# Enhancing the Quality of Video Streaming over Unreliable Wireless Networks

A Thesis Submitted for the Degree of

Doctor of Philosophy

By

Muhammad Usman

in

Faculty of Engineering and Information Technology

UNIVERSITY OF TECHNOLOGY, SYDNEY

AUSTRALIA

October 2017

**UNIVERSITY OF TECHNOLOGY, SYDNEY**

**Faculty of Engineering and Information Technology**

The undersigned hereby certify that they have read this thesis entitled "**Enhancing the Quality of Video Streaming over Unreliable Wireless Networks**" by Muhammad Usman and that in their opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Doctor of Philosophy.

| Principal Supervisor | Co-Supervisor |
| --- | --- |
| Prof. Xiangjian He | Dr. Min Xu |

# CERTIFICATE OF AUTHORSHIP

Date: **16$^{th}$ October 2017**

Author: Muhammad Usman
Title: Enhancing the Quality of Video Streaming over Unreliable Wireless Networks
Degree: PhD

I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of requirements for a degree except as part of the collaborative doctoral degree and/or fully acknowledged within the text.

I also certify that the thesis has been written by me. Any help that I have received in my research work and the preparation of the thesis itself has been acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This research is supported by UTS International Research Scholarship (IRS).

Signature of Author

# Acknowledgements

Pursing a doctoral degree is a long journey that one cannot make alone. I would like to thank all those who have assisted me in one way or another along this journey.

First and foremost, I wish to thank my supervisor Prof. Xiangjian (Sean) He. He has been supportive since the days I began working as a Ph.D. researcher. I remember, he always used to say that "Yes, I know, you can do this" to encourage me to move on and focus on my work. Ever since, Prof. Sean has supported me not only by providing a research assistantship, but also academically and emotionally through the rough road to finish this thesis. And during the most difficult times when writing this thesis, he gave me the moral support and freedom I needed to move on.

Similar profound gratitude goes to Dr. Min Xu, who has been truly dedicated to her role as my secondary supervisor. I am particularly indebted to Dr. Min for her constant faith in my research work. Her valued suggestions, constant support, and encouragement has kept me moving ahead at critical times. Without her help, I would not have been able to complete this thesis.

I am very glad to have worked with wonderful researchers, such as Dr. Mian Ahmad Jan and Dr. Syed Mohsin Matloob Bokhari. I have worked with them closely and learned a lot. They are my best buddies and I have shared many happy moments with them. Without their valuable research experiences, I would not have been able to balance my research and finish this degree. I am thankful to both of them for their continuous willingness to collaborate and discuss on various ideas during my research studies.

Finally, my deep and sincere gratitude to my family for their continuous and unparalleled love, help, and support. I am grateful to both my sisters (Shaista Irum and Seema Irum) for always being there for me as my best friends. I love our funny arguments and fights and the way how we make fun of each other. Special thanks to my funny, shy, and flirty brother (Zohaib Imran) who is always ready to help me. I am forever indebted to my

brother-in-law (Zulfiqar Khan Khattak) for giving me the opportunities and experiences that have made me who I am. He selflessly encouraged me to explore new directions in life and seek my own destiny. This paragraph would be incomplete if I do not say anything about the young devils of my family, i.e., my nieces (Maryam Niazi, Zunairah Zulfiqar, and Zoha Zulfiqar) and my nephews (Shaheer Niazi, Abdul-Hadi Niazi, Zulkifal Khattak, Zain-ul-Abdin Khattak, Daud Suleman, and Muzammil Wadood). The bond I share with these girls and boys, no-one can ever have. This journey would not have been possible without all these people, and I dedicate this milestone to them.

Dedicated to my family

# Table of Contents

# List of Tables

# List of Figures

# Abbreviations

| Abbreviations | Descriptions |
|---|---|
| **HEVC** | High Efficiency Video Coding |
| **QoS** | Quality of Service |
| **EC** | Error Concealment |
| **HD** | High Definition |
| **UHD** | Ultra High Definition |
| **ME** | Motion estimation |
| **WMSN** | Wireless Multimedia Sensor Network |
| **PLR** | Packet Loss Ratio |
| **QoE** | Quality of Experience |
| **VSP** | Video Service Provider |
| **ITU-T** | International Telecommunication Union-Telecommunication |
| **VCEG** | Video Coding Experts Group |
| **MPEG** | Moving Picture Experts Group |
| **VCL** | Video Coding Layer |
| **NAL** | Network Abstraction Layer |
| **MIMO** | Multiple-Input-Multiple-Output |
| **ER** | Error Resilience |
| **VoD** | Video on Demand |
| **FEC** | Forward Error Correction |

| | |
|---|---|
| **ARQ** | Automatic Repeat Request |
| **QP** | Quantisation Parameter |
| **SHVC** | Scalable High Efficiency Video Coding |
| **MANETs** | Mobile Ad-hoc Networks |
| **BL** | Base Layer |
| **EL** | Enhancement Layer |
| **JCT-VC** | Joint Collaborative Team on Video Coding |
| **CTU** | Coding Tree Unit |
| **MB** | MacroBlock |
| **CTB** | Coding Tree Block |
| **CB** | Coding Block |
| **CU** | Coding Unit |
| **SCB** | Smallest Coding Block |
| **LCB** | Largest Coding Block |
| **PB** | Prediction Block |
| **PU** | Prediction Unit |
| **MV** | Motion Vector |
| **TB** | Transform Block |
| **DCT** | Discrete Cosine Transform |
| **DST** | Discrete Sine Transform |
| **FS** | Full Search |
| **TSS** | Three Step Search |
| **FSS** | Four Step Search |

| | |
|---|---|
| **DS** | Diamond Search |
| **HS** | Hexagonal Search |
| **MC** | Motion Compensation |
| **URQ** | Uniform Reconstruction Quantisation |
| **SAO** | Sample Adaptive Offset |
| **CABAC** | Context Adaptive Binary Arithmetic |
| **HM** | HEVC test Model |
| **AI** | All-Intra |
| **LDP** | Low-Delay P |
| **LDB** | Low-Delay B |
| **RA** | Random Access |
| **GoP** | Group of Pictures |
| **RDO** | Rate Distortion Optimisation |
| **RD** | Rate Distortion |
| **SAD** | Sum of Absolute Difference |
| **TSAD** | Hadamard Transformed SAD |
| **SSE** | Sum of Square Error |
| **SVC** | Scalable Video Coding |
| **PSNR** | Peak-Signal-to-Noise-Ratio |
| **fps** | frames per second |
| **FIR** | Finite Impulse Response |
| **SHM** | SHVC test model |
| **LUT3D** | 3D Look-Up-Table |

| | |
|---|---|
| **SEC** | Spatial Error Concealment |
| **TEC** | Temporal Error Concealment |
| **HEC** | Hybrid Error Concealment |
| **PDF** | Probability Density Function |
| **KDE** | Kernel Density Estimation |
| **MESE** | Mean Error Square Estimator |
| **DP** | Dynamic Programming |
| **MFI** | Motion Field Interpolation |
| **FBM** | Flexible Block Matching |
| **TLRA** | Tensor Low Rank Approximation |
| **MVE** | Motion Vector Extrapolation |
| **MDC** | Multiple Description Coding |
| **EII** | Examplar-based Image Inpainting |
| **SI** | Spatial Interpolation |
| **PDIP** | Primal-Dual Interior Point |
| **SO** | Spare Optimisation |
| **RoI** | Region of Interest |
| **PIR** | Passive Infra Red |
| **BS** | Base Station |
| **WSN** | Wireless Sensor Network |
| **UWB** | Ultra Wide Band |
| **MSE** | Mean Square Error |
| **CSP** | Cloud Service Provider |

| | |
|---|---|
| **SaaS** | Software as a Service |
| **PaaS** | Platform as a Service |
| **IaaS** | Infrastructure as a Service |
| **CD** | compact Disc |
| **DVD** | Digital Video Disc |
| **BMA** | Block Matching Algorithm |
| **FC** | Frame Copy |
| **FI** | Frame Interpolation |
| **IV** | Input Video |
| **OV** | Output Video |
| **IB** | Input Block |
| **RB** | Reference Block |
| **SW** | Search Window |
| **TDLS** | Two Dimensional Logarithmic Search |
| **SES** | Simple and Efficient Search |
| **ARPS** | Adaptive Rood Pattern Search |
| **SSD** | Sum of Square Difference |
| **SHD** | Sum of Hamming Distance |
| **MAD** | Minimum Absolute Difference |
| **LSAD** | Locally scaled SAD |
| **LSSD** | Locally scaled SSD |
| **ZSAD** | Zero-mean SAD |
| **ZSSD** | Zero-mean SAD |

| | |
|---|---|
| **NCC** | Normalised Cross Correlation |
| **ZNCC** | Zero-mean NCC |
| **IFC** | Intelligent Frame Creation |
| **CF** | Concealed Frame |
| **AFO** | Adaptive Filter Output |
| **AMF** | Adaptive Mean Filter |
| **RAM** | Random Access Memory |
| **SNR** | Signal-to-Noise-Ratio |
| **SFI** | Scalable Frame Interpolation |
| **TZS** | Test Zone Search |
| **RS** | Raster Search |
| **FME** | Forward Motion Estimation |
| **FMV** | Forward Motion Vector |
| **BME** | Backward Motion Estimation |
| **BMV** | Backward Motion Vector |
| **MSN** | Multimedia Sensor Node |
| **IPS** | Indirect Path from Sender |
| **ACK** | Acknowledgement |
| **NACK** | Negative ACK |
| **IoT** | Internet of Things |
| **SDN** | Software Defined Network |

# Abstract

Real-time video transmission over unreliable wireless networks remains a serious challenge due to bandwidth limitation and sensitive nature of video bitstreams generated by today's complex video encoders, e.g., High Efficiency Video Coding (HEVC/H.265). These compressed video bitsreams face packet-drop problem when transmitted over unreliable wireless networks. The effect of packet-drop on the received video quality can be minimised in two ways 1) increasing Quality of Service (QoS) by adopting efficient routing schemes between source and destination, and 2) maintaining video quality at receiver's side by applying smart and real-time-based Error Concealment (EC) techniques. The QoS refers to the capability of a transmission network to provide better service to selected network traffic. It is a generic term and can be applied to any data transmission network. The term video quality refers to perceived video degradation and is compared to the original video. In this dissertation, we explore the above mentioned two ways and propose a comprehensive solution for real-time video transmission over unreliable networks with the contributions as follows.

1. An efficient, lightweight and real-time EC algorithm is proposed to conceal the missing/lost video frames in H.265 encoded HD videos. The EC algorithm is based on threshold-based distributed Motion Estimation (ME) scheme and utilises only two video frames to estimate the missing one, thus eliminating the need for a large buffer and processing of a bundle of video frames to estimate the missing one.

2. Scalable video coding produces multiple interrelated bitstreams of a single video with different bitrates. For scalable bitstreams, we propose a lightweight and real-time EC algorithm to cover up the effects of missing/lost video frames. Due to complicated nature of scalable video bitstreams, our proposed EC algorithm utilises three previously processed video frames along with their master video frames to perform threshold-based distributed ME to estimate the missing video frames in

enhancement layer.

3. We propose a feed-back-based on-demand multipath routing scheme over a multi-hop Wireless Multimedia Sensor Network (WMSN) to ensure the QoS. The feed-back helps in deciding the optimum path between sources and destinations and reduces the Packets Loss Ratio (PLR) during the transmissions. On-demand connection assists in saving the available network resources while multipath routing aids in maintaining the connection between sources and destinations.

The proposed research makes notable contributions to designing a QoS-supported HD video streaming paradigm to deliver HD videos over unreliable networks and to maintain the received video quality on resource-constrained mobile terminals.

# List of Publications

## Journal Papers

1. Muhammad Usman, Xiangjian He, Kin-Man Lam, Min Xu, Syed Mohsin Matloob Bokhari, and Jinjun Chen, Frame Interpolation for Cloud-Based Mobile Video Streaming, IEEE Transactions on Multimedia, vol. 18, No. 05, pp.831-839, 2016.
2. Muhammad Usman, Xiangjian He, Kin-Man Lam, Min Xu, Syed Mohsin Matloob Bokhari, and Jinjun Chen, Cloud-Hosted and Assisted Scalable Video Coding and Concealment for High Definition Videos, IEEE Transactions on Cloud Computing, (Accepted for Publication on 22 July 2017).
3. Muhammad Usman, Mian Ahmad Jan, Xiangjian He, Min Xu, and Kin-Man Lam, A Joint Framework for QoS and QoE for Video Transmission over Wireless Multimedia Sensor Networks, IEEE Transactions on Mobile Computing, (Accepted for Publication on 03 August 2017).

## Conference Papers

1. Muhammad Usman, Xiangjian He, Min Xu, and Kin-Man Lam, Survey of Error Concealment Techniques, Research Directions and Open Issues, 21st International Packet Video Workshop in Conjunction with 31st Picture Coding Symposium PCS, pp.233-238, 2015, IEEE).
2. Muhammad Usman, Mian Ahmad Jan, Xiangjian He, and Priyadarsi Nanda, Data Sharing in Secure Multimedia Wireless Sensor Networks, 15th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), 2016, IEEE.

Chapter 1

# Introduction

Multimedia applications and services are becoming common nowadays due to rapid development in telecommunication and Information Technology sectors. With the availability of high bandwidth wireless broadband networks and technological growth in mobile devices, people are able to watch and stream live programs, news, sports, on-demand video clips, movies, multimedia chats and calls. Latest mobile devices are able to capture very high resolution images, audio and videos. With the introduction of public platforms, such as Facebook, Twitter and Instagram, people like to share their status through images, audio and videos. With the evolution of wireless technologies and constant connectivity, devices, like smartphones, tablets, laptops and even portable TV can be connected anywhere at any time without cable mess.

However, multimedia applications and services demand a certain level of Quality of Service (QoS) and Quality of Experience (QoE). The QoS refers to the capability of a network to provide better service to selected network traffic while the QoE is a measure of the delight or annoyance of a customer's experiences with a service. Real-time multimedia applications require a high bandwidth and seamless connectivity while traditional protocols stack and network architecture are basically designed to provide a robust distribution of the bandwidth to a variety of applications. The situation becomes more challenging in a multi-

hop communication scenario. A multi-hop communication can be a wired, wireless or ad-hoc communication. In a multi-hop communication, intermediate nodes are known as relay nodes. In a communication network, it is not necessary that all nodes have the same energy level and computational power. The practical examples of such type of networks are mobile networks and Mobile Ad-hoc Networks (MANETs). Before transmission, the multimedia data is always compressed due to its large size. For videos, many video coding standards are proposed in past decades to compress raw video data. These video standards usually exploit both spatial and temporal redundancies to compress the videos. For video streaming, there can be different modes, such as store and play, on-demand or live streaming. Smooth video streaming depends upon various network related factors, such as availability of bandwidth, network congestion and transmission errors. Streaming protocols play an important role in controlling network congestion and minimising transmission errors, thus enhancing the overall QoS by providing a smooth end-to-end video streaming. QoE is dependent on various factors, such as video compression ratio, dropped video packets and network delays. Overall, QoE is used to rate a service or product from customer's satisfaction perspective.

## 1.1 Background

Concurrent transmission of audio and video is usually known as video streaming. Unlike buffer and play situation, video streaming can be either live or on-demand and does not require the waiting time. It can be a one-to-one or one-to-many streaming.

Conventional video streaming systems consist of three entities, i.e., streaming server, transmission medium and the receiver. In recent years, Video Service Providers (VSPs) have shifted their infrastructure to the cloud platform. The cloud platform offers sufficient computing and storage resources at an affordable cost. Due to sufficient computing resources, the streaming servers on cloud platform are able to process user's requests in real-time. The users access these streaming servers through the Internet. The Internet is

a combination of both wired and wireless networks and is never considered as a reliable transmission medium due to its best-effort delivery service. In recent years, we have witnessed drastic change in the development of mobile devices. Now, the users/receivers can stream the videos on their mobile devices without paying for cable connection.

The latest technological growth in video coding standards, e.g., HEVC/H.265 [1], has made it possible to stream videos over a broad range of bandwidth-constrained networks. However, smooth video streaming over bandwidth-constrained networks is not an easy task and poses many technical challenges. The multimedia applications are mostly capable of adjusting coding parameters with time-varying bandwidth but cannot deal with network impairments, such as packets loss, latency and queuing delays, at networking devices [2]. In this section, first we provide a generic overview of the HEVC. Next we briefly describe video streaming along with streaming protocols followed by QoS and QoE.

### 1.1.1 Video Coding Scheme

The HEVC, formally known as H.265 video coding standard, has been developed jointly by International Telecommunication Union-Telecommunication (ITU-T)'s Video Coding Experts Group (VCEG) and ISO/IEC's Moving Picture Experts Group (MPEG) [3]. It is a successor of the very popular H.264/AVC video coding standard [4]. Compared with H.264, H.265 promises to provide the same or an improved visual quality but at half bitrate [5]. It is basically designed to support videos of $1920 \times 1080$ or higher resolution. Depending on the multimedia application requirements, HEVC encoder can adjust encoding time, delay, compression rate and computational complexity. Unlike H.264, HEVC supports parallel processing. Similar to other video codecs, HEVC consists of two layers, i.e., Video Coding Layer (VCL) and Network Adaptation Layer (NAL). The former layer exploits spatial and temporal redundancies, encodes videos using Inter/Intra predictions and transform coding and then translates into bitstreams. On the other hand, NAL is responsible for transforming bitsreams produced by VCL into packets in a byte-oriented format, known as NAL units.

#### 1.1.1.1 Spatial Redundancy

Spatial redundancy means redundancy or excess of information inside a video frame. In this case, pixels contain either similar or almost similar values to their adjacent neighbours. If properly exploited, smart predictions can be made about the movement of neighbourhood pixels. Due to the correlation between neighbour pixels, less information needs to be encoded and transmitted, leading to frame compression. Such type of compression is known as Intra compression, in which each video frame is treated independently. Spatial redundancy can be explained with the following examples.

- In a facial image, pixels in specific regions of a face, e.g., cheeks, chin, forehead, etc., contain either the same or similar values. Hence, in these regions, the image contains lots of redundant information which can be removed during compression.
- In a crowded sport ground, there is enough information , e.g., ground colour, crowd seated, etc., which can be considered as redundant. In this case, the prediction about neighbourhood pixels become easy and can efficiently be utilised in video frame compression.

#### 1.1.1.2 Temporal Redundancy

Temporal redundancy is the amount of redundant information which exists in a group of video frames. For example, in a video shot, the numerical values of chrominance and luminance of all the pixels in an entire frame remain either the same or similar to the values of pixels in consecutive frames. During video compression, this type of redundancy is exploited and the idea is not to encode values of those pixels which are similar to those, which have already been encoded and transmitted. This type of compression is known as Inter compression. Temporal redundancy can be explained with the following examples.

- In a video conference, the movement is usually in the forms of hand gestures, body language and facial expressions but the background remains same in all the frames of a video session. The pixels of the background repeat their values in each frame

and can be encoded and transmitted just once during an Inter compression.

- In a live tennis match, the movement is represented by players and the ball that they play while the background, tennis court and audience remain the same in the entire match. This redundant information can be utilised to perform an Inter compression.

### 1.1.2 Video Streaming

Unlike download and play mode, video streaming mode is based on how the stream is received, decoded and played. In either mode, smoothness of transmission is essential. In the case of former mode, network latency produces long delays during downloading. Unless the download is completed, a user is unable to play the video. In the later mode, as soon as a bunch of video frames are received, the receiving terminal decodes and plays them. Due to networking delays during transmission, a user is unable to watch the video smoothly and needs to wait for the arrival of next bunch of video frames. The dynamic nature of channels of wireless networks, networking delays and packet-drop cannot be known in advance, and hence video streaming over wireless channels becomes a challenging task [6].

Wireless networks are always based on multi-channel transmission [7]. Usage of multi-channel transmission can help in improving the network performance by increasing network capacity for multimedia communication. Unlike single-channel-based communication, multi-channel communication can help in minimising the network congestion problem by load distribution through the Multiple-Input-Multiple-Output (MIMO) technology [8]. To stand against bit errors over unreliable wireless networks, the bitstreams of encoded videos are protected through an Error Resilience (ER) scheme [9]. An ER scheme is an essential part of all video encoders and is applied before forwarding a bitstream to a transmission network.

In the case of video streaming, video service providers lease multimedia servers from cloud service providers and use high speed data lines to handle millions of users to request for video streaming across the globe. The requests for video streaming vary over time. At

some particular moment or day, servers may become overloaded by user's requests, and increase the demand for more computing resources. An increasing demand for computing resources increases investment costs, making the streaming services expensive for users. Hence, a balance is required between an efficient management of computing resources and an investment to grab a wide range of customers.

Video-on-Demand (VoD) is another special case of video streaming. In VoD, a user watches a video in parts, i.e., requesting different parts of a video at different time intervals. Many solutions have been proposed to reduce the computational load of VoD servers [10]. Usually, each solution creates a group of users making requests for the same video. Due to such grouping, users need to wait for an initial connection time and become unable to rewind and fast-forward the video.

### 1.1.3 Streaming Protocols

Network impairments in the form of throughput fluctuations, packet-drops and delays affect smooth video streaming. These impairments produce effects, like frame freezing and distortion. There are many reasons behind these network impairments, such as mobility, inconsistencies in wireless channels, interference, shared medium and environmental changes.

Video streaming can be seen from two perspectives, at application layer and at lower layers, respectively. In the former perspective, it is assumed that the error control schemes are already installed and multimedia applications do not need to focus on errors. In this scenario, network impairments, like inconsistencies in wireless channels and fluctuations in network throughput become a serious problem and need to be tackled with care. To control these network impairments, lower layers play an important role. Lower layers can also offer error control mechanisms for smooth video streaming over wireless channels.

Latest trends in IEEE 802.11 standard offer more bandwidth and are able to transmit HD videos [11]. However, channel inconsistency in 802.11 standard is the biggest issue

for delay-sensitive video streaming applications. Wireless channels of 802.11 standard can easily get disturbed by other nearby wireless signals of almost the same frequency, environmental changes and mobility. Due to these problems, the PLR increases at physical layer. Most of the error control schemes at lower layers are based on retransmission [12]. Multiple retransmissions in a shared bandwidth scenario not only consume bandwidth but also causes more packet-drop.

Schemes, like Forward Error Correction (FEC) [13] and Automatic Repeat Request (ARQ) [14] already exist to control transmission errors. In ARQ, a sender sends the data and wait for a positive or negative acknowledgement from the destination. In the case of negative acknowledgement or if no acknowledgement is received for a predetermined period of time, the sender resends the data packet. On the contrary, the FEC is based on data redundancy. This scheme adds extra bits with data packets. This redundant data helps the receiver to fix the errors in the original data rather than putting a request for a retransmission. Although these techniques have been adopted in modern wireless technologies in hybrid forms, they are not suitable for real-time video streaming due to their extra processing and bandwidth consumption.

### 1.1.4 Quality of Service

The QoS is the overall performance of a network seen by its users and was defined by ITU in 1994 [15]. There are different quantitative metrics used to measure the QoS, such as jitter, transmission delay, bitrate, PLR, throughput and availability of service. In packet switched networks, resource reservation and traffic prioritisation are mostly used to provide QoS. The QoS is usually considered in special data transmission scenarios and video streaming is one of them. Best-effort networks, such as the Internet, cannot support QoS. In best-effort-based networks, the QoS can be ensured through heavy communications between the source and the destination. On the contrary, the heavy communications create extra traffic load and may become a significant issue at peak hours of network traffic.

Real-time video streaming over wireless networks can be disturbed by several parameters which can be distributed into two broad categories, i.e., sender-side parameters and network-based parameters. In the sender-side category, the major issue is the transmission of huge volume of data. In the case of HD and UHD videos, the situation becomes worse. To transmit videos over bandwidth limited networks, compression needs to be applied. HEVC/H.265 standard promises to provide half bitrate at the same visual quality as compared to H.264 standard. H.265 uses complex compression algorithms and various configuration parameters, such as Inter-frame coding and variable Quantisation Parameters (QPs) to get such a low bitrate. Low bitrate means the compression is lossy and most of the information is thrown out during the encoding procedure. Compressed video bitstreams generated by H.265 are quite complex and sensitive to bit errors. A single bit modification can change the entire video bitstream, resulting in either drop of an entire video bitstream by the decoder or disturbed video quality after decoding.

On the other hand, network-based category parameters are very difficult to control, especially when it comes to the Internet and wireless networks. Mostly, the video streaming applications demand guaranteed end-to-end communication during the entire communication session. Due to best-effort nature of the Internet, providing guaranteed end-to-end communications in an entire communication session is not possible without doing extra efforts. Similarly, the wireless networks offer a shared medium for communication and other nodes of the network may suffer due to the guaranteed end-to-end communication. Due to unreliable nature of the Internet and the wireless networks, the packets loss problem is very common and can turn into burst errors during peak traffic hours. A data packet may contain either a part of a video frame or an entire video frame. In either situation, a data packet drop means an entire frame loss, which disturbs the received video quality.

### 1.1.5 Quality of Experience

The QoE/QoX represents the overall level of customer's satisfaction with a service and to some extent, is user-dependent. QoE is different from QoS. In QoS notion, the service can

be improved and somehow can be guaranteed through modifications at both hardware and software level. On the contrary, QoE expresses user's satisfaction both from subjective and objective points of view. Based on that, QoE can be applied to any product or service. There are many factors that can influence QoE, such as reliability, security, privacy, cost, user's terminal hardware, user friendliness, importance of application and transmission environment (wired/wireless).

From end-to-end performance perspective, bad network performance can certainly produce major effects on user's QoE. While designing any communication system for customers, an expected QoE is always taken into account. As the QoE is a feedback from customers and is measured at the end-device side, we can say that it is calculated after a distortion is introduced in a network. There are many factors that can bring down the QoE of video streaming, such as encoding system, transmission system, multimedia application and end devices. In videos, it is usually measured by trying to find out the artefacts in the received videos rather than comparing the received videos with the original ones. The objective and subjective evaluations have their own pros and cons. The subjective evaluation is considered as more authentic but requires a large amount of human resources and is a time consuming process. On the other hand, the objective evaluation is a machine-based solution and takes less time. The verification of objective evaluation requires lots of computing resources, and sometimes it is considered unreliable.

