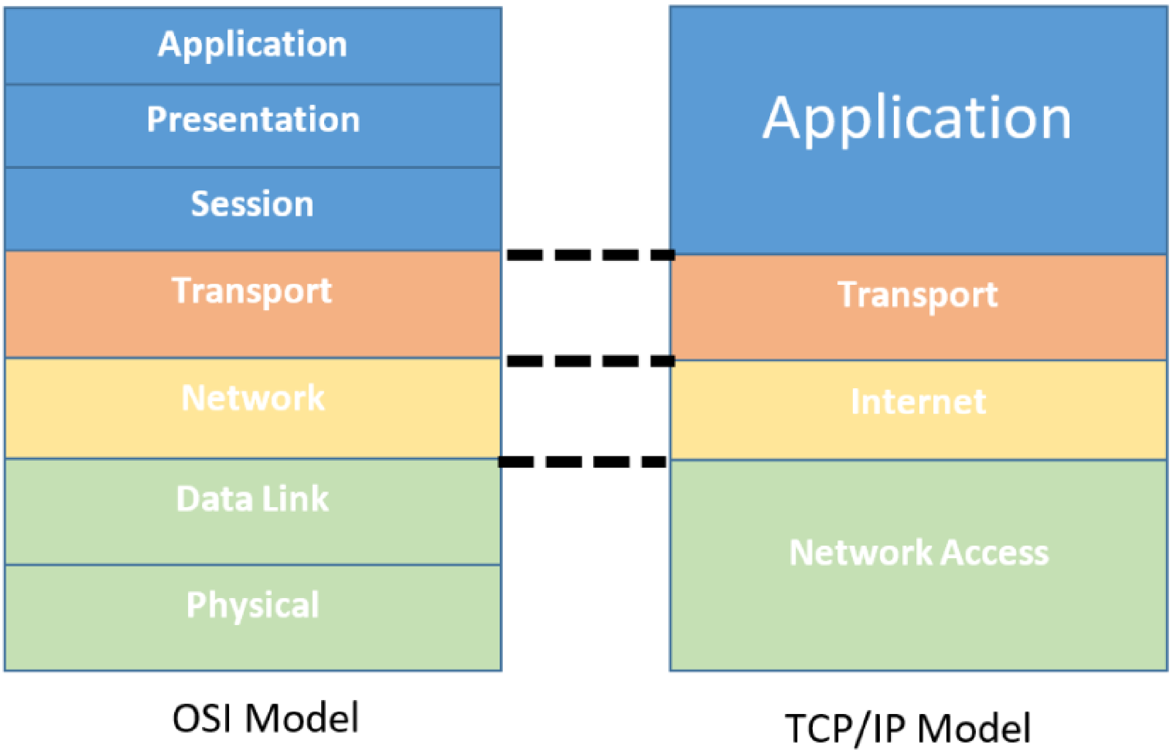

Figure 3.2: Two layered architecture models OSI and TCP/IP [8]

and degradation, high initial infrastructure costs, and the limited availability of orbital slots and frequency spectrum.

4 Cross-layer approach

As seen in the seven-layer Open Systems Interconnection (OSI) model or the TCP/IP model depicted in Figure 3.2, the concept of a layered architecture involves dividing the complex network functions into distinct layers. Each layer provides a specific set of services and functionality within the overall network framework. Protocols are designed to implement these services at each layer. One key characteristic of a layered architecture is the restriction on direct communication between non-adjacent layers. This architecture also limits communication between adjacent layers to procedure calls and corresponding responses. Any deviation from this structured approach, such as allowing direct communication between non-adjacent layers or sharing variables between layers, violates the layered architecture and is known as cross-layer design.

Cross-layer design in the context of protocol design involves actively exploiting the interdependencies between different protocol layers to enhance performance. This approach enables one layer to leverage specific design elements of another layer, even if they are not directly adjacent in the protocol stack. Cross-layer design can be viewed as a collaborative strategy wherein adaptation is synchronized across multiple layers within a communication system. Cross-layer designs initially involved integrating two adjacent layers to fulfill similar functions. An illustrative instance is the integration of the PHY and MAC layers.

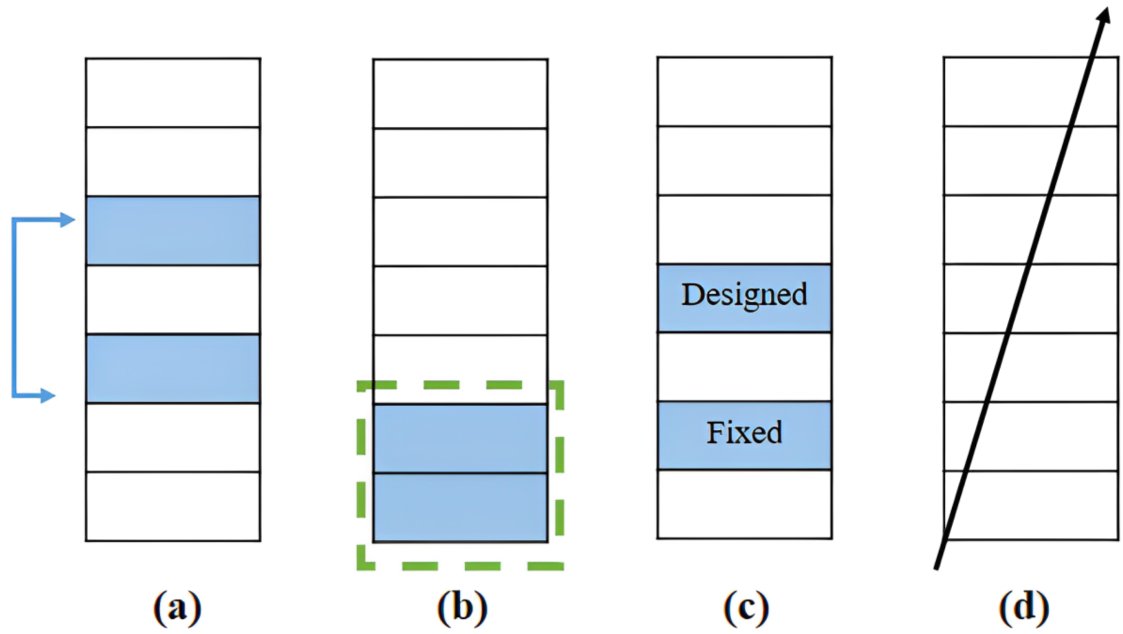

Figure 3.3: The different inter-layer designs [14].

4.1 Overview of design proposals a cross-layer system

The existing literature contains various proposals for cross-layer designs. The authors have conducted a comprehensive study of multiple cross-layer design proposals from the literature, with a specific emphasis on the interconnections between the layers. Our focus is on understanding the intricate connections between these layers. It's important to highlight that the layered architecture can be customized in numerous different ways.

- Creation of new interfaces 3.3.a,
- Fusion of adjacent layers 3.3.b,
- Design coupling without new interfaces 3.3.c,
- Vertical calibration across multiple layers 3.3.d.

Most cross-layer design proposals in the literature fall into one of these categories. Figure 3.3 illustrates the different designs. We will now discuss the four categories in more detail. However, the identified architectural violations can be combined to create more complex cross-layer designs [14].

4.1.1 Creation of New Interfaces:

Numerous cross-layer designs suggest novel interfaces between different layers [119]. These fresh interfaces transmit information between layers while the program is running. The introduction of a new interface violates the traditional layered architecture. This particular category can be subdivided into three subcategories based on the direction of information flow along the new interfaces: upward, downward, and bidirectional.

- When a higher-layer protocol needs information from one or more lower layers while running, a new interface is created from the lower layer(s) to the upper layer. For example, suppose there's a wireless link in the end-to-end TCP path. In that case, errors on the wireless link can cause the TCP sender to make incorrect assumptions about network congestion, which can result in degraded performance. To address this, interfaces are created from the lower layers to the transport layer to provide explicit notifications. An example is explicit congestion notification (ECN), where the router sends explicit indications to the TCP sender about network congestion, helping the sender differentiate between wireless link errors and network congestion.

- Cross-layer design proposals often involve setting parameters at the lower layer of the technology stack in real-time using a direct interface from the upper layer. For instance, applications can communicate their delay requirements to the link layer, allowing it to prioritize processing packets from delay-sensitive applications. Upward information flow is used to inform upper layers about the underlying network conditions, while downward information flow guides lower layers on handling application data.

- In bidirectional information flow, two layers collaborate to carry out different tasks. This collaboration typically involves an iterative exchange of information between the layers. In this scenario, the architecture is violated by the presence of two complementary new interfaces.

4.1.2 Merging of Adjacent Layers:

In cross-layer design, a different approach is to combine two or more neighboring layers to form a new super-layer. This super-layer would then offer a merged set of services, incorporating those provided by the individual constituent layers. This approach doesn't necessitate the introduction of new interfaces within the stack. From an architectural viewpoint, the super-layer can interact with the remainder of the stack by utilizing the existing interfaces in the original architecture.

4.1.3 Design Coupling Without New Interfaces:

Another category involves integrating multiple layers during the design phase without adding new interfaces for sharing information during execution. This approach can incur architectural costs because replacing one layer may require corresponding changes in another layer, even though no new interfaces are created.

4.1.4 Vertical Calibration Across Multiple Layers:

The final category focuses on adapting cross-layer design proposals to align with specific standards, which is known as vertical calibration across layers. This process entails making adjustments to parameters that extend across multiple layers. The main idea behind this approach is that the performance experienced at the application level is influenced by the parameters of all the underlying layers. As a result, fine-tuning these parameters collectively could potentially lead to improved overall performance.

5 Proposed Scheme

This section will delve into the proposed system [13], beginning with an in-depth discussion of MDC-based video encoding. Next, we will explore the enhanced version of the MPQUIC protocol, focusing on the modifications incorporated into its new scheduler. Lastly, we will analyze the error compensation block and the process of reconstructing the two descriptions in order to produce the received video.

Expanding on the methodologies described previously, we are introducing an innovative approach to improve the quality of streaming videos and reduce latency in vehicular environments. Our system utilizes the MDC technique to address the challenges posed by inaccurate arrival-time estimates due to network jitter and packet loss. This technique involves splitting the desired video into two distinct descriptions before transmission through the network. The first description comprises the even frames of the video, while the second description comprises the odd frames. Each description is assigned a different priority – one is designated as "high priority," and the other as "low priority". It's important to note that both sequences possess equal importance and data size within their respective descriptions.

The two descriptions both demonstrate lower but still acceptable quality, and when they are reconstructed at the receiving end, the resulting quality is higher. This approach improves the overall transmission quality and boosts error resilience by ensuring that successive frames are not affected by the same errors. In the context of this paper, the prioritization of the sequence doesn't matter due to their similarity, so the even sequence is selected as the high-priority level. Once these two descriptions are generated and one is prioritized over the other, an HEVC encoder encodes them using a fixed QP and an AI configuration to minimize the occurrence of errors in consecutive frames.

Our system incorporates MDC technology, complemented by a modified extension of the MPQUIC protocol. This combination ensures the reliable transmission of one of the generated high-priority data descriptions while deliberately sacrificing the transmission of the other description in cases of network constraints. The modified MPQUIC protocol facilitates data transmission using hybrid mechanisms across heterogeneous networks, such as satellite and LTE networks, allowing each data sequence to be transmitted independently. The 5G and LTE links were specifically designed to provide a reliable transmission mechanism. The 5G network is an ideal choice for streaming video applications due to its exceptional capacity, high data rate, low latency, and minimal packet loss rate. However, its limited coverage area may impact its overall performance. On the other hand, the LTE network offers a broader coverage area but suffers from reduced capacity, lower data rate, increased latency, and higher packet loss rate. The combination of these two networks within a single connection allows each network to compensate for the weaknesses of the other. This approach is utilized to ensure the reliable transmission of high-priority sequences. In contrast, non-priority sequences are transmitted via satellite communication due to its extensive coverage area, making it an excellent option for reaching patients residing in remote or isolated areas. In summary, the modified MPQUIC protocol checks the priority level of each transmitted packet. Highly prioritized packets are transmitted reliably through either the 5G or LTE networks, while non-priority packets are transmitted unreliably via the satellite link.

The objective is to ensure the reliable transmission of high-priority medical video

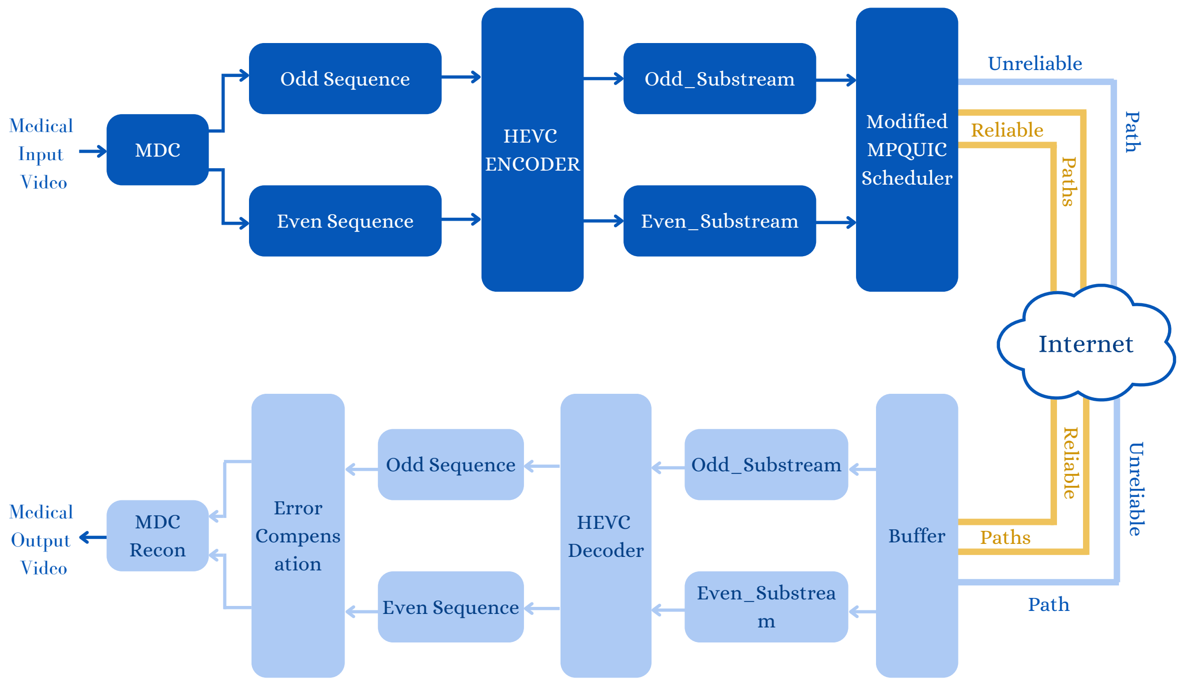

Figure 3.4: Adaptation scheme for streaming medical videos hybridly over heterogeneous paths [13].

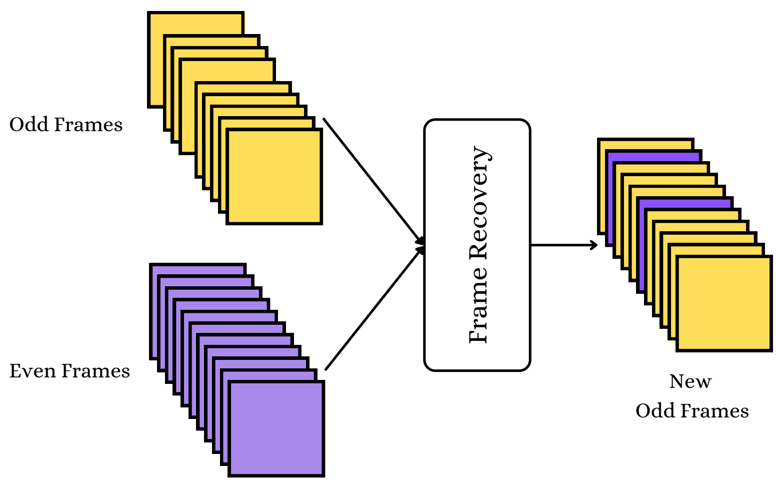

Figure 3.5: Error Compensation for lost frames [13].

descriptions to enhance streaming video quality in vehicular environments. Within multipath transport protocols, schedulers play a crucial role in determining the optimal path for packet delivery based on packet loss rate, delay, and available window size. In the context of MPQUIC, packets can contain either a single frame or multiple frames, such as stream and window-update frames, which carry encrypted payload or control information. Tommy Pauly introduced the DATAGRAM frame, a new type designed to facilitate the unreliable transmission of data within a QUIC connection. This feature was subsequently integrated into the MPQUIC protocol to enable the non-priority descriptions to be sent unreliably over the network using a satellite link. Should packet loss be detected during transmission, the new scheduler disregards it, opting to send the subsequent packet without ensuring the proper reception or retransmission of the lost packet. This approach aims to reduce the overall number of retransmissions and minimize transmission delays.