## 1.2 Motivation

In the past few decades, wireless networks and communications have become the fastest growing industries. Standards and technologies, like IEEE 802.11-based Wireless LANs, IEEE 802.16-based Wireless MANs, IEEE 802.15-based Wireless PANs and 4G mobile communication, have opened doors for many multimedia applications and the ways in which those applications can be used. On the other hand, in multimedia applications, download and play, video streaming, VoD and live streaming have become very common

and popular in recent years. Fig. 1.1 shows the global 4K video traffic forecast from 2015 till 2020 [16]. With advancement in technology, videos are getting higher in resolution, such as HD ($1920 \times 1080$) and UHD (4K). Due to higher resolution, UHD videos are more sharper and clearer than HD videos. A UHD display offers at least 8 million active pixels and is usually standardised as $3840 \times 2160$ for television and $4096 \times 2160$ for digital cinema.

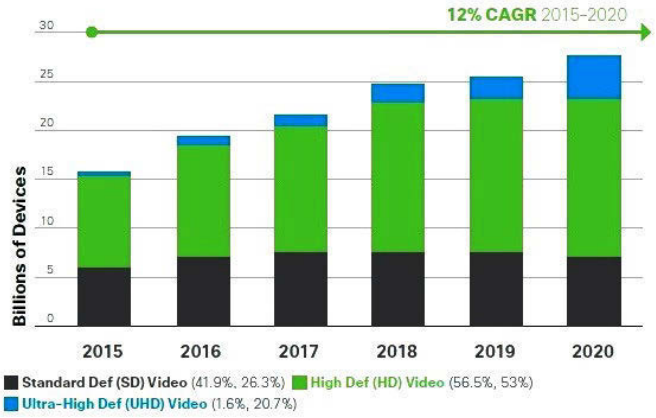



Figure 1.1: Global 4K video traffic [16]

Due to an increasing resolution, the videos are getting higher in volume and require sufficient amount of bandwidth for transmission. Popular multimedia service provider Netflix recommends steady connection of at least 25Mbps downstream speed for 4K video streaming [17]. This downstream speed is only possible if we get a direct connection to the streaming server. Video streaming over the Internet requests the use of low and high speed networks, so video compression is a suitable option. Recent research and development in the video compression domain have made video compression algorithms robust and efficient. The most recent H.265 standard offers better resolution at half bitrate as compared to its predecessor standard. Due to variations in bandwidth, environment and underlying technology, the developers of H.265 has introduced its scalable version, formally known as Scalable High Efficiency Video Coding (SHVC). Before transmission, the SHVC changes the bitrate of the compressed video according to the user's network bandwidth and device's capabilities.

Due to advancements in multimedia technology and wireless communications, it will not be far when most of the data travelling over the next generation Internet will be of multimedia as shown in Fig. 1.2. Mobile device vendors are focusing on multimedia enriched features in their devices to cope with growing demand of multimedia supported devices from the users in recent years. Currently available mobile devices in market can support HD video streaming and allow users to capture images, record videos in HD form and share multimedia data over the Internet as shown in Fig. 1.3.

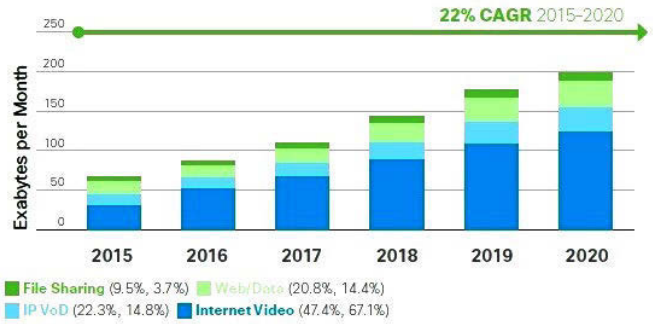



Figure 1.2: Global IP traffic by application category [16]

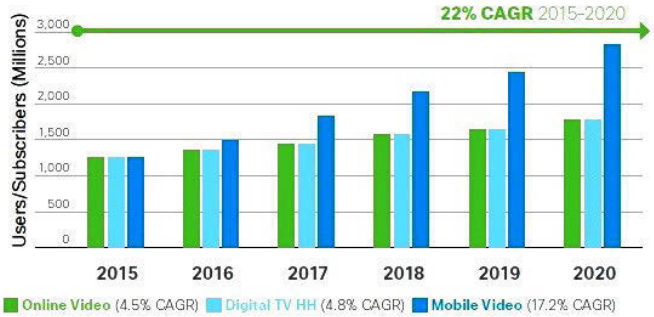



Figure 1.3: Mobile video traffic [16]

However, smooth streaming of HD and UHD videos over the Internet not only demand sufficient bandwidth but is also sensitive to other network impairments, such as delay, packet-drop and interference in wireless environment. The idea of the Internet was to provide reliable communication between computers for non-multimedia form of data.

However, the Internet cannot offer any guarantee for reliable end-to-end video communications. These factors have raised challenges for researchers and developers, working in telecommunication and networking domains, to think, design and develop solutions to provide reliable end-to-end video communications over the current and the future Internet.

To ensure QoS and QoE over wireless networks, cross-layer design can be considered as an effective and optimal approach. Cross-layer designs can be used to provide a balance between network throughput and received video quality. In the past, cross-layer designs have been applied in may areas, such as networking, energy efficiency and scheduling and queuing [18–20]. However, cross-layer design combined with EC technique for H.265 encoded HD videos has not been explored yet and raises following research problems:

- Bitrate of pre-encoded videos cannot be adjusted according to varying network conditions and user terminal's capabilities. In recent years, usage of portable devices is becoming very common. These portable devices mostly rely on chargeable batteries. Due to mobility, varying channel conditions and limited battery power, a user's device may not be able to process the same bitrate all the time. In order to achieve an optimal performance in a cross-layer design, wireless communications need to be combined with video coding and EC. Previously, some solutions have been proposed for a cross-layered design using multimedia streaming over wireless networks [21–23]. However, those proposals only focus on QoS and do not consider HD/UHD video coding and EC.
- To increase or maintain a certain level of QoS, many factors need to be considered, e.g., throughput, latency, jitter, energy consumption, queuing, buffering, etc. The existing work for QoS targets on some of these factors while ignoring the rest, thus leaving a need for a complete package [24]. The situation can become worse in the case of transmitting a large volume of real-time multimedia data over a long distance in a multi-hop communication. The QoS for a multi-hop multimedia communication demands to consider all those factors that can help in providing a guaranteed end-to-

end multimedia communication.

- The users of a video streaming service usually do not understand the technical details and problems that may happen during video streaming. They can just rate the service as either good or bad. A positive review from the users can definitely help in increasing popularity and marketing of a video service provider. The QoS alone is insufficient to achieve this target due to unreliable nature of wireless networks and best-effort delivery of the Internet. However, a balanced combination of QoS and QoE can help in increasing the overall rating of a video streaming service. Apart from the best design to achieve higher QoS, network impairments, like packet-drop, delay, power failure, buffer overloading, etc., will always be there. These impairments can be minimised but cannot be eliminated. To cover up the effects of these impairments, EC techniques can play an important role. The EC techniques always reside and run on the receiver's terminal. The main role of an EC technique is to estimate/recover missing video frames lost/damaged during transmission, thus can help in increasing the level of QoE. Due to the popularity of mobile technology, users are mostly relying on wireless technology and are using portable devices, such as smartphones, tablets and laptops. These portable devices are dependent on chargeable batteries and vary in hardware configurations. Due to these limitations, there is a need to propose a lightweight and energy-efficient EC technique for real-time HD/UHD video streaming.

## 1.3 Research Objectives and Contribution

In Section 1.2, we highlighted various research issues related to maintain QoS and QoE. These research issues motivate us to investigate them and to set objectives and contributions for this thesis. The most relevant objectives and their novelties are explained as follows.

1. A packet loss model based on random distribution is designed for H.265-based Intra-

encoded HD videos. During encoding, each video frame is treated independently and is encoded as one data packet. The proposed packet loss model reads the encoded video bitstream and discards video packets consecutively/randomly. Unlike the existing models where the PLR was set to be 1%, 3% or 5% [2], the PLR in our proposed model can be set manually to any ratio.

2. An EC algorithm is proposed to conceal a whole video frame, lost/dropped during a video streaming for H.265 Intra-encoded HD videos. The proposed algorithm does not demand any modification in the decoder architecture and runs as a separate module on a user's portable terminal. Due to computational and power limitations, the proposed EC algorithm is based on threshold-based parallel processing. HD video frames are large in size and rich in resolution. The threshold-based parallel processing helps in speeding up the concealment process by dividing the whole video frame into equal-sized partitions and dealing each partition independently and in parallel. The threshold-based parallel processing decreases the computational time and helps in saving the stored charge.

3. An EC algorithm is proposed to conceal a whole video frame, lost/dropped during a cloud-assisted SHVC encoded HD video streaming. In a cloud-assisted scalable video streaming, multiple layers of various bitrates are generated. These layers are classified as Base Layer (BL) and Enhancement Layer (EL). There is always one BL while multiple ELs can be generated. The ELs are always dependent on a BL. A single video frame loss in BL may bring down an entire EL dependent on it. Therefore, care must be taken while concealing the lost BL video frame. The proposed EC algorithm is able to conceal both BL and EL video frames through a threshold-based parallel processing, and it doesn't require any alteration in the decoder architecture. Due to a threshold-based parallel processing, it can readily be applied on chargeable mobile devices.

4. We propose an efficient HD video streaming scheme over a secured WiFi-based WMSN. To control the amount of traffic in the network introduced by malicious

nodes, a cluster-based authentication scheme is applied. This authentication scheme not only authenticates the malicious or non-member nodes but also authenticates the data. In this scheme, any unauthorised data is simply discarded by the cluster head who is responsible for forwarding the data to the base station. This authentication scheme helps in increasing the network throughput by forwarding the authentic data only. Due to the unreliability of wireless channels, the packet-drop issue is always there. To cover-up the packet-drop issue and to improve the received video quality, a lightweight EC scheme is combined with the proposed authentication scheme.

5. We propose a framework to meet QoS and to maintain the received video quality for HD video streaming over multi-hop WMSN. In this proposed framework, an SHVC-based efficient video encoding scheme is introduced for multimedia sensor nodes. The main purpose of this encoding scheme is to adaptively adjust the video bitrate based on the current transmission channel status. This encoding scheme is combined with a feedback-based multipath packet scheduling algorithm to improve the QoS. A lightweight EC is introduced as a backup technique to cover-up the packet-loss effect in received videos.

## 1.4 Structure of the Thesis

The rest of this thesis is organised as follows. A review of the prior research works on EC for Intra and Inter-encoded videos and video streaming over multi-hop wireless networks is conducted in Chapter 2. Chapter 3 proposes a lightweight and threshold-based frame interpolation algorithm for cloud-assisted H.265 encoded HD videos. A cloud platform is used to encode and store HD videos while an EC algorithm is running on user's mobile terminals to conceal Intra frames only. Chapter 4 proposes a smart and energy-efficient EC algorithm for cloud-assisted SHVC encoded HD videos. A cloud platform is used to encode and store HD videos while the proposed EC algorithm deals with both Intra and Inter frames on user's terminals. This EC algorithm is based on the threshold-based

frame interpolation algorithm of Chapter 3 as an underlying platform for its operation. In Chapter 5, a feedback-based multi-hop transmission framework is proposed for HD video streaming over a WMSN. The proposed framework helps in increasing the QoS and in maintaining the level of received video quality. Finally, a summary and future work are drawn together with the thesis conclusion in Chapter 6.

Chapter 2

# Literature Review

In the past few years, the Internet and technology are becoming essential parts of our daily lives. Most of us carry technological gadgets that can help us to access and to share multimedia data, like audio, videos and images, through the Internet on public platforms. Unlike simple data applications, multimedia applications are bandwidth sensitive and demand a certain level of QoS. A rapid development has been seen in recent years in the wireless technology domain to support bandwidth sensitive applications. However, these Internet-based wireless technologies promise best-effort services and cannot guarantee end-to-end communication. Wireless channels face many network impairments, such as interference, fading and path loss during a transmission. Error rates caused by these impairments become significant during end-to-end communications. A higher error rate produces notable artefacts on the received video quality and impacts the reputation of video service providers. The situation becomes more difficult in multi-hop networks. The most complex form of multimedia data is a video. Together, videos and wireless networks, make it difficult and challenging to maintain a certain level of QoS and received video quality.

In this chapter, we present a brief survey of related works. In Section 2.1, we provide a brief overview of the work flows of H.265 video coding standard in simple and scalable domains. In Section 2.2, various video error concealment techniques are discussed. An

overview of QoS-ensured video streaming over a WMSN is provided in Section 2.3. A detailed description of layered and cross-layered architectures is also discussed in this section and it reflects our goal to develop a smart feedback-based multipath routing protocol for a multi-hop WMSN. We conclude this chapter with a discussion of multimedia computing on a cloud platform in Section 2.4.

## 2.1 Overview of Video Coding Standard

### 2.1.1 High Efficiency Video Coding

High Efficiency Video Coding (HEVC/H.265) is a globally recognised video coding standard, developed by Joint Collaborative Team on Video Coding (JCT-VC), a joint venture of ISO/IEC MPEG and ITU-T VCEG [1]. The main objective behind the development of this video coding standard is to improve encoding performance, achieve a better compression rate and produce a higher visual quality as compared to its predecessor video coding standard H.264/AVC. Unlike H.264 standard, the H.265 standard can support up to 8K resolution with bit depth of up to 16 bits per sample.

#### 2.1.1.1 Workflow of Encoding and Decoding

Both in H.264 and H.265 standards, encoder is more complex than the decoder. Compared to H.264, the encoder of H.265 offers more options to encode videos according to the user's application requirements. Standard block diagrams of encoders and decoders of H.265 standard are shown in Fig. 2.1 and Fig. 2.2, respectively, where decoding is the reverse procedure of encoding.

H.265 is a hybrid block-based video coding standard. During encoding, the encoder partitions a video frame into non-overlapping regions, formally known as blocks. Then, these blocks are predicted by either an Intra or an Inter prediction method. Intra prediction keeps the scope of prediction within the same video frame while the Inter prediction

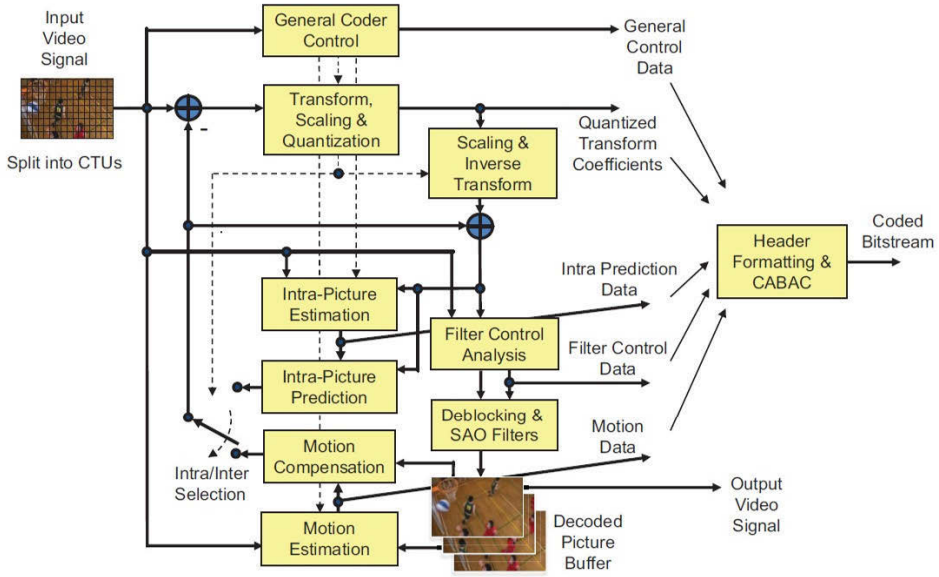



Figure 2.1: HEVC encoder [1]

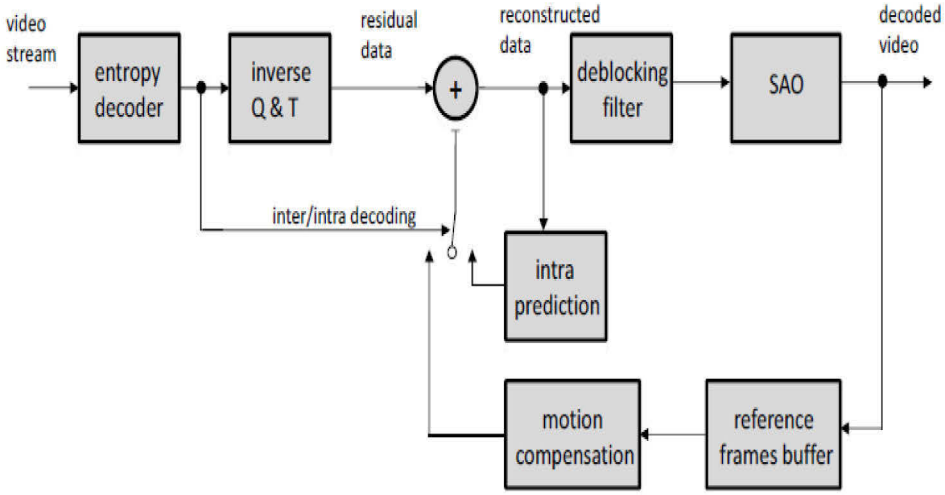



Figure 2.2: HEVC decoder [25]

exploits the temporal redundancy and involves more than one video frame during the prediction process.

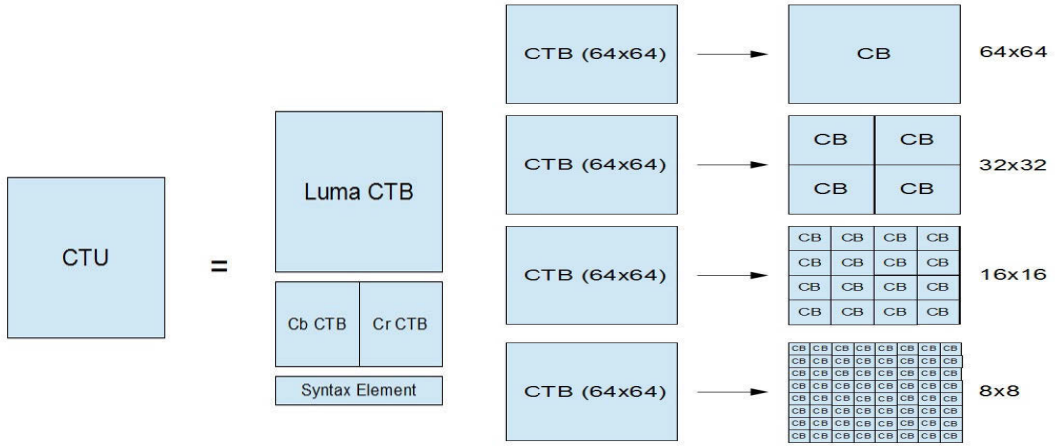
#### 2.1.1.2 Structure of H.265 Encoder

A high-level framework of H.265 allows to carry coded video data along with associated information over transmission networks. The coded video data along with associated information is known as bitstream. Following coding units play important roles in generating the video bitsreams.

##### 2.1.1.2.1 Video Frame Partitioning

**2.1.1.2.1.1 Coding Block** During the coding process, the Coding Tree Unit (CTU) plays the same role as MacroBlock (MB) played in H.264. The CTU consists of Coding Tree Blocks (CTBs) from luminance and chrominance elements. The CTB is of a square size, i.e., $L \times L$, where $L$ can be chosen as 16, 32 or 64. A larger CTB size offers a better compression. In a flexible framework, a CTB can further be divided into smaller blocks, known as Coding Blocks (CBs) and can be represented by a tree or quad-tree structure. One luma CB and two chroma CBs, together with associated syntax, form a Coding Unit (CU). The size of a CB is variable with defined minimum and maximum limits. The minimum size is known as Smallest CB (SCB) while the maximum size is called Largest CB (LCB). Fig. 2.3 and Fig. 2.4 show examples of a CTU with sub-CBs and a CU, respectively.

**2.1.1.2.1.2 Prediction Block** The decision to predict a region of a video frame either in Intra or Inter domain is made at the CU level. In order to make such prediction, the luma and chroma components of a CB is divided into smaller regions and predicted from luma and chroma Prediction Blocks (PBs). The PBs of a CB together form the Prediction Unit (PU) of a CU. For prediction, Motion Vectors (MVs) and prediction modes play important roles. Intra redundancy is removed by applying Intra prediction on the pixels of the nearby PBs in the same video frame. On the other hand, motion estimation and compensation are used to exploit temporal redundancies. A PB can be of symmetric or asymmetric shape and there can be multiple PBs within a single CB. H.265 allows variable PB sizes from $4 \times 4$ to $64 \times 64$ samples. Fig. 2.5 shows an example of a CB with PBs.




(a) Coding tree unit (b) Coding tree block

Figure 2.3: Coding tree unit architecture [26]

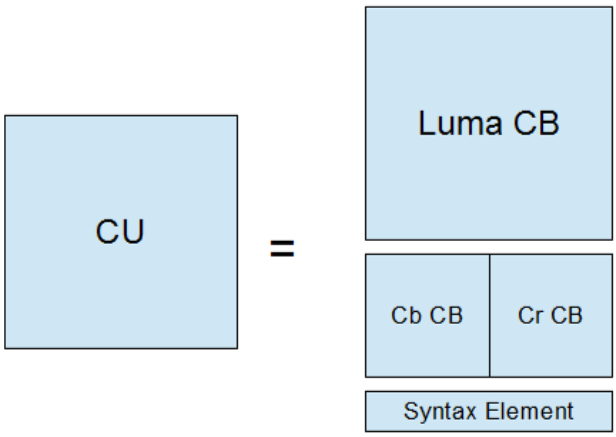



Figure 2.4: Coding unit [26]

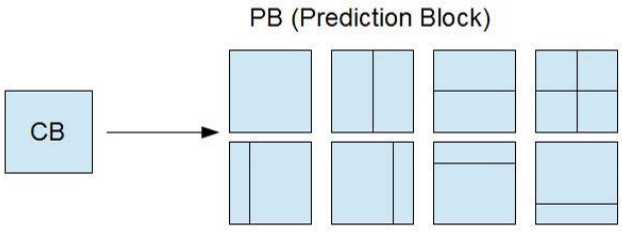



Figure 2.5: Prediction blocks [26]

**2.1.1.2.1.3 Transform Block** Block transforms are used to code prediction residuals. The residual of a luma CB may seem identical to a luma Transform Block (TB) or may further be divided into smaller TBs. The same applies to a chroma TB. Integer basis functions are defined for symmetric sizes of a TB, such as $4 \times 4$, $8 \times 8$, $16 \times 16$ and $32 \times 32$. These integer basis functions are similar to Discrete Cosine Transform (DCT). In Intra prediction, an integer basis function similar to Discrete Sine Transform (DST) can alternatively be applied on a $4 \times 4$ sized TB. These TBs form a TU at the CU level. Fig. 2.6 shows an example of a CB with TBs.

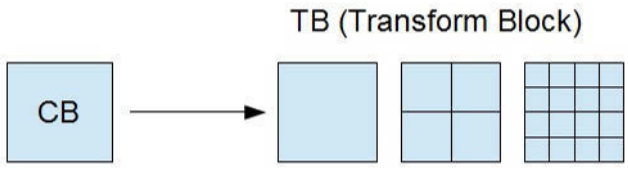



Figure 2.6: Transform blocks [26]

**2.1.1.2.2 Coding in H.265** The video coding layer of H.265 applies the same hybrid approach, as was applied in previous video coding standards. Video frames are processed through Intra/Inter prediction blocks to exploit the spatial and temporal redundancies. The Inter prediction is supported by motion estimation and compensation blocks. The outputs of these blocks are passed through transformation, quantisation, in-loop filtering and entropy coding phases to generate the bitstream for network abstract layer.

**2.1.1.2.2.1 Intra Prediction** In PB regions, the boundary samples of neighbourhood blocks are used for spatial predictions in the absence of Inter prediction. This situation is known as Intra prediction. H.265 offers 33 directional Intra prediction modes along with planar prediction and DC prediction modes as shown in Fig. 2.7. Planar prediction modes are used to improve subjective visual quality by removing blocking artefacts. A DC prediction mode is used to generate a mean value by taking the average of referenced samples and can be utilised for flat regions of a PB. The selection of a specific Intra prediction modes depends on the Intra prediction mode selected for previously processed PBs.

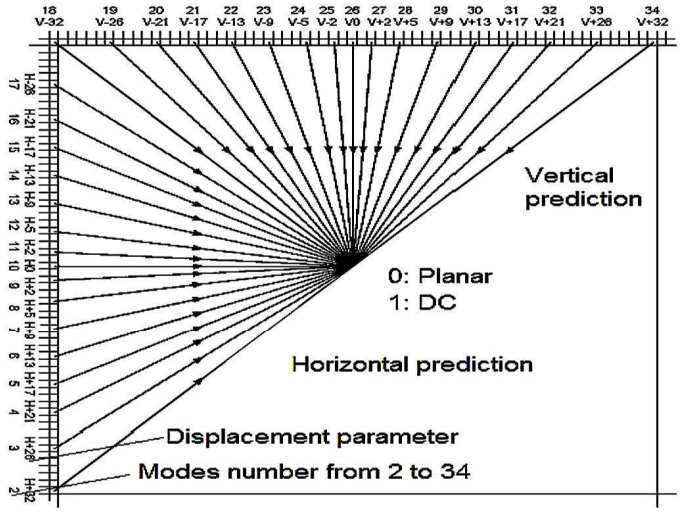



Figure 2.7: Intra prediction modes [27]

**2.1.1.2.2.2 Inter Prediction** In H.265, conventional motion estimation and compensation are utilised for Inter prediction. Temporal correlation between video frames is exploited to extract motion compensation information. Just like Intra prediction, Inter prediction is also block-based. For a block of pixels in the current video frame, a block of pixels in previously processed video frames is used as a predictor. The concept of Inter prediction is based on translational motion model where MVs help in tracking the motion of individual pixel or block of pixels in consecutive video frames as shown in Fig. 2.8.

**2.1.1.2.2.3 Motion Estimation** The process of matching the block of the current video frame with the blocks of reference video frame and finding the best match is known as Motion Estimation (ME). The displacement between the block in the current video frame and the best matched block in the reference video frame is represented by an MV and the difference value between the current and the reference blocks is known as residue. Block distortion is used to find the best match and can be defined as

$$D_{(x,y)} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |f_t(x,y) - f_{t-1}(x+i, y+j)| \tag{2.1}$$

where $f_t(x,y)$ represents the intensity value of a pixel in the current video frame while

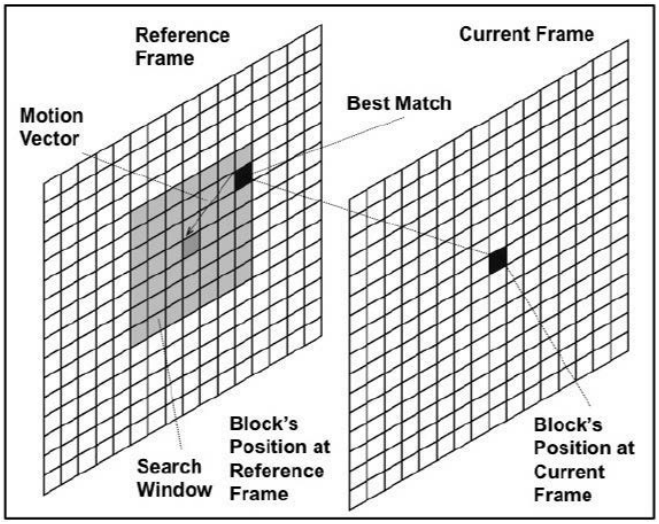



Figure 2.8: Inter prediction [28]

$f_{t-1}(x,y)$ represents the intensity value in the reference video frame.

The MV representing the minimum distortion within a Search Window (SW) can be found as

$$MV(x,y) = \min_{x,y \in SW} D(x,y) \tag{2.2}$$

There are many search algorithms to find the MVs, such as Exhaustive or Full Search (FS), Three Step Search (TSS), Four Step Search (FSS), Diamond Search (DS), Hexagonal Search (HS) and many others [29]. The most popular searching technique is FS. It produces the highest accuracy at the cost of computational complexity.

**2.1.1.2.2.4 Motion Compensation** Motion Compensation (MC) is an algorithmic technique and is used to predict a video frame with respect to a reference video frame. The reference video frame can be a previous and/or future video frame. If the current video frame is predicted from the previous video frames only, then such a prediction is called P-frame based prediction. On the other hand, if the prediction is bidirectional, i.e., relying on both previous and future video frames, then such a prediction is known as B-frame based prediction. The MC is a compulsory block in video standards to compress videos. The MC

can be either block-based or pixel-based. In a block-based MC, the current video frame is divided into fixed/variable size blocks and these blocks are predicted from the blocks of the same size in the reference video frame. The block-based MC is fast in processing but introduces discontinuities at the edges of blocks in the video frames and can produce false results in sharp regions. On the other hand, the pixel-based MC produces accurate results and can reach quarter-pixel levels at the cost of computational complexity. Usually, in a video shot, most of the video frames are almost representing the same data, i.e., either one or more objects are moving but the remaining parts in the scene are same or the camera is moving and everything in the scene is stationary. In such situations, the videos can efficiently be compressed using a MC procedure where the compressed video bitstream contains only few video frames along with important information used to retrieve the other video frames during decompression.

**2.1.1.2.2.5 Transformation** Block-based transforms, such as DCT or DST, are used in H.265 to convert residual signals produced from Intra or Inter prediction to frequency domain in order to compact and decorrelate the information. This transform process is based on fixed point integer operations. The transform matrices supported by H.265 are of size of $4 \times 4$, $8 \times 8$, $16 \times 16$ and $32 \times 32$, and are known as core transform matrices. Intra predicted $4 \times 4$ matrix is processed by a DST while the remaining matrices are processed through a DCT. For simplicity, these 2D transforms can be separated and processed in 1D spaces, i.e., being processed in vertical and horizontal directions at a time. For transformation, each CB is divided into TBs using a quad-tree scheme where the largest TB has the same size as that of a CB. Large TBs are divided into $4 \times 4$ blocks. In these matrices, the processing starts with last significant value and ends up at the DC coefficient in reverse scanning order to obtain the transform coefficients. The scanning patterns are shown in Fig. 2.9. The transform matrices can be processed quickly with fewer mathematical operations due to their symmetric properties.

**2.1.1.2.2.6 Quantisation** After motion estimation and compensation, transformation is applied to obtain transform coefficients. These transform coefficients cannot help

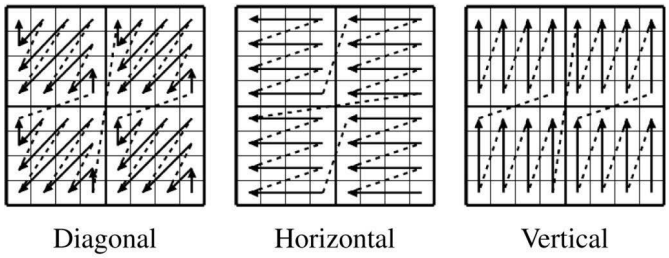



Figure 2.9: Scanning patterns [30]

in compression. However, when the DCT is combined with quantisation, lossy compression can be achieved. In H.265, quantisation is performed using Uniform-Reconstruction Quantisation (URQ) procedure where the transform coefficients are represented by specific values in quantisation matrix. The URQ is controlled by QPs, ranging from 0 to 51. Smaller values of QPs can hold more details and produce less compression as compared to higher values. For example, in a frequency domain, high frequencies represent sharp edges while low frequencies represent average brightness. Because a human vision system is more sensitive to low frequency regions, smaller QP values are selected for low frequency components while high frequency components are represented by higher QP values.

**2.1.1.2.2.7 In-loop Filtering** Two filters are used in H.265, i.e., deblocking and Sample Adaptive Offset (SAO) filters, as shown in Fig. 2.1. Due to the block-based nature of coding schemes in H.265, deblocking filter is used to minimise the visible structures of blocks in the edges of transform blocks. On the other hand, SAO is a sampling-based filter and works at CTU level. It is designed for sample values of local neighbourhood and does not consider the edge values. It is usually used in combination with deblocking filter to reduce the ringing effects. The DCT can give better results in flat regions, however, on contours and in the presence of noise, the performance of the DCT is compromised. It can perform well for large sized blocks. In small sized blocks, the visual artefacts are visible. These artefacts can be reduced by deblocking filters on the edges of TBs while the artefacts inside the TBs are minimised by SAO filter.

**2.1.1.2.2.8 Entropy Coding** The quantised transform coefficients combined with prediction modes, MVs and other related data are coded through the Context Adaptive Binary Arithmetic (CABAC) to construct the video bitstream. This CABAC technique is used as an entropy coding in H.265. CABAC is based on an arithmetic coding and provides good compression performance. It uses probability models to estimate the context coded bins. The three major components are depicted in Fig. 2.10.

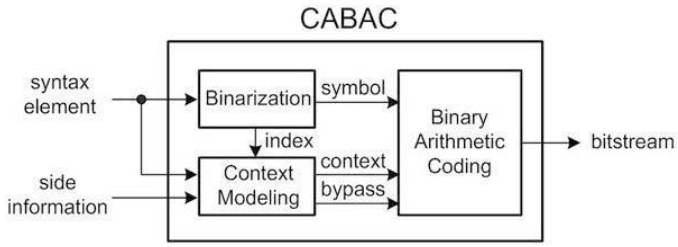



Figure 2.10: Modules of CABAC [31]

### 2.1.1.3 Reference Software

Reference software is used to encode and decode videos in video standards. In H.265, it is called HEVC test Model (HM) with recent version HM 16.14 [32]. Several encoding configuration modes are available in this version, such as All-Intra (AI), Low-Delay P and B (LDP and LDB) and Random Access (RA). In the AI mode, only Intra prediction is considered and all video frames are treated independently without referring to any previous and/or future video frames. The LDP and LDB modes are suitable for low delay scenarios where the video frames are organised as a group of I-frame, P-frame and B-frame. In each group, there can be multiple P and B-frames but there will be only one I-frame. The RA mode is suitable for a broadcasting scenario and the group of frames consists of P and B-frames only. This group of frames is formally known as Group of Pictures (GoP). In HEVC, the quality of encoding is tested through a Rate-Distortion Optimisation (RDO) process [33]. The RDO is used to select the encoding mode which helps in generating the least Rate-Distortion (RD) cost. Multiple options are available in H.265 to measure the RD cost, such as Sum of Absolute Difference (SAD), Hadamard Transformed SAD (TSAD)

and Sum of Square Error (SSE).

### 2.1.2 Scalable Video Coding

With the evolution in digital video industry and rapid development in streaming applications, the usage of interactive multimedia is becoming popular day by day. In order to receive the best quality of streaming video, transmission channels should be able to provide constant connectivity and reliability of connection. Nondeterministic transmission channels and heterogeneous networks often are unable to provide best network support. In such situations, the video streams should be adaptive with network conditions in terms of bitrate, frame rate and resolution. Conventional video encoders cannot deal with such scenarios, thus posing a great challenge for many video streaming applications.

This challenge can easily be dealt by Scalable Video Coding (SVC). In this style of coding, a video stream is divided into multiple streams with distinguishable and additional components. These streams are mostly known as layers. An encoder can create multiple layers with different bitrates, frame rates and resolutions. On the other hand, a receiver can select and decode either one or multiple layers according to the network and device capabilities as shown in Fig. 2.11.

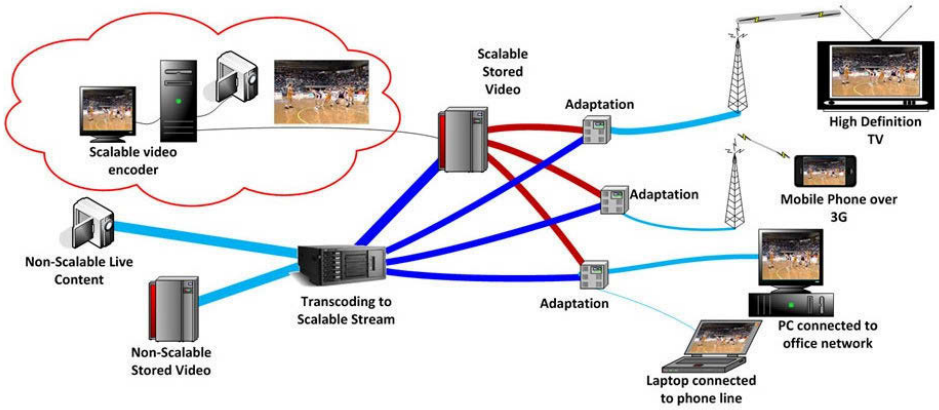



Figure 2.11: Scalable video coding [34]

In scalable video coding, there is always one BL and one or more ELs. These ELs are

encoded based on the encoded BL. The scalability is usually available in spatial, temporal and quality-based flavours. Spatial scalability produces video streams with less spatial resolution, temporal scalability disturbs the frame rate and quality scalability produces bitstreams with lower Peak-Signal-to-Noise-Ratio (PSNR) as shown in Fig. 2.12. These flavours help in creating different versions of the same video in terms of bitrate, frame rate and resolution. These different versions of a video help in dealing with problems, like storage space, data rate in broadcast environment and video streaming over unreliable multi-hop networks.

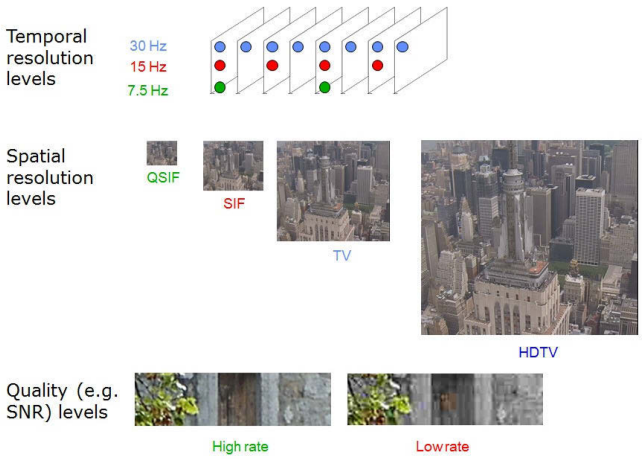



Figure 2.12: Scalability types [35]

### 2.1.2.1 Overview of Scalable High Efficiency Video Coding

Scalable High Efficiency Video Coding (SHVC) is an extension of HEVC/H.265, jointly developed by ISO/IEC-MPEG and ITU-T VCEG and was standardised in 2014 [36]. In SHVC, a video is encoded as a BL with basic version and then as ELs with enhanced versions. It uses HEVC encoding features to create these multiple layers. The HEVC slice headers of video bitstreams are modified to get the SHVC architecture. The compression tools of HEVC are combined with inter-layer predictions to generate highly compressed SHVC video bitstreams. SHVC architecture is multi-loop based. At the decoder side, the BL is decoded first followed by ELs, so that the BL should be available for prediction

references. Compared to single-loop design of an HEVC decoder, the multi-loop design of an SHVC decoder is more complex and demands much more memory and computational power.

Compared with the SVC of H.264, SHVC can support videos up to 8K resolution [37] and comes with following enhanced coding options:

- Support for higher frame rate in EL, i.e., 60 frames per second (fps) in EL while 30 fps in BL
- Support for higher resolution in EL, i.e., UHD in EL while HD in BL
- Support for a combination of higher PSNR in EL and lower PSNR in BL
- Support for a combination of higher bit depth in EL and lower bit depth in BL
- Support for a combination of interlaced and progressive format, i.e., BL in interlaced format while the EL in progressive format
- Support for a combination of higher colour gamut in EL and lower colour gamut in BL
- Support for a combination of external and internal coding, e.g., BL encoded by SVC while EL encoded by SHVC

#### 2.1.2.2 Architecture of Scalable High Efficiency Video Coding

By default, the SHVC encoder encodes a video into two layers, i.e., one BL and one EL, where more ELs can be obtained by manual configuration. If there are more than one EL, then the decoding of the particular $M^{th}$ EL demands the decoding of a BL and the previous $M-1$ ELs, where $M$ represent the ID of a layer. Fig. 2.13 and Fig. 2.14 show the base architecture of a two-layered SHVC encoder and decoder, respectively.

#### 2.1.2.3 Upsampling Filter

The purpose of an upsampling filter is to map the values of reconstructed samples of the reference layer to the higher-resolution sampling grid of EL [39]. This filter is considered

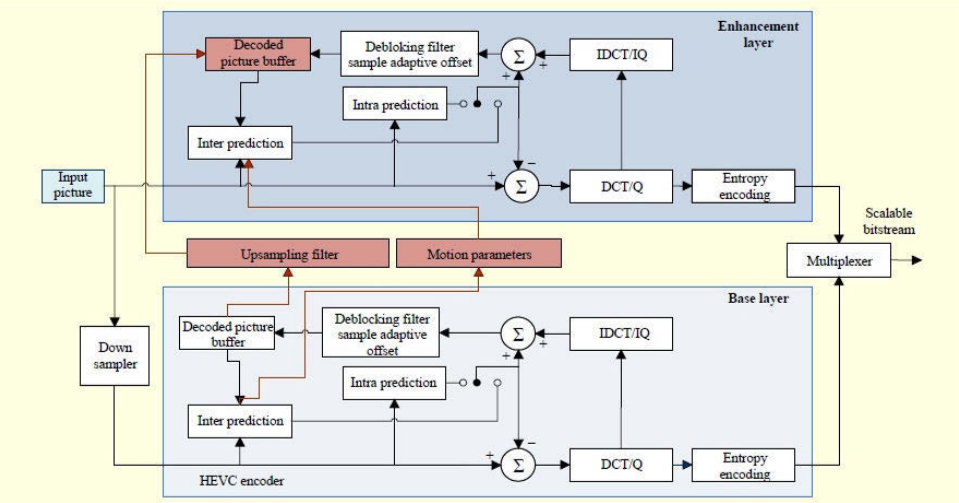



Figure 2.13: SHVC encoder [38]

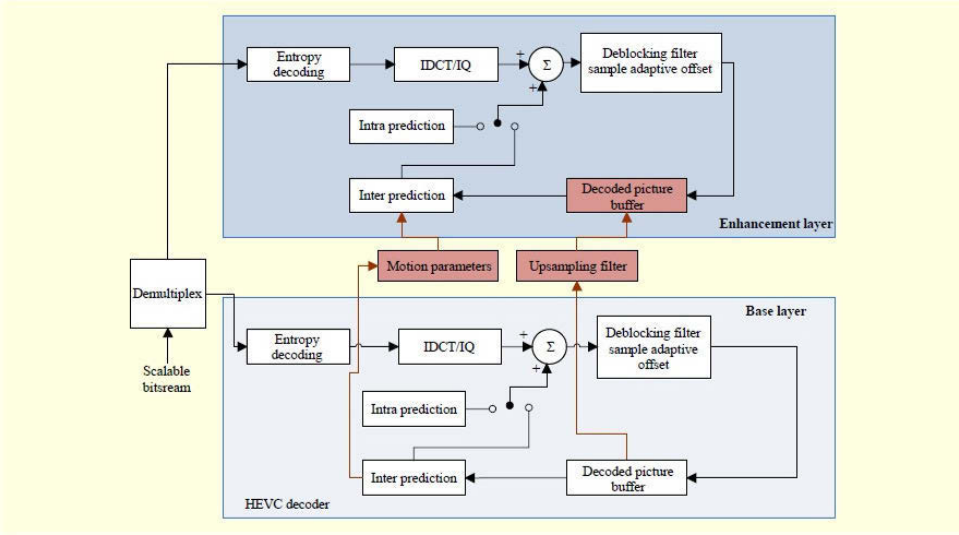



Figure 2.14: SHVC decoder [38]

as an established part of the SHVC standard. In SHVC, an upsampling filter is basically an 8-tap polyphase Finite Impulse Response (FIR) filter for luma resampling and a 4-tap polyphase FIR filter for chroma resampling [36]. In the initial versions of HEVC, the multiple taps help in fractional-position interpolation with at least $\frac{1}{4}$ accuracy precision for a luma sample and $\frac{1}{8}$ accuracy precision for a chroma sample [39]. In luma upsampling, sample location and sample phase of a corresponding layer can be determined by a scaling

factor and a scaled offset of the reference layer. This helps in estimating the filter tap coefficient values which correspond to the determined sample phase. However, the selection of tap coefficients values corresponds to the HEVC's MC interpolation process. On the other hand, chroma upsampling demands nine phases of polyphase FIR filter due to inherent phase shift between luma and chroma samples.

#### 2.1.2.4 Intra Prediction

To perform Intra prediction, SHVC offers the following options.

**2.1.2.4.1 IntraBL Mode** The IntraBL mode is similar to inter-layer Intra mode of scalable version of H.264 [36]. In HEVC, this style of mode can be utilised at the CU level. In this mode, the prediction signal of EL can be formed by either upsampling or copying the reconstructed samples from the co-located areas in the BL. If the spatial resolution used in the EL is higher than the spatial resolution used in the BL, then upsampling filters will be required for the prediction of EL signals from the BL. Due to variable motions in video frames, noise and quantisation errors may happen after applying the upsampling filters. These effects can be minimised by another mode, known as IntraBLFilt mode [36]. This mode contains an extra smoothing filter which is applied after copying or upsampling the reconstructed BL samples.

**2.1.2.4.2 Difference Signal Mode** Both Intra BL and IntraBLFilt modes depend on the reconstructed BL samples to predict the EL samples. Due to prediction, most of the high frequency signals representing the difference between a BL and an EL will be missing. Moreover, the quantisation step used in a BL is usually greater than the quantisation step used in an EL. Due to these systematic errors, the final reconstructed EL signals will contain errors. These errors can be reduced by imposing two additional steps, i.e., generation of two intermediate signals. The first intermediate signal is generated by using the data of the reconstructed BL signal while the second intermediate signal is generated by using the EL data. Another Intra prediction mode, known as DiffIntra mode, is introduced by taking a difference of these obtained signals [36]. Like IntraBL and IntraBLFilt modes,

the DiffIntra mode can also be used at CU level in HEVC.

**2.1.2.4.3 Weighted Intra Prediction Mode** In this mode, the BL reconstructed signal is combined with the prediction signal of EL [36]. The upsampled BL signal constitutes one component of the prediction signal. Upsampling is achieved through the same interpolation filters used in the IntraBL mode. The second component is achieved through Intra prediction from already reconstructed blocks of the EL samples. The final signal is obtained by first passing the BL signal through a low pass filter and the EL signal through a high pass filter and then adding them up. In this low and high pass filtering process, signals are first transformed by a DCT to generate transform coefficients. Next, the transform coefficients are weighted in such a way that the low frequency components of BL and EL signals are retained while the high frequency components are suppressed. Finally, the EL prediction signal is obtained by adding the weighted transform coefficients of the BL and the EL and then applying an inverse DCT [39].

#### 2.1.2.5 Inter Prediction

Just like the DiffIntra mode, there is a DiffInter mode supported by SHVC [36]. In this mode, reconstructed BL samples from a BL signal are upsampled to derive the first component of the EL prediction signal. MC prediction of difference pictures is used to derive the second component of an EL prediction signal. Difference pictures can be obtained by taking the differences of already reconstructed pictures from the EL reconstruction and corresponding BL pictures from the BL reconstruction. The MC difference signal is added up with the corresponding BL signal to form an EL prediction signal. For the transmission of motion parameters, regular HEVC syntax is used in MC prediction based on difference pictures. At fractional sample positions, a simple bilinear interpolation is used instead of tap-based interpolation filters.

#### 2.1.2.6 Reference Software

The HEVC test Model (HM) is used as a base to design an SHVC test Model (SHM) with recent version SHM-10.0-Dev, and is still under development [40]. For each layer, the SHM generates a separate encoder instance from base HM model. These instances communicate with an inter-layer processing unit to extract and exchange the information in the inter-layers and reference layer. The inter-layer processing unit is responsible for resampling and mapping of information between different layers. A 3D Look-Up-Table (LUT3D) is implemented in the SHM to improve the colour mapping. The upsampling and MV mapping processes are also included to improve the coding efficiency.

## 2.2 Video Error Concealment

Error Concealment (EC) techniques are used to estimate the missing information. For videos EC, the aim of EC is to recover the missing part of a video frame or an entire video frame. With HEVC, high compression is applied on video sequences. However, it still leaves enough redundancies to recover the missing data in an HEVC encoded video. Videos are usually encoded in Sliced or Unsliced mode. Sliced mode is the default mode of encoding. In this mode, a video frame is distributed into multiple blocks of either a regular or an irregular shapes. If a block of a video frame gets lost during transmission, it can be concealed by either utilising the nearby blocks within the same video frame (e.g., I-frame) or utilising the blocks at the same position in previously processed video frames (e.g., B or P-frame). If multiple consecutive blocks are missing in a video frame, nearby blocks cannot produce quality reconstruction. In this scenario, MVs of the surrounding blocks are used to estimate the MVs of the missing blocks. These MVs are estimated by performing ME on the previously processed video frames. On the other hand, Unsliced mode can be selected manually during video coding. In this mode, a data packet contains an entire video frame. If a data packet gets lost during transmission, an entire video frame will be lost. In the case of missing an I/B/P-frame, the only option is to utilise previously

processed frames.

There are three broad categories of EC techniques, i.e., Spatial Error Concealment (SEC), Temporal Error Concealment (TEC) and Hybrid Error Concealment (HEC). There categories are further explained below.

## 2.2.1 Spatial Error Concealment

Spatial Error Concealment (SEC) is also known as Intra-frame EC. The SEC techniques are pixel-based techniques, i.e., the information from the surrounding group of pixels in the same frame is used to estimate missing/damaged pixels. Such EC techniques use either texture features or edge/object-shape information. A special case occurs when most of the surrounding pixels are also missing/damaged, and cannot provide useful information for prediction. In such a case, previously processed frame(s) can be used to estimate the lost information. A summary of SEC techniques, along with their advantages and disadvantages, can be found in Table 2.1.

### 2.2.1.1 Texture-based Approaches

The textures-based techniques can be classified into statistical, structural and model-based techniques. In a statistical approach, statistical results of selected features are merged and processed to predict the features of a missing region. In a structural approach, textures are classified as either visually strong or visually weak. This categorisation of structures helps in estimating the texture of foreground and background regions. In a texture modelling approach, the texture of an image or a texture image is modelled as a linear combination of basis function(s) and a probability model. The coefficients of such models help in categorising different textures [49].

In [41], a 3-D head-and-shoulder texture model, i.e., Candid-3, was used to reconstruct the corrupted facial regions with reduced artefacts in video conferencing applications. This technique works well in the presence of lost MBs. However, it is suitable to reconstruct facial regions only. Probability Density Function (PDF) from Kernel Density Estimation

| Approaches | Techniques | Advantages | Disadvantages |
|---|---|---|---|
| Statistical-based approaches | Candid-3 model [41] | Suitability for consecutive packets loss, better reconstruction quality, support for different block sizes | Computational complexity, unsuitability for real-time processing |
| | PDF and MESE [42] | | |
| | Adaptive linear predictor [43] | | |
| Edge-based approaches | Weighted average of edges [44] | Simple mathematical operations, controlled computational complexity, ability to reconstruct complex edges, linear and non-linear textures | Dependent on significant edge details, iterative processing, incompatibility with consecutive packet loss, unsuitability for real-time processing |
| | Canny edge detector [45] | | |
| | Hough transform [46] | | |
| Embedded approaches | Histogram shifting [47] | Suitability for random errors | Degraded reconstruction results, incompatibility for consecutive packet loss |
| | Frame copy [2, 48] | | |

Table 2.1: Spatial error concealment techniques

(KDE) and minimum Mean Error Square Estimator (MESE) were used to reconstruct the lost portion of an Intra-frame with vector formalism in [42]. This approach requires computationally complex techniques to improve the reconstruction quality. Adaptive linear predictor based on Bayesian information criterion was used to reconstruct pixels of missing blocks in [43]. The predictor is auto-tunable to control the order and can support different shapes at the cost of computational complexity. The approaches presented in [41–43] produce better reconstruction quality but are not suitable for real-time processing.