To ensure the reliable transmission of even-sequence packets, the system will assign frames of type STREAM and prioritize transmitting them through reliable links. This frame type requires retransmission upon loss detection, which helps maintain the data's integrity. On the other hand, for the odd-substream packets, the system will use frames of type DATAGRAM, designed to minimize latency by avoiding the retransmission of lost packets. This approach helps optimize data delivery while prioritizing real-time communication for these packets.

The proposed scheduler is designed to guarantee the reception of the high-priority sequence without any loss by sending them reliably through 5G/LTE cellular networks. This ensures that critical data is delivered with minimal risk of loss or corruption, meeting the demands for reliability and quality of service. In contrast, the other sequence will be sent unreliably through the satellite link, acknowledging the inherent challenges of satellite communication and adjusting the transmission strategy accordingly to account for potential packet loss.

The SendPacket function plays a crucial role in transmitting packets along the appropriate path. When transmitting a DATAGRAM, the scheduler sends it directly without ensuring its reception through the satellite link. In the event of any loss of a DATAGRAM packet, there is no provision for retransmission. Conversely, if the frame is a STREAM, the scheduler will transmit it reliably through the remaining paths, such as the 5G and LTE networks. Before transmission, the scheduler verifies and rectifies any lost packets. To send the next packet, the sender must adhere to the Max-Ack-Delay time to receive the ACK packet. If the ACK packet does not arrive, indicating loss, the packet will be deemed lost and retransmitted.

The proposed algorithm selects two out of three paths to establish reliable connections, resulting in longer but higher-quality communications. The third path will be designated for unreliable connections with lower latency and degraded quality. The substreams will be received separately via different paths on the receiver side. Each substream will be decoded using the HEVC decoder. Additionally, a frame recovery algorithm will compensate for lost frames by copying them from the reliably transmitted frames, thus enhancing the overall quality. The general mechanism is depicted in Figure 3.4, while Figure 3.5 illustrates the adaptation scheme of the system designed and utilized for a realistic investigation. Furthermore, Algorithm 1 clearly outlines our approach's process in a simple pseudocode.

Function SendPacket

```
/* Check the frame type                                          */
if packet.frame == Datagrame then
  | SendDatagrame(packet, UnrelPath)
else if packet.frame == Stream then
  | // Check for Retransmission first
  | HasRetrn ← HasFramesRetransmission()
  | if HasRetrn == true then
  |   | Dealing with retransmissions before sending any data
  | else
  |   | RelPath ← SelectPath() sendpackedpacket(packet, RelPath)
  | end
end
```

Algorithm 1 Pseudocode for Streaming Medical Video using a Modified MPQUI Combined with MDC [13].

Input: Medical video.
Output: Received medical video.
Preprocessing:
1: Divide the video into two descriptions using MDC.
2: Mark one description as "high priority" and the other as "low priority."
3: Encode both descriptions using the HEVC encoder.
Sender:
4: Initialize MPQUIC connection.
5: Send high-priority descriptions reliably through the 5G and LTE cellular networks.
6: Send low-priority description over the satellite network.
Receiver:
7: Receive high-priority description.
8: Decode it using the HEVC decoder.
9: Receive low-priority description.
10: Decode it using the HEVC decoder.
Error Compensation:
11: Implement Error Compensation for lost frames.
Combine descriptions:
12: Combine the decoded high-priority and low-priority descriptions using MDC techniques to generate the total received medical video.

6 Framework and simulation set-up

6.1 Environment setup

The proposed approach entails an innovative system comprising an ambulance and a remote hospital that are interconnected via three different networks, each with unique characteristics. Two networks were dedicated to providing a consistent and dependable connection to ensure reliable connectivity, namely the 5G and LTE networks. In contrast, the third network, which was the Satellite network, was used to provide an unreliable and less consistent connection. Realistic experimentation was carried out to validate the efficacy of the proposed solution, especially under challenging conditions. Specifically, we assumed that the connection between the connected ambulance and a mobile medical department was successfully established. Another scenario we considered was in the event of an accident, where a car could send comprehensive accident information to the connected ambulance to enable remote treatment.

To emulate network conditions in a Mininet-Wifi [95] environment, we utilized Netem [97] as our tool of choice. Mininet-WiFi is an extension of the Mininet network emulator that allows for virtual WiFi networks by incorporating virtualized WiFi Stations and Access Points. Additionally, this extension supports the creation of VANET networks. The pre-configured map is already integrated with Mininet-WiFi, so the remaining step is running the VANET simulation. By default, each car in the simulation has only one wireless interface dedicated solely to V2V communication. However, we decided to increase the number of interfaces to three to replicate real-world conditions more accurately. To simulate realistic network conditions further, we connected the sender and receiver to different networks - specifically, 5G, LTE, and satellite connections. To determine the chosen characteristics for each path, we consulted several sources, including [98], [99], [100], and [101]. The characteristics were grouped and organized in Table 3.1.

	Bandwidth (Mbps)	Delay (ms)	Loss Rate (%)
LTE	50	50	0.5
5G	1000	1	0.1
LEO Satellite	100	3	0

Table 3.1: Different paths characteristics

6.2 Video encoders

In order to ensure high-quality video encoding, we have carefully evaluated various options and have ultimately decided to use the latest joint video coding standardization project of the ITU-T Video Coding Experts Group (ITU-T Q.6/SG 16) and ISO/IEC Moving Picture Experts Group (ISO/IEC JTC 1/SC 29/WG 11), which is known as the HEVC standard. This standard significantly improves its predecessor, H.264, in terms of compression efficiency and quality (See Section 3.8.7 for more information).

To implement HEVC, we have utilized the open-source reference software, the HEVC Test Model (HM) [120]. This software provides a comprehensive and reliable toolset for encoding and decoding HEVC-compliant video streams. We have extensively tested the HM software and found it highly accurate and efficient, enabling us to achieve excellent results regarding video quality and compression ratios. Overall, we are confident that using the HEVC standard and the HM software allow us to provide our users with the best possible video experience.

Throughout our thesis, we utilized HM version 16.24 [28] in conjunction with FFMPEG [102] to decode specific binary files that could otherwise be decoded with HM due to data loss. Using these two powerful software tools, we were able to carry out a range of video stream modifications, including adjustments to the GOP structure, QP, type of temporal prediction, output video bitrate, and much more. Our ability to perform such extensive modifications was crucial in enabling us to achieve the desired results for our research.

6.3 Video description

In order to gain insight into the various effects in realistic scenarios, selecting suitable sequences for study is crucial. Video resolution is intricately linked to the intended application of the video. For this reason, the Video Coding Collaboration Team (JCT-VC) has developed a classification system to aid in selecting sequences for processing. The test video sequences are classified into six distinct classes, ranging from Class A to Class F, each with unique characteristics that make it suitable for specific research purposes. This method ensures that researchers can accurately and efficiently analyze the effects of video coding on different types of content [121].

Class A sequences are a particular type of video sequences utilized in various applications, such as reducing computation time or evaluating the performance of 4K/8K video encoding. These sequences have a remarkably high resolution that surpasses the standard high-definition 1080p. Specifically, Class A sequences have a resolution of 2,560 x 1,600 pixels, considerably higher than traditional video sequences' resolution. Due to their high resolution, Class A sequences are ideal for applications that demand exceptional clarity and detail. For instance, they can be leveraged in scientific research, medical imaging, and video encoding to achieve unprecedented detail and precision. Class B is a standardized method for evaluating 1080p High Definition Television (HDTV) performance. The image size for Class B is set to 1920×1080 pixels, which means that the picture is made up of over 2 million pixels. This evaluation class is typically used to assess the quality of larger screen displays, such as those found in homes or public viewing areas.

On the other hand, Classes C and D are designed to evaluate the performance of mobile video applications. Class C is intended for evaluating the video quality of mobile devices with an image resolution of 832×480 pixels. Meanwhile, Class D assesses the video quality of even lower-end mobile devices with an image resolution of 416×240 pixels. These classes help evaluate the performance of mobile devices, which have become increasingly popular for video streaming and other types of multimedia content. Class E test sequences have an image size of 1280×720 pixels and are specifically designed for assessing coding performance in low-latency applications such as video surveillance or interactive visual communications. These test sequences ensure that the encoded video signals are transmitted and received with minimal delay, making them ideal for real-time applications. Class F sequences evaluate the coding performance of non-camera-generated content like text and graphics. These test sequences are essential for determining the efficiency of video coding algorithms in accurately preserving the details of such content. By assessing the coding performance of camera and non-camera-generated content, these test sequences help ensure that video compression standards are optimized for a wide range of applications.

Furthermore, ITU-T Recommendation P.910 suggests that when selecting test sequences, comparing the relative spatial information and temporal information found in the different available sequences [122] can be helpful. Generally, the difficulty of compression is directly related to a sequence's spatial information (SI) and temporal information (TI). SI is a measure that typically indicates the amount of spatial detail in an image and is generally higher for more spatially complex scenes. On the other hand, perceptual temporal information (TI) is a measure that indicates the number of temporal changes in a video sequence and is generally higher for sequences with high motion. If a small number of sequences were used in a given test, choosing sequences that cover a wide range of the spatiotemporal information plane could be essential.

To meet the ITU-T's requirements [123] [124], its recommended YUV videos were used for evaluation. In Table 3.2, videos were classified into three different classes as follows:

Class	Video Name	Resolution
C	BasketballDrill	832x480
B	BasketballDrive	1920x1080
A	Traffic	2560x1600

Table 3.2: Used videos

6.4 Video Quality Assessment Metrics

The PSNR has been the most widely used metric for assessing the quality of images and videos for a long time. It compares the original and received images or videos pixel-by-pixel and then measures their differences [125]. However, while the PSNR has been quite popular, it does not always accurately reflect the quality of an image or video as perceived by human observers. This is because the PSNR only considers the difference between the original and received pictures or videos and does not account for the structural components of the image.

To address this issue, the SSIM has emerged as a popular alternative for assessing the quality of still video images [126]. Unlike the PSNR, the SSIM analyzes the structural components of a pixel, such as luminance, contrast, and structure, to measure the image distortion. This approach provides a more comprehensive view of the image quality from a human perspective. The SSIM score ranges from 0 to 1, with a score closer to 0 indicating less image distortion [104]. This score is calculated by comparing the original and received images or videos' luminance, contrast, and structure. Considering these factors, the SSIM provides a more accurate image quality assessment, making it a popular choice for many image and video quality assessment applications.

The VMAF has emerged as a cutting-edge method in video quality assessment. Developed by Netflix, VMAF leverages the power of machine learning algorithms to combine several elementary metrics into a final metric that predicts subjective video quality. Unlike traditional metrics such as PSNR and SSIM, VMAF compensates for the lack of any complete reflection of the subjective view of the human eye, making it a more reliable and accurate method for video quality assessment. With its ability to consider various factors influencing video quality, VMAF has become increasingly popular among video streaming services and content creators who aim to deliver the best possible video experience to their audiences [106, 127, 128].

7 Performance evaluation

Firstly, before presenting the simulation results, the most suitable scheduler must be selected to decide which existing state-of-the-art schedulers perform better in a vehicular environment. This could be done by evaluating the mentioned schedulers in Section 1.4 by comparing each schedule's latency with others when streaming videos in a vehicular environment. Then, the simulation results are presented and discussed for various scenarios based on different metrics: completion time, data rate, packet loss, PSNR, SSIM, and VMAF.

7.1 Schedulers evaluation in vehicular environments

An assessment was carried out to determine the most appropriate scheduler for data scheduling over varied paths in vehicular environments using the MPQUIC protocol. The evaluation was performed between four schedulers, namely minRTT [84], Blest [76], ECF [75], and Peekaboo [74]. The results obtained from streaming recommended videos using one of the mentioned schedulers implemented in MPQUIC are presented in Table 3.3.

	MinRTT	Blest	ECF	Peekaboo
BasketballDrill	19	25	20	19
BasketballDrive	60	69	60	58
Traffic	65	66	65	64

Table 3.3: Completion Time of streaming videos using different schedulers in vehicular environments.

According to the information presented in Table 3.3, it can be observed that the performance of the schedulers in vehicular environments was quite similar, except for the Peekaboo scheduler. This particular scheduler exhibited a completion time that was a few seconds faster than the others. Given this finding, the Peekaboo scheduler was selected as the proposed approach. This scheduler will efficiently and reliably schedule data transmission through cellular networks, specifically 5G/LTE.

7.2 Simulation results

Three distinct scenarios were considered to assess the suggested methodology thoroughly. In each scenario, the sender initiates the process by transmitting three video files, beginning with the lowest resolution and gradually progressing toward the highest. This sequential approach enables a comprehensive evaluation of the proposed approach, providing valuable insights into its feasibility and efficacy.

First Scenario: Streaming video using multipath QUIC through 5G and LTE networks.

Second Scenario: sending data through satellite link using Unreliable QUIC.

Third Scenario: streaming data using the proposed approach explained in Section 5.

7.3 Completion time assessment

As illustrated in Figure 3.6, the completion time measures reveal that the first scenario, which employs MPQUIC as the transport protocol to transmit the entire data, exhibits the poorest performance in terms of completion time. The time duration for sending the video ranged from 19 seconds to 68 seconds, depending on the type of video being transmitted. It is a well-known fact that MPQUIC lacks a reliable lost recovery mechanism, which results in the retransmission of packets in the event of packet loss, thereby increasing latency. This, in turn, leads to increased completion time. Additionally, MPQUIC's efficiency is hampered by high mobility, which elevates the rate of data loss and leads to connectivity issues. Therefore, it is not surprising that MPQUIC performed poorly in this scenario.

It's interesting to note that the unreliable QUIC protocol lacks retransmission data upon loss detection. As a result, the second scenario had the most favorable outcomes regarding completion time. The time it took for the video to be sent varied between 10 to 21 seconds, depending on the specific video being sent. This finding highlights the importance of considering the specific protocol used when evaluating network performance.

In the third scenario, we evaluated our proposed approach, which combines satellite communication and cellular technologies. Our core concept is based on the implementation of multiple description coding technologies, which generate two distinct descriptions of the data. These descriptions are then packetized individually using different mechanisms, allowing efficient traffic dispersion and load balancing in the network. Ultimately, this approach significantly reduces latency, leading to an increase in overall network efficiency.

To illustrate the effectiveness of this approach, we used the BasketballDrill transmission as an example. The transmission was completed in 10 seconds, similar to the second scenario's completion time. We analyzed the results and found that each video arrived at its destination between 10 to 28 seconds, depending on the substream file size, as depicted in Figure 3.6. Overall, our proposed approach holds significant potential for optimizing network performance and reducing latency in various applications.

	SSIM		
	BasketballDrill	BasketballDrive	Traffic
First Scenario	0.98	0.99	0.99
Second Scenario	0.55	0.67	0.69
Third Scenario	0.94	0.93	0.93
	VMAF		
	BasketballDrill	BasketballDrive	Traffic
First Scenario	88	91	82
Second Scenario	26	35	43
Third Scenario	82	82	78

Table 3.4: SSIM and VMAF results

7.4 Video quality assessment

Video quality assessment is an integral part of video transmission and reception. One widely used metric for this purpose is the PSNR. The results of the PSNR evaluation are presented in Figure 3.7. The recorded measurements from the first scenario indicate that when we used MPQUIC to send data, it outperformed the other scenarios

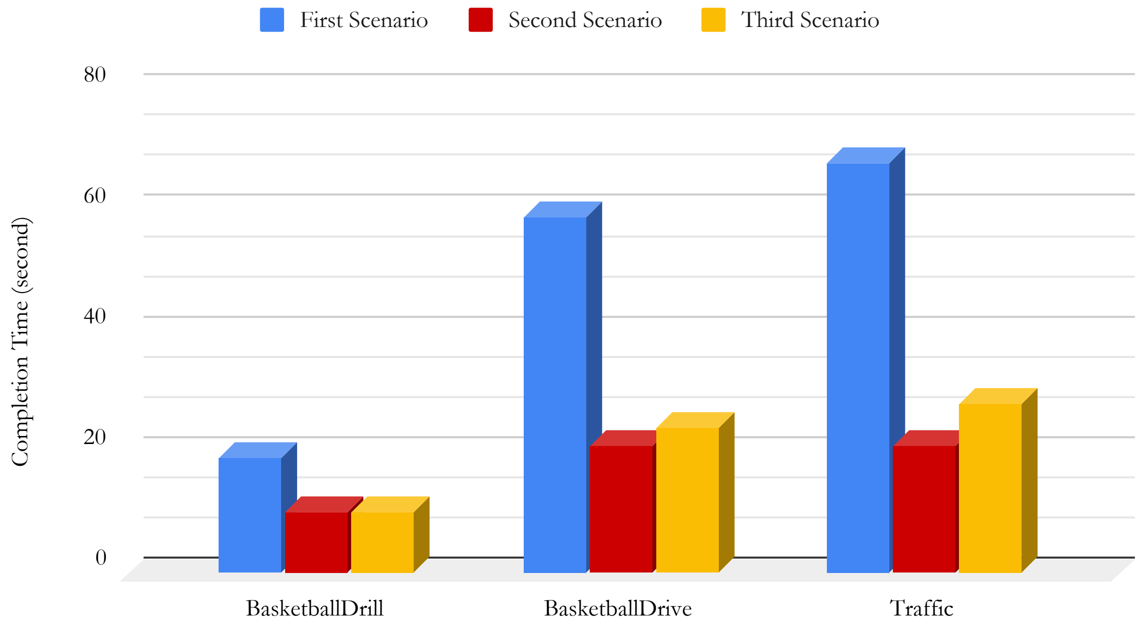

Figure 3.6: Streaming completion time

regarding video quality. However, it is essential to note that achieving high video quality comes at the cost of taking the highest completion time. This means that while we may have excellent video quality, we may also experience an increase in latency. In mobile health applications, minimizing latency is a critical challenge that must be addressed. A few seconds delay could make a significant difference and could even save lives in real-life situations. Therefore, it is essential to balance video quality and latency to ensure that we can provide the best possible service to end-users.

The second scenario did not provide any guarantees for receiving all the necessary data amounts or for reordering the arrived packets. As a result, the recorded video quality was subpar, with PSNR values ranging from 20dB to 25 dB. It is important to note that this lack of guarantees and mechanisms for data delivery can significantly impact the overall quality of the recorded video.

The third scenario, which utilized the proposed scheme, showed a significant improvement in video quality compared to QUIC. The improvement ranged from 36% to 70%, and the PSNR for each video was recorded at 34dB. The MDC played a vital role in enhancing error resilience, where an acceptable quality was received as long as the same error did not affect the two substreams. Furthermore, the reliable side of the proposed scheme compensated for missed frames on the unreliable side. To further validate the results, the study presented a table that showed how the proposed scheme improved the PSNR results. The table supported the notion that the proposed scheme could provide better video quality than traditional streaming protocols.

Finally, the study presented a sample frame from the received video, where each column represented a scenario and each row represented a specific video. The sample frame showed that the second scenario had the worst quality, while the first and third gave almost the same quality. Overall, the proposed scheme showed promising results and could be a viable solution for streaming high-quality videos over unreliable networks.

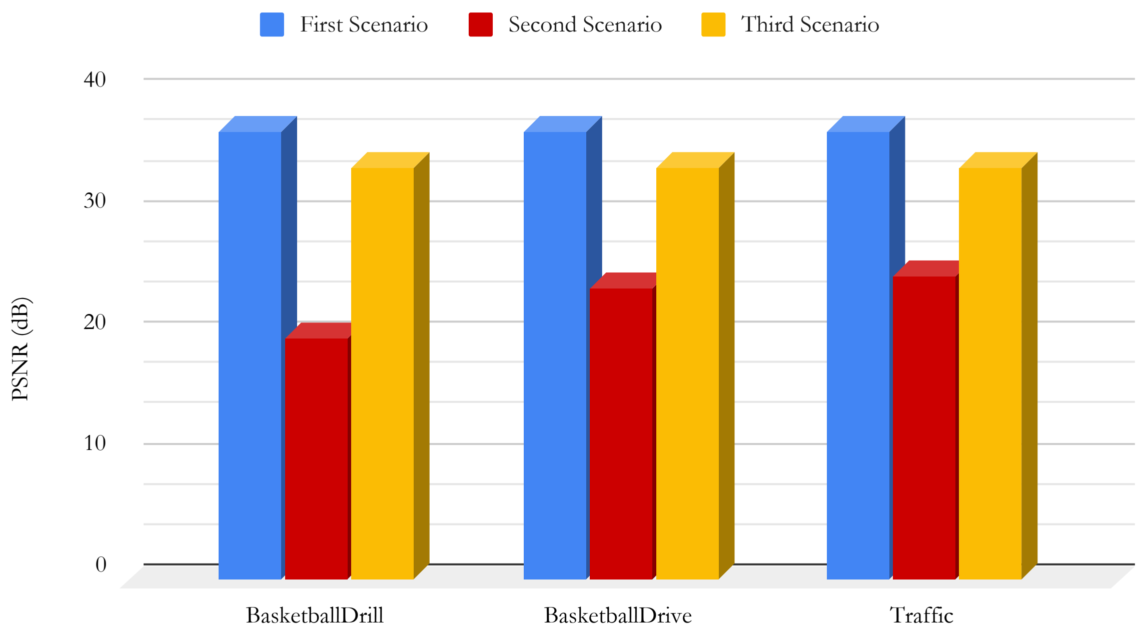

Figure 3.7: PSNR values of different videos

Through the utilization of the modified MPQUIC and the integration of multiple description coding techniques, the suggested approach has been shown to enhance streaming video quality by a remarkable 70% across various transmission resolutions. Moreover, it has been observed to significantly diminish network latency by up to 60%, thereby providing a seamless and improved streaming experience.

7.5 Receiving data rate and packet loss assessment

The data presented in Table 3.5 compares transmitted and received data amounts for different scenarios. Regardless of the scenario, the transmitted data remains consistent, while the received data varies. When utilizing the classical MPQUIC protocol to stream videos over 5G and LTE cellular networks, the first scenario demonstrates that all data reaches the destination correctly due to the protocol's robust data recovery mechanism. This allows for lossless transmission even in challenging vehicular environments characterized by high dynamicity and mobility, leading to increased packet loss rates. However, the unreliable QUIC protocol faces data loss during video streaming as it lacks a reliable data recovery mechanism. This results in packet loss percentages ranging from 0.5% to 2.2% for different videos. The unreliable QUIC protocol experiences the highest packet loss rate when streaming traffic sequences due to a large amount of data transmission, which is more heavily impacted by the demanding conditions in vehicular environments. After analyzing the data presented in Figures 3.9, 3.10, and 3.11, we have observed that our proposed approach effectively minimizes packet loss rates in comparison to the unreliable QUIC measurements, this reduction ranges from 20% to 50% owing to the reliable links, which employ a mechanism to detect

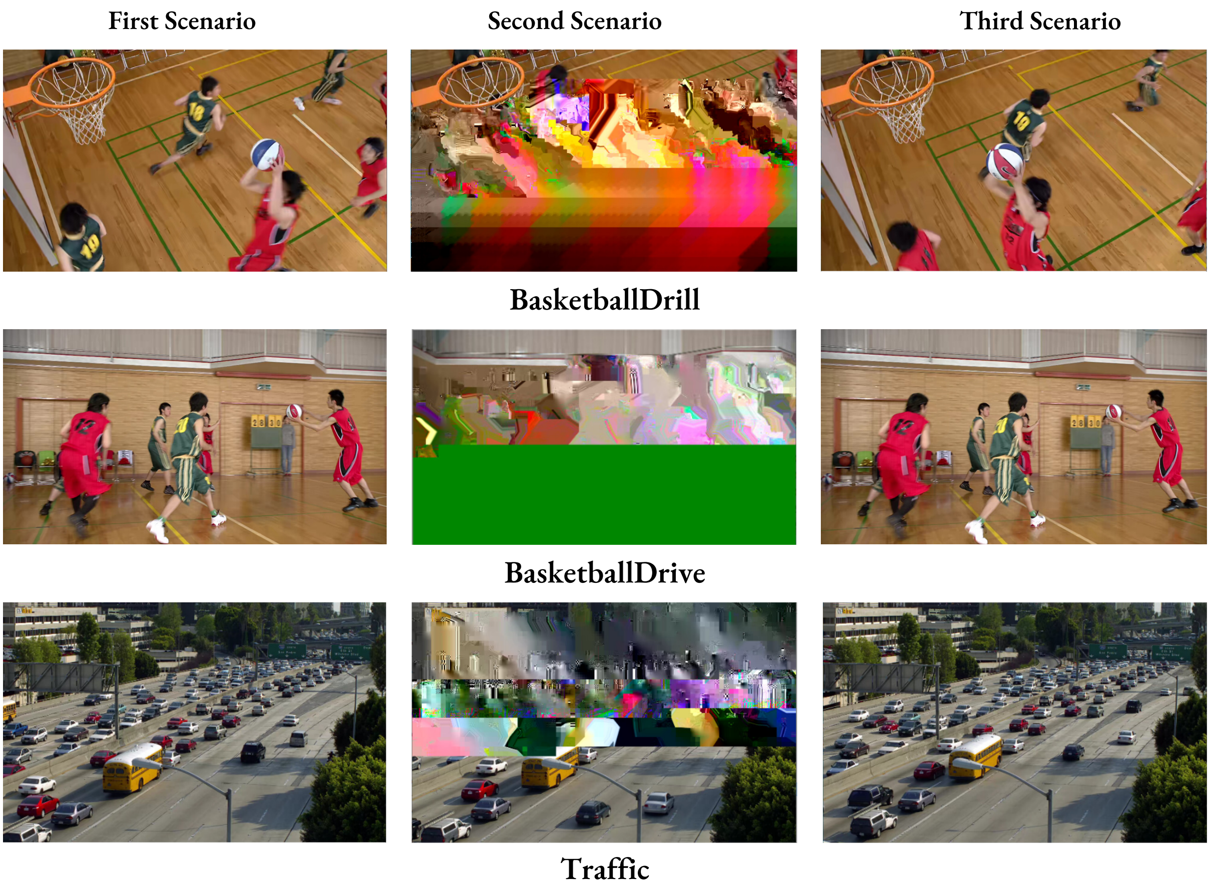

Figure 3.8: Error Compensation for lost frames.

and retransmit lost packets, ensuring the complete and accurate reception of priority packets, as a result, our system achieves higher visual quality and reduced data loss rates. It's important to note that our system does not guarantee the correct reception of non-priority sequence data, as it is sent through an unreliable satellite link, leading to potential data loss during transmission. Nevertheless, compared to the packet loss and visual quality recorded while using the unreliable QUIC, our system has shown an improvement of up to 50%. Our system significantly improves the receiving data rate compared to MPQUIC, with enhancements of up to 143% without sacrificing quality. The graphs 3.9, 3.10, and 3.11 depict the receiving data rate for each specific scenario. Receiving data rate refers to the amount of data transmitted through a network within a given period. The traditional MPQUIC resulted in the lowest receiving data rate due to significant delays in streaming data. In contrast, the second scenario achieved the highest data rate due to minimal delays in streaming data using the unreliable QUIC, albeit at the expense of visual quality.

8 Conclusion

In this chapter, a highly intelligent approach has been presented to significantly enhance the quality of video streaming in mobile telemedicine units. This approach has been designed to overcome the challenges of transmitting high-quality data simultaneously across multiple paths in rural regions. The proposed method is based

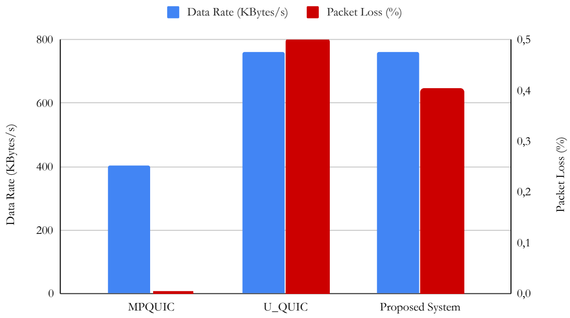

Figure 3.9: Packet loss and data rate measurements for BasketballDrill sequence.

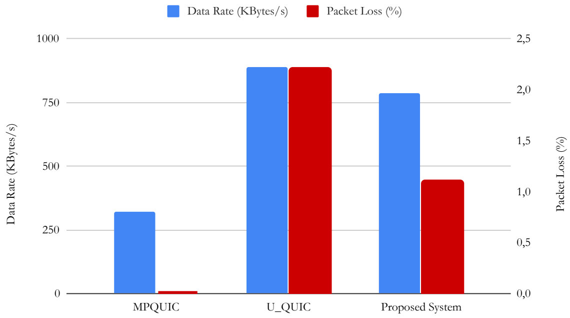

Figure 3.10: Packet loss and data rate measurements for BasketballDrive sequence.

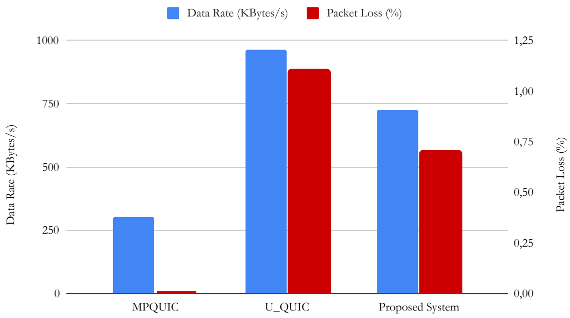

Figure 3.11: Packet loss and data rate measurements for Traffic sequence.

Sequence name	Protocol	Transmitted Data size (Bytes)	Received Data size (Bytes)
BasketballDrill	MPQUIC	7632324	7632324
	U_QUIC	7632324	7597582
	Proposed Systeme	7632324	7602723
BasketballDrive	MPQUIC	19077994	19077994
	U_QUIC	19077994	18660394
	Proposed Systeme	19077994	18852393
Traffic	MPQUIC	20474647	20474647
	U_QUIC	20474647	20255047
	Proposed Systeme	20474647	20325846

Table 3.5: transmitted and the received data in byte

on a modified extension of the MPQUIC protocol that utilizes two different mechanisms for sending data reliably through 5G/LTE cellular networks and unreliably via satellite networks. The proposed scheme employs an advanced video compression technique called MDC to enhance the efficiency of the overall transmission quality and error resilience. This technique is highly effective in reducing packet loss's impact and ensuring the quality of the transmitted data. The study focuses on a specific use case: a connected ambulance transmits patient data to a mobile expert unit. With this intelligent approach, it is possible to ensure that the patient's data is transmitted in the highest possible quality, even in areas with poor network coverage. Overall, this research presents a promising solution that has the potential to revolutionize the field of mobile telemedicine and improve healthcare access for rural communities.

The proposed approach has demonstrated a considerable improvement in video quality, boasting a 70% increase compared to the unreliable QUIC transport protocol. Furthermore, it has also significantly reduced the completion time required for video transmission by up to 61% when compared to the MPQUIC protocol alone and improves the receiving data rate by up to 143% compared to the classical MPQUIC protocol. It also minimizes packet loss by up to 50% compared to the unreliable QUIC.

CHAPTER 4

Enhancing Real-Time Mobile Health Video Streams: A Cross-Layer Region-of-Interest Based Approach

1 Introduction

The COVID-19 pandemic highlighted the importance of telemedicine and telecommunications in healthcare delivery. Wireless telemedicine technologies have transformed emergency care, reducing diagnosis-to-treatment time and improving patient outcomes. These advancements have been especially valuable in isolated areas where in-person consultations are challenging. Integrating m-health video communication technology into healthcare services has greatly benefited telemedicine applications. This technology has allowed for fast crisis interventions and access to specialized care for individuals in remote areas and has reduced the need for frequent checkups. It has also played a crucial role in improving health outcomes by enabling timely diagnosis and reducing re-hospitalization rates while contributing to substantial cost savings within healthcare expenditure.

However, because of the large amount of information contained in medical videos, streaming them over wireless communication technologies presents several challenges, including limited resources, efficiency issues, and the need for high bandwidth to ensure high-quality and low-latency video delivery. For that reason, video compression technology is crucial in minimizing the transmission of data and reducing overall transmission delays. Ensuring that medical professionals receive high-quality video feeds for accurate diagnoses and treatment decisions. Furthermore, video compression is significant in low bandwidth environments, such as remote diagnoses from isolated locations, developing countries, and disaster sites.HEVC is a popular video compression standard, outperforming its predecessor, H.264/AVC, by providing the same visual quality while requiring only half the bit rate.

During video communication, the original video sequence goes through stages before transmission. This includes encoding, packetization, channel encoding, and network transmission. While video streaming protocols generally tolerate packet loss, it's crucial to provide reliability to specific parts of the video known as the Region Of Interest (ROI) to maintain high video quality during transmission. These regions are more critical than the non-ROI area (Background). Encoding and transmitting the ROI with higher quality allows for superior video quality with minimal interruptions. Therefore, implementing reliability techniques in transmitting video content is important.

The deployment of multipath transport protocols can enhance video streaming performance by optimizing data delivery. Devices with multiple network interfaces can simultaneously use WiFi and 4G/5G connections to transmit data. Various transport protocols like MPTCP and MPQUIC harness the capabilities of multiple network interfaces. Our proposal involves combining ROI separation technology with multipath video streaming. This includes transmitting a segmented video stream, encoded independently using HEVC encoder at different bitrates, over distinct wireless paths to facilitate bandwidth aggregation based on priority.

This chapter introduces the Region-of-Interest approach, including detection and encoding ROI-based techniques and the reason behind choosing the GMM algorithm to separate ROI regions from medical videos. It then discusses the ROI-based solution to improve video streaming over low-latency networks. Finally, a conclusion summarizes the contributions of this document and suggests potential future work.

2 Related Work

In recent years, the scientific community has shown increasing interest in enhancing video streaming performance within vehicular ad-hoc networks. Numerous solutions have been proposed, targeting various layers such as source encoding, channel transmission, and channel encoding. These solutions can be categorized based on the abstraction layer they address. This related work section is divided into two parts: the first part covers research on source encoding solutions within m-health video communication technology, while the second part examines adaptive methods for improving video streaming in vehicular environments through multipath solutions.