#### 2.2.1.2 Edge-based Approaches

There are certain points in images where the brightness changes rapidly. Such points are grouped into curved lines, known as edges. Mathematical techniques to detect edges are called edge or contour or boundary detection techniques [50]. Directional edge analysis

was used to locate edges of missing areas, and the weighted average of the two corresponding edges was used to conceal missing areas in [44]. This approach produces a defined number of computations, and is suitable for images having significant edges with details. The Canny edge detector was used to detect relevant edges in order to conceal digital drop-out errors for the content of digital video tapes in [45]. This technique can restore complex edges and non-linear features with controlled computational complexity. However, it relies on pathological motion. Relevant edges were found by using the Hough Transform, and fine concealment was achieved through repeated interpolations in [46]. This approach can reconstruct complex textures, but is computationally complex and requires accelerated hardware. The approaches presented in [44–46] produce a better reconstruction performance with simple operations and are suitable for the random loss of data.

#### 2.2.1.3 Embedded Approaches

Histogram shifting technique was used to embed the MVs of an MB into another MB within the same video frame in [47]. If an MB is lost/damaged during video transmission, its MVs can be recovered from neighbourhood MBs at the decoder side to conceal it with a better reconstruction quality. However, this data hiding produces extra non-zero residual blocks in video frames and degrades the received video quality. Frame Copy was used to conceal missing frames for scalable video streaming over vehicular networks based on a cooperative LTE/802.11p communication in [48]. Similarly, Frame Copy was used to increase the QoE of HEVC video streaming over wireless networks in [2]. Frame Copy is the default technique, embedded in video decoders to conceal missing video frames/blocks. However, it produces visible blocky artefacts and is not suitable for videos containing fast moving objects or moving cameras .

### 2.2.2 Temporal Error Concealment

Temporal Error Concealment (TEC) is also known as Inter-Frame EC. TEC techniques are usually used to recover lost MVs, which are used to recover damaged MBs by considering

the MVs of the corresponding MBs in the reference video frames. However, these techniques can also use algorithms from other domains of image processing to refine or filter out estimated MVs. Temporal EC techniques perform better than SEC techniques when the motion is continuous in the consecutive video frames. Such EC techniques are used in both 2-D and 3-D video domains. Table 2.2 summarises current TEC techniques along with their advantages and disadvantages.

#### 2.2.2.1 2-D Video Approaches

A 2-D image is an image, captured through a camera with height and width information. Videos made up of a combination of such images are called 2-D videos [67]. In [51], object detection, the boundary matching score and the MVs from a reference video frame were used to recover lost variable sized MBs in a video frame. This approach performs better on frames having multiple objects. An iterative Dynamic Programming (DP)-based approach was used to estimate the lost MVs of corrupted MBs in [52]. This technique works well with both Intra and Inter-frames. In [53], the MVs from 20 nearby sub-blocks were used to restore the MVs of a lost MB within the same video frame. This technique can be applied to real-time applications, but is not suitable for videos having slow motion and a consecutive loss of MBs. The MVs of missing MBs were predicted by using a Kalman filter in combination with modified bilinear Motion Field Interpolation (MFI) in [54]. This approach works well with heavily corrupted videos, but is suitable for videos having linear motions only. Flexible Block Matching (FBM) scheme along with Tensor Low Rank Approximation (TLRA) was used to figure out and conceal the corrupted blocks in a video bitstream in [55]. This approach works well for large and continuous corrupted blocks of video frames. Encoding partition information in H.264 encoded video sequences combined with Motion Vector Extrapolation (MVE) was utilised to recover lost MVs in [56]. This technique produces better reconstruction quality in the presence of random packet drops only. Region-based homography was used to inpaint a group of video frames for object removal and error concealment in video sequences in [57]. A group of 20 video frames is used

| Approaches | Techniques | Advantages | Disadvantages |
|---|---|---|---|
| 2-D video approaches | Object detection using MVs from reference frames [51] | Suitability for both Intra and Inter frames, better reconstruction quality, can conceal heavily corrupted regions | Incompatibility with consecutive packet loss, suitability for linear motions only, computationally expensive, unsuitability for real-time processing |
| | Iterative DP [52] | | |
| | MVs of nearby blocks [53] | | |
| | Kalman filter and MFI [54] | | |
| | FBM and TLRA [55] | | |
| | Encoding partition and MVE [56] | | |
| | Region-based homography [57] | | |
| 3-D video approaches | Past Intra-frames [58] | Simple computations, suitability for real-time processing, better reconstruction results | Suitability for 3-D/Multi-view videos only, incompatibility with consecutive packet loss |
| | MV variances and texture histograms [59] | | |
| | Intra and Inter MVE [60] | | |
| | MVs, disparity vectors and SAD [61] | | |
| | Depth-based image rendering [62] | | |
| | True ME and depth maps [63] | | |
| | MVE and colour and depth information [64] | | |
| | Spatial Intra and Inter-view techniques [65] | | |
| | Consistency model [66] | | |

Table 2.2: Temporal error concealment techniques

to improve the performance of error concealment. The approaches presented in [51–57] produce a better reconstruction quality but are computationally expensive and unsuitable for real-time processing.

#### 2.2.2.2 3-D/Multi-view Video Approaches

Unlike a 2-D image, a 3-D image includes the perception of depth. Videos made up of such images are called 3-D videos. In past, dual cameras were used to produce such videos. However, nowadays this is usually done by using software technology, which transforms 2-D videos to 3-D videos [67]. An analysis of 3-D video compression, transmission and quality evaluation is summarised in [68]. Past Intra-frames were used as a reference in 3-D videos to recover the lost MVs in an Intra-frame in [58]. In [59], the MV variances and texture histograms from nearby MBs were used to estimate the similarity between the lost and the nearby MBs in 3-D videos. The lost frames of a 3-D video were recovered by using Inter and Intra MVE while ignoring the depth information in multimedia wireless sensor networks in [60]. Motion and disparity vectors were combined with Sum of SAD to recover missing MVs and conceal a missing/corrupted whole video frame in [61]. Similarly, depth-based image rendering was used to conceal a whole video frame in a multi-view video in [62]. True ME and depth maps were exploited to recover lost MVs of objects and conceal a lost frame in 3-D videos in [63]. MVE was utilised along with colour and depth information of 3D videos to conceal a missing video frame in [64]. The techniques presented in [58–64] produce less computational complexity. Spatial Intra and inter-view techniques were used to exploit error-free neighbouring data of depth and colour frames to conceal corrupted depth maps in multi-view videos in [65]. A consistency model for EC is proposed for multi-view videos in [66]. This model maintains inter-view and temporal consistencies between video frames. The approaches presented in [58–66] produce a better reconstruction quality, but are suitable for 3-D/multi-view video transmission only.

### 2.2.3 Hybrid Error Concealment

The EC techniques proposed in spatial and temporal domains have their own advantages and disadvantages. Sometimes, it is better to combine established EC techniques from spatial and temporal domains or to combine techniques from other domains of signal and image processing to perform EC. Table 2.3 summarises the Hybrid EC (HEC) techniques along with their advantages and disadvantages.

| Approaches | Techniques | Advantages | Disadvantages |
|---|---|---|---|
| Spatio-temporal approaches | Temporal extrapolation refinement [69] | Better reconstruction results, less computational overhead, suitability for real-time processing | Incompatibility with consecutive packet loss |
| | Encoder controlled transition scheme [70] | | |
| | MDC [71] | | |
| Random approaches | EII and SI [72] | Better reconstruction results, compatibility with consecutive packet loss | High bandwidth demand, high computational complexity |
| | PDIP and SO [73] | | |
| | Multiple streams [74] | | |

Table 2.3: Hybrid error concealment techniques

#### 2.2.3.1 Spatio-Temporal Approaches

A comparison between switching EC algorithms and optimal switching EC algorithms was provided in [75] to demonstrate the trade-off between optimal performance and computational complexity. Spatio-temporal EC were combined to estimate the lost information in H.264 encoded videos in [69]. A switch-able HEC technique was proposed to utilise MVs, SADs and motion compensated histogram differences to conceal the missing data

in encoded video sequences in [70]. Spatio-temporal Multiple Description Coding (MDC) based adaptive EC technique was proposed to conceal the missing data in H.264 encoded video sequences in [71]. The techniques proposed in [69–71] perform well and produce a better reconstruction quality with less computational overhead, but cannot support the consecutive loss of data.

#### 2.2.3.2 Random Approaches

Examplar-based Image Inpainting (EII) was combined with Spatial Interpolation (SI) to reconstruct pixels and patches lost during video communication in [72]. Primal-Dual Interior Point (PDIP) method was combined with Sparse Optimisation (SO) to efficiently utilise spatial and temporal correlations between pixels in a group of video frames for EC purpose in [73]. High and low-resolution versions of the same video were streamed together to perform EC for Regions of Interest (RoI) in [74]. The techniques presented in [72–74] support consecutive loss of data with a high bandwidth demand and computational complexity.

## 2.3 Video Streaming over Wireless Multimedia Sensor Networks

Real-time Wireless Multimedia Sensor Networks (WMSNs) are gaining the attention of researchers, as they can support many application layer services to solve real-life problems, such as health monitoring, weather forecasting, transport management, industrial automation and surveillance [76]. A traditional WMSN consists of a variety of sensor nodes, including both simple and multimedia sensor nodes along with a Base Station (BS). In this section, we summarise the existing studies covering fully or partially QoS support for WMSNs. The QoS studies cover both simple and cross-layer approaches for video streaming over WMSNs.

### 2.3.1 Layered Vs. Cross-Layered Architecture

In a traditional TCP/IP-based network, the end-to-end communication is possible through the collaboration of all five layers, i.e., physical, data-link, network, transport and application layer. The communication through this architecture can either be bottom-up or top-down [77]. In either form of communication, the exchange of services and data takes place between two adjacent layers only, thus leading to a black-box or information hiding architecture. Such strict boundaries between layers make the architecture easy to deploy and prevents sharing of unnecessary information between multiple layers. However, there are some drawbacks too. For example, channels in wireless networks face noisy situations frequently. These noises cause a connection termination and the re-establishment of a connection can be a time consuming process. Moreover, wireless channels are vulnerable to many network impairments and security threats [78]. Cross-layered designs offer flexibilities to cover-up the weaknesses of simple layered design. Unlike the TCP/IP architecture, where the layers and protocols target a specific set of problems, a cross-layered design allows to reconfigure the layer functionalities to fight against multiple network problems. In cross-layered architectures, two non-adjacent layers can directly communicate with each other. It also allows layers to share internal information, status and parameters without disturbing the traditional five-layered architecture. The cross-layered design makes the hidden information of a layer visible to other layers and can help in achieving multiple network related goals, such as security, QoS and mobility, as shown in Fig. 2.15 [79–81].

### 2.3.2 Quality of Service

WMSN is by default a layered architecture, just like the standard TCP/IP architecture, having the same set of layers [76]. A WMSN is comprised of heterogeneous multimedia sensor nodes, capable of dealing with different forms of multimedia data, e.g., audio, videos and images. To maintain a guaranteed level of QoS in such networks, the heterogeneous sensor nodes need to interact with each other by sharing different forms of data. A

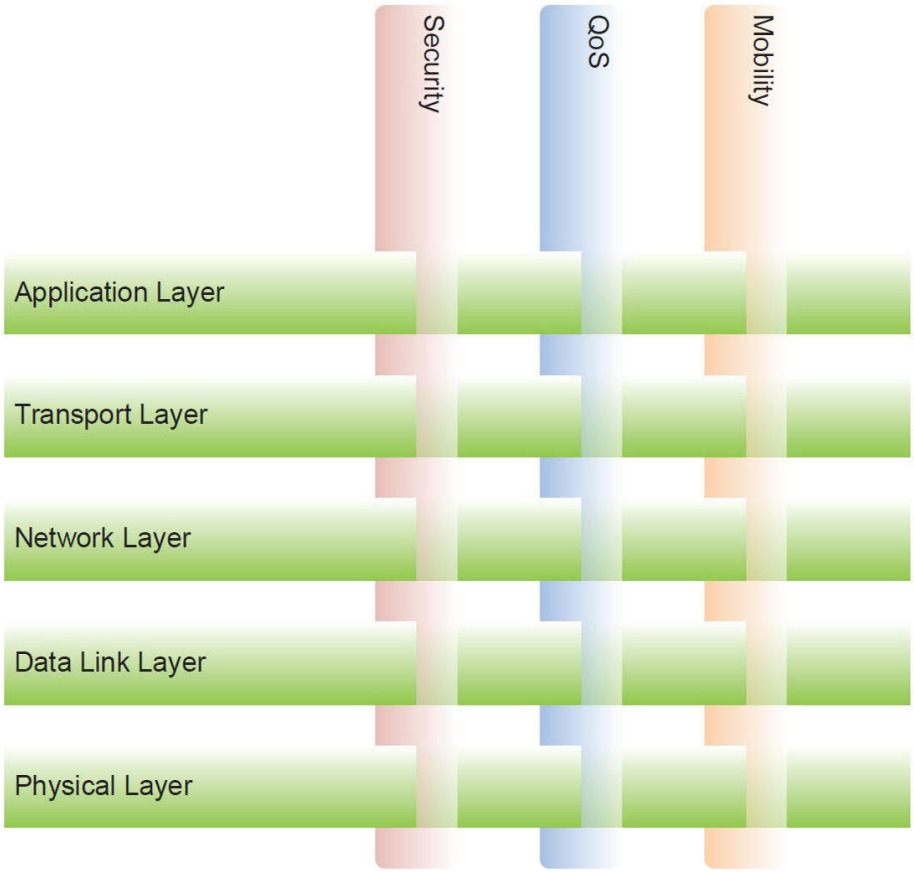



Figure 2.15: Goals of cross-layer design [81]

surveillance system based on Wireless Sensor Networks (WSNs) is an example of layered architecture, in which different types of sensors, e.g., Passive Infra Red (PIR), audio and video sensors communicate with each other [82]. To achieve a guaranteed QoS in a layered architecture, different research efforts focus on different layers [83–85]. In contrast, a cross-layer approach utilises the information gathered from all layers to jointly improve the QoS performance of a system [80, 86]. In such an architecture, the application layer takes responsibility of directing the lower layers by defining QoS criterion. However, in both layered and cross-layered platforms, MAC and routing layers play important roles in maintaining QoS for video transmission.

#### 2.3.2.1 MAC Layer

IEEE 802.15.4 standard is designed to support low power and short range sensor devices with a data rate of 250kbps [87]. In order to increase the data rate, other technologies, like ZigBee and Ultra Wide Band (UWB), are proposed with a data rate around 250kbps to 480Mbps [88, 89]. Although, these technologies are suitable for simple WSNs, they cannot be applied to WMSNs due to their short range communication. Among these technologies, the IEEE 802.15.4 standard has become popular and been applied in short range wireless ad-hoc networks. It has been applied as a physical layer technology with a support for video devices and designed for security surveillance and health monitoring applications. To provide a long range communication with higher data rate, IEEE 802.11ah has recently been proposed, thus making it an ideal platform to provide an acceptable QoS for WMSNs [90, 91]. A cross-layer platform based on IEEE 802.15.4 was proposed for WMSNs to improve throughput and reduce energy consumption for a better QoS in [92]. Although this platform outperforms the traditional IEEE 802.15.4 architecture by reducing energy consumption, it is incompatible for a long range multimedia communication. Another cross-layer framework, SchedEx-GA was proposed to identify network configuration in order to maintain application specific QoS services in [93]. This approach works well under a centralised controlling mechanism to provide high end-to-end reliability. However, it demands the involvement of additional sink sources, so it increases cost of network management. A MAC protocol was proposed for a cross-layer sensor network platform to improve the battery life of sensor nodes by reducing the energy consumption in [94]. This scheme works well for unidirectional communication and is unsuitable for bidirectional video communication. Network lifetime for energy constrained WSN was improved by a joint physical, MAC and network layer design in [95]. The scheme produces a better performance under a specific data rate and packet scheduling approaches. Performance of physical layer in WSNs was improved by balancing the links between sensor nodes and the BS in [96]. Although this approach produces minimal Mean Square Error (MSE), it cannot

support higher data rates. An analysis was performed for interaction between MAC layer and wireless channels in [97]. This analysis summarises the benefits of channel fading on the overall network performance. Challenges of IEEE 802.11ah along with suggested improvements were summarised in [98]. This report highlights major issues related to link reliability, throughput and interference problems in WLANs. Physical layer aided MAC framework was proposed to enhance the uplink throughput for data generated by sensor nodes in [99]. This framework enhances the throughput by optimising physical layer components without addressing group scheduling. A MAC layer scheme was proposed to allow parallel transmission of sensor data on a busy channel in [100]. This scheme reduces packets latency in WSNs and is suitable for short range data communication, thus making it unsuitable for WMSNs.

#### 2.3.2.2 Routing Layer

The QoS can be maintained by improving network performance through proper optimisation of routing protocols. A comparison of geographic routing protocols for mobile, ad-hoc and sensor networks was presented in [101–103] which shows their efficiency for shortest path selection between sources and destinations. However, in WMSNs, these protocols need to be optimised to better utilise available bandwidth for multimedia contents delivery. Pair-wise directional geographical routing was proposed to solve an energy bottleneck issue while performing multi-path routing in WMSNs in [104]. This strategy improves network lifetime with a compromise on delay time. A straight line routing was proposed in WSNs to conserve sensor nodes energy in [105]. Although this style of routing produces better performance, it cannot be applied for a long range video transmission over WMSNs. To cover-up the delay constraint and to provide end-to-end reliability in WSNs, geographic opportunistic routing was proposed in [106]. This protocol is energy efficient and reduces network latency for scalar data communication only. Therefore, it is unsuitable for multimedia communications in WMSNs. A hybrid multicast routing protocol for large scale sensor networks was proposed in [107]. Although, it reduces energy consumption and

minimises end-to-end delays by detecting network holes, it is suitable for simple and tiny sensor nodes. A seamless streaming data delivery protocol was proposed for cluster-based multi-hop WSNs with mobile nodes in [108]. Cross-cluster handover and path redirection strategies are applied to minimise the delivery latency and the energy consumption. However, the performance goes down as the scale of the sensor network increases with addition of powerful multimedia sensor nodes. An energy efficient and spectrum aware routing protocol for multimedia contents delivery was proposed for cognitive radio sensor networks in [109]. The QoS is improved by exploiting clustering scheme with a compromise on network latency. An energy-balanced routing scheme based on forward aware factoring was proposed for WSNs in [110]. The QoS is improved with the selection of link weight and node's energy which prolongs the network life time. However, this scheme is only applicable for simple WSNs.

## 2.4 Multimedia Cloud Computing

Cloud platforms offer on-demand and convenient access to a pool of computing resources to a diverse range of applications through the Internet [111]. The data of an application is owned by a third organisation, i.e., Cloud Service Provider (CSP), and all the processing and the storage take place at cloud platform. These processing and storage resources can be at different geographic locations across the globe. The owners of applications now just need to focus on their business strategies and renting of cloud resources. Issues related to resources management, security, backups, upgrading of technology, etc., are dealt with CSP. Multiple cloud platforms are available in the market, such as Amazon, Microsoft Azure and Google App Engine [112]. A cloud platform offers different services, such as Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS) for different purposes [113]. In the past years, multimedia contents were stored and distributed through Compact Disc (CD) and Digital Video Disc (DVD). Due to increasing resolution and size of multimedia data, especially videos, users cannot purchase bundles

of CDs and DVDs to store their data. For real-time multimedia services, VSPs are shifting their infrastructures to public clouds [114]. These public clouds offer powerful and reliable computing and storage platforms to process large volumes of multimedia data. Alongside computing and storage facilities, these cloud platforms also ensure to provide a support for application-specific QoS and QoE services [115, 116]. The video data is continuous in nature and needs to be dealt with carefully. Such a massive and continuous form of data requires a scalable storage architecture along with analytical tools, which should allow users to store multimedia data along with the context information [117]. The scalable and multimedia-aware cloud architecture should also provide categorisation and indexing facilities for its users to properly manage their multimedia data. In an any storage system, data integrity and redundancy removal are critical issues [118]. It is common that a cloud server may get the same type of information from multiple devices. If data is properly handled and filtered, storage resources can be utilised efficiently. In cloud computing, resource allocation and scheduling are two sensitive issues [119]. Resource allocation deals with appropriate allocation of cloud resources to a group of applications. Cloud computing allows its users to request and dynamically release the resources [120]. For better performance, type, amount and placement of resources need to be decided smartly. Scheduling, on the other hand, maintains time slots of the allocated resources. This time management is important as the resources are shared and need to be released on time in order to be allocated to other user applications. In traditional multimedia systems, the data is usually encoded once and is decoded multiple times [121]. Based on this fact, an encoder is always complex as compared to a decoder. If combined with cloud computing, the computational complexity of an encoder can be shifted to a decoder on a cloud platform [114]. Furthermore, low-powered devices can perform basic encoding on multimedia data and can send it to a cloud platform. The parallel and distributed nature of cloud computing makes it easier to carry out such complex computing tasks in real-time.

## 2.5 Summary

This thesis aims to address smooth video streaming over a multi-hop WMSN and to increase received video quality at an end-user side. There are different styles of video coding to compress high resolution videos before transmission. The resource-constrained nature of mobile devices requires light and energy-efficient EC techniques to maintain the quality of received videos. The multi-hop nature of wireless networks makes it challenging to design routing protocols to support smooth and real-time video streaming. This chapter has presented a background of video coding, EC and video streaming over layered and cross-layered architectures to support QoS. In the video coding section, first we discuss the architecture of high efficiency vide coding, highlight its different components and its reference software. The scalable architecture of high efficiency video coding is also discussed to highlight the difference between simple and scalable video coding. Various EC techniques from spatial, temporal and hybrid domains are discussed to highlight their pros and cons and applications. Video streaming over WMSNs is discussed in terms of layered and cross-layered architectures. Issues like, the importance of QoS in video streaming and how to maintain/improve it using MAC and routing layers approaches are also discussed. In the end, a brief summary on the introduction and background of multimedia cloud computing and how the multimedia cloud computing resources can be utilised in video streaming, is presented.

Chapter 3

# Error Concealment for HEVC Encoded Videos

## 3.1 Introduction

There is a tremendous growth in the usage of streaming videos nowadays. Current video processing standards, such as H.264 and H.265, are dealing with immense amount of video data, which ultimately demand either parallel or distributed processing platforms. In the case of cloud computing, many VSPs can rent out the distribution architectures from CSPs to deliver video streams to a large number of mobile end users [122, 123]. The most famous cloud-based media services include Youtube, Dailymotion, Microsoft Azure and Putlocker. A general framework of cloud-based media streaming is shown in Fig. 3.1.

In the case of cloud server-based media streaming, unreliable transmission channels are the biggest issue. As H.265 encoded videos are highly compressed, packet loss or bit errors during video streaming may lead to huge quality degradations. The main idea behind an EC technique is to estimate the video packet lost/damaged during video streaming. The lost/damaged video packet may contain some parts of a video frame or one complete video frame, depends on the encoding parameters. The objective of our proposed frame

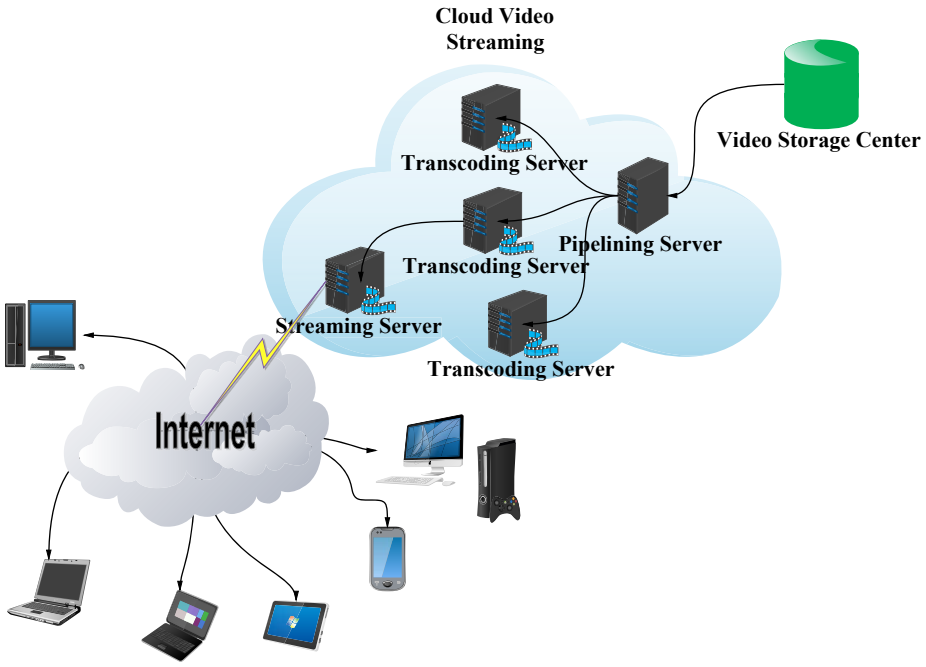



Figure 3.1: Generic architecture of cloud-based media streaming

interpolation algorithm is to recover the missing video frame and to improve the received video quality at the receiver side. The proposed frame interpolation algorithm considers H.265/HEVC-based Intra-encoded videos and recovers an entire Intra-frame lost during video streaming. The Unsliced mode of H.265 is targeted for the proposed approach. We specifically target the Unsliced mode approach to experience the effect of lost video frames on the received video quality and to compare the quality of concealed video frame with the original one. In Sliced mode, a video frame is distributed into multiple parts of equal/unequal length and each part is encapsulated into one data packet. If a data packet gets lost during transmission, the data packets containing the other parts of the same video frame can be utilised to conceal the missing part of the video frame and the concealment effect might not be notable. On the contrary, in the Unsliced mode, the only possibility to conceal the missing video frame is to utilise the neighbour frames which is more challenging and time consuming, as the received whole video frames need to be processed to conceal the missing one. The main emphasis in the proposed approach is to recover MVs

of a lost video frame in real-time. The search to find the optimum MV is performed in parallel in nearby four sub-blocks in the reference video frame. There are many state-of-the-art searching techniques, such as Two Dimensional Logarithmic Search (TDLS), Simple and Efficient Search (SES), Diamond Search (DS) and Adaptive Rood Pattern Search (ARPS), among which ARPS is the latest one [124]. These techniques are very popular in term of computational time. However, their searching accuracies do not beat the searching accuracy of Full Search (FS) [125]. To boost-up the search process for lost MVs, both a bigger block size and searching in parallel are considered. The frame interpolation algorithm discussed in Section 3.2 is based on our work published in [126]. In the following sections, we explain the proposed frame interpolation algorithm followed by discussion on the experimental results.