Panayides et al. [129] introduced a novel approach that involves identifying ROIs crucial for the diagnosis process. These regions, known as d-ROIs, can be manually determined by medical experts or automatically detected using computer-based segmentation techniques. The approach ensures that these d-ROIs are encoded with all the necessary diagnostic information, allowing for high-quality diagnostics, while less critical areas are encoded at a lower quality. Similarly, Romain Delhaye et al. [130] developed an encoding solution based on the H.264 compression standard designed to stream medical videos over low-bandwidth networks. Their approach focuses on delivering high-quality video of the expert-identified ROIs, with the background quality being reduced. This method is particularly relevant for limited bandwidth situations, such as air-to-ground communication channels. Debono et al. [131] explored how to minimize the computational power required for advanced error concealment techniques to recover lost video information on wireless channels. They proposed applying these techniques to specific ROIs identified by medical experts, using a cross-layer design within a mobile WiMAX environment to optimize the quality of experience for these critical regions while ensuring real-time delivery. Cavero et al. [132] introduced the Echocardiogram Transmission Protocol (ETP), a new protocol for real-time transmission of echocardiograms over IP networks. The protocol uses an encoding scheme that considers the data classification, visualization attributes, and diagnostic importance of each area within the echocardiogram, allowing for different error protection strategies to be applied to each region. This approach is particularly beneficial for medical applications requiring secure and reliable echocardiogram transmission. Wu et al. [133] presented two innovative techniques for improving the compression of medical ultrasound videos. The first technique involves an ROI extraction algorithm based on image textural features, identifying the most diagnostically relevant regions in the videos. The second technique is a hierarchical coding method that adjusts transform coefficients and quantization parameters to ensure higher quality compression for ROIs compared to non-ROIs. In military surgical scenarios, where bandwidth is often limited, Khire et al. [134] addressed the challenge of real-time video communication by proposing an ROI video coding solution. This solution allocates more bits to the ROI regions, providing a low-complexity, H.264-compliant, and diagnostically lossless encoding method. Maharjan et al. [135] aimed to enhance the quality of transmitted video during real-time surgical tele-training by introducing an algorithm that reduces video frame distortion, optimizes flow rate allocation, and improves transmission reliability. The proposed EVQDMBRM algorithm dynamically adjusts the video rate at the source to reduce packet loss and efficiently manage bandwidth. Lastly, Zahra et al. [136] proposed a surveillance framework for smart cities, which efficiently handles video surveillance in environments with limited bandwidth. The framework leverages a deep learning-based approach to extract salient regions from video frames and compresses them to reduce data size without losing critical information. This approach ensures high visual quality for effective analysis in challenging environments. Similarly, Zhenjie Deng et al. [108] introduced a strategy for optimizing parameters across different network layers in a cross-layer manner, particularly integrating 802.11ac and LTE technologies, to improve video streaming performance and user experience. In another study, Salva-Garcia et al. [137] introduced a 5G video optimizer virtual network function (vOptimizerVNF) to mitigate the negative impacts of heavy network loads on essential services. This solution leverages cutting-edge 5G and video processing technologies, combining advanced traffic filtering and scalable H.265 video coding for dynamic, bandwidth-efficient video optimization while preserving QoS. Using kernel-space video processing further enhances system performance, and deploying virtual network functions (VNFs) offers scalability and flexibility. Further, a study presented in [138] focused on optimizing QoS in UAV-assisted vehicular ad hoc networks (VANETs) through a multipath-based approach designed to reduce streaming video delays. This method employs scalable video coding (SVC) and AODV-Multipath (AODVM) with disjoint nodes, ensuring reliable video transmission despite high bandwidth demands. Kim et al. [111] tackled the issue of HTTP adaptive streaming within 5G networks by proposing a multipath transmission approach that employs a collective segment request mechanism. This method improves bandwidth estimation accuracy and network responsiveness through block-based and segment-based evaluations. In addressing QoS requirements for VANET entertainment and traffic safety applications, Vafaei et al. [139] introduced a routing protocol based on adaptive junctions. This protocol leverages an ant colony optimization algorithm and a fuzzy logic-based algorithm to select optimal routing paths, ensuring high packet delivery ratios, connectivity, and low delay. Their subsequent study [140] proposed a two-path model for vehicular communications, enhancing global QoS metrics by categorizing video information and transmitting it across distinct paths based on priority. The model also employs different protocols for inter-frames and intra-frames to optimize transmission quality and minimize delays. Finally, Lopes et al. [141] developed an IP-based multihomed approach for RTP video stream transmission in VANETs. Their method improves video performance even under suboptimal signal conditions by dividing the video into different frame types and routing critical frames through more reliable paths.

This study reviews literature that delves into multipath transport protocols and ROI-based strategies to enhance video streaming performance in heterogeneous networks. Some works primarily focus on video source coding for transmission, neglecting the influence of network conditions and transport protocols on video quality. Factors like high mobility, frequent topology changes, and elevated packet loss rates can considerably degrade the viewing experience. The choice of transmission method—whether reliable or unreliable—also plays a critical role in overall quality. Conversely, other research emphasizes developing advanced multipath schedulers at the transport layer. Still, it does not fully address the significant delays encountered when streaming high-resolution medical videos with large data volumes, particularly when using protocols that ensure reliable transmission. However, the existing studies have yet to propose a unified approach that integrates these technologies to maximize their benefits in vehicular environments. Consequently, this analysis led to combining a multipath transport protocol with ROI separation technology, allowing for reliable and unreliable data transmission across heterogeneous networks. Furthermore, this paper specifically encodes high-priority regions to ensure superior video quality in these areas. Table 4.1 compares existing concepts and the proposed innovative approach.

3 Region Of Intrest approach (ROI)

Segmenting medical videos is crucial for improving the visibility of anatomical or pathological features within the videos, enhancing the efficiency and accuracy of computer-aided diagnosis and intelligent treatment. This segmentation process involves identifying specific ROIs, such as the liver and liver tumors, the brain and brain tumors, the optic disc, individual cells, the lungs, pulmonary nodules, and cardiac images. Medical professionals use various imaging techniques, including X-rays and ultrasound, to not only diagnose diseases but also assess the progression of illnesses and plan appropriate treatments. The first crucial step in ROI-based video coding techniques is identifying the Region of Interest (ROI), typically defined by moving regions or objects within the video frames. Numerous studies have proposed various methods for detecting these moving regions using established Moving Object Detection (MOD) techniques. For example, some methods use background subtraction, employing models like Gaussian Mixture Models (GMMs) to effectively separate moving objects from the background effectively [142]. Traditionally, computer vision techniques relied on manually designed features and computationally intensive processes. For instance, the sliding window approach was used for region selection, where a fixed-size window moved across the image to detect objects. On the other hand, feature extraction techniques were employed to identify and measure essential image patterns for further processing. These techniques include Histogram of Oriented Gradients (HOG) and Scale-Invariant Feature Transform (SIFT). However, the traditional methods faced difficulties handling a wide range of object characteristics and required significant computational resources. In contrast, computer vision has experienced a significant shift with the rise of deep learning models, mainly due to the advancements in Deep Convolutional Neural Networks (DCNNs) and the enhanced computational capabilities of GPUs. Models like Faster RCNN, SSD, and YOLO can automatically extract feature representations from data without the need for manual intervention, effectively

Aspect	Existing Works	Proposed Approach
Source Encoding	Utilizes methods like H.264 and H.265/HEVC for compression, focusing on ROI for better diagnostic quality and lower bandwidth usage.	Combines ROI-based encoding with multipath transport protocols to ensure high video quality in VANETs.
Channel Transmission	Incorporates multipath solutions and cross-layer designs to enhance QoS and reduce delays.	Integrates a modified extension of multipath transport with ROI encoding, optimizing both reliable and unreliable data streams.
Focus on Network Conditions	Some studies address network conditions like high mobility and packet loss, but often separately from source coding and transport protocols.	Unifies network condition considerations with encoding and transport, optimizing performance in vehicular environments.
Multipath Transport Protocols	Explores innovative multipath schedulers and protocols but may overlook the integration with source encoding techniques.	Integrates multipath transport protocols with ROI separation, leveraging their advantages collectively.
Reliability and Quality	Ensures high diagnostic quality and low distortion for critical regions in medical applications but may not fully address vehicular-specific challenges.	Focuses on reliable, high-quality video reception in vehicular environments, addressing mobility issues.
Innovation	Introduces novel encoding schemes, error protection techniques, and cross-layer designs for specific applications, but often in isolation.	Innovatively merges multipath transport with ROI encoding for comprehensive performance enhancement.

Table 4.1: Comparison of Existing Works and Proposed Approach.

overcoming the constraints of traditional techniques [143, 144].

3.1 Techniques for ROI Detection and Encoding

Region of Interest (ROI) detection and encoding are critical techniques in the field of video streaming, particularly when optimizing video quality in resource-constrained environments like vehicular networks. These methods focus on identifying and prioritizing specific areas of a video frame that are most relevant or important to the viewer. By selectively encoding these regions at higher quality while compressing less important areas more aggressively, ROI-based techniques help to efficiently utilize bandwidth and improve the viewer's Quality of Experience (QoE). This approach is especially beneficial in environments with fluctuating network conditions, such as vehicular networks, where maintaining consistent video quality can be challenging [33, 145, 146].

3.1.1 Automatic ROI Detection

Machine Learning-Based Detection: Machine learning models, particularly deep learning techniques like Convolutional Neural Networks (CNNs), can automatically detect ROI within video streams. These models can be trained on large datasets to identify specific objects, faces, or movements that are likely to be of interest to the viewer. In vehicular environments, this might include detecting vehicles, pedestrians, or road signs.
Edge Detection and Motion Analysis: Traditional image processing techniques, such as edge detection and motion analysis, can also be used for ROI detection. Edge detection algorithms like Canny or Sobel can highlight significant boundaries within a video frame, while motion analysis can track moving objects, identifying regions where movement is most significant.
Hybrid Approaches: Combining machine learning with traditional methods can improve the accuracy and efficiency of ROI detection. For example, motion analysis might first identify potential ROIs, which are then confirmed and refined by a deep learning model.

3.1.2 ROI-Based Video Compression

Selective Encoding: In ROI-based video compression, the video encoder allocates more bits to the ROI, ensuring that these areas are transmitted with higher quality. Non-ROI areas are compressed more aggressively, reducing their bit allocation and thus saving bandwidth.
Layered Coding Techniques: Scalable video coding (SVC) can be used to encode ROI and non-ROI regions at different layers. The base layer might contain the entire frame at lower quality, while enhancement layers contain higher-quality ROI. This allows for flexible adaptation based on network conditions or device capabilities in vehicular environments.
Adaptive Bitrate Streaming (ABR) with ROI: ABR techniques can be adapted to prioritize ROIs during streaming. When network bandwidth fluctuates, the streaming protocol can adjust the bitrate of the non-ROI areas first, ensuring that the quality of the ROI remains high .

3.1.3 Dynamic ROI Adjustment

Real-Time User Feedback: In some systems, user feedback is utilized to adjust the ROI dynamically. For example, eye-tracking data can be used to identify where the user is looking, allowing the system to adjust the ROI accordingly. This could be

particularly useful for in-car entertainment systems or heads-up displays in vehicular environments.
Context-Aware ROI Selection: The selection of ROI can be context-aware, meaning that the system can adjust the ROI based on external factors, such as the type of content being viewed or the vehicle's location and speed. For example, a navigation system might prioritize the ROI around road signs or upcoming turns.

3.1.4 Energy-Efficient ROI Encoding for Vehicular Networks

Low-Power Encoding Techniques: Given the energy constraints in vehicular environments, it is important to use low-power encoding techniques for ROI processing. Techniques such as hardware acceleration and parallel processing can be leveraged to reduce the energy consumption of ROI encoding while maintaining high-quality output.
Edge Computing for ROI Processing: Offloading the computationally intensive tasks of ROI detection and encoding to edge servers can also be beneficial. Edge computing allows for faster processing and reduces the computational burden on the vehicle's on-board systems, which is particularly advantageous in resource-constrained vehicular environments.

These techniques provide a comprehensive approach to detecting and encoding ROIs in video streams, particularly in the challenging conditions of vehicular environments. By focusing on the most relevant parts of the video, these methods ensure that bandwidth and computational resources are used efficiently, enhancing the overall Quality of Experience (QoE) for users.

4 Region-of-Interest Based solution to improve Real-Time Mobile Health Video Streaming.

In this section, we will provide a detailed explanation of our proposed system. We will start by explaining the ROI-based video encoding and then move on to discussing our ROI-based original cross-layer system.

4.1 ROI-based video encoding

In the initial phase of our approach, we focus on identifying the Regions of Interest (ROI) within the video data. This involves detecting moving regions or objects. To accomplish this, we utilized two methods: a conventional approach using the Gaussian Mixture Model (GMM) technique, and a second approach using YOLOv3, a Convolutional Neural Network (CNN) model trained on the video dataset we collected. The validation of these methods was carried out by a medical team from Henri Mondor University Hospital in Créteil, France. Upon evaluation, both approaches yielded nearly identical results in identifying image areas as ROI. However, the processing time required for the YOLOv3-based solution was found to be considerably longer. As a result, for real-time applications, we chose to proceed with ROI extraction based on the GMM technique.

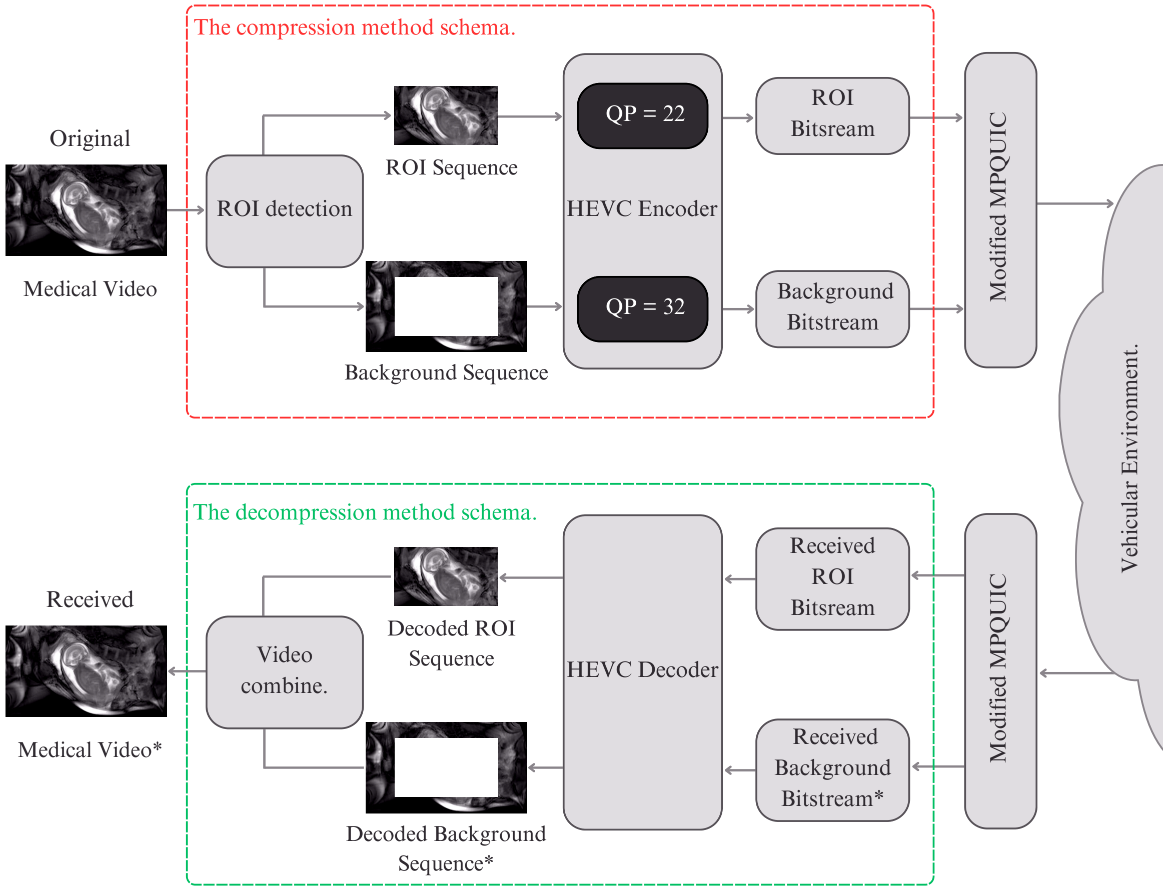

Figure 4.1: The compression method schema [12]

In the proposed compression method, we segment each frame of the medical video into two parts using the GMM algorithm. This creates the region of interest (ROI), which represents the frame's foreground. The resulting ROI frames are concatenated to form a separate video sequence called the region of interest (ROI) sequence. Similarly, the background of each frame is also segmented, creating the background sequence. In this paper, we define the entire fetus presentation area as the ROI because we want to encode the two sequences separately using different qualities for medical image compression. The proposed compression method's flowchart is shown in Figure 4.1. However, other applications could choose a different specific part as their region of interest. In the second step of the method, as shown in the framework, an HEVC encoder compresses the ROI video sequence with a small QP value to ensure high quality and avoid losing too much information in the compression process. On the other hand, the background video sequence is encoded with lower quality.

4.2 The proposed cross-layer system

In m-Health environments, it is important to ensure high-quality streaming of medical video content. This content contains vital information that is crucial for medical professionals to make informed decisions. Existing research has primarily focused on improving video streaming at the source coding level, overlooking the challenges posed by transmission in the network. The dynamic nature and mobility of ambulances

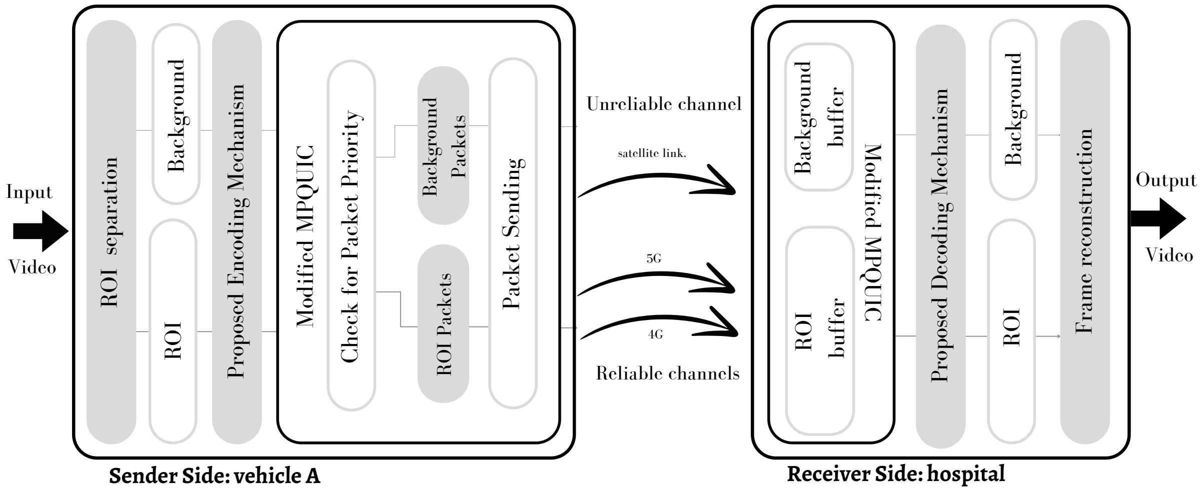

Figure 4.2: The proposed ROI-based schema for multipath video streaming using hybrid data recovery mechanisms [12].

contribute to a high packet loss rate and reduced data delivery efficiency, rendering solely source coding solutions inadequate. To address this gap, this paper aims to enhance the streaming of video content in vehicular environments. The proposed approach focuses on improving video streaming by separating the crucial and comprehensive information, known as ROI (region of interest), and addressing challenges at the source coding and channel transmission levels.

The process of separating the input video into two sub-videos of different sizes on the sender side involves an ROI separation block. This separation can be achieved through the use of automatic or semi-automatic algorithms, often with the assistance of a medical expert. In our system, we employ the GMM algorithm to facilitate this process. The resulting ROI sequence has a smaller video size compared to the background. For specific details regarding the resolutions of the ROI and background videos used in our experiments, please refer to Table 4.2.

In this particular approach, two separate mechanisms are utilized to stream an entire video. Emphasizing the significance of the ROI parts, they are transmitted via a reliable channel, while the background (or non-region of interest) is conveyed through an unreliable channel. This strategy is employed to minimize the overall transmission delay. As discussed in Subsection 1.5, MPQUIC is a multipath transport protocol that facilitates the concurrent transmission of data through multiple paths. The paper introduces a modified version of MPQUIC to accommodate both mechanisms and facilitate the transmission of data through reliable and unreliable paths. Notably, the reliable channel employs the QUIC protocol, which incorporates an error recovery process to guarantee packet delivery. On the other hand, the unreliable channel utilizes Unreliable QUIC, which does not feature any reliability mechanism. For a visual representation of our approach, please refer to Figure 4.2.

The modified multipath QUIC (MPQUIC) protocol is designed to enhance video streaming performance by utilizing three different communication channels, or paths, simultaneously connected to satellite, 4G, and 5G cellular networks. This multipath approach allows the protocol to dynamically select the optimal path for transmitting data based on the specific characteristics of each channel, such as packet loss rate, latency, and available bandwidth. A crucial component of multipath transport protocols is the path scheduler, which is responsible for determining the most suitable path for sending packets. In the case of MPQUIC, it uses the min RTT (for more information see Section 1.5) algorithm as the default path scheduler. This algorithm prioritizes transmitting data over the fastest path with the lowest RTT, provided that path has an available congestion window [84]. In the event that a packet fails to reach its destination, it will need to be retransmitted, resulting in a delay equivalent to one round-trip time (RTT) for delivery. This mechanism ensures reliable data transmission across the multiple paths of the MPQUIC protocol. These features collectively contribute to the protocol's ability to deliver seamless video streaming experiences across diverse network environments.

The challenge is exacerbated by the fact that the ratio of segment delivery in VANETs is lower compared to traditional wireless networks. This is primarily due to the dynamically changing network topology and the presence of intermittent connections. Consequently, the MPQUIC protocol encounters additional delays in delivering data owing to the reduced delivery ratio. To address this issue, we have developed a system that employs an additional channel exclusively dedicated to the transmission of unreliable data, with the aim of minimizing the overall delay associated with streaming video content in vehicular environments. This supplementary channel is tasked with delivering non-Region of Interest (ROI) background sequences in an unreliable manner via a satellite link, given that these regions are deemed less critical compared to the ROI ones. The choice of satellite communication was deliberate, driven by its extensive coverage capabilities, making it a viable option for establishing connections with individuals residing in remote and isolated rural areas.

In order to support unreliable data delivery, the default path scheduler of MPQUIC requires some modifications. The key modification involves the integration of a new frame type known as the "Datagram" frame into MPQUIC. This particular frame, developed by [147] as part of the unreliable extension of the QUIC protocol, enables end-hosts to stream data unreliably over a QUIC connection. Through the integration of this frame and its support by the min RTT scheduler, the enhanced version of MPQUIC will have the capability to transmit data using various mechanisms for streaming both ROI (Region of Interest) and background sequences. The QUIC protocol is designed to transfer data efficiently and reliably over the internet. In QUIC, each packet can contain one or multiple frames, with different types of frames used to ensure the protocol works correctly. One such frame type is the *Stream* frame, which is responsible for carrying encrypted payload (data). If there are any issues in delivering these frames to their destination, retransmission is necessary to ensure the data reaches its intended recipient. Due to the critical nature of the ROI region, we have made the decision to prioritize these regions and assign them to the *Stream* frame type, ensuring that the data in these regions is given special attention and guaranteed lossless data delivery through cellular (4G/5G) networks.

The system manages the transmission of different types of frames to ensure reliable and efficient delivery. Background regions are designated to *Datagram* frames, which are then transmitted unreliably via satellite link to their respective destinations. Prior to sending any data, the scheduler performs checks to determine if any *Stream* frames need to be retransmitted. If not, the frame type for each transmitted packet is examined. If the packet is of type *Stream* it is sent reliably via 4G and 5G cellular networks, ensuring lossless transmissions and in-order reception. This process is crucial for maintaining a high-quality base frame sequence. On the other hand, if the packet is of type *Datagram*

Video Sequence	ROI frame size	Background frame size
Eyes examination	728x240	848x464
Ultrasound	600x528	1272x720
Amoeba	864x864	1920x1080

Table 4.2: ROI and Background resolutions (pixel).

it is sent unreliably through the satellite path, with no retransmission efforts made if the frame fails to reach its destination. This strategy is designed to enhance temporal quality by improving the frame rate.

When transmitting videos over networks, it's important to consider data compression to handle the large amount of information and redundancies within and between frames. In our system, we make use of the high-efficiency video coding standard (HEVC) to compress videos before transmission. As described in Section 3.8.7, we utilize the HEVC test model (HM-16.24) for encoding sequences, while the decoding functions are performed by the x265 FFmpeg's library. We've chosen to use the x265 library for decompression instead of the HEVC test model due to its ability to handle lossy data, a feature not available in the HEVC test model. To ensure the highest quality and minimize data loss during the compression process, we've encoded the ROI frames using a small QP value (QP = 22) and the background frames using a larger QP value (QP = 32). The frame reconstruction algorithm on the receiver side reconstructs the ROI and background sequences based on the order of information in the packets, combining them to produce the received video. This process allows for the efficient transmission of high-quality videos over networks.

Upon reaching the receiver side, the reverse process begins by first implementing the Region of Interest (ROI) and background buffers to compensate for delay variations. This involves organizing the received packets according to their sequence numbers, ensuring that they are placed in their initial order. Once all packets have been received, they are passed through an HEVC decoder, which decodes the sequences. Finally, the decoded sequences are reconstructed to produce the complete video.

5 Performance Evaluation

In order to measure the effectiveness of the proposed approach in vehicular environments, a series of experiments were conducted to assess five different scenarios. Each scenario was distinguished from the others by either the encoded parameters or the method used to stream the video. Our proposal was tested using two protocols. The first protocol, unreliable QUIC, transmits data without guaranteeing their arrival at their destination due to the lack of a reliable mechanism to ensure their arrival or to raise an alarm in case of lost packets during transmission. The second protocol, MPQUIC, is a reliable transport protocol designed to ensure lossless packet delivery, unlike the unreliable QUIC protocol. The second variable refers to the compression method used for encoding each video. It involves the use of constant QP (quantization parameter) values or different QP values assigned to regions of interest (ROIs) and background areas. The QP values are adjusted to achieve distinct encoding for ROIs and non-ROI regions. We specifically chose to use a lower QP value (QP=22) to

	Metrics	U_QUIC (QP=27)	MPQUIC (QP=27)	U_QUIC (QP=22/32)	MPQUIC (QP=22/32)	Proposed
Encoding ROI	PSNR	43.40	43.40	48.41	48.41	48.41
	SSIM	0.995	0.995	0.998	0.998	0.998
Received ROI	PSNR	39.13	43.40	39.77	48.41	48.41
	SSIM	0.986	0.995	0.986	0.998	0.998
Encoding Background	PSNR	46.84	46.84	43.48	43.48	43.48
	SSIM	0.999	0.999	0.997	0.997	0.997
Received Background	PSNR	44.19	46.84	41.85	43.48	41.85
	SSIM	0.996	0.998	0.991	0.997	0.991
Encoding Total	PSNR	44.70	44.70	42.36	42.36	42.36
	SSIM	0.997	0.997	0.996	0.996	0.996
Received Total	PSNR	40.50	44.70	38.70	42.36	41.00
	SSIM	0.991	0.997	0.988	0.996	0.993

Table 4.3: PSNR and SSIM results for Eyes Examination sequence.

encode the ROI, ensuring higher-quality compression for these important areas within the video sequence. In contrast, we utilized a higher QP value (QP=32) for encoding the background regions. This approach ensures that the ROIs are compressed with higher quality, while the non-ROIs are compressed with lower quality. Our method was compared with the traditional encoding technique, where the entire video was encoded using a single QP value (QP=27).

To effectively evaluate the performance of the system, it is essential to measure various objective and subjective metrics. These metrics play a crucial role in assessing different aspects of the system's performance. In our experiments, we utilized several key metrics including Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), Video Multimethod Assessment Fusion (VMAF), completion time for video streaming, and receiving data rate. The PSNR, SSIM, and VMAF metrics were specifically calculated to compare the quality of the transmitted video. These metrics provide valuable insights into the fidelity and perceptual quality of the video content. Meanwhile, the completion time for video streaming represents the total delay caused by the protocols during the transmission of video data. Additionally, the quality of video streaming is intricately tied to the receiving data rate, which serves as a critical benchmark for comparing the performance of each protocol. By meticulously considering these metrics, we were able to gain a comprehensive understanding of the system's performance and make informed assessments about its capabilities.

5.1 Video quality assessment

In the analysis, we compared the simulation results in terms of PSNR and SSIM values for the eye examination, Ultrasound, and Amoeba videos. The results are presented in

	Metrics	U_QUIC (QP=27)	MPQUIC (QP=27)	U_QUIC (QP=22/32)	MPQUIC (QP=22/32)	Proposed
Encoding ROI	PSNR	42.76	42.76	46.73	46.73	46.73
	SSIM	0.998	0.998	0.999	0.996	0.999
Received ROI	PSNR	33.81	42.67	37.47	46.73	46.73
	SSIM	.975	0.998	0.968	0.999	0.999
Encoding Background	PSNR	46.16	46.16	43.84	43.84	43.84
	SSIM	0.998	0.998	0.998	0.998	0.998
Received Background	PSNR	38.73	46.16	38.39	43.84	38.39
	SSIM	0.988	0.998	0.982	0.998	0.982
Encoding Total	PSNR	44.5	44.5	42.64	42.64	42.64
	SSIM	0.998	0.998	0.998	0.998	0.998
Received Total	PSNR	35.63	44.5	35.35	42.64	37.80
	SSIM	0.984	0.998	0.976	0.998	0.987

Table 4.4: PSNR and SSIM results for Ultrasound sequence.

	Metrics	U_QUIC (QP=27)	MPQUIC (QP=27)	U_QUIC (QP=22/32)	MPQUIC (QP=22/32)	Proposed
Encoding ROI	PSNR	38.20	38.20	46.57	46.57	46.57
	SSIM	0.981	0.981	0.995	0.995	0.995
Received ROI	PSNR	21.81	38.20	26.28	46.57	46.57
	SSIM	0.691	0.981	0.907	0.995	0.995
Encoding Background	PSNR	47.5	43.25	43.25	43.25	43.25
	SSIM	0.999	0.755	0.999	0.999	0.999
Received Background	PSNR	29.44	47.5	26.98	43.25	31.49
	SSIM	0.951	0.999	0.907	0.999	0.954
Encoding Total	PSNR	44.39	44.93	39.84	39.84	39.84
	SSIM	0.998	0.998	0.997	0.997	0.997
Received Total	PSNR	24.71	44.38	23.55	39.84	34.70
	SSIM	0.830	0.998	0.827	0.997	0.914

Table 4.5: PSNR and SSIM results for Amoeba sequence.

Figure 4.3: Eyes examination VMAF measurements

Figure 4.4: Ultrasound VMAF measurements

Figure 4.5: Amoeba VMAF measurements

Tables 4.3, 4.4, and 4.5 respectively. Each table specifically illustrates the encoding video quality (before transmission) for Regions of Interest (ROI), background, and the entire video in rows 2, 4, and 6, while the subsequent rows depict the received video quality after transmission in a vehicular environment. The comparison reveals consistent results with minor variations among the cases. Notably, the seventh row, which indicates the received video quality, demonstrates excellent quality in all scenarios when transmitting the encoded medical video with a specific Quantization Parameter (QP) value using the Multipath QUIC (MPQUIC) protocol, achieving a PSNR value of 44dB. This outcome aligns with expectations, as the MPQUIC protocol ensures lossless data transmission through its robust data recovery mechanism to address packet loss detection.

The choice of encoding method is extremely important for maintaining high video quality across all regions, regardless of their significance. In our analysis, we observed that the ROI regions (third row) showed a decrease in PSNR and SSIM values when compared to the received video encoded using different QP values. This was because the ROI regions in the latter were compressed at a higher quality using a smaller QP value. On the other hand, when using the unreliable QUIC protocol to send the video encoded with different QP values for each region, we observed the poorest PSNR and SSIM values. This was attributed to losses associated with unreliable transmission and the impact of degraded background quality, which occupies 56 to 64% of the frame surface. However, it is worth noting that there was a slight improvement in the PSNR and SSIM values for the ROI regions when compared to the ROI quality in the case of sending the video compressed using the same QP value.

Based on the simulation results presented in Tables 4.3, 4.4, and 4.5, our proposed method, as shown in the last column, demonstrates superior performance compared to the unreliable QUIC in terms of recorded video quality. Specifically, our proposed approach improves video quality by up to 40% and enhances the quality of the Region of Interest (ROI) by up to 50%. This improvement is attributed to the modification we made to the MPQUIC protocol, which ensures the reliable transmission of ROI sequences through dependable paths. Furthermore, the high-quality encoding applied during the compression process significantly contributes to achieving the best video quality. These findings are consistent with the results obtained from the Video Multimethod Assessment Fusion (VMAF) scores, as illustrated in Figures 4.3, 4.4, and 4.5.

When comparing the results, it is evident that the Eyes Examination video maintains an acceptable visual quality even when using the unreliable QUIC. This is due to the lower data size required to be transmitted through the network, which mitigates the impact of the challenging vehicular environment on the streaming video and its received quality. On the other hand, the impact of these challenging conditions becomes more pronounced when streaming a large amount of data, such as in the case of the Amoeba video, which is a Full HD video with intricate details. As a result, the unreliable QUIC produces poor video quality when sending the Amoeba sequence.