## 3.2 Frame Interpolation Algorithm

Our proposed EC technique is based on multiple threads and is motivated by the Block Matching Algorithm (BMA) and the Frame Copy (FC) technique. The proposed EC technique, named as Frame Interpolation (FI) is illustrated in Fig. 3.2. For each lost Intra-frame, MVs are derived first. Lost MVs are estimated through ME scheme. The ME is the heaviest process in video encoding and consumes approximately 40% to 80% of computational time [127]. In our approach, the main emphasis is to reduce the computational complexity and time. These two factors need to be reduced, as we cannot expect a high computational power in the mobile devices. This reduction in the complexity of computations ultimately reduces the computational time, which helps in supporting the real-time processing. The ME process is performed by using two video frames at a time, i.e., the previous video frame and the next video frame. The blocks in the previous video frame and the next video frame represent the tail and the head of estimated MVs, respectively. The starting point of any MV is known as a tail while the ending point is known as a head as shown in Fig. 3.2. The BMA is used as a base algorithm for further enhancement in

the ME process. Estimated MVs then help to interpolate lost video frames. Interpolated frames are passed through an adaptive filter to remove false estimation noises, if any. The details of the FI are discussed in the following subsections.

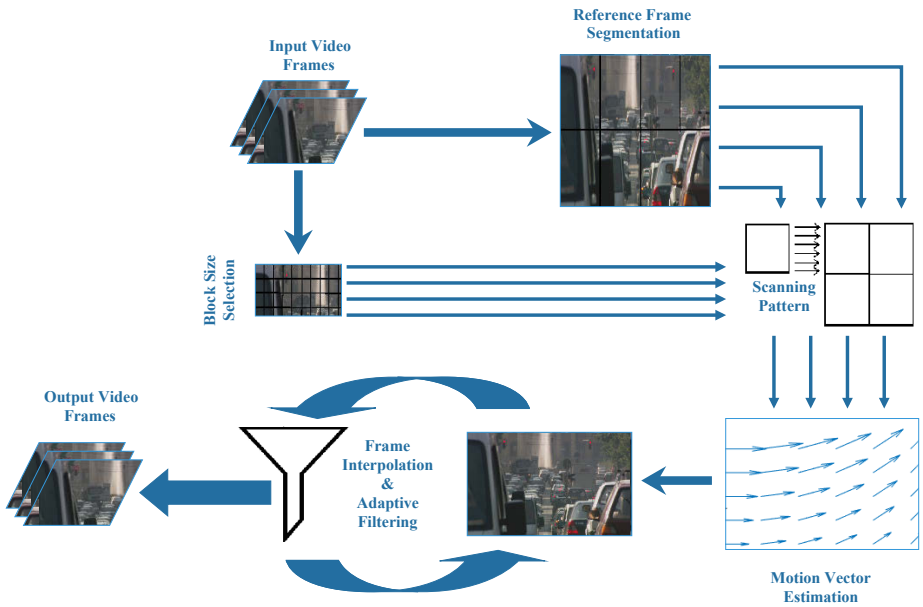



Figure 3.2: Block diagram of the proposed algorithm

### 3.2.1 Block Size Selection

Before starting the procedure of the EC, it is important to divide the current video frame into blocks but there is no specific criterion for that. This division totally depends upon the nature of information and applications. Sizes and shapes of segmented blocks need to be considered carefully when the current video frame is divided into blocks.

For the division of a video frame into blocks, there are two options to determine the size of blocks. We can either consider the whole video frame as one block or divide the video frame into blocks of equal/unequal size. The first option demands large sized buffers and may slow down the performance of the system. This option also requests to process a whole video frame, although there may be specific regions that can easily be skipped based upon the repetitive information. The above drawbacks can be minimised using the

second option, which is to segment a video frame into number of blocks. Depending upon the nature of contents in a video frame, either fixed or variable sized segmentation can be opted. In the proposed FI, we select the second option, and divide a video frame into number of blocks of fixed size. During experiments, $16 \times 16$ is proved to be an optimal block size on average. As our algorithm is basically designed for HD and UHD videos, a larger block size creates minimal computational overhead. On the contrary, smaller and variable sized divisions produce fine results but at the cost of processing overhead and higher computational complexity.

The shape of a block can be square, rectangular, triangular and irregular. Out of all these possibilities, the square shape is widely adopted. One main reason behind the popularity of square shape is the even resolution of test videos. Test videos are always available in even resolution as shown in Table 3.2 and Table 3.3. Traditional BMAs, such as FS, TSS and FSS, use square shape while others use diamond, cross or hexagonal shapes [128]. Computational time depends upon the number of pixels in a selected block shape. A non-squared shape, e.g., hexagonal and/or diamond, saves coordinate information of pixels of a video frame in compared to a square shape, and hence requires further processing, especially in an HD and a UHD video. In the case of a square shape, only starting and ending points are required while in hexagonal and/or diamond shapes, checks and breaks are required in rows and columns to create hexagonal and/or diamond shapes. In our proposed algorithm, to reduce the amount of information required during the processing of a block, a square block shape is considered. Experiments show that using a square shape produces better results and requires less computational time as compared to other block shapes.

In our proposed algorithm, each Input Video ($IV$) is read in frame by frame during the decoding process. If $N^{th}$ video frame is missing, then $N - 1^{th}$ and $N + 1^{th}$ video frames are selected for the EC process. Otherwise, the video frame is written to the Output Video ($OV$) file. To quickly process HD video frames, a block size of $16 \times 16$ is used to perform EC at the decoder side. The main reason to choose this block size is to make the proposed FI scheme compatible with prior standards where the maximum supported block size is

$16 \times 16$ [129]. The blocks of the current video frame are denoted as the Input Blocks ($IBs$) and the blocks of the reference video frame are denoted as the Reference Blocks ($RBs$).

### 3.2.2 Reference Frame Segmentation

The second step in any EC scheme is to divide each reference video frame into areas, known as Search Windows ($SWs$). Like the division of the current video frame, there is no specific criterion for the division of a reference video frame and it totally depends upon the nature of information and applications. Two factors need to be taken into account when dividing a reference video frame into $SWs$, and they are the size and the shape of the $SW$.

In our proposed approach, we divide each reference video frame into multiple $SWs$ of a fixed size. When determining the size of a $SW$, we assume that the motion of pixels is not very fast in consecutive video frames. Therefore, the size of a $SW$ is selected to be four times of the size of the $IB$, i.e., $64 \times 64$. It has been proven that the size of the $SW$ can further be reduced for better results and less computations if needed [130].

In traditional BMAs, such as FS, TSS and FSS, patterns of a $SW$ are of square shape. However, it is not necessary to have a square shape. The shape can be either irregular or adaptive [131, 132]. However, irregular and/or adaptive shapes usually require extra processing time due to resolution variation from one test video to another test video. Therefore, a square shape is adopted in our proposed algorithm.

### 3.2.3 Scanning Pattern

After defining the size and the shape for blocks and $SWs$, the next step is to find the best matching block in a reference video frame. In BMA, the most famous searching technique is FS. This technique is linear and tries to find the best match exhaustively. The main advantage of FS is its high accuracy while the drawbacks are its computational complexity and unsuitability for real-time applications.

In our proposed technique, the FS is used with modifications. Unlike the traditional FS, our approach does not search every pixel in a $SW$. In our approach, we assume that in consecutive video frames, pixels cannot move very fast and the movement of pixels is always very slow. It is also possible that more than one match are found during the matching process at different locations. To overcome this problem, we set a threshold and terminate the searching process as soon as the first near and best match is found.

To further speed up the searching process, multiple threads-based parallel processing is considered in our proposed method. Each $SW$ is divided into $N$ partitions, known as $RBs$. The size of each $RB$ is equal to the size of an $IB$. A random value is set as an initial threshold. The block matching process is performed in parallel in each partition. The matching is performed pixel by pixel just like the FS. If the matching value is lower than the initial threshold, the search moves to the next pixel in sequence until a matching value greater than the initial threshold is found to terminate the searching process. As soon as the searching process is terminated, the threshold value is replaced by the minimum value, found in the set of the previously searched minimum values.

### 3.2.4 Motion Vector Estimation

The MV is considered as a key element in the ME process. It is used to represent a similarity match between a block in the current and the reference video frames [1]. In other words, it defines the distance covered by a block between the current and the reference video frames. To find the best match, there are many mathematical techniques, such as Sum of Absolute Difference (SAD), Sum of Squared Differences (SSD), Sum of Hamming distances (SHD), Minimum Absolute Differences (MAD), Locally scaled Sum of Absolute Differences (LSAD), Locally scaled Sum of Squared Differences (LSSD), Zero-mean Sum of Absolute Differences (ZSAD), Zero-mean Sum of Squared Differences (ZSSD), Normalised Cross Correlation (NCC) and Zero-mean Normalized Cross Correlation (ZNCC). These techniques involve operations, like addition, subtraction, multiplication and division. The time complexity is $N$ for addition and subtraction and is $N^2$ for multiplication

and division [133].

In our proposed approach, we try to avoid multiplications and divisions to reduce the computational complexity and the time. Among the above mentioned matching techniques, we use MAD in our proposed approach. During the searching process, the MAD is calculated for each pixel. If the difference is higher than the threshold in a row, the searching process continues and moves to the next row in the $SW$. This process is repeated until all the rows in an $SW$ are searched. After that, the minimum value is searched from the set of calculated values. Next, $X$-$Y$ coordinates of pixels, in the current and the reference video frames, respectively, are found out. These $X$-$Y$ coordinates determine the head and the tail locations of the corresponding MV. After that, another search is performed to find out an MV with minimum value from the set of calculated MVs. The MV with minimum value is selected as the representing MV of the current block. The above process is known a Forward Motion Estimation ($FME$), i.e., to find the Forward Motion Vector ($FMV$) for each $IB$. The same process as that shown in subsections 3.2.1 to 3.2.4 is performed for a Backward Motion Estimation ($BME$) to find out the Backward Motion Vector ($BMV$) for each $IB$ by considering $N + 1^{th}$ and $N - 1^{th}$ video frames as the current and the reference video frames, respectively.

### 3.2.5 Frame Interpolation

The frame or motion interpolation is a process in which intermediate animated video frames are generated between existing ones to provide a clear, fluid or smooth motion. This feature is very common in many latest HD displays and media players, manufactured by different companies, such as LG, Mitsubishi, Panasonic, Phillips, Samsung, Sony, Toshiba, Sharp and many others [134–141]. Although these techniques lead to clear visual quality, they request complex hardware. Furthermore, from marketing point of view, the manufacturers never reveal the technical processes of their techniques.

To perform a simple frame interpolation between a pair of video frames $F_{N-1}$ and

$F_{N+1}$, at time $t_{N-1}$ and $t_{N+1}$, respectively, the video frame $F_i$ can be inserted as

$$t_i = \frac{t - t_{N-1}}{t_{N+1} - t_{N-1}}, \tag{3.1}$$

$$F_{i,x,y} = (1 - t_i)F_{N-1,x,y} + t_i F_{N+1,x,y}. \tag{3.2}$$

Let $V_{x,y}$ be the MV between blocks in the current video frame and the reference video frame. $V_{x,y}$ defines the MV of pixels between the current and the reference video frames, so we let the tail coordinates be $[x_b, y_b]$ and the head coordinates be $[x_f, y_f]$. The values of $[x_b, y_b]$ and $[x_f, y_f]$ can be computed by

$$\begin{aligned} [x_b, y_b] &= [x, y] - t_i V_{x,y}, \\ [x_f, y_f] &= [x, y] + (1 - t_i) V_{x,y}, \end{aligned} \tag{3.3}$$

where the point $[x, y]$ represents a point in the interpolated frame [142]. This approach is a pixel-based approach, explores entire video frame and consumes lot of processing time.

In our proposed algorithm, we use the above technique but at a block-level. In our proposed approach, MVs are estimated at a block-level rather than at a pixel-level, thus, our approach saves the computational time and does not demand processing of all pixels in a video frame. After estimating MVs for all blocks of a missing video frame, missing blocks are interpolated by modifying Eq. 3.2

$$B_{i,x,y} = (1 - t_i)B_{N-1,x_b,y_b} + t_i B_{N+1,x_f,y_f}, \tag{3.4}$$

where $B_i$ is the block of a missing video frame, $B_{N-1}$ is the block in the current video frame and $B_{N+1}$ is the block in the reference video frame. The final Concealed Frame

($CF$) can be estimated by

$$CF = \frac{F_{i,x,y}}{2}. \tag{3.5}$$

The main reasons behind adopting the simple frame interpolation are its mathematical simplicity, less computational complexity, and less processing time. However, it is observed in experiments that in some interpolated frames, there are thin lines at the edges of reconstructed blocks because of the placement of pixels to false locations.

### 3.2.6 Adaptive Filtering

Sometimes it may happen that a $CF$ contains blocky artefacts or thin hair-like lines. These types of distortions are very common in block-based approaches. While smoothing down a video frame, it is very important to maintain the significant details without introducing blurring effects. In homogeneous regions, intensities of distorted pixels can easily be replaced by the average intensities of neighbourhood pixels. Sometimes, the distortions occur at edges or boundaries of objects. In this situation, averaging operation produces blurring and blocky artefacts. Therefore, following two solutions are proposed for such cases.

- In the first solution, the adaptive neighbourhood can be formulised as:

$$f(x) = \frac{1}{M} \sum_{y \in N(x)} f(y), \tag{3.6}$$

  where $f(y)$ represents an input video frame, $f(x)$ is the smoothed video frame, $N(x)$ is the adaptive neighbourhood and $M$ is the size of the adaptive neighbourhood. In Eq. 3.6, $N(x)$ consists of pixels from a set $\Omega$, where $\Omega$ contains pixels which are homogeneous and have intensities similar to $x$. Therefore,

$$N(x) = \{y : R(x,y) \leq T, x, y \in \Omega\}, \tag{3.7}$$

where $R$ represents the relationship between $x$ and $y$, $T$ is a real positive threshold and $\Omega \subset R^2$ [143].

First main problem in this approach is to find the required neighbourhood intelligently because if a false neighbourhood is selected, it automatically leads to false results. Another significant problem is the computational complexity, as this approach involves division operations frequently.

- In the second solution, the corrupted pixels need to be detected first before applying the adaptive filter. To declare a pixel as a corrupted one, a difference of $\Delta_1$ and $\Delta_{(1)}$ is used, where $\Delta_1$ is the cumulated weighted distance allocated to the central pixel of a filtering window and $\Delta_{(1)}$ is the output of the weighted vector median filter. This difference shows the strength of impulsive noise. Let this difference be denoted by $\delta$. The next step is to define a threshold value $T_h$ for comparison purpose. The threshold value needs to be set carefully. If it is very large, it will pass many corrupted pixels. In the case of very low value, uncorrupted pixels will also be declared as corrupted ones. The Adaptive Filter ($AF$) can be defined as:

$$AFO = \begin{cases} CF_{AMF}, & \textit{if } \Delta_1 - \Delta_{(1)} > T_h \\ CF, & \textit{otherwise} \end{cases} \tag{3.8}$$

where $AFO$ is the Adaptive Filter Output, $CF$ is the uncorrupted pixels of the concealed frame and $CF_{AMF}$ is the output of Arithmetic Mean filter ($AMF$). $CF_{AMF}$ is computed from those pixels, declared as corrupted in the $CF$ [144]. Although this approach is very simple and produces less computational complexity, it may produce false results.

In our proposed algorithm, we choose and modify the adaptive filter approach. The main reason for choosing that approach is its light computational complexity. In thin hair-like lines, it is very obvious that there will be corrupted pixels in more than one row. Based upon that assumption, the filtering process may become fast and may produce less com-

putational time if the filtering process is applied in parallel. Another check is to detect the thin hair-like lines in homogeneous regions. If the hair-like lines are in homogeneous regions, then intensity averaging is applied to save the computational time. In our proposed technique, the estimated video frame is passed through the modified adaptive filter to remove line errors, if there are any. Another reason to use this modified filter is the presence of noise, which may or may not be present in the concealed video frame. In this case, the adaptive filter approach proves to be very quick. The entire procedure of the frame interpolation algorithm is described in pseudo code in Algorithm 1.

## 3.3 Experimental Setup and Results

To evaluate the performance of the proposed technique, experiments are performed using HEVC test model HM 16.2 [32] and Matlab $R2015a$. The experiments are carried out using various test video sequences, including both HD and non-HD. Fifteen popular test video sequences are used for simulation purposes. The details of experimental model and platform can be found in Table 3.1.

| | |
|---|---|
| Hardware | CPU: Intel Core$^{TM}$ i5-3470<br>CPU @ 3.20 GHz<br>RAM: 8 GB |
| Software | HEVC test model HM 16.2<br>Matlab R2015a |
| Video Format | 4:2:0 |
| QP | 10 to 37 |
| PLR | 1%, 3%, 5% |
| Methods | FC, BMA, Proposed technique |

Table 3.1: Simulation environment

The details of test video sequences, such as resolution, total number of frames, frame rate, QP and encoding bitrates are illustrated in Table 3.2 and Table 3.3. All videos are encoded in Unsliced and Intra only modes, and the first 150 video frames are used from each

**Algorithm 1** Frame Interpolation

**procedure :**CONCEAL CORRUPTED $IV$
  **for** all frames in $IV$ **do**
    Read frame $N-1$
    Read frame $N+1$
    **Call algorithm 2 to compute FME**
    **Call algorithm 3 to compute BME**
    Duplicate $N-1$ as Forward Duplicate ($FD$)
    **while** true **do**
      Calculate Forward Length ($FL$) of $FMV$
      $FD(\frac{FL}{2}) \longleftarrow N-1(IB)$
    **end while**
    Duplicate $N+1$ as Backward Duplicate ($BD$)
    **while** true **do**
      Calculate Backward Length ($BL$) of $BMV$
      $BD(\frac{BL}{2}) \longleftarrow N+1(IB)$
    **end while**
    $CF = \frac{BD+FD}{2}$
    Apply $AF$ on $CF$
  **end for**
**return** $AFO$
**end procedure**

| Sequence | Resolution | Total Number of Frames | Frame Rate | QP | Bitrate (Kbps) |
|---|---|---|---|---|---|
| Blue_sky | $1920 \times 1080$ | 250 | 25 | 27, 32, 37 | 23.982, 14.977, 9.261 |
| BQTerrace | $1920 \times 1080$ | 600 | 60 | 27, 32, 37 | 81.257, 43.334, 23.746 |
| Cactus | $1920 \times 1080$ | 500 | 50 | 27, 32, 37 | 49.908, 27.077, 14.652 |
| Kimono | $1920 \times 1080$ | 240 | 24 | 27, 32, 37 | 13.262, 7.864, 4.628 |
| Rush_hour | $1920 \times 1080$ | 500 | 25 | 27, 32, 37 | 6.219, 3.745, 2.291 |
| Tractor | $1920 \times 1080$ | 761 | 25 | 27, 32, 37 | 22.311, 12.958, 7.528 |

Table 3.2: HD video sequences

test video sequence. The test video sequences are encoded using different QPs ranging from 10 to 37 as shown in Table 3.2 and Table 3.3. There are two main reasons to use different QPs. The first reason is to generate different bitrates for testing the efficiency of our proposed approach. The second reason is to follow the same experimental setup, as that

**Algorithm 2** Forward Motion Estimation

**procedure :**COMPUTE $FME$
  Divide $N-1$ into $IBs$
  Divide $N+1$ into $SWs$
  Divide each $SW$ into $RBs$
  **while** true **do**
    **while** true **do**
      Match $IB$ with $RB_1$
      Calculate $FMV$
      $FMV_1 \longleftarrow \sum_{M=1}^{N} RB_1$
    **end while**
    **while** true **do**
      Match $IB$ with $RB_2$
      Calculate $FMV$
      $FMV_2 \longleftarrow \sum_{M=1}^{N} RB_2$
    **end while**
    **while** true **do**
      Match $IB$ with $RB_3$
      Calculate $FMV$
      $FMV_3 \longleftarrow \sum_{M=1}^{N} RB_3$
    **end while**
    **while** true **do**
      Match $IB$ with $RB_4$
      Calculate $FMV$
      $FMV_4 \longleftarrow \sum_{M=1}^{N} RB_4$
    **end while**
    Assign best $FMV_i \longrightarrow IB$
  **end while**
**end procedure**

used in [2]. To simulate the packet-loss, we use H.265 RTP loss model proposed in [145], which is a well-known packet-loss model and is widely used in many research articles. In our experiments, three different PLRs, 1%, 3% and 5%, are used. We also implement the FC technique [2] and the BMA [146] under the same experimental conditions, as proposed in [2, 146].

Tables 3.4 and Table 3.5 summarise the simulation results for the BMA, the FC and the proposed FI approaches in terms of average PSNR with different PLRs for selected

**Algorithm 3** Backward Motion Estimation

**procedure :**COMPUTE $BME$
  Divide $N+1$ into $IBs$
  Divide $N-1$ into $SWs$
  Divide each $SW$ into $RBs$
  **while** true **do**
    **while** true **do**
      Match $IB$ with $RB_1$
      Calculate $BMV$
      $BMV_1 \longleftarrow \sum_{M=1}^{N} RB_1$
    **end while**
    **while** true **do**
      Match $IB$ with $RB_2$
      Calculate $BMV$
      $BMV_2 \longleftarrow \sum_{M=1}^{N} RB_2$
    **end while**
    **while** true **do**
      Match $IB$ with $RB_3$
      Calculate $BMV$
      $BMV_3 \longleftarrow \sum_{M=1}^{N} RB_3$
    **end while**
    **while** true **do**
      Match $IB$ with $RB_4$
      Calculate $BMV$
      $BMV_4 \longleftarrow \sum_{M=1}^{N} RB_4$
    **end while**
    Assign best $BMV_i \longrightarrow IB$
  **end while**
**end procedure**

HD and non-HD test video sequences. The MSE metric is used to calculate the distortion effects. The PSNR in simulations is based on the MSE value for each estimated block. The test video sequences used in simulations have varieties of motions, ranging from slow speed to high speed. The videos are distributed into three main categories, i.e., moving objects with static camera, static objects with moving camera and moving objects with moving camera. Tables 3.6 and Table 3.7 describe the average computational time with different PLRs, both for the BMA and the proposed FI algorithms for selected HD and non-HD videos, respectively. As shown in Table 3.4 and Table 3.5, the proposed FI al-

| Sequence | Resolution | Total Number of Frames | Frame Rate | QP | Bitrate (Kbps) |
|---|---|---|---|---|---|
| BasketballDrillText | 832 × 480 | 500 | 50 | 26, 29, 38 | 13.437, 9.443, 3.429 |
| BasketballPass | 416 × 240 | 500 | 50 | 21, 23, 30 | 5.563, 4.537, 2.126 |
| BlowingBubbles | 416 × 240 | 500 | 50 | 21, 23, 30 | 10.858, 8.93, 4.176 |
| Flowervase | 416 × 240 | 300 | 30 | 10, 11, 17 | 4.416, 4.067, 2.544 |
| | 832 × 480 | 300 | 30 | 19, 22, 27 | 7.426, 5.61, 3.492 |
| Keiba | 416 × 240 | 300 | 30 | 13, 17, 24 | 6.285, 4. 526, 2.434 |
| | 832 × 480 | 300 | 30 | 33, 37, 42 | 2.483, 1.558, 0.827 |
| RaceHorses | 416 × 240 | 300 | 30 | 17, 21, 29 | 7.516, 5.418, 2.358 |
| | 832 × 240 | 300 | 30 | 27, 29, 37 | 11.593, 9.347, 3.148 |

Table 3.3: Non-HD video sequences

| Sequence | Algorithm | PSNR with 1% PLR | | | PSNR with 3% PLR | | | PSNR with 5% PLR | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | QP 27 | QP 32 | QP 37 | QP 27 | QP 32 | QP 37 | QP 27 | QP 32 | QP 37 |
| BQTerrace | FC | 25.25 | 25.22 | 25.39 | 26.25 | 26.70 | 25.69 | 26.92 | 26.18 | 26.68 |
| | BMA | 29.40 | 29.36 | 29.04 | 29.07 | 28.48 | 29.48 | 28.03 | 26.90 | 28.32 |
| | Proposed | 34.70 | 34.22 | 32.95 | 34.23 | 33.42 | 33.04 | 32.86 | 31.29 | 32.11 |
| Cactus | FC | 23.38 | 23.63 | 22.19 | 21.88 | 21.78 | 21.02 | 21.32 | 20.71 | 20.39 |
| | BMA | 27.82 | 27.64 | 27.24 | 26.90 | 26.86 | 26.98 | 26.43 | 26.14 | 26.56 |
| | Proposed | 30.25 | 30.43 | 30.34 | 29.44 | 29.44 | 29.62 | 28.91 | 28.65 | 29.27 |
| Rush_hour | FC | 21.35 | 21.30 | 19.96 | 21.26 | 21.07 | 20.53 | 22.09 | 21.72 | 20.79 |
| | BMA | 29.16 | 29.16 | 27.44 | 28.09 | 29.27 | 28.02 | 28.43 | 27.75 | 28.15 |
| | Proposed | 32.99 | 32.98 | 32.16 | 32.07 | 33.33 | 32.54 | 32.71 | 31.65 | 32.49 |

Table 3.4: Average PSNR for HD video sequences

gorithm produces better performance as compared to the FC and the BMA in terms of average PSNR by approximately $12dB$ and $2.5dB$, respectively. In Table 3.6 and Table 3.7, the proposed FI algorithm leads the BMA in average computational time, by approximately $1,788$ seconds. The two main motives behind the development of our proposed FI algorithm are to produce an acceptable/affordable average PSNR and to have less computational cost. Less computational time is particularly focused because the EC algorithms execute at the end-user's side. In the case of mobile users, the end-users may have low

| Sequence | Algorithm | PSNR with 1% PLR | | | PSNR with 3% PLR | | | PSNR with 5% PLR | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | QP 21 | QP 23 | QP 30 | QP 21 | QP 23 | QP 30 | QP 21 | QP 23 | QP 30 |
| BlowingBubbles | FC | 25.25 | 25.22 | 25.39 | 26.25 | 26.70 | 25.69 | 26.92 | 26.18 | 26.68 |
| | BMA | 29.40 | 29.36 | 29.04 | 29.07 | 28.48 | 29.48 | 28.03 | 26.90 | 28.32 |
| | Proposed | 34.70 | 34.22 | 32.95 | 34.23 | 33.42 | 33.04 | 32.86 | 31.29 | 32.11 |
| Flowervase ($832 \times 480$) | FC | QP 19 | QP 22 | QP 27 | QP 19 | QP 22 | QP 27 | QP 19 | QP 22 | QP 27 |
| | | 23.38 | 23.63 | 22.19 | 21.88 | 21.78 | 21.02 | 21.32 | 20.71 | 20.39 |
| | BMA | 27.82 | 27.64 | 27.24 | 26.90 | 26.86 | 26.98 | 26.43 | 26.14 | 26.56 |
| | Proposed | 30.25 | 30.43 | 30.34 | 29.44 | 29.44 | 29.62 | 28.91 | 28.65 | 29.27 |
| RaceHorses ($832 \times 480$) | FC | QP 27 | QP 29 | QP 37 | QP 27 | QP 29 | QP 37 | QP 27 | QP 29 | QP 37 |
| | | 21.35 | 21.30 | 19.96 | 21.26 | 21.07 | 20.53 | 22.09 | 21.72 | 20.79 |
| | BMA | 29.16 | 29.16 | 27.44 | 28.09 | 29.27 | 28.02 | 28.43 | 27.75 | 28.15 |
| | Proposed | 32.99 | 32.98 | 32.16 | 32.07 | 33.33 | 32.54 | 32.71 | 31.65 | 32.49 |

Table 3.5: Average PSNR for non-HD video sequences

| Sequence | Algorithm | Time (Seconds) with 1% PLR | | | Time (Seconds) with 3% PLR | | | Time (Seconds) with 5% PLR | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | QP 27 | QP 32 | QP 37 | QP 27 | QP 32 | QP 37 | QP 27 | QP 32 | QP 37 |
| BQTerrace | BMA | 1748.53 | 1734.27 | 1744.73 | 1856.04 | 1817.65 | 1806.66 | 3383.66 | 3415.27 | 3364.38 |
| | Proposed | 12.48 | 12.49 | 12.59 | 12.51 | 12.54 | 12.77 | 12.58 | 12.66 | 12.72 |
| Cactus | BMA | 1812.33 | 1815.76 | 1813.48 | 1865.33 | 1826.65 | 1841.92 | 3657.99 | 3474.17 | 3522.72 |
| | Proposed | 12.61 | 12.72 | 12.74 | 12.66 | 12.79 | 12.82 | 12.61 | 12.82 | 12.78 |
| Rush_hour | BMA | 1455.27 | 1451.43 | 1444.11 | 1456.64 | 1460.32 | 1477.69 | 2757.98 | 2776.22 | 2714.78 |
| | Proposed | 12.68 | 14.85 | 13.89 | 12.69 | 14.74 | 14.36 | 12.29 | 14.88 | 14.47 |

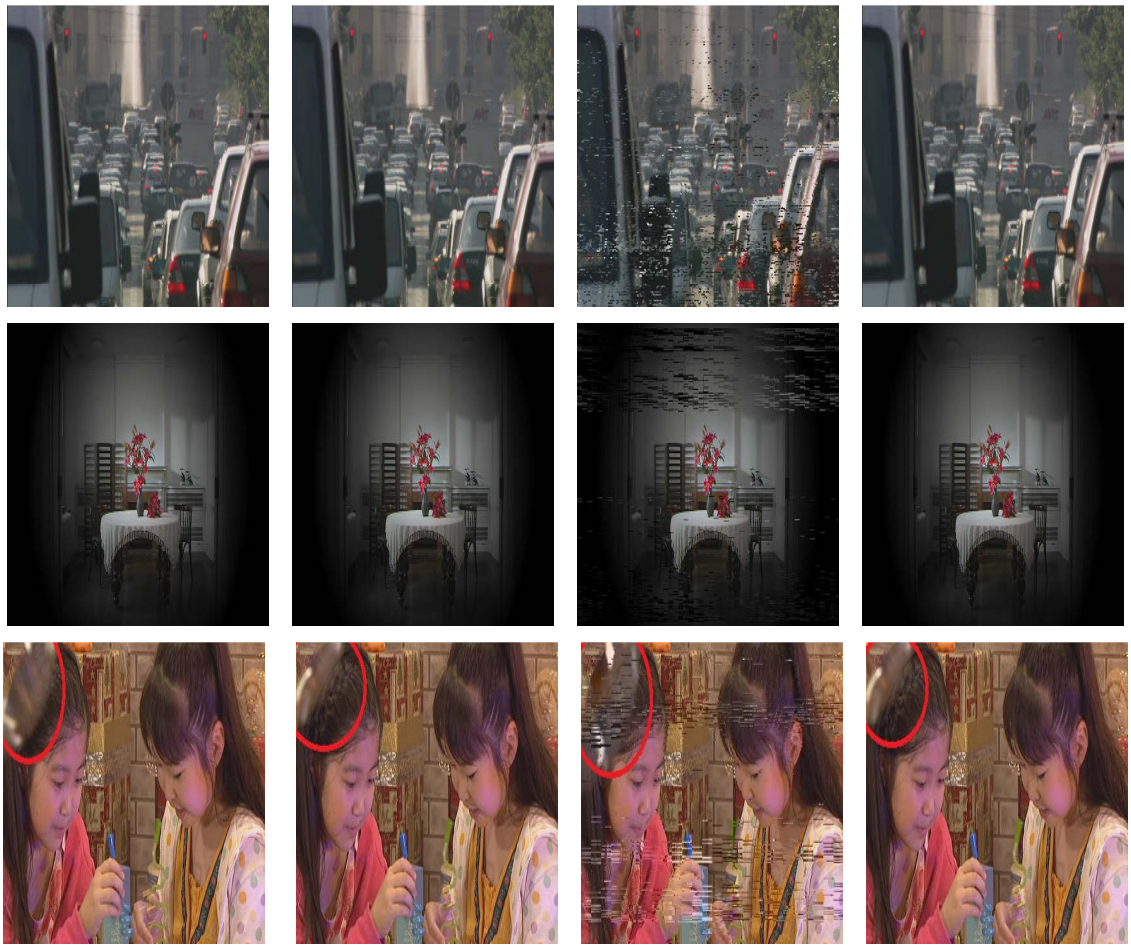
Table 3.6: Average computational time for HD videos

processing devices, such as smartphones, tablets and personal laptops. These end-user devices have slow processors with limited hardware resources, such as a small amount of Random Access Memory (RAM) and a limited number of temporary storage buffers with low speed data buses. In real-time streaming, delays are always time critical, so low processing time and the requirement for minimum hardware resources for the EC applications are always recommended. The visual comparison between the proposed and the reference algorithms is presented in Fig. 3.3 under different PLRs and QPs with different test video sequences. In Fig. 3.3, column (*a*) represents the lost video frames, (*b*) represents the

| Sequence | Algorithm | Time (Seconds) with 1% PLR | | | Time (Seconds) with 3% PLR | | | Time (Seconds) with 5% PLR | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | **QP 21** | **QP 23** | **QP 30** | **QP 21** | **QP 23** | **QP 30** | **QP 21** | **QP 23** | **QP 30** |
| BlowingBubbles | BMA | 79.92 | 63.48 | 102.89 | 71.51 | 71.20 | 71.38 | 141.59 | 156.47 | 161.44 |
| | Proposed | 1.51 | 1.52 | 1.52 | 1.52 | 1.52 | 1.51 | 2.01 | 2.02 | 2.02 |
| Flowervase | BMA | **QP 19** | **QP 22** | **QP 27** | **QP 19** | **QP 22** | **QP 27** | **QP 19** | **QP 22** | **QP 27** |
| (832 × 480) | | 398.59 | 448.77 | 420.15 | 508.34 | 526.45 | 520.88 | 1321.22 | 1305.41 | 1308.06 |
| | Proposed | 3.14 | 3.16 | 3.13 | 3.14 | 3.15 | 3.14 | 4.20 | 4.19 | 4.20 |
| RaceHorses | BMA | **QP 27** | **QP 29** | **QP 37** | **QP 27** | **QP 29** | **QP 37** | **QP 27** | **QP 29** | **QP 37** |
| (832 × 480) | | 379.18 | 488.47 | 296.20 | 508.49 | 471.11 | 476.65 | 1333.08 | 1221.59 | 1161.14 |
| | Proposed | 3.12 | 3.15 | 3.15 | 3.13 | 3.14 | 3.13 | 4.18 | 4.18 | 4.17 |

Table 3.7: Average computational time for non-HD videos

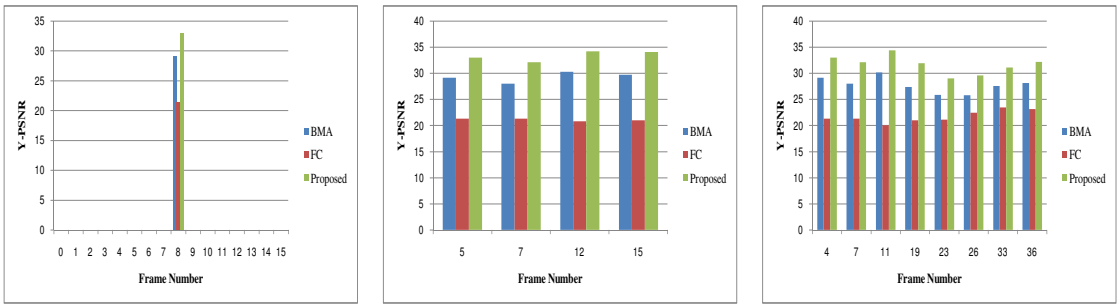
outputs of the FC algorithm, ($c$) represents the outputs of the BMA and ($d$) represents the outputs of the proposed FI algorithm. These sample images are taken from three different test videos. These videos have different resolutions and motions of cameras and/or objects. The first sample is taken from the HD video, *Rush_hour* with QP 32. The second one is from the non-HD video, *Flowervase* with resolution 832 × 480 and QP 22. The last one is also from the non-HD video, *BlowingBubbles* with QP 30. It can be seen very clearly that there is no bigger noticeable visual difference between the original, the output of the FC and the output of the proposed FI technique while the output of the BMA is showing lots of blurred lines. The only noticeable difference is present in the *BlowingBubbles* samples, which is red encircled. As there is no significant motion between consecutive video frames, the visual difference cannot be noticed much in the FC algorithm. However, in the case of higher PLRs, the FC algorithm may not perform well and may produce frame-freezing effects.



(a) Original (b) FC (c) BMA (d) Proposed

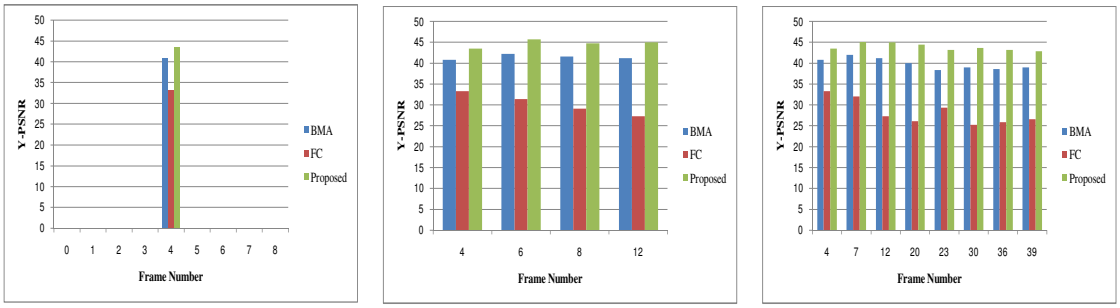
Figure 3.3: Visual comparisons

Fig. 3.4, Fig. 3.5 and Fig. 3.6 show the performance graphs of the FC, the BMA and the proposed FI algorithms in terms of average PSNR under different PLRs for different videos. Fig. 3.7, Fig. 3.8 and Fig. 3.9 show the performance graphs of the BMA and the proposed FI algorithms in terms of average computational time under different PLRs for different videos. It can be seen very easily that the average PSNR performance of the proposed FI algorithm is better than the FC and the BMA techniques. The proposed FI algorithm outperforms the BMA technique in terms of average computational time, proving its suitability for real-time processing.



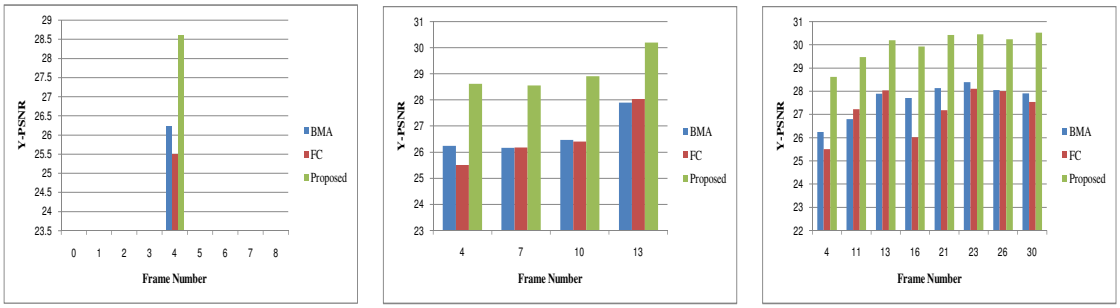

(a) 1% PLR (b) 3% PLR (c) 5% PLR

Figure 3.4: PSNR comparisons for Rush_hour



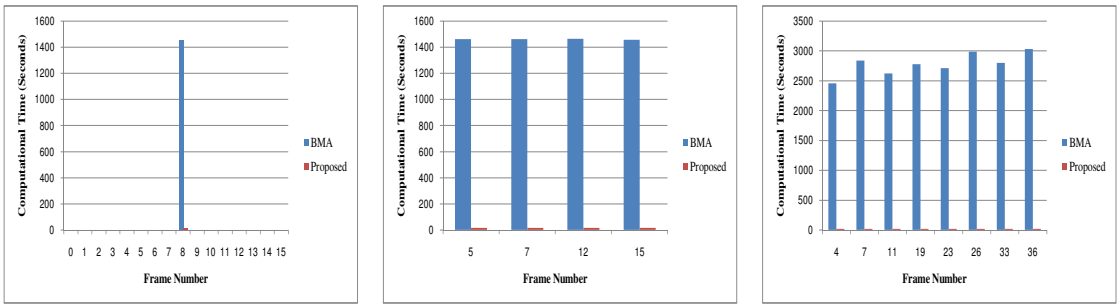

(a) 1% PLR (b) 3% PLR (c) 5% PLR

Figure 3.5: PSNR comparisons for Flowervase



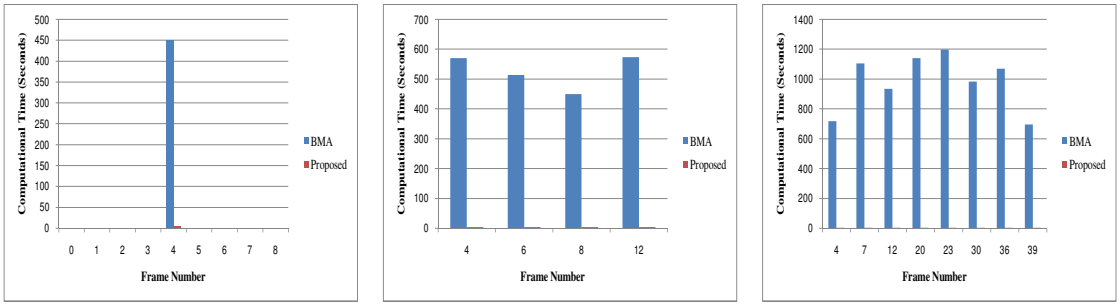

(a) 1% PLR (b) 3% PLR (c) 5% PLR

Figure 3.6: PSNR comparisons for BlowingBubbles



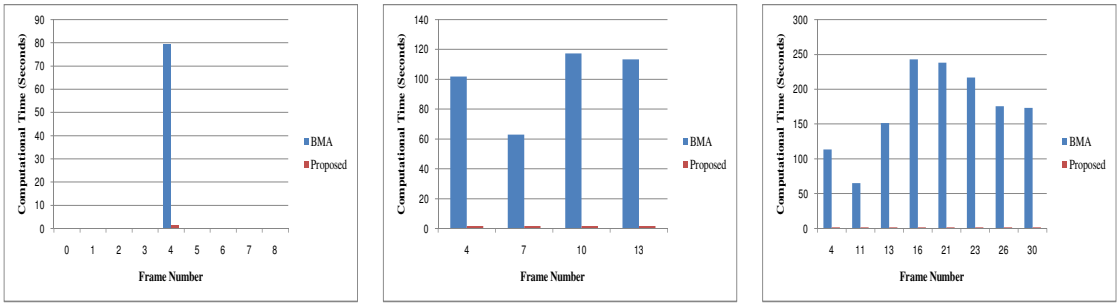

(a) 1% PLR (b) 3% PLR (c) 5% PLR

Figure 3.7: Computational time comparisons for Rush_hour




(a) 1% PLR (b) 3% PLR (c) 5% PLR

Figure 3.8: Computational time comparisons for Flowervase




(a) 1% PLR (b) 3% PLR (c) 5% PLR

Figure 3.9: Computational time comparisons for BlowingBubbles

## 3.4 Summary

In this chapter, a frame interpolation-based EC algorithm has been proposed to conceal lost Intra-frames in HEVC encoded HD and non-HD videos. The proposed EC algorithm utilises two video frames only to conceal the missing one. The goal of the proposed EC algorithm is to recover the missing Intra-frame with less computational complexity in real-time. The proposed EC algorithm is a block-based approach and segments the current and reference video frames into fixed size blocks, known as $IB$s and $SW$s, respectively. Best match for each $IB$ is searched in the corresponding $SW$ using a linear search. The searching process is threshold-based and is performed in a parallel style to save the computational time. After finding the best match, MVs of missing video frame are estimated using MVE. Next step is to utilise the estimated MVs to interpolate the missing video frame. Due to the estimation, it is possible that the concealed video frame might contain thin-hair like lines in various regions. To remove the effect of thin-hair like lines, an adaptive filter is applied. Unlike the existing EC approaches, our algorithm does not require complex hardware and can produce better reconstruction result in real-time.

Chapter 4

# Error Concealment for Scalable Encoded Videos

## 4.1 Introduction

Designing a video transmission system over heterogeneous transmission networks faces many challenges, such as fluctuating bandwidth, PLR in transmission channels and user's device capabilities. SVC is considered as one of the easiest approaches to address these challenges [147]. The SVC enables VSPs to encode videos in parts, according to spatial, temporal, and Signal-to-Noise-Ratio (SNR) requirements of the user. To meet end user demands, a VSP can rent cloud resources, such as virtual CPUs and a storage space, in order to avoid hardware investments and maintenance costs. Nowadays, many VSPs are switching their set-ups to cloud platforms, such as YouTube [148]. Public cloud platforms help VSPs to improve their QoS at a lower cost by creating multiple agents to deal with user requests in real-time. However, such improvements still require many factors to consider, such as fluctuating wireless channels and transmission errors [149]. Fig. 4.1 shows a traditional scenario, in which video bitstreams are transmitted from cloud servers over heterogeneous networks to end users. In this chapter, a novel EC scheme is proposed for

quality-oriented scalable video streaming using the Amazon cloud computing platform. This EC scheme is an extension of the EC scheme proposed in Chapter. 3. In this scheme, HD videos are encoded using scalable version of H.265 and stored on the cloud platform. The missing video frames are estimated on the receiver side using the modified EC scheme. The EC scheme is designed to recover a whole video frame lost during an SHVC video transmission. The scalable version encodes an HD video into multiple data inter-linked streams. In order to estimate the missing video frame, the EC scheme first finds the relevant reference video frame from the inter-linked streams. After finding the relevant reference video frame, it follows the same steps, as performed in the EC scheme proposed in Chapter 3. However, instead of following the FS algorithm to compute the missing MVs, it follows Test-Zone Search (TZS) algorithm. In the following sections, we explain the proposed EC scheme followed by a discussion on experimental results.

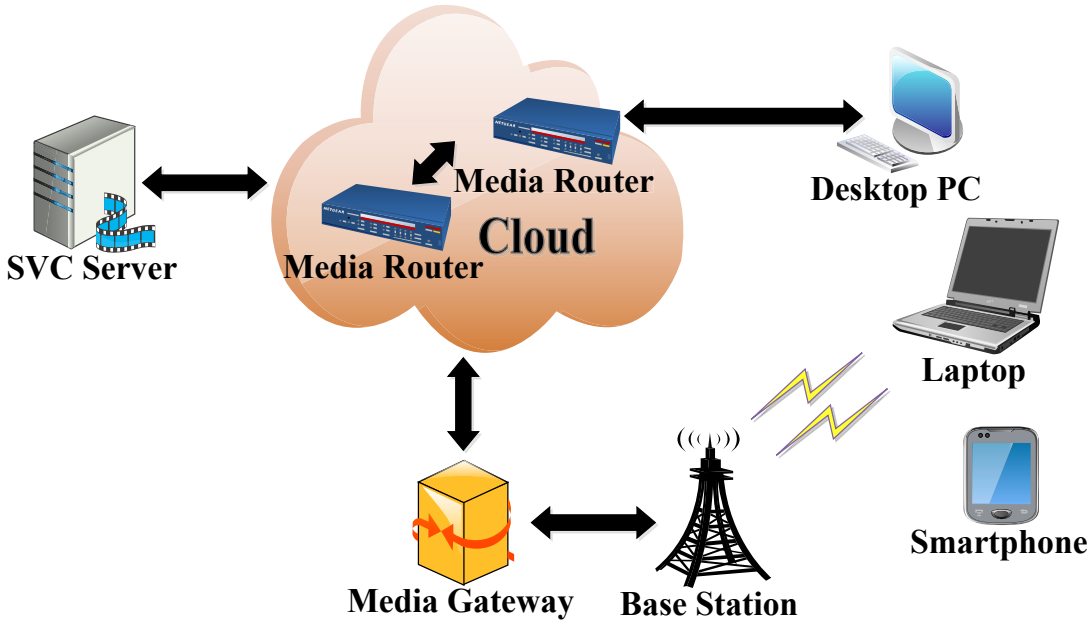



Figure 4.1: Traditional cloud-based video streaming

## 4.2 Scalable Frame Interpolation

Scalable Frame Interpolation (SFI) is an extension of our proposed technique presented in Chapter. 3. The targeted applications of SFI technique are cloud-hosted and assisted scalable video streaming. The encoded videos are stored in a video storage centre. In SVC, there are always multiple streams, of which one is known as BL, while the rest are called ELs. The BL is always encoded in Intra mode, while ELs are encoded in Inter mode [150]. The video streams are encoded using various transcoding servers, under the supervision of a pipe-lining server. The streaming server is responsible to deal with user's requests. For each lost BL or EL frame, MVs are derived first using an ME procedure. Usually, the ME process is performed on the encoder side and is considered as the most time-consuming task, consuming 40% to 80% of the encoding computational time [127]. However, in order to apply the ME process on the decoder side, it needs to be modified to support real-time processing. In our proposed approach, the main targets are to minimise the computational time and to improve the visual quality. Less computational time is highly desired for scalable video streams, as we cannot expect high processing capability from user's mobile device. Massive processing has a direct effect on the stored battery charge. Minimum computational time can easily be achieved, if the computational process is less complex and involves simple and fewer operations. Minimum computational time also helps to support real-time video analysis and communication. In our proposed SFI, the ME process considers two video frames at a time, i.e., immediate previous video frame and the next video frame. The blocks in the previous video frame and the next video frame represent heads and tails of the estimated MVs. The tail represents the starting point of a MV, while the head represents the ending point, as shown in Fig. 4.2. The $IB$ is known as the Input Block in the current video frame while the $SW$ is the Searching Window in the reference video frame. The default algorithm for the ME process is always the BMA on the encoder side. Therefore, we also use the same algorithm for further enhancements. The estimated MVs help in estimating the lost video frames. The estimated frames are

further processed by an adaptive filter to remove minor blurring and hair effects produced by the estimation process. All these phases are performed by our proposed SFI technique, whenever called by an SVC decoder, as shown in Fig. 4.3.