	Sequence	Ultrasound	Eyes Examination	amoeba
U_QU IC QP = 27	Delay(s)	11	7	36
	Transmitted Data	8042008	4015606	46206033
	Received Data	7952008	3997606	45282033
MPQUIC QP=27	Delay(s)	23	14	152
	Transmitted Data	8042008	4015606	46206033
	Received Data	8042008	4015606	46206033
U_QU IC QP = 22/32	Delay(s)	14	13	48
	Transmitted Data	9324251	4909522	52659987
	Received Data	9198251	4866322	51549187
MPQUIC QP=22/32	Delay(s)	32	17	167
	Transmitted Data	9324251	4909522	52659987
	Received Data	9324251	4909522	52659987
Proposed	Delay(s)	17	9	121
	Transmitted Data	9324251	4909522	52659987
	Received Data	9279851	4890322	52533987

Table 4.6: Completion time results, accompanied by the transmitted and received amount of data.

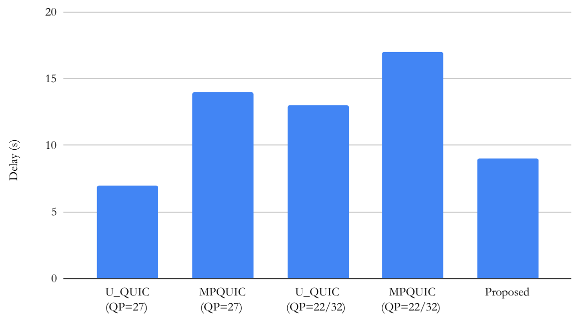

Figure 4.6: Eyes examination delays

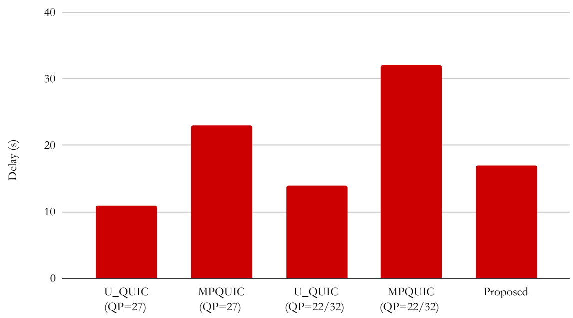

Figure 4.7: Ultrasound delays

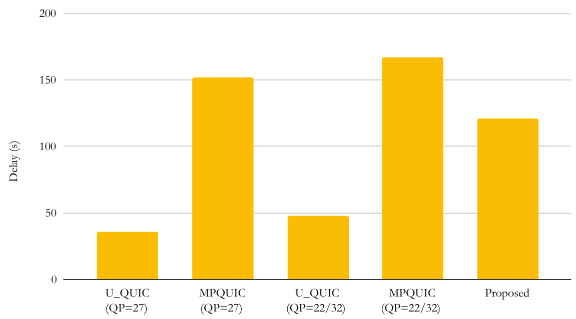

Figure 4.8: Amoeba delays

5.2 Completion time assessment

The MPQUIC protocol delivers excellent video quality, but it suffers from high transmission delays, making it less than ideal for real-time video streaming applications in wireless networks like VANETs. To illustrate this issue, Figures 4.6, 4.7, and 4.8 show the total delay for each protocol in transmitting an entire video. The transmission delay of MPQUIC across three different paths demonstrates that the delay is significant, even when using only two paths. This delay persists even when the entire video is transmitted using full MPQUIC due to the considerable challenges posed by vehicular environments. MPQUIC shows even poorer completion times when the transmitted video is compressed with different QP values for the regions of interest (ROI) and background, as this requires the transmission of more significant data. In Table 4.6, you can find the data size of each video encoded using unified or distinct QP values. Our simulation results demonstrate that our approach reduces delays by up to 35% compared to the classical MPQUIC. This reduction in delay is achieved by transmitting part of the video over the unreliable QUIC channel, which offers a higher data rate.

In the experiments conducted, it was observed that in Figures 4.6, 4.7, and 4.8, the time required to transmit the video using unreliable QUIC was found to be comparably shorter in contrast to other protocols. This can be attributed to the absence of any recovery mechanisms in the unreliable QUIC. However, it was noted that the quality of the transmitted video was significantly compromised due to its inherent unreliability. Furthermore, referencing Table 4.6, it is evident that the data size of the transmitted and received encoded video in bytes serves as an illustrative measure of the impact of lossy transmission in vehicular environments for each protocol, given the high levels

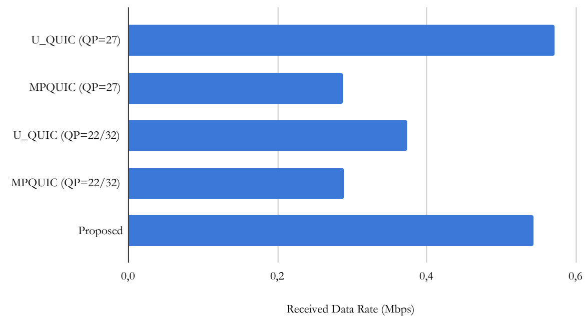

Figure 4.9: Received data rate values for Eyes examination.

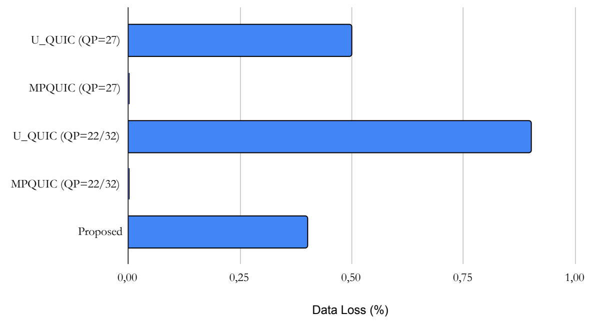

Figure 4.10: Eyes examination data loss measurements.

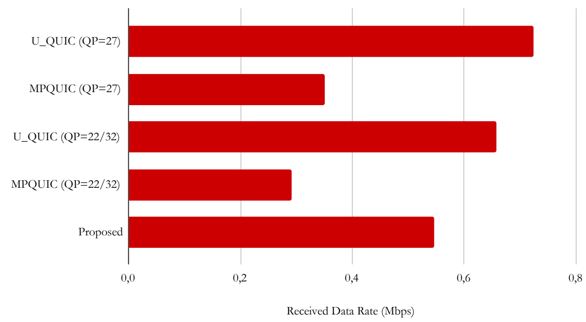

Figure 4.11: Received data rate values for Ultrasound.

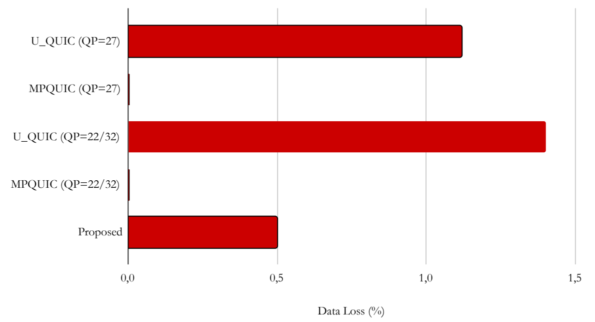

Figure 4.12: Ultrasound data loss measurements.

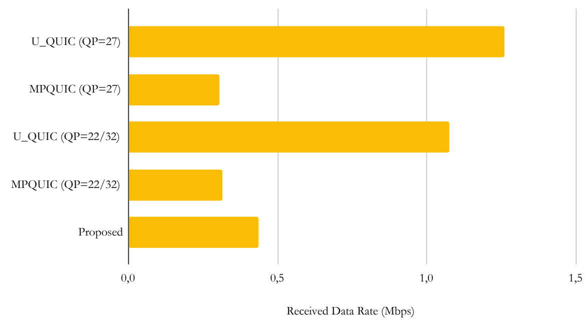

Figure 4.13: Received data rate values for Amoeba.

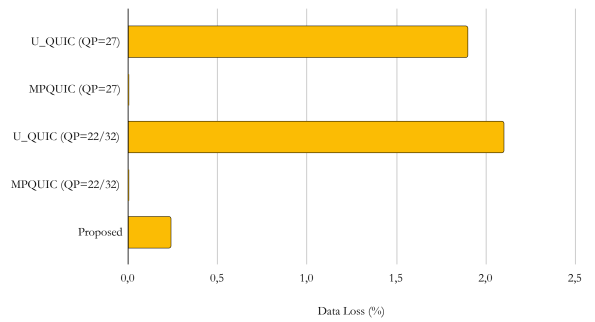

Figure 4.14: Amoeba data loss measurements.

of dynamicity and mobility exhibited by the vehicles. Notably, when transmitting videos encoded using different QP values for regions of interest (ROI) and background regions, a larger volume of data needs to be transmitted, leading to a considerably higher data loss rate.

5.3 Receiving data rate and packet loss assessment

The receiver data rate is a crucial factor in determining the quality of streaming video. In order to evaluate the effectiveness of each protocol, we conducted measurements of the receiving data rate for each protocol, as presented in Figures 4.9, 4.11, and 4.13. Our simulations indicate that unreliable QUIC demonstrates the best receiving data rate due to its minimal delay. However, it is important to note that a significant portion of the successfully received data originates from regions outside the region of interest (ROI), which may not be necessary. Conversely, MPQUIC exhibits the lowest receiving data rate values across the three use cases due to its packet delivery mechanism, ensuring the proper reception of all data without any loss, leading to high transmission delays. Conversely, our modified MPQUIC shows a noteworthy increase in data rate values, improving by up to 86% compared to the traditional MPQUIC. This improvement is attributed to its capability to effectively schedule video packet transmission, ensuring reliable transmission in ROI regions. However, it still lags behind unreliable QUIC in terms of data rate, which can be attributed to the latter's data delivery mechanism, resulting in higher data loss, as depicted in Figures 4.10, 4.12, and 4.14. It's important to note that unreliable QUIC exhibits lower PSNR values and degraded received frames, affecting the overall video quality.

In the data represented by Figure 4.9, Figure 4.11, and Figure 4.13, we can observe the received data rate values and packet loss percentage for each video. The Eyes Examination video, followed by the Ultrasound and then the Amoeba sequence, all demonstrate the performance of the proposed approach in comparison to Unreliable QUIC under different scenarios. The comparison shows that our proposed approach performs similarly to Unreliable QUIC when QP = 27 and outperforms it in other scenarios. During the streaming of the Ultrasound and Amoeba sequences, Unreliable QUIC outperforms our proposal when QP = 27 and QP = 22/32. This difference can be attributed to several factors. Firstly, the Eyes Examination sequence contains a relatively small amount of data and has the smallest resolution compared to the other sequences. This means that the transport protocol requires the shortest duration to complete its transmission for this video. Additionally, the use of multipath to stream the video minimizes the total end-to-end delays compared to single path protocols. Another contributing factor is the lack of retransmissions when detecting a packet loss during background transmission, which is due to enabling the unreliable transmission of the non-ROI regions. All these reasons make the proposed algorithm perform similarly to the Unreliable QUIC protocol when sending the Eyes Examination sequences for various reasons. However, it differs when streaming the Ultrasound and the Amoeba sequences. The Amoeba sequence contains a significantly larger amount of data due to its high resolution (full HD video), resulting in larger regions of interest (ROI). This causes the transmission of the data to take more time, especially in challenging network conditions and lossy networks. Each packet loss can increase the total end-to-end delay by one round-trip time (RTT) after retransmitting it to ensure proper reception. In the comparison of transmitted and received packets for each protocol and encoding scheme, the proposed approach streamed the Eyes Examination sequence in just 9 seconds, while it took 17 seconds and 121 seconds to send the Ultrasound and the Amoeba sequences, respectively. On the other hand, using Unreliable QUIC with QP=22/32, the same videos were transmitted in 14 seconds and 48 seconds, resulting in higher received data rates regardless of the degraded received quality.

5.4 Comparison between the proposed system and the state-of-the-art schemes.

In this section, we aim to conduct a subjective comparison between our proposed system and the state-of-the-art schemes of video streaming techniques. Our comparison will be based on the experiment results obtained from the following papers: [139, 108, 136, 135, 141, 109]. To ensure a comprehensive and detailed comparison, we will specifically identify critical metrics such as objective video quality and end-to-end delay. The choice of the PSNR metric has been made to evaluate video quality. It is important to note that before delving into the comparison, it is necessary to select relevant papers with similar environments and calculate similar metrics under identical network conditions. We have deliberately chosen a specific research paper to compare involving systems based on deep-learning algorithms, multipath techniques, an adaptive mapping mechanism applied at the MAC (Media Access Control) layer, Dynamic Adaptive Streaming over HTTP (DASH), and the fifth generation in vehicular environments. For more comprehensive details, please refer to Section 2.

In the results shown in Figure4.15, it is clear that our proposed system outperforms most state-of-the-art methods, with a notable PSNR improvement ranging from 1.3 dB to 16.8 dB. The EVQDMBRM algorithm introduced in [135] achieved a higher PSNR value of 51.13 dB compared to our proposal, but only improved the quality of surgical video transmission by 8.14%, while our system achieved an impressive 40% enhancement in visual quality. Furthermore, the comprehensive analysis of end-to-end delay comparisons presented in Figure 4.16 demonstrates that our proposal exhibits minimal transmission latency in streaming videos over the vehicular network. This is in contrast to [139], which recorded a latency of 11 seconds. The difference in latency can be attributed to the lower video resolution (352×288) used in the study, resulting in a significantly smaller data size compared to the videos transmitted in our system, which encompass a range of resolutions from SD to Full HD.

6 Conclusion

The paper presents an innovative method to improve the performance of video streaming in emerging m-health networks by combining region of interest (ROI) separation with multipath video streaming. The system utilizes a cross-layer approach, merging an adaptive algorithm with the multipath QUIC (MPQUIC) protocol to ensure the reliable transmission of priority data through cellular networks and non-priority data through satellite links. Moreover, it introduces an ROI detection system based on Gaussian Mixing Models (GMMs) to reduce latency and bitrate while maintaining high-quality focus on ROIs. These ROIs are selected based on their significance in medical decision-making within specific medical domains and are given special encoding treatment to preserve their high quality. Simulations and comparisons with traditional techniques demonstrate that the proposed algorithm outperforms MPQUIC and the unreliable QUIC, resulting in a 40% increase in overall video quality and up to 50% enhancement in ROI region quality compared to the unreliable QUIC. Furthermore, the end-to-end delay has been reduced by up to 35% compared to the delays associated with MPQUIC. Future research can explore optimizing the proposed system for real-world deployment, addressing practical challenges such as integration with

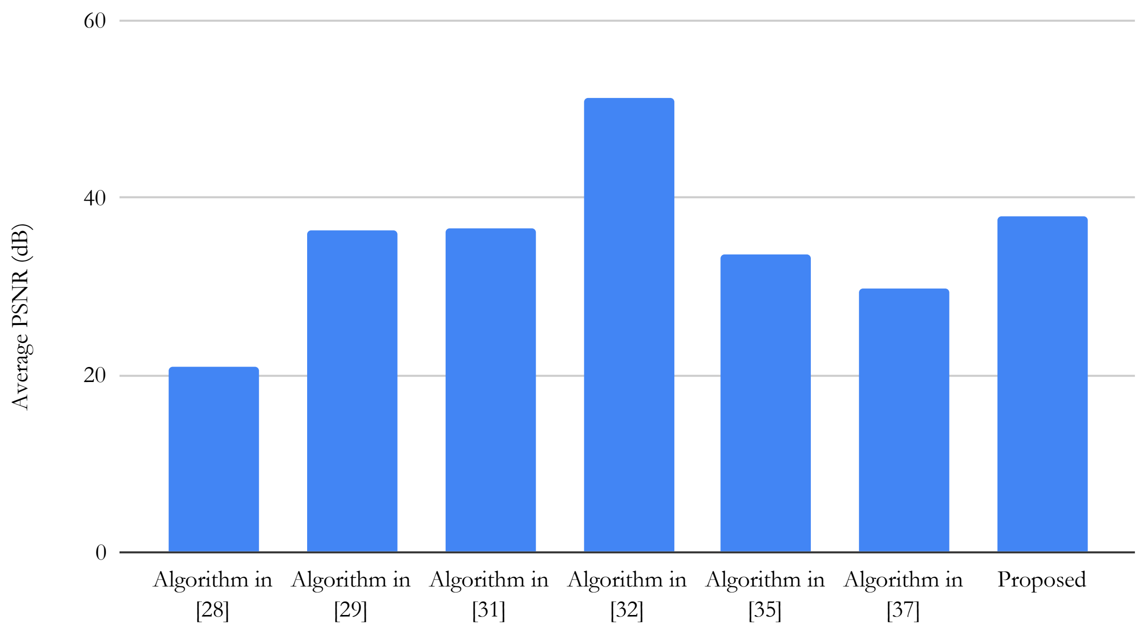

Figure 4.15: Average PSNR values obtained by the proposed approach and the state-of-the-art algorithms.