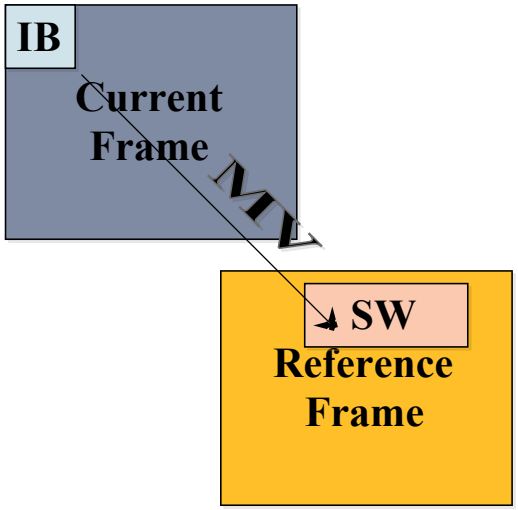



Figure 4.2: The motion vector

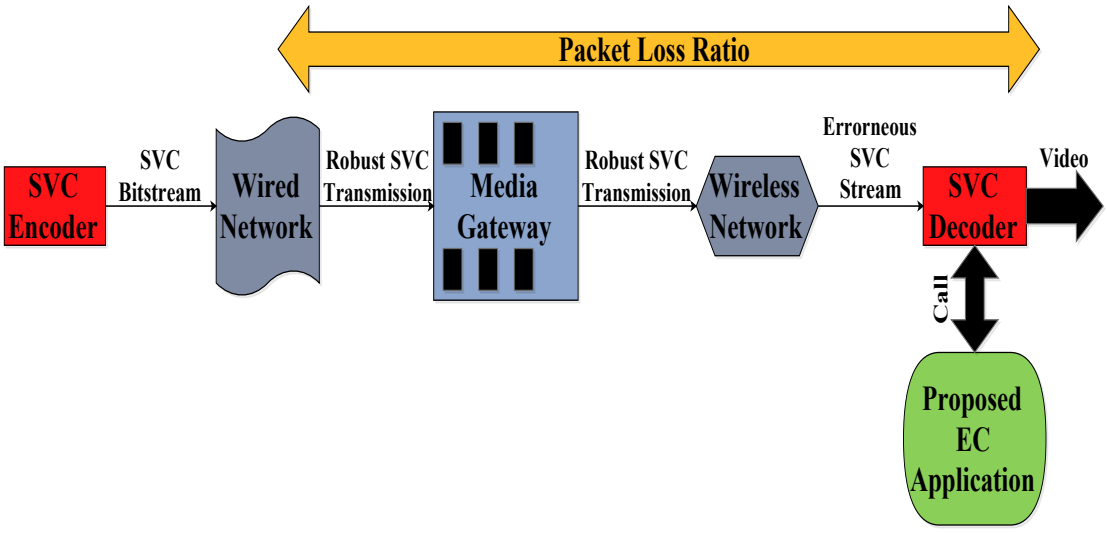



Figure 4.3: Scalable video streaming

### 4.2.1 Reference Frame Selection For Enhancement Layer

In the case of a BL video frame concealment, the steps shown in Subsections 4.2.2 to 4.2.7 are followed. However, the case of an EL video frame concealment is slightly different. EL is an Inter-frame based. To conceal a missing video frame in an EL stream, we first need to find the master BL or the EL video frames, on which the missing video frame is dependent on. Care must be taken at this stage, as the input video sequences are of HD or UHD, and are compressed significantly in the EL streams [150]. A flag in the header of an EL video frame indicates whether it is predicted from a BL or an EL frame. Therefore, the first step in this phase is to find out the master video frames for a missing video frame. The block diagram of an SHVC decoder is shown in Fig. 4.4. The *DEMUX* extracts the EL video frames from an incoming video stream. The *Inter Layer Processing* gives us a knowledge about the dependency of an EL video frame on the BL or the EL video frames.

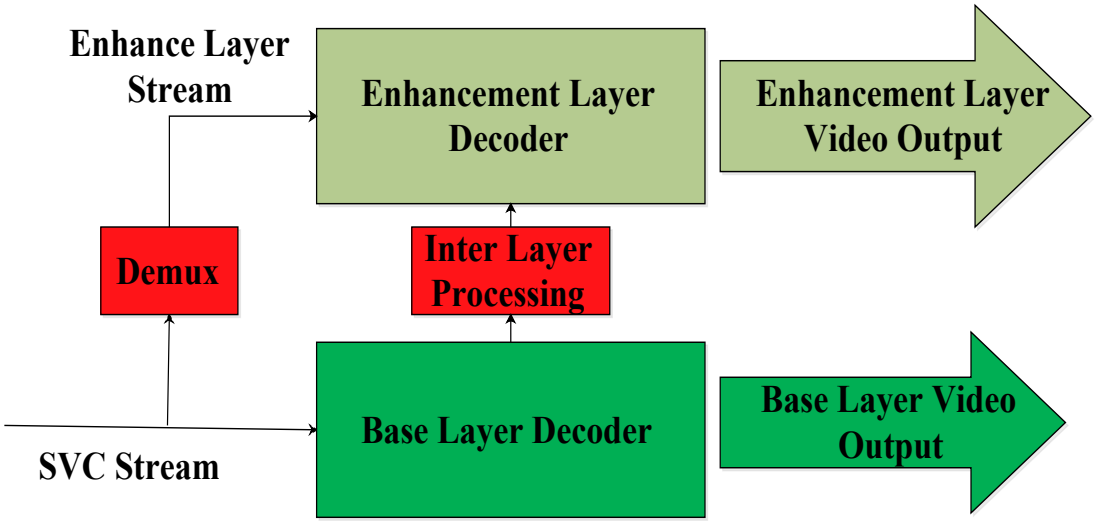



Figure 4.4: SVC decoder

In our proposed approach, we modify the decoder to extract the information about the dependencies of EL video frames and forward it to our SEC scheme. In the SEC scheme, the first step is to determine whether the missing video frame is from the BL or the EL

stream. If the missing video frame belongs to the BL stream, then steps from Subsections 4.2.2 to 4.2.7 are followed. If the missing video frame belongs to the EL stream, the information received from the decoder is utilised to find the master or reference video frames for the missing video frame. If the master video frame is from the BL stream, then only one video frame will be involved. In that case, the SEC scheme follows the steps from Subsections 4.2.2 to 4.2.7. If the master video frame is from the EL stream, then it means that more than one master frames are involved. Usually, the SEC scheme keeps copies of at least three previously processed EL video frames. The SEC scheme first finds the master video frames of the three previously processed EL video frames. Once the master video frames are found, The SEC scheme computes an average of intensities in those master video frames. This computed average is used to produce a master video frame for the missing EL video frame, as shown in Fig. 4.5.

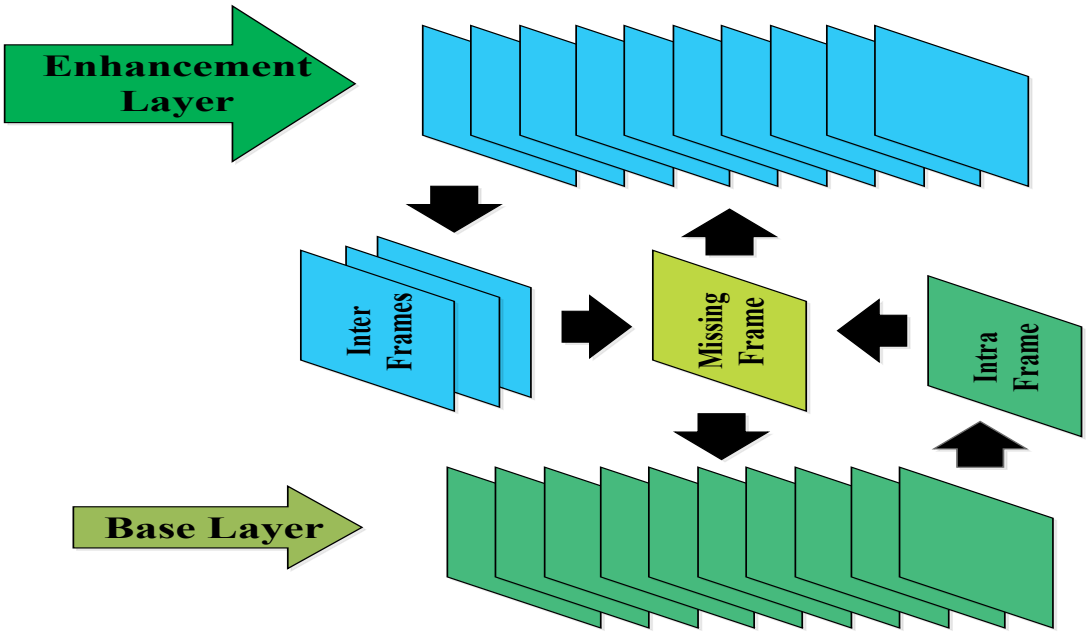



Figure 4.5: Master video frame referencing

### 4.2.2 Block Size Selection

Just like FI proposed in Chapter. 3, the first phase in the proposed SFI is the division of a current video frame into a number of blocks. During this phase, the SEC scheme first divides every video frame into $n$ blocks of equal size. In our experiments, a block size of $16 \times 16$ is selected. The main reason to choose this block size is to make the SEC scheme compatible with prior standards where the maximum supported block size is $16 \times 16$ [129].

Applying threads is very common in the video coding and decoding, as it is always done to speed up the computations drastically. When the Motion Estimation (ME) is applied on the blocks of a video frame, it can be done in parallel by generating multiple threads. These threads can be generated using any appropriate underlying thread library, such as *pthreads (C/C++)*, *java.lang.Thread (JAVA/Andorid)* or *Windows Threads*, and assigned to different blocks in the video frame to process them simultaneously.

In the SEC scheme, the Input Video (IV) stream is processed on frame by frame basis rather than as a group of video frames. The SEC scheme always keeps copies of the recently processed video frames. If frame $N$ of a video is lost during a transmission, it is immediately notified through the sequence numbers stored in the header of other video frames previously received. The SEC scheme brings back the $(N-1)^{th}$ video frame and waits for the $(N+1)^{th}$ video frame to arrive. Once the $(N+1)^{th}$ video frame arrives, the concealment procedure starts immediately by partitioning the $(N+1)^{th}$ video frame into blocks of size $16 \times 16$. The $(N+1)^{th}$ video frame is considered as the current video frame and its blocks are denoted as Input Blocks (IBs), while the $(N-1)^{th}$ video frame is considered as the reference video frame and its blocks are denoted as Reference Blocks (RBs).

### 4.2.3 Construction of Searching Windows for Block Matching

In this phase, for each input block of size $16 \times 16$, a searching window (SW) of size $64 \times 64$ is constructed in the reference video frame in such a way that the location of

the input block is located in the middle of the corresponding searching window. All SWs have the same size. In determining the size of SWs, we assume that the motion in the consecutive video frames is not very fast. Therefore, we set the size of an SW to be four times (i.e., $64 \times 64$) of the size of an IB [129].

### 4.2.4 Searching Patterns

In this phase, a process to find the best match through searching is performed between the current and the reference video frames. This process works on the block level. The best match for blocks in the current video frame is searched in the reference video frame.

In the SEC scheme, TZS is modified. By default, TZS starts with an 8-point Diamond Search (DS). After finding a best match, it still goes further for three more rounds. In the SEC scheme, a 16-point DS is proposed to cover a bigger area without the three extra rounds. The modified TZS works under the assumption of slow and linear motion in the consecutive video frames. Secondly, the searching pace is changed from two pixels of TZS to one pixel in this paper in the searching direction, in order to better detect the minor changes. Thirdly, the searching process is performed in parallel using multiple threads in all the blocks within the searching window.

In the SEC scheme, multiple Working Threads (WTs) are generated during the ME process and are assigned to each IB. Each WT processes one row of the IB. The size of each IB is $16 \times 16$, so 16 WTs are assigned to each IB. To further speed up the searching process, an adaptive threshold is set to terminate the DS process. It changes every time when a new searching process triggers. Each RB is defined as a block inside searching window, and the sizes of RBs and IBs are the same. In the beginning, a random but a small value is set as a termination threshold. The matching process is based on one pixel movement. If the best match is found and the distance (i.e., the SAD value) is lower than the termination threshold, the searching process immediately terminates. If the best match is found and the distance is equal to or higher than the termination threshold, the searching

process continues with the raster search to find an optimum match. The distance between an IB and the optimum match is used as a termination threshold for the searching process of next IB.

### 4.2.5 Motion Vector Estimation

During the searching patterns phase, SAD is calculated for each pixel, and then compared with the termination threshold. There are three possible scenarios. Firstly, if the computed SAD is greater than the termination threshold in three unit steps (i.e., three pixel paces), the searching process leaves that row of pixels by assuming that there is no match in this row. Secondly, if the computed SAD is equal to the termination threshold, the searching process continues to find a better match up to three unit steps. Thirdly, if the computed SAD is less than the termination threshold, the searching process stops immediately. This process is repeated for all the possible locations in the targeted SW. This process may produce more than one better match, thus creating a set of best matches. Next, the $(X\text{-}Y)$ coordinates of the pixels representing the best matches are obtained. The $(X\text{-}Y)$ coordinates in the reference and the current video frames represent the tail and the head of the desired MVs, respectively. In the end, another search is performed to find an MV having a minimum length from the set of the estimated MVs. The MV with the minimum length is used to represent the motion of a block of pixels from the reference to the current video frames. This process is known as Forward Motion Estimation (FME) and the MVs estimated in this process are known as Forward Motion Vectors (FMVs). The same process is repeated from Subsections 4.2.2 to 4.2.5 by transforming the $(N+1)^{th}$ and the $(N-1)^{th}$ video frames as the reference and the current video frames, respectively. This process is known as Backward Motion Estimation (BME) and the MVs estimated in this process are known as Backward Motion Vectors (BMVs).

### 4.2.6 Frame Interpolation

Let frame $N$ be the frame to be concealed, $F_i$ represents the $i^{th}$ frame and $F_i(x,y)$ represent the intensity at pixel location $(x,y)$ of $F_i$ ($i = 1,2,\cdots$), and $D$ represent the image domain. Then, the intensity values are initialised using the simple interpolation algorithm in Eq. 4.1 [151].

$$F_N(x,y) = 0.5F_{N-1}(x,y) + 0.5F_{N+1}(x,y), (x,y) \in D \tag{4.1}$$

Let $V_{x,y}$ represent the MV of the block containing $(x,y)$, and $(x_b,y_b)$ and $(x_f,y_f)$ represent the tail and the head of the MV, respectively. Then, it is obvious that

$$\begin{aligned}(x_b,y_b) &= (x,y) - 0.5V_{x,y},\\ (x_f,y_f) &= (x,y) + 0.5V_{x,y}.\end{aligned} \tag{4.2}$$

Then, the intensity value at pixel location $(x,y)$ of the missing video frame (i.e., frame $N$) is updated using a modified interpolation algorithm, taking into account the block MV, as Eq. 4.3

$$F_N(x,y) = 0.5F_{N-1}(x_b,y_b) + 0.5F_{N+1}(x_f,y_f), \tag{4.3}$$

when both $(x_b,y_b)$ and $(x_f,y_f)$ are in $D$.

### 4.2.7 Adaptive Filtering

Thin black-lines commonly appear in block-based approaches. The Arithmetic Mean Filter (AMF) [126] that is averaging pixel values in each pixel's neighbourhood can be used to suppress this problem of thin black-lines, but it will also have blurring effects at image contours and edges. To propose a method that can smooth the mis-concealed pixel values

while keeping the sharpness of edges and contours, an Adaptive Denoising Filter (ADF) is applied and computed at each pixel location $(x, y)$ by

$$ADF(x,y) = \begin{cases} CP_{AMF}(x,y), & \text{if } \Delta_1 - \Delta_{(1)} > T_h \\ CP(x,y), & \text{otherwise} \end{cases} \tag{4.4}$$

where $ADF(x,y)$ is the output pixel value at $(x,y)$ generated by an ADF, $CP(x,y)$ is the concealed pixel at $(x,y)$ computed by Eq. 4.3, $CP_{AMF}(x,y)$ is the pixel value at $(x,y)$ concealed using AMF, $\Delta_1$ is the cumulatively weighted distance (CWD) allocated to the central pixel of a filtering window, $\Delta_{(1)}$ is the smallest CWD within the window, and $T_h$ is a threshold value. Detailed description of $\Delta_1$, $\Delta_{(1)}$ and the selection of $T_h$ value can be referred to [144]. The adaptive filter is shown in Fig. 4.6. Algorithm 4 summarises the entire SEC scheme detailed above in this section.

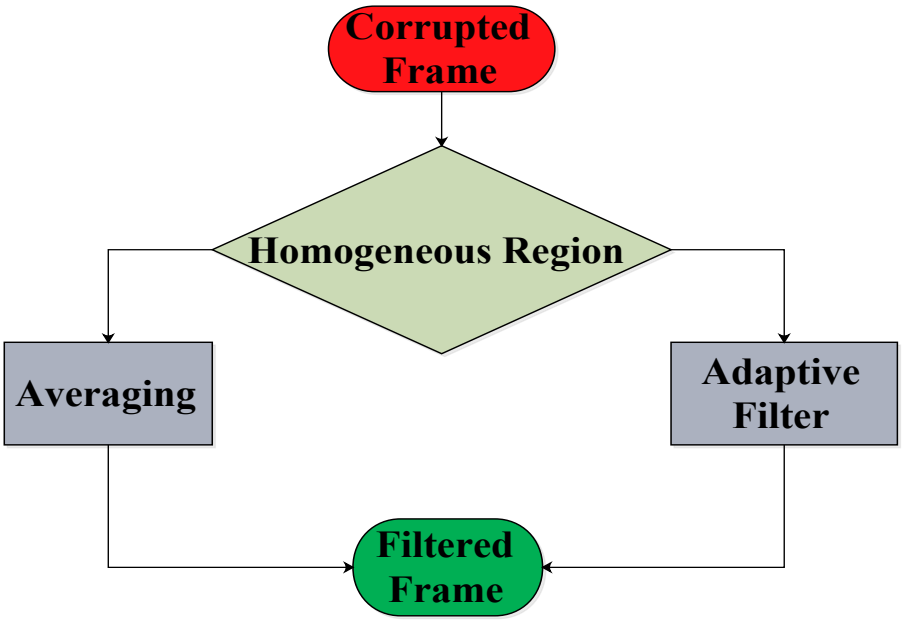



Figure 4.6: Adaptive filter

**Algorithm 4** Scalable Error Concealment

```
1: procedure SCALABLE–ERROR–CONCEALMENT
2:     IV: Input Video, IB: Input Block, SW: Search Window, RB: Reference Block
3:     for Missing F_N ∈ IV do
4:         if F_N ∈ BL then
5:             goto Concealment
6:         else
7:             Find Master Frame
8:             goto Concealment
9:         end if
10:        goto Filtering
11:    end for
12: end procedure
13: Concealment:
14: while F_{N-1}, F_N, F_{N+1} ∈ IV do
15:    Read F_{N+1}
16:    Read F_{N-1}
17:    goto FME
18:    goto BME
19:    Conceal F_N using Eq. 4.3
20: end while
21: FME:
22: while IB ⊂ F_{N-1} do
23:    Construct SW in F_{N+1}
24:    for RB ⊂ SW do
25:        Generate Working Threads (WTs)
26:        Perform Mapping between IB and RB Using WTs
27:        Compute FMV for IB
28:    end for
29: end while
30: BME:
31: while IB ⊂ F_{N+1} do
32:    Construct SW in F_{N-1}
33:    for RB ⊂ SW do
34:        Generate WTs
35:        Perform Mapping between IB and RB Using WTs
36:        Compute BMV for IB
37:    end for
38: end while
39: Filtering:
40: while True do
41:    Generate WTs
42:    Apply Adaptive Filter on Concealed F_N Using WTs
43: end while
```

## 4.3 Implementation and Evaluation

Test video sequences are chosen from an SHVC standard documentation [152]. Six test video sequences are used for simulation purpose in our experiments. Details of the test video sequences, such as resolution, total number of frames and frame rate, are listed in Table 4.1.

| **Sequence** | **Resolution** | **Total Number of Frames** | **Frame Rate** |
|---|---|---|---|
| PeopleOnstreet | $2560 \times 1600$ | 150 | 30 |
| BasketballDrive | $1920 \times 1080$ | 500 | 50 |
| BQTerrace | $1920 \times 1080$ | 600 | 60 |
| Cactus | $1920 \times 1080$ | 500 | 50 |
| Kimono | $1920 \times 1080$ | 240 | 24 |
| ParkScene | $1920 \times 1080$ | 240 | 24 |

Table 4.1: Test video sequences

Encoding details, such as Quantisation Parameters (QPs) used during the encoding process and the bit-rates obtained at the end of the encoding process, are tabulated in Table 4.2.

| **Sequences** | **QP Pairs** | | | | |
|---|---|---|---|---|---|
| | **(20,22)** | **(24,26)** | **(28,30)** | **(32,34)** | **(36,38)** |
| PeopleOnStreet | 152.799 | 711.312 | 505.382 | 319.452 | 205.801 |
| BasketballDrive | 102.077 | 412.136 | 214.05 | 129.638 | 805.679 |
| BQTerrace | 191.936 | 123.574 | 675.929 | 406.758 | 249.337 |
| Cactus | 156.169 | 744.634 | 412.003 | 250.373 | 151.744 |
| Kimono | 234.06 | 135.395 | 861.646 | 561.255 | 357.66 |
| ParkScene | 646.129 | 394.476 | 235.31 | 137.842 | 775.943 |

Table 4.2: QP pairs and video sequences' bitrates

The QPs regulate how much detail within a frame is saved. The bit-rate means the number of bits that make up a section of a video per unit of time, i.e., a bit-rate of 1kbs means that every second of video is represented by 1 thousand bits. In SHVC encoding, QP values always come in pairs. In a QP pair, first value is used to encode BL and second value is used to encode EL. QP pairs are chosen according to the SHVC standard documentation [152]. The test videos are encoded using the standard settings of SHVC in an Unsliced mode. Group of Pictures (GoP) is set to 1 because of the Intra coding mode [152].

Details of the experimental platform are summarised in Table 4.3. The experiments and performance evaluations of the SEC scheme are conducted using SHM-Dev 10.0 [40] and Matlab R2015a. To simulate packet-losses, H.265 RTP loss model proposed in [145] is adopted. In our experiments, we selected PLRs of 1%, 3% and 5%. For comparison purposes, we also implemented the FC [153] and the BMA [154] techniques under the same experimental settings.

| | |
|---|---|
| Hardware | CPU: Intel Core$^{TM}$ i5-4590<br>CPU @ 3.30 GHz<br>RAM: 16 GB |
| Software | SHVC Codec SHM-10.0-Dev<br>Matlab R2015b |
| Video Format | 4:2:0 |
| QP | 20 to 38 |
| PLR | 1%, 3%, 5% |
| Methods | FC, BMA, Proposed technique |

Table 4.3: Simulation environment

The test videos are encoded in the cloud environment, provided by Matlab. Current version of Matlab offers Amazon EC2 cloud facility and programmers can easily use built-in functions to perform cloud-based computing [155]. In our experiments, the Amazon cluster consists of 20 virtual machines of type medium, having 2 virtual CPUs and 16GB RAM. SHVC encoder is executed on one virtual machine of type medium. The maximum

size of a heap is set to 8GB.

### 4.3.1 Experimental Results

The simulation results in terms of average PSNRs under different QPs and PLRs, are shown in Table 4.4.

| Sequence | Algorithm | 1% PLR | | | 3% PLR | | | 5% PLR | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | QP (20,22) | QP (28,30) | QP (36,38) | QP (20,22) | QP (28,30) | QP (36,38) | QP (20,22) | QP (28,30) | QP (36,38) |
| PeopleOnStreet | FC | 24.46 | 24.88 | 25.76 | 25.98 | 26.22 | 26.56 | 25.34 | 25.43 | 24.21 |
| | BMA-FS | 30.03 | 29.89 | 30.11 | 28.78 | 28.88 | 27.38 | 28.55 | 29.94 | 30.37 |
| | Proposed-TZS | 35.46 | 34.33 | 35.73 | 34.59 | 34.85 | 35.63 | 35.41 | 35.36 | 34.71 |
| BasketballDrive | FC | 24.22 | 24.97 | 24.59 | 25.27 | 25.42 | 26.79 | 26.28 | 25.96 | 25.78 |
| | BMA-FS | 30.49 | 30.66 | 31.42 | 31.93 | 29.72 | 29.65 | 30.62 | 31.95 | 30.66 |
| | Proposed-TZS | 35.83 | 35.66 | 36.49 | 36.68 | 34.19 | 34.35 | 35.07 | 36.69 | 35.82 |
| BQTerrace | FC | 25.88 | 25.94 | 26.3 | 27.17 | 26.68 | 26.49 | 26.53 | 26.31 | 26.39 |
| | BMA-FS | 29.4 | 30.07 | 29.11 | 26.82 | 27 | 27.91 | 28.11 | 25.54 | 28.51 |
| | Proposed-TZS | 34.19 | 34.72 | 32.5 | 31.42 | 31.54 | 31.77 | 32.81 | 33.13 | 32.14 |
| Cactus | FC | 22 | 21.6 | 21.11 | 21.2 | 20.91 | 20.16 | 21.6 | 21.26 | 20.64 |
| | BMA-FS | 26.71 | 26.87 | 27.36 | 25.94 | 26.18 | 26.31 | 26.32 | 26.53 | 26.84 |
| | Proposed-TZS | 29.14 | 29.46 | 29.93 | 28.52 | 28.75 | 28.96 | 28.83 | 29.11 | 29.44 |
| Kimono | FC | 23.06 | 23.11 | 23.36 | 22.51 | 22.19 | 22.86 | 22.79 | 22.65 | 23.11 |
| | BMA-FS | 25.65 | 25.77 | 26.33 | 25.27 | 25.64 | 25.39 | 25.46 | 25.71 | 25.86 |
| | Proposed-TZS | 28.42 | 28.58 | 29.07 | 28.07 | 28.52 | 28.13 | 28.25 | 28.55 | 28.6 |
| ParkScene | FC | 25.47 | 25.94 | 26.63 | 24.9 | 24.59 | 25.29 | 26.66 | 26.17 | 25.58 |
| | BMA-FS | 28.57 | 29.39 | 29.99 | 30.72 | 30.69 | 30.22 | 29.09 | 29.77 | 30.51 |
| | Proposed-TZS | 34.66 | 35.93 | 35.62 | 36.41 | 36.89 | 35.69 | 35.16 | 34.7 | 35.44 |

Table 4.4: Average PSNR for test video sequences

To calculate the average PSNRs, Mean Square Error (MSE) metric is used. The test video sequences contain different types of motions, such as moving objects with a static camera, a moving camera with static objects and a moving camera with moving objects. It is shown clearly in Table 4.4 that the SEC scheme outperforms the FC in terms of the average PSNRs by approximately 5dB to 8.5dB under a 1% PLR, 4.5dB to 7.5dB under a 3% PLR and 6.5dB to 9.5dB under a 5% PLR. In comparison with the BMA, the SEC

scheme outperforms in terms of the average PSNRs by approximately 3.5dB to 5.5dB under a 1% PLR, 3.7dB to 4.6dB under a 3% PLR and 2.5dB to 4.5dB under a 5% PLR.