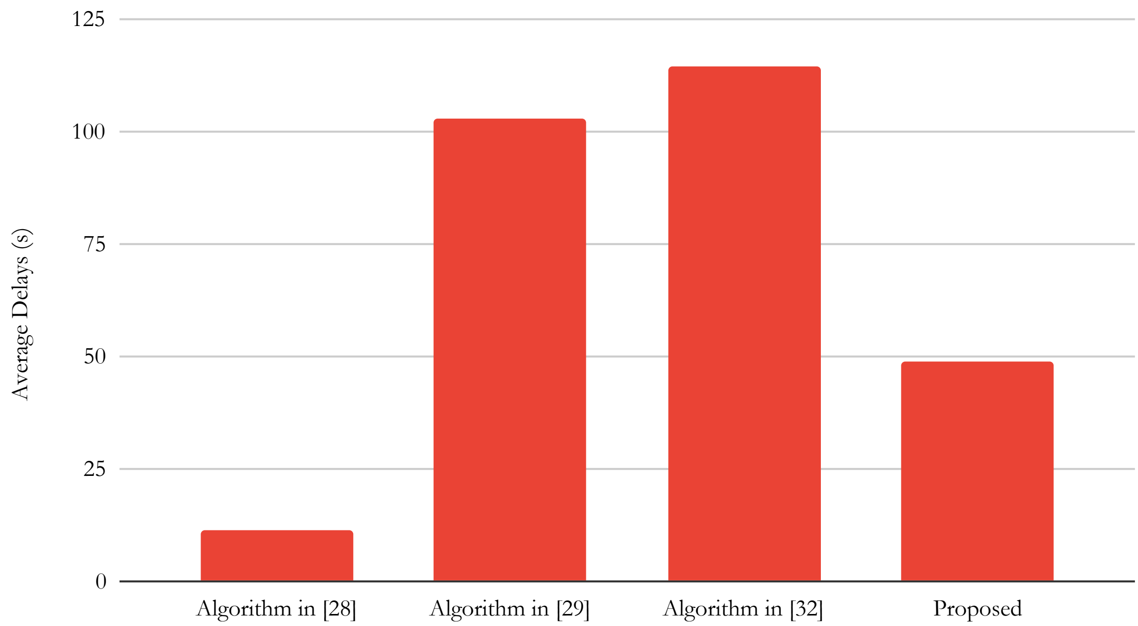

Figure 4.16: Average delay values obtained by the proposed approach and the state-of-the-art algorithms.

existing healthcare infrastructures, and ensuring scalability. Additionally, the integration of emerging technologies like artificial intelligence and edge computing could further enhance the performance and applicability of the solution in various healthcare scenarios.

General Conclusion

This thesis has addressed the critical challenge of enhancing video streaming quality over Vehicular Ad Hoc Networks (VANETs) by proposing a novel cross-layer approach that integrates multiple description coding with a modified Multipath QUIC (MPQUIC) protocol extension. The primary objective was to develop a system capable of delivering high-quality, low-latency video content in the inherently unstable and heterogeneous environment of VANETs, characterized by fluctuating network conditions and high packet loss rates.

Through the course of this research, the proposed cross-layer scheme demonstrated significant improvements in the Quality of Service (QoS) for video streaming applications within VANETs. By leveraging multiple description coding and region of interest-based coding techniques, the approach provided robust protection against data loss and transmission errors, ensuring that video quality remains acceptable even under adverse network conditions. Additionally, the extension of the MPQUIC protocol to support hybrid mechanisms—combining both reliable and unreliable transmission paths—proved effective in optimizing available network resources. This hybrid approach allowed the system to balance the trade-offs between latency and reliability, making it well-suited for the diverse range of video applications envisioned for VANETs. The proposed solution's evaluation through extensive simulations in realistic vehicular scenarios further validated its efficacy. The results demonstrated that the cross-layer approach not only enhances video streaming performance compared to traditional single-path and single-description methods but also adapts dynamically to the varying conditions of VANETs, providing a more resilient and flexible solution.

In conclusion, this thesis contributes to the growing body of research on vehicular communications by presenting a comprehensive solution that addresses the unique challenges of video streaming in VANETs. Integrating multiple description coding with a hybrid MPQUIC protocol extension represents a significant advancement in the field, offering a viable path forward for deploying reliable and high-quality video streaming services in vehicular environments. Future research could explore further optimizations, such as refining the coding strategies or enhancing the protocol's adaptability to even more complex vehicular scenarios. Additionally, real-world testing and the development of more sophisticated simulation models would provide deeper insights into the proposed system's practical deployment, potentially leading to its adoption in next-generation intelligent transportation systems.

List of Publications

- H. Elhachi, F. Boumehrez, M. Aymen Labiod, S. Redadaa, A. Mellouk, Smart cross-layer approach to multi-access terrestrial and non-terrestrial networks (ntns): Real-time mobile-health use case, International Journal of Communication Systems (2024) e5941. https://doi.org/10.1002/dac.5941
- Elhachi, H., Labiod, M. A., Boumehrez, F., & Redadaa, S. (2024). Enhancing real-time mobile health video streams: A cross-layer Region-of-Interest based approach. Computer Networks, 111014. https://doi.org/10.1016/j.comnet.2024.111014

List of Communications

- H. Elhachi, F. Boumehrez, M. Aymen Labiod, A. sahour, S. Redadaa, Comparative Analysis of QUIC, Unreliable QUIC, and MPQUIC for Video Streaming in VANETs, International Conference On Electrical Engineering and Control Applications, ICEECA2024, November 19-21 2024, University of Abbes Laghrour, Khenchela, Algeria (Springer publication)
- H. Elhachi, F. Boumehrez, M. Aymen Labiod, S. Redadaa, , Real time video transmission with terrestrial and non-terrestrial Networks, 1st International Workshop on Telecommunications «IWT'2024» , 1st & 2nd December 2024, Guelma, Algeria

Bibliography

[1] H.265 (hevc) video compression standard tutorial, [access jan 2025], https://www.linkedin.com/pulse/h265-hevc-video-compression-standard-tutorial-mirko-vojnovic.

[2] I. E. Richardson, H. 264 and MPEG-4 video compression: video coding for next-generation multimedia, John Wiley & Sons, 2004.

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

[4] M. Kazemi, S. Shirmohammadi, K. H. Sadeghi, A review of multiple description coding techniques for error-resilient video delivery, Multimedia Systems 20 (2014) 283–309.

[5] Chapter 10. user datagram protocol (udp) and ip fragmentation, [access jan 2025], https://notes.shichao.io/tcpv1/ch10/.

[6] The tcp/ip stack, [access jan 2025], https://www.geeksforgeeks.org/tcp-3-way-handshake-process/.

[7] Tcp 3-way handshake process, [access jan 2025], https://adacomputerscience.org/concepts/internet_tcp_ip.

[8] A. Rabitsch, Evaluation of packet schedulers for multipath quic.

[9] Building blocks of tcp, [access jan 2025], https://hpbn.co/building-blocks-of-tcp/.

[10] A. Langley, A. Riddoch, A. Wilk, A. Vicente, C. Krasic, D. Zhang, F. Yang, F. Kouranov, I. Swett, J. Iyengar, et al., The quic transport protocol: Design and internet-scale deployment, in: Proceedings of the conference of the ACM special interest group on data communication, 2017, pp. 183–196.

[11] M. Coudron, Nouvelles approches pour les communications multichemins.

[12] H. Elhachi, M.-A. Labiod, F. Boumehrez, S. Redadaa, Enhancing real-time mobile health video streams: A cross-layer region-of-interest based approach, Computer Networks (2024) 111014.

[13] H. Elhachi, F. Boumehrez, M. Aymen Labiod, S. Redadaa, A. Mellouk, Smart cross-layer approach to multi-access terrestrial and non-terrestrial networks (ntns): Real-time mobile-health use case, International Journal of Communication Systems (2024) e5941.

[14] M. A. Labiod, Contribution à l'amélioration des transmissions vidéo dans les réseaux ad-hoc véhiculaires (vanet).

[15] G. J. Sullivan, J.-R. Ohm, W.-J. Han, T. Wiegand, Overview of the high efficiency video coding (hevc) standard, IEEE Transactions on Circuits and Systems for Video Technology 22 (12) (2012) 1649–1668. doi:10.1109/TCSVT.2012.2221191.

[16] R. C. Gonzalez, Digital image processing, Pearson education india, 2009.

[17] A. B. Atitallah, Etude et implantation d'algorithmes de compression d'images dans un environnement mixte matériel et logiciel, Ph.D. thesis, UNIVERSITE BORDEAUX I (2007).

[18] The joint photographic experts group (jpeg) committee [accessed jun 2023], `https://jpeg.org/jpeg/index.html`.

[19] Mpeg committee documentation [accessed jun 2023], `http://www.chiariglione.org/`.

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

[21] D. Hands, M. Wilkins, A study of the impact of network loss and burst size on video streaming quality and acceptability, in: Interactive Distributed Multimedia Systems and Telecommunication Services: 6th International Workshop, IDMS'99 Toulouse, France, October 12–15, 1999 Proceedings 6, Springer, 1999, pp. 45–57.

[22] F. Ghozzi, Optimisation d'une bibliothèque de modules matériels de traitement d'images: conception et test vhdl, implementation sous forme fpga, Ph.D. thesis, Bordeaux 1 (2004).

[23] A. Martin, Représentations parcimonieuses adaptées à la compression d'images, Ph.D. thesis, Université Rennes 1 (2010).

[24] A. Robert, Transformées orientées par blocs pour le codage vidéo hybride, Ph.D. thesis, Télécom ParisTech (2008).

[25] F. Urban, Implantation optimisée d'estimateurs de mouvement pour la compression vidéo sur plates-formes hétérogènes multicomposants, Ph.D. thesis, INSA de Rennes (2007).

[26] M. Alvarez, E. Salami, A. Ramirez, M. Valero, A performance characterization of high definition digital video decoding using h. 264/avc, in: IEEE International. 2005 Proceedings of the IEEE Workload Characterization Symposium, 2005., IEEE, 2005, pp. 24–33.

[27] Découverte du mpeg-4 part 10 / avc, ou h.264, https://www.planetenumerique.com/guides-dossiers/Decouverte-du-MPEG-4-part-10-AVC.html?Decouverte-du-MPEG-4-part-10-AVC=.

[28] High-efficiency video coding (hevc) [accessed jun 2024], https://vcgit.hhi.fraunhofer.de/jvet/HM/.

[29] X265 homepage: [accessed jun 2023], https://bitbucket.org/multicoreware/.

[30] Libde265 homepage : [accessed jun 2023], https://www.libde265.org/.

[31] Le codage des images et de la couleur: [accessed jun 2023], http://obligement.free.fr/articles/codagedesimages.php/.

[32] C. Meinel, H. Sack, Digital communication: Communication, multimedia, security, Springer Science & Business Media, 2014.

[33] S. Vivienne, B. Madhukar, J. Gary, High efficiency video coding (hevc): algorithms and architectures, Cham, Switzerland: Springer International Publishing (2014).

[34] Z. Cheng, L. Ding, W. Huang, F. Yang, L. Qian, Subjective qoe based hevc encoder adaptation scheme for multi-user video streaming, in: 2016 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB), IEEE, 2016, pp. 1–6.

[35] M. Pereira, M. Antonini, M. Barlaud, Multiple description image and video coding for wireless channels, Signal Processing: Image Communication 18 (10) (2003) 925–945.

[36] K. R. Fall, W. R. Stevens, TCP/IP illustrated, volume 1: The protocols, addison-Wesley, 2011.

[37] D. An, Find out how you stack up to new industry benchmarks for mobile page speed, Think with Google-Mobile, Data & Measurement (2018) 24.

[38] The need for mobile speed: How mobile latency impacts publish, https://www.thinkwithgoogle.com/intl/en-154/marketing-strategies/app-and-mobile/need-mobile-speed-how-mobile-latency-impacts-publisher-revenue/.

[39] A. Dunkels, Design and implementation of the lwip tcp/ip stack, Swedish Institute of Computer Science 2 (77) (2001).

[40] J. Postel, User datagram protocol, Tech. rep. (1980).

[41] K. R. Kurose, Computer networking: A top-down approach by james, Kurose, Keith W. Ross.— (2017) 601.

[42] J. Postel, et al., Transmission control protocol rfc 793 (1981).

[43] I. Grigorik, High Performance Browser Networking: What every web developer should know about networking and web performance, " O'Reilly Media, Inc.", 2013.

[44] R. Nastase, Computer Networking: Beginners guide for Mastering Computer Networking and the OSI Model, Independently published, 2017.

[45] W. Eddy, Rfc 9293: Transmission control protocol (tcp) (2022).

[46] E. Blanton, D. Paxson, M. Allman, Tcp congestion control, rfc 5681 (2009).

[47] A. Rabitsch, Evaluation of packet schedulers for multipath quic (2018).

[48] Ietf, quic (quic), [accessed jull 2023, https://datatracker.ietf.org/wg/quic/about/.

[49] M. Thomson, S. Turner, Using transport layer security (tls) to secure quic, Internet Engineering Task Force, Internet-Draft draft-ietf-quic-tls-02 (2017).

[50] J. Salowey, A. Choudhury, D. McGrew, Aes galois counter mode (gcm) cipher suites for tls, Tech. rep. (2008).

[51] D. McGrew, An interface and algorithms for authenticated encryption, Tech. rep. (2008).

[52] K. Pittevils, Quicker: On the design and implementation of the quic protocol (2018).

[53] J. Iyengar, M. Thomson, Rfc 9000 quic: A udp-based multiplexed and secure transport, Omtermet Emgomeeromg Task Force (2021).

[54] L. C. et al., A quic implementation in pure go, 2017 [accessed jun 2023], https://github.com/lucas-clemente/quic-go.

[55] S. Ha, I. Rhee, L. Xu, Cubic: a new tcp-friendly high-speed tcp variant, ACM SIGOPS operating systems review 42 (5) (2008) 64–74.

[56] I. Rhee, L. Xu, S. Ha, A. Zimmermann, L. Eggert, R. Scheffenegger, Cubic for fast long-distance networks, Tech. rep. (2018).

[57] K. Pittevils, Quicker: On the design and implementation of the quic protocol, Ph.D. thesis (01 2018).

[58] S. Agarwal, C.-N. Chuah, R. H. Katz, Opca: Robust interdomain policy routing and traffic control, in: 2003 IEEE Conference onOpen Architectures and Network Programming., IEEE, 2003, pp. 55–64.

[59] C. Pelsser, L. Cittadini, S. Vissicchio, R. Bush, From paris to tokyo: On the suitability of ping to measure latency, in: Proceedings of the 2013 conference on Internet measurement conference, 2013, pp. 427–432.

[60] M. Coudron, Nouvelles approches pour les communications multichemins, Ph.D. thesis, Université Pierre et Marie Curie-Paris VI (2016).

[61] X. Meng, Z. Xu, B. Zhang, G. Huston, S. Lu, L. Zhang, Ipv4 address allocation and the bgp routing table evolution, ACM SIGCOMM Computer Communication Review 35 (1) (2005) 71–80.

[62] S. K. Singh, T. Das, A. Jukan, A survey on internet multipath routing and provisioning, IEEE Communications Surveys & Tutorials 17 (4) (2015) 2157–2175.

[63] H. Wu, S. Ferlin, G. Caso, Ö. Alay, A. Brunstrom, A survey on multipath transport protocols towards 5g access traffic steering, switching and splitting, IEEE Access 9 (2021) 164417–164439.

[64] R. Khalili, N. Gast, M. Popovic, J.-Y. Le Boudec, Mptcp is not pareto-optimal: Performance issues and a possible solution, IEEE/ACM Transactions On Networking 21 (5) (2013) 1651–1665.

[65] A. Nikravesh, Y. Guo, F. Qian, Z. M. Mao, S. Sen, An in-depth understanding of multipath tcp on mobile devices: Measurement and system design, in: Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking, 2016, pp. 189–201.

[66] A. Ford, C. Raiciu, M. Handley, O. Bonaventure, C. Paasch, Rfc 8684: Tcp extensions for multipath operation with multiple addresses (2020).

[67] R. Van Der Pol, S. Boele, F. Dijkstra, A. Barczyk, G. van Malenstein, J. H. Chen, J. Mambretti, Multipathing with mptcp and openflow, in: 2012 SC Companion: High Performance Computing, Networking Storage and Analysis, IEEE, 2012, pp. 1617–1624.

[68] M. Honda, Y. Nishida, L. Eggert, P. Sarolahti, H. Tokuda, Multipath congestion control for shared bottleneck, in: Proc. PFLDNeT workshop, Vol. 357, Citeseer, 2009, p. 378.

[69] H. Han, S. Shakkottai, C. V. Hollot, R. Srikant, D. Towsley, Multi-path tcp: a joint congestion control and routing scheme to exploit path diversity in the internet, IEEE/ACM Transactions on networking 14 (6) (2006) 1260–1271.

[70] D. Wischik, C. Raiciu, A. Greenhalgh, M. Handley, Design, implementation and evaluation of congestion control for multipath {TCP}, in: 8th USENIX Symposium on Networked Systems Design and Implementation (NSDI 11), 2011.

[71] Q. Peng, A. Walid, J. Hwang, S. H. Low, Multipath tcp: Analysis, design, and implementation, IEEE/ACM Transactions on networking 24 (1) (2014) 596–609.

[72] Y. Cao, M. Xu, X. Fu, Delay-based congestion control for multipath tcp, in: 2012 20th IEEE international conference on network protocols (ICNP), IEEE, 2012, pp. 1–10.

[73] Christoph paasch and sebstien barre et al. multipath tcp - linux kernel implementation, 2009-2018 [accessed jun 2023]., `https://multipath-tcp.org/`.

[74] H. Wu, [U+FFFD] Alay, A. Brunstrom, S. Ferlin, G. Caso, Peekaboo: Learning-based multipath scheduling for dynamic heterogeneous environments, IEEE Journal on Selected Areas in Communications 38 (10) (2020) 2295–2310. doi:10.1109/JSAC.2020.3000365.

[75] S. Ferlin, O. Alay, O. Mehani, R. Boreli, BLEST: Blocking estimation-based MPTCP scheduler for heterogeneous networks, 2016 IFIP Networking Conference (IFIP Networking) and Workshops, IFIP Networking 2016, Vienna, Austria (2016) 431–439doi:10.1109/IFIPNetworking.2016.7497206.

[76] Y. S. Lim, E. M. Nahum, D. Towsley, R. J. Gibbens, ECF: An MPTCP Path Scheduler to Manage Heterogeneous Paths, Proceedings of the 2017 ACM SIGMETRICS / International Conference on Measurement and Modeling of Computer Systems 45 (1) (2017) 33–34. doi:10.1145/3078505.3078552.

[77] C. Paasch, S. Ferlin, O. Alay, O. Bonaventure, Experimental evaluation of multipath tcp schedulers, in: Proceedings of the 2014 ACM SIGCOMM workshop on Capacity sharing workshop, 2014, pp. 27–32.

[78] S. Ferlin, Ö. Alay, O. Mehani, R. Boreli, Blest: Blocking estimation-based mptcp scheduler for heterogeneous networks, in: 2016 IFIP networking conference (IFIP networking) and workshops, IEEE, 2016, pp. 431–439.

[79] Y.-s. Lim, E. M. Nahum, D. Towsley, R. J. Gibbens, Ecf: An mptcp path scheduler to manage heterogeneous paths, in: Proceedings of the 13th international conference on emerging networking experiments and technologies, 2017, pp. 147–159.

[80] D. Wischik, M. Handley, M. B. Braun, The resource pooling principle, ACM SIGCOMM Computer Communication Review 38 (5) (2008) 47–52.

[81] C. Paasch, G. Detal, F. Duchene, C. Raiciu, O. Bonaventure, Exploring mobile/wifi handover with multipath tcp, in: Proceedings of the 2012 ACM SIGCOMM workshop on Cellular networks: operations, challenges, and future design, 2012, pp. 31–36.

[82] S. Cheshire, D. Schinazi, C. Paasch, Advances in networking, part 1.(june 2017) (2017).

[83] O. Bonaventure, S. Seo, Multipath tcp deployments, IETF Journal 12 (2) (2016) 24–27.

[84] Q. De Coninck, O. Bonaventure, Multipath quic: Design and evaluation, in: Proceedings of the 13th international conference on emerging networking experiments and technologies, 2017, pp. 160–166. doi:10.1145/3143361.3143370.

[85] E. Bouzid Smida, S. Gaied Fantar, H. Youssef, Link efficiency and quality of experience aware routing protocol to improve video streaming in urban vanets, International Journal of Communication Systems 33 (3) (2020) e4209.

[86] A. A. Ahmed, S. J. Malebary, W. Ali, O. M. Barukab, Smart traffic shaping based on distributed reinforcement learning for multimedia streaming over 5g-vanet communication technology, Mathematics 11 (3) (2023) 700.

[87] M. Okpok, B. Kihei, Challenges and opportunities for multimedia transmission in vehicular ad hoc networks: A comprehensive review, Electronics 12 (20) (2023) 4310.

[88] A. Benmir, A. Korichi, A. Bourouis, M. Alreshoodi, L. Al-Jobouri, Geoqoe-vanet: Qoe-aware geographic routing protocol for video streaming over vehicular ad-hoc networks, Computers 9 (2) (2020) 45.

[89] R. Immich, E. Cerqueira, M. Curado, Efficient high-resolution video delivery over vanets, Wireless Networks 25 (2019) 2587–2602.

[90] O. A. Hammood, M. Nizam, M. Nafaa, W. A. Hammood, Resp: Relay suitability-based routing protocol for video streaming in vehicular ad hoc networks, International Journal of Computers Communications & Control 14 (1) (2019) 21–38.

[91] M. Wang, J. Mao, W. Zhao, X. Han, M. Li, C. Liao, H. Sun, K. Wang, Smart city transportation: A vanet edge computing model to minimize latency and delay utilizing 5g network, Journal of Grid Computing 22 (1) (2024) 25.

[92] Y. Sahraoui, A. Ghanam, S. Zaidi, S. Bitam, A. Mellouk, Performance evaluation of tcp and udp based video streaming in vehicular ad-hoc networks, in: 2018 International Conference on Smart Communications in Network Technologies (SaCoNeT), IEEE, 2018, pp. 67–72.

[93] Simulation of urban mobility, `https://www.eclipse.org/sumo/`.

[94] M. Fogue, P. Garrido, F. J. Martinez, J.-C. Cano, C. T. Calafate, P. Manzoni, A realistic simulation framework for vehicular networks, in: Fifth International Conference on Simulation Tools and Techniques, 2012.

[95] C. R. E. R. Ramon dos Reis Fontes, Mininet-wifi, `https://mininet-wifi.github.io/` (March 2022).

[96] Mininet emulator, `http://mininet.org/`.

[97] H. P. P. Fabio Ludovici, Netem - network emulator at linux, `https://www.linux.org/docs/man8/tc-netem.html` (Sep 2022).

[98] Y. Zhai, X. Xu, B. Chen, H. Lu, Y. Wang, S. Li, X. Shi, W. Wang, L. Shang, J. Zhao, 5g-network-enabled smart ambulance: architecture, application, and evaluation, IEEE Network 35 (1) (2021) 190–196. doi:10.1109/MNET.011.2000014.

[99] S. Dananjayan, G. M. Raj, 5g in healthcare: how fast will be the transformation?, Irish Journal of Medical Science (1971-) 190 (2) (2021) 497–501. doi:10.1007/s11845-020-02329-w.

[100] S. Liu, Z. Gao, Y. Wu, D. W. K. Ng, X. Gao, K.-K. Wong, S. Chatzinotas, B. Ottersten, Leo satellite constellations for 5g and beyond: How will they reshape vertical domains?, IEEE Communications Magazine 59 (7) (2021) 30–36. doi:10.1109/MCOM.001.2001081.

[101] V. A. Vu, B. Walker, On the latency of multipath-quic in real-time applications, in: 2020 16th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), IEEE, 2020, pp. 1–7. doi:10.1109/WiMob50308.2020.9253402.

[102] Ffmpeg developers, "ffmpeg tool (version be1d324) [software]," (2023), `https://ffmpeg.org/`.

[103] A. Nasrabadi, M. Shirsavar, A. Ebrahimi, M. Ghanbari, Investigating the psnr calculation methods for video sequences with source and channel distortions, in: 2014 IEEE international symposium on broadband multimedia systems and broadcasting, IEEE, 2014, pp. 1–4.

[104] Z. Wang, A. Bovik, H. Sheikh, E. Simoncelli, Image quality assessment: from error visibility to structural similarity, IEEE Transactions on Image Processing 13 (4) (2004) 600–612. doi:10.1109/TIP.2003.819861.

[105] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, Image quality assessment: from error visibility to structural similarity, IEEE transactions on image processing 13 (4) (2004) 600–612.

[106] Z. Li, C. Bampis, J. Novak, A. Aaron, K. Swanson, A. Moorthy, J. Cock, Vmaf: The journey continues, Netflix Technology Blog 25 (2018) 1.

[107] W. H. Organization, Global diffusion of ehealth: Making universal health coverage achievable, `https://www.afro.who.int/publications/global-diffusion-ehealth-making-universal-health-coverage-achievable` (Dec. 2020).

[108] Z. Deng, Y. Liu, J. Liu, A. Argyriou, Cross-layer dash-based multipath video streaming over lte and 802.11 ac networks, Multimedia Tools and Applications 80 (2021) 16007–16026. doi:10.1007/s11042-020-10393-8.

[109] M. A. Labiod, M. Gharbi, F.-X. Coudoux, P. Corlay, N. Doghmane, Cross-layer scheme for low latency multiple description video streaming over vehicular ad-hoc networks (vanets), AEU-International Journal of Electronics and Communications 104 (2019) 23–34. doi:10.1016/j.aeue.2019.03.001.

[110] Y. Xing, K. Xue, Y. Zhang, J. Han, J. Li, J. Liu, R. Li, A low-latency mptcp scheduler for live video streaming in mobile networks, IEEE Transactions on Wireless Communications 20 (11) (2021) 7230–7242. doi:10.1109/TWC.2021.3081498.

[111] H. Kim, K. Chung, Multipath-based http adaptive streaming scheme for the 5g network, IEEE Access 8 (2020) 208809–208825. doi:10.1109/ACCESS.2020.3038854.

[112] F. Boumehrez, R. Brai, N. Doghmane, K. Mansouri, Quality of experience enhancement of high efficiency video coding video streaming in wireless packet networks using multiple description coding, Journal of Electronic Imaging 27 (1) (2018) 013028–013028. doi:10.1117/1.JEI.27.1.013028.

[113] S. Zaidi, S. Bitam, A. Mellouk, Hybrid error recovery protocol for video streaming in vehicle ad hoc networks, Vehicular communications 12 (2018) 110–126. doi:10.1016/j.vehcom.2018.01.002.

[114] M. Chen, J. Yang, J. Zhou, Y. Hao, J. Zhang, C.-H. Youn, 5g-smart diabetes: Toward personalized diabetes diagnosis with healthcare big data clouds, IEEE Communications Magazine 56 (4) (2018) 16–23. doi:10.1109/MCOM.2018.1700788.

[115] M. S. Hossain, G. Muhammad, Emotion-aware connected healthcare big data towards 5g, IEEE Internet of Things Journal 5 (4) (2017) 2399–2406. doi:10.1109/JIOT.2017.2772959.

[116] D. Soldani, F. Fadini, H. Rasanen, J. Duran, T. Niemela, D. Chandramouli, T. Hoglund, K. Doppler, T. Himanen, J. Laiho, et al., 5g mobile systems for healthcare, in: 2017 IEEE 85th vehicular technology conference (VTC Spring), IEEE, 2017, pp. 1–5. doi:10.1109/VTCSpring.2017.8108602.

[117] I. U. Rehman, M. M. Nasralla, A. Ali, N. Philip, Small cell-based ambulance scenario for medical video streaming: A 5g-health use case, in: 2018 15th International conference on smart cities: Improving quality of life using ICT & IoT (HONET-ICT), IEEE, 2018, pp. 29–32. doi:10.1109/HONET.2018.8551336.

[118] Q. Wang, J. Alcaraz-Calero, R. Ricart-Sanchez, M. B. Weiss, A. Gavras, N. Nikaein, X. Vasilakos, B. Giacomo, G. Pietro, M. Roddy, et al., Enable advanced qos-aware network slicing in 5g networks for slice-based media use cases, IEEE transactions on broadcasting 65 (2) (2019) 444–453. doi:10.1109/TBC.2019.2901402.

[119] V. Srivastava, M. Motani, Cross-layer design: a survey and the road ahead, IEEE communications magazine 43 (12) (2005) 112–119.

[120] Hevc hm reference software [software]," (2023), `https://vcgit.hhi.fraunhofer.de/jvet/HM`.

[121] W. Mathias, High efficiency video coding coding tools and specification (2015).

[122] I. T. S. Sector, Subjective video quality assessment methods for multimedia applications, ITU-T Recommendation (2008) 910.

[123] V. Sze, M. Budagavi, G. J. Sullivan, High efficiency video coding (hevc), in: Integrated circuit and systems, algorithms and architectures, Vol. 39, Springer, 2014, p. 40. doi:10.1007/978-3-319-06895-4.

[124] M. Wien, High efficiency video coding, Coding Tools and specification 24 (2015). doi:10.1007/978-3-662-44276-0.

[125] O. B. Maia, H. C. Yehia, L. de Errico, A concise review of the quality of experience assessment for video streaming, Computer communications 57 (2015) 1–12. doi:10.1016/j.comcom.2014.11.005.

[126] W. Zhou, Image quality assessment: from error measurement to structural similarity, IEEE transactions on image processing 13 (2004) 600–613.

[127] Z. Li, A. Aaron, I. Katsavounidis, A. Moorthy, M. Manohara, et al., Toward a practical perceptual video quality metric, The Netflix Tech Blog 6 (2) (2016) 2.

[128] M. A. Usman, M. G. Martini, On the suitability of vmaf for quality assessment of medical videos: Medical ultrasound & wireless capsule endoscopy, Computers in biology and medicine 113 (2019) 103383.

[129] A. Panayides, M. S. Pattichis, C. S. Pattichis, C. P. Loizou, M. Pantziaris, A. Pitsillides, Atherosclerotic plaque ultrasound video encoding, wireless transmission, and quality assessment using h. 264, IEEE Transactions on Information Technology in Biomedicine 15 (3) (2011) 387–397.

[130] R. Delhaye, R. Noumeir, G. Kaddoum, P. Jouvet, Compression of patient's video for transmission over low bandwidth network, IEEE Access 7 (2019) 24029–24040.

[131] C. J. Debono, B. W. Micallef, N. Y. Philip, A. Alinejad, R. S. H. Istepanian, N. N. Amso, Cross-layer design for optimized region of interest of ultrasound video data over mobile wimax, IEEE Transactions on Information Technology in Biomedicine 16 (6) (2012) 1007–1014.

[132] E. Cavero, A. Alesanco, J. García, Real-time echocardiogram transmission protocol based on regions and visualization modes, IEEE Journal of Biomedical and Health Informatics 18 (5) (2014) 1668–1677.

[133] Y. Wu, P. Liu, Y. Gao, K. Jia, Medical ultrasound video coding with h. 265/hevc based on roi extraction, PLoS One 11 (11) (2016) e0165698.

[134] S. Khire, S. Robertson, N. Jayant, E. A. Wood, M. E. Stachura, T. Goksel, Region-of-interest video coding for enabling surgical telementoring in low-bandwidth scenarios, in: MILCOM 2012-2012 IEEE Military Communications Conference, 2012, pp. 1–6.

[135] P. Maharjan, A. Alsadoon, P. Prasad, A. B. Al-Khalil, O. D. Jerew, G. Alsadoon, B. Chapagain, An enhanced algorithm for improving real-time video transmission for tele-training education., Multimedia Tools and Applications 81 (6) (2022) 8409–8428.

[136] A. Zahra, M. Ghafoor, K. Munir, A. Ullah, Z. Ul Abideen, Application of region-based video surveillance in smart cities using deep learning., Multimedia Tools and Applications (2021) 1–26.

[137] P. Salva-Garcia, J. M. Alcaraz-Calero, Q. Wang, M. Arevalillo-Herráez, J. B. Bernabe, Scalable virtual network video-optimizer for adaptive real-time video transmission in 5g networks, IEEE Transactions on Network and Service Management 17 (2) (2020) 1068–1081.

[138] C. Zhao, Q. Zeng, Y. Tang, B. Yang, Multi-path routing protocol for the video service in uav-assisted vanets, in: 2021 IEEE 94th Vehicular Technology Conference (VTC2021-Fall), 2021, pp. 1–5.

[139] M. Vafaei, A. Khademzadeh, M. A. Pourmina, A new qos adaptive multi-path routing for video streaming in urban vanets integrating ant colony optimization algorithm and fuzzy logic, Wireless Personal Communications 118 (2021) 2539–2572.

[140] M. Vafaei, A. Khademzadeh, M. A. Pourmina, Qos-aware multi-path video streaming for urban vanets using aco algorithm, Telecommunication Systems 75 (2020) 79–96.

[141] R. Lopes, M. Luís, S. Sargento, Real-time video frame differentiation in multi-homed vanets, Wireless Networks 27 (2021) 2559–2575.

[142] B. Garcia-Garcia, T. Bouwmans, A. J. R. Silva, Background subtraction in real applications: Challenges, current models and future directions, Computer Science Review 35 (2020) 100204.

[143] R. Kaur, S. Singh, A comprehensive review of object detection with deep learning, Digital Signal Processing 132 (2023) 103812.

[144] M. Mandal, S. K. Vipparthi, An empirical review of deep learning frameworks for change detection: Model design, experimental frameworks, challenges and research needs, IEEE Transactions on Intelligent Transportation Systems 23 (7) (2021) 6101–6122.

[145] I. Himawan, W. Song, D. Tjondronegoro, Impact of automatic region-of-interest coding on perceived quality in mobile video, Multimedia Tools and Applications 76 (2017) 785–813.

[146] B. G. Haskell, P. G. Howard, Y. A. LeCun, A. Puri, J. Ostermann, M. R. Civanlar, L. Rabiner, L. Bottou, P. Haffner, Image and video coding-emerging standards and beyond, IEEE Transactions on Circuits and Systems for Video Technology 8 (7) (1998) 814–837.

[147] D. S. Tommy Pauly, Eric Kinnear, An unreliable datagram extension to quic [accessed 20 jul 2024], `https://datatracker.ietf.org/doc/rfc9221/` (Mar. 2022).