The average SSIM index values under different QPs and PLRs are tabulated in Table 4.5. The SSIM index is calculated for FC, BMA and SEC schemes. As shown in Table 4.5, on average, BMA produces the least amount of index values due to the presence of blurred lines in the outputs of the BMA. On average, SEC scheme shows consistent performance and produces better index values as compared to both FC and BMA under the same values of QPs and PLRs.

| **Sequence** | **Algorithm** | **1% PLR** | | | **3% PLR** | | | **5% PLR** | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | **QP (20,22)** | **QP (28,30)** | **QP (36,38)** | **QP (20,22)** | **QP (28,30)** | **QP (36,38)** | **QP (20,22)** | **QP (28,30)** | **QP (36,38)** |
| PeopleOnStreet | FC | 0.9076 | 0.9077 | 0.9096 | 0.9078 | 0.9000 | 0.9009 | 0.9087 | 0.9032 | 0.9088 |
| | BMA-FS | 0.8917 | 0.8916 | 0.8996 | 0.8960 | 0.8958 | 0.8915 | 0.8983 | 0.8944 | 0.8959 |
| | Proposed-TZS | 0.9189 | 0.9123 | 0.9156 | 0.9152 | 0.9178 | 0.9139 | 0.9167 | 0.9137 | 0.9184 |
| BasketballDrive | FC | 0.9045 | 0.9053 | 0.9071 | 0.9081 | 0.9017 | 0.9026 | 0.9083 | 0.9040 | 0.9007 |
| | BMA-FS | 0.8992 | 0.8909 | 0.8937 | 0.8938 | 0.8904 | 0.8945 | 0.8911 | 0.8954 | 0.8947 |
| | Proposed-TZS | 0.9169 | 0.9112 | 0.9183 | 0.9110 | 0.9196 | 0.9124 | 0.9196 | 0.9147 | 0.9188 |
| BQTerrace | FC | 0.9036 | 0.9057 | 0.9092 | 0.9090 | 0.9004 | 0.9036 | 0.9037 | 0.9098 | 0.9001 |
| | BMA-FS | 0.8944 | 0.8951 | 0.8913 | 0.8955 | 0.8919 | 0.8948 | 0.8954 | 0.8997 | 0.8928 |
| | Proposed-TZS | 0.9136 | 0.9164 | 0.9187 | 0.9197 | 0.9160 | 0.9188 | 0.9134 | 0.9147 | 0.9152 |
| Cactus | FC | 0.9071 | 0.9065 | 0.9082 | 0.9017 | 0.9006 | 0.9073 | 0.9025 | 0.9027 | 0.9083 |
| | BMA-FS | 0.8917 | 0.8946 | 0.8936 | 0.8928 | 0.8961 | 0.8983 | 0.8990 | 0.8963 | 0.8999 |
| | Proposed-TZS | 0.9165 | 0.9133 | 0.9138 | 0.9195 | 0.9199 | 0.9140 | 0.9138 | 0.9138 | 0.9175 |
| Kimono | FC | 0.9015 | 0.9060 | 0.9021 | 0.9046 | 0.9099 | 0.9042 | 0.9092 | 0.9082 | 0.9086 |
| | BMA-FS | 0.8916 | 0.8910 | 0.8934 | 0.8998 | 0.8988 | 0.8936 | 0.8961 | 0.8908 | 0.8945 |
| | Proposed-TZS | 0.9137 | 0.9154 | 0.9165 | 0.9185 | 0.9163 | 0.9132 | 0.9186 | 0.9152 | 0.9142 |
| ParkScene | FC | 0.9093 | 0.9080 | 0.9098 | 0.9090 | 0.9070 | 0.9081 | 0.9086 | 0.9003 | 0.9004 |
| | BMA-FS | 0.8940 | 0.8927 | 0.8962 | 0.8942 | 0.8931 | 0.8962 | 0.8933 | 0.8925 | 0.8950 |
| | Proposed-TZS | 0.9102 | 0.9115 | 0.9172 | 0.9198 | 0.9142 | 0.9142 | 0.9167 | 0.9154 | 0.9137 |

Table 4.5: Average SSIM for test video sequences

The average computational times under different QPs and PLRs for the SEC, TZS and BMA schemes are tabulated in Table 4.6.

| Sequence | Algorithm | 1% PLR | | | 3% PLR | | | 5% PLR | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | QP (20,22) | QP (28,30) | QP (36,38) | QP (20,22) | QP (28,30) | QP (36,38) | QP (20,22) | QP (28,30) | QP (36,38) |
| PeopleOnStreet | FC | 88.63 | 90.39 | 98.93 | 160.77 | 152.97 | 120.66 | 140.62 | 130.32 | 120.72 |
| | BMA-FS | 75.83 | 77.31 | 76.86 | 135.83 | 129.49 | 102.76 | 115.72 | 105.85 | 100.82 |
| | Proposed-TZS | 0.2103 | 0.2111 | 0.2103 | 0.2333 | 0.2119 | 0.2003 | 0.2321 | 0.2121 | 0.2133 |
| BasketballDrive | FC | 83.55 | 85.87 | 80.22 | 148.29 | 135.62 | 105.53 | 115.83 | 110.39 | 90.29 |
| | BMA-FS | 61.77 | 66.66 | 62.37 | 129.34 | 107.75 | 77.26 | 90.91 | 88.55 | 70.73 |
| | Proposed-TZS | 0.2072 | 0.2122 | 0.2999 | 0.2243 | 0.2191 | 0.2133 | 0.2121 | 0.2201 | 0.2103 |
| BQTerrace | FC | 77.46 | 82.88 | 78.21 | 146.26 | 132.69 | 93.67 | 111.86 | 107.79 | 85.94 |
| | BMA-FS | 60.63 | 64.87 | 61.22 | 114.48 | 103.86 | 73.32 | 87.55 | 84.37 | 67.27 |
| | Proposed-TZS | 0.2108 | 0.211 | 0.209 | 0.2113 | 0.2096 | 0.2115 | 0.211 | 0.2106 | 0.2103 |
| Cactus | FC | 99.14 | 99.14 | 89.65 | 125.6 | 115.07 | 61.72 | 112.37 | 107.1 | 75.68 |
| | BMA-FS | 77.59 | 77.59 | 70.17 | 98.31 | 90.01 | 48.31 | 87.95 | 83.83 | 59.24 |
| | Proposed-TZS | 0.2123 | 0.2123 | 0.2068 | 0.2084 | 0.2091 | 0.2109 | 0.2103 | 0.2107 | 0.2089 |
| Kimono | FC | 82.52 | 86.15 | 82.52 | 146.78 | 158.18 | 101.6 | 114.65 | 122.17 | 92.06 |
| | BMA-FS | 64.59 | 67.43 | 64.59 | 114.89 | 123.81 | 79.53 | 89.74 | 95.63 | 72.06 |
| | Proposed-TZS | 0.2121 | 0.2173 | 0.2121 | 0.2143 | 0.2114 | 0.2097 | 0.2132 | 0.2143 | 0.2109 |
| ParkScene | FC | 84.88 | 85.01 | 83.39 | 147.33 | 155.23 | 110.43 | 109.23 | 125.88 | 100.52 |
| | BMA-FS | 62.73 | 63.11 | 64.84 | 139.66 | 119.7 | 80.39 | 91.88 | 93.06 | 70.56 |
| | Proposed-TZS | 0.2109 | 0.2045 | 0.2111 | 0.2211 | 0.2009 | 0.2056 | 0.2101 | 0.2078 | 0.2177 |

Table 4.6: Average computational time (minutes) of test video sequences

As shown in Table 4.6, on average, the SEC scheme requires approximately 99.75% to 99.76% less time under a 1% PLR, 99.76% to 99.85% less time under a 3% PLR and 99.75% to 99.82% less time under a 5% PLR, compared to the original TZS and the BMA approaches. Hence, the SEC scheme performs better in terms of both the average PSNRs and the computational times. The computational time remains a major target in the SEC scheme, as the main motive is to support the real-time processing of SHVC video streams. We cannot expect high computational power and unlimited memory at an end-user's mobile device. Delays are always critical in real-time applications, so light-weight processing and requirement for less hardware resources are recommended.

For a visual comparison, we randomly select three different video sequences from the test videos dataset. The visual results of the referenced and the SEC schemes are depicted

in Fig. 4.7 under different QPs and PLRs.

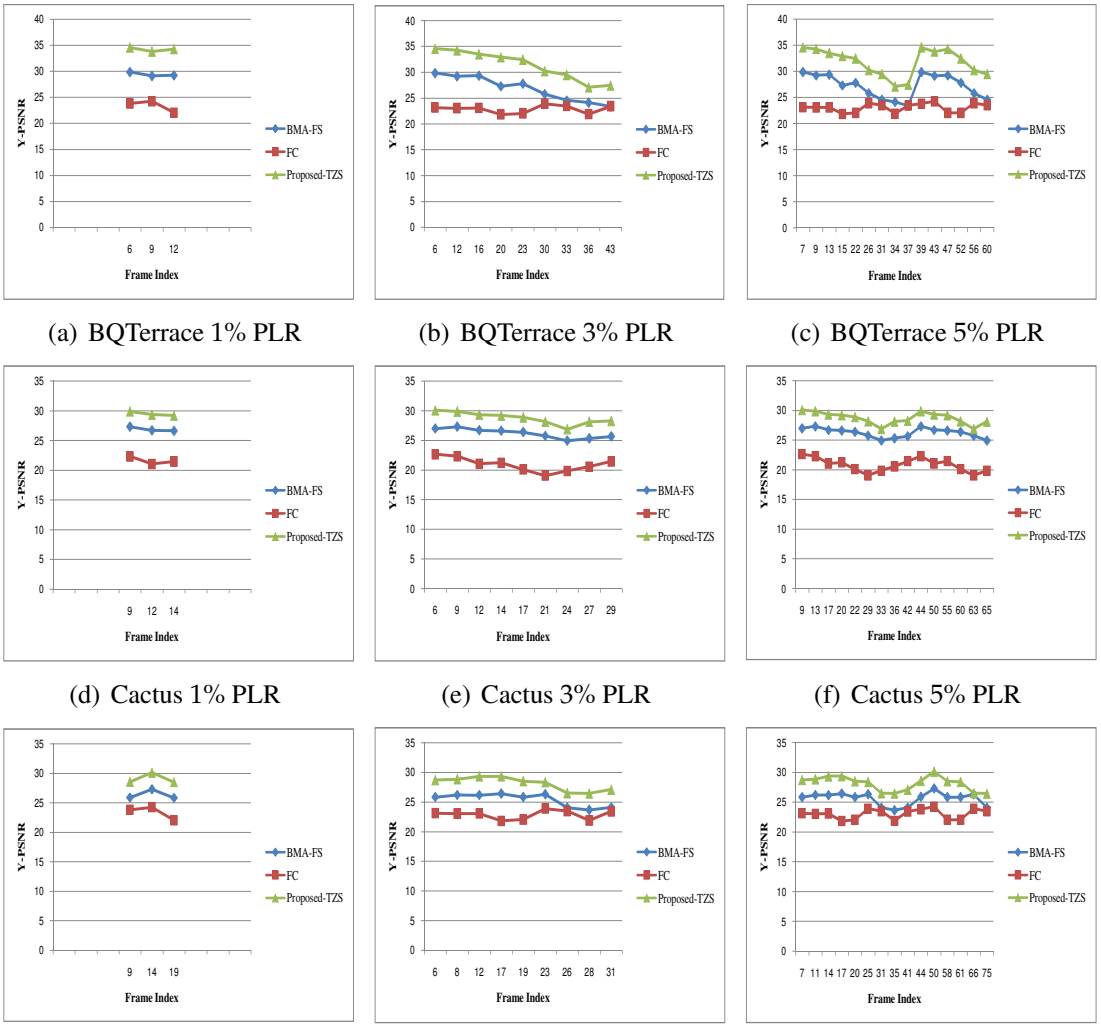
(a) Original (b) FC (c) BMA (d) Proposed

Figure 4.7: Visual comparisons

In Fig. 4.7, first column (from left to right) represents the original video frame lost during a transmission and second, third and fourth columns represent the outputs of the FC, the BMA and the SEC schemes, respectively. The sample images are taken from the test video sequences, having different types of motions and are encoded with different QPs. The first sample is taken from the video sequence BQTerrace with QP pair $(20, 22)$, the second one is taken from the video sequence Cactus with QP pair $(28, 30)$ and the third one is taken from the video sequence Kimono with QP pair $(36, 38)$. These comparisons

show clearly that there is not much visible difference between the original video frame and the outputs of the FC and the SEC schemes. However, blurred lines are quite visible in the outputs of the BMA technique. In the consecutive video frames, the motion of objects is usually slow and the FC does not show noticeable differences. However, in the experiments with higher PLRs, visible fluctuations and video freezing effects are experienced when the FC technique is applied. Because of their large sizes, the videos showing the effects are not possible to be uploaded in this paper to show the effects of errors, so we have included only the specific video frames instead.

The average PSNRs and the average computational times, under different QP pairs, of the three video sequences are plotted in Fig. 4.8 and Fig. 4.9, respectively. Begin from the left, the three columns in the sequential order, represent 1%, 3% and 5% PLRs, respectively. It is clear, in both figures, that the SEC scheme outperforms both FC and the BMA techniques in terms of average PSNRs and computational times. The significantly smaller computational times of the proposed SEC proves its suitability for real-time processing of SHVC video streams.



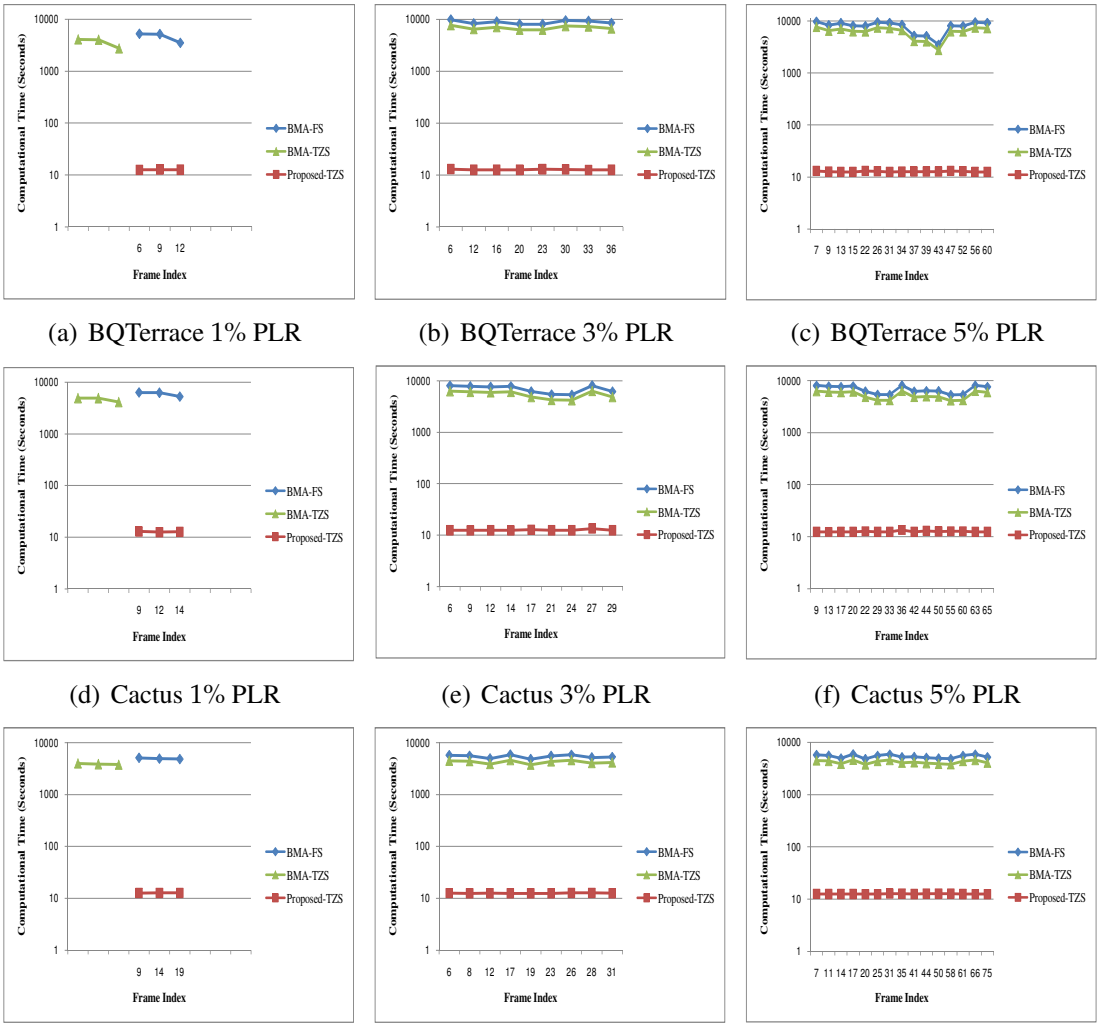

(a) BQTerrace 1% PLR (b) BQTerrace 3% PLR (c) BQTerrace 5% PLR

(d) Cactus 1% PLR (e) Cactus 3% PLR (f) Cactus 5% PLR

(g) Kimono 1% PLR (h) Kimono 3% PLR (i) Kimono 5% PLR

Figure 4.8: PSNR comparison

Our results in Fig. 4.7 and Fig. 4.8 demonstrate that, by applying the proposed SEC scheme, we can greatly enhance the QoE of the users. The subjective gains are shown in Fig. 4.7 and the objective gains are shown in Fig. 4.8 and Fig. 4.9 in terms of PSNRs and computational costs using the proposed SEC scheme. They demonstrate that the SEC scheme not only achieves much better quality but also has much lower computational costs.




(a) BQTerrace 1% PLR (b) BQTerrace 3% PLR (c) BQTerrace 5% PLR

(d) Cactus 1% PLR (e) Cactus 3% PLR (f) Cactus 5% PLR

(g) Kimono 1% PLR (h) Kimono 3% PLR (i) Kimono 5% PLR

Figure 4.9: Average computational time

### 4.3.2 Computational Complexity

The execution time of the proposed SEC scheme is much less than the standard classic BMA technique. It is demonstrated, in Table 4.6 and Fig. 4.9, that our proposed SEC scheme achieves approximately several hundred times faster speed than the standard BMA technique. Note that the values quoted demonstrate only the 'relative' reduction in terms of computation costs. The experiments are performed on a computer using Matlab, and not

on a DSP/DIP processor using C or other hardware languages. In real applications with optimisation, the computational costs can be in micro or nano seconds and the information of implementing the video algorithms on a hardware kit with timing details can be found in [156,157]. Therefore, from the mobile devices point of view, the overhead caused by the computational costs is imperceptible as the proposed scheme just involves implementing a motion estimation algorithm and an adaptive filter when it receives an alert for a missing video frame. Modern mobile devices are easily able to encode and decode videos at a speed up to 60fps, so adoption of the proposed scheme is not an issue for them at all. It will be our future work to provide those details of implementing this proposed scheme (whose concepts are proved and verified by the experiments in this paper) on an embedded kit and code in a low level programming language, such as C.

Let $m \times m$ be the block size in the SEC scheme. Then, the time complexity of SAD is $O(m^2)$, as stated in [125, 158]. Note that, in a BMA based on FS or TZS. TZS is a combination of a diamond search and a raster search. The raster search goes through every pixel. Furthermore, each pixel in an RB is searched and the RB and the IB have the same size. Therefore, in the worst case, the computational complexity of raster search is $m^2$, and the computation complexity of BMA is O($m^2 \times m^2$), i.e., $O(m^4)$.

On the other hand, in the SEC scheme, the search does not go till the end of an RB. The search ends in the first few steps once the nearest match is found, so the computational complexity of the searching process in the SEC scheme is $O(p)$, where $p$ represents the total number of unit steps and $p < m^2$.Furthermore, the matching process based on SAD in the SEC scheme is performed in parallel in rows using multiple threads, so the the time complexity of performing SAD in the SEC is $O(m)$. Therefore, the computation complexity of SEC is $O(p \times m)$.

## 4.4 Summary

In this chapter, we have used the frame interpolation algorithm of Chapter 3 to design a frame interpolation algorithm for SHVC encoded videos. The main motive is to design an efficient and lightweight EC algorithm to conceal missing video frames in scalable encoded videos. In the scalable coding domain, the reference video frame can be from either base or enhancement layer. Therefore, the first step in the EC algorithm is to find the reference frame for the current video frame. After finding the reference video frame, both current and reference video frames are divided into equal size blocks, known as $IB$s and $SW$s, respectively. Best match for each $IB$ is searched in the corresponding $SW$ using a linear search. The searching process is threshold-based and is performed in a parallel style to save the computational time. After finding the best match, MVs of missing video frame are estimated using MVE. Next step is to utilise the estimated MVs to interpolate the missing video frame. Due to the estimation, it is possible that the concealed video frame might contain thin-hair like lines in various regions. To remove the effect of thin-hair like lines, an adaptive filter is applied. Unlike the existing EC approaches, our algorithm does not require complex hardware and can produce better reconstruction result in real-time.

Chapter 5

# High Definition Video Streaming over Wireless Multimedia Sensor Network

## 5.1 Introduction

With the emergence of WMSNs, the distribution of multimedia contents have now become a reality. In the olden days, a WMSN usually includes nodes of small cameras and battery sources, and also contain low-power transceivers to transmit and receive multimedia data, such as images, audio, and videos [159]. Recently, other types of nodes, such as Lotus motes, perform much better and have superior hardware configurations as shown in Table 5.1.

| Generation | Microcontroller | Frequency | Tranceiver | Memory |
|---|---|---|---|---|
| Old (TelosB) | Atmega128/ARM7 | 8-48 MHz | ChipCon CC2420 | 4-10 KB RAM, 512 KB Flash |
| New (Lotus Mote) | Cortex M3 | 10-100 MHz | 802.15.4 Radio Antenna | 64 KB SRAM, 512 KB Flash, 64 MB Serial Flash |

Table 5.1: Generations of sensors

A WMSN usually consists of a variety of sensor nodes, including both simple and multimedia sensor nodes, and a BS as shown in Fig. 5.1. Unlike a traditional multimedia transmission system, a WMSN faces many challenges in various aspects, such as bandwidth, computation, storage and energy. Moreover, a WMSN uses a many-to-one communication paradigm, i.e., the upstream data traffic from multimedia sensor nodes is routed to a single entity, also known as sink.

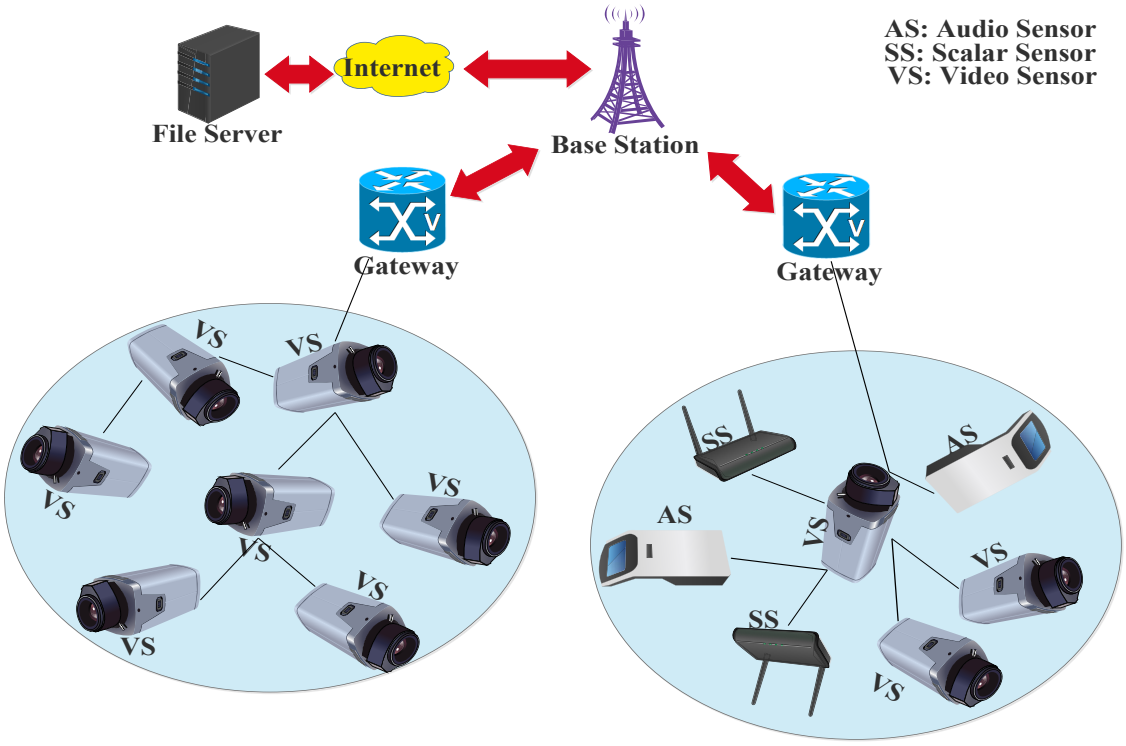



Figure 5.1: Traditional wireless multimedia sensor network

The situation gets worse with an increase in the density of multimedia sensor nodes because they produce a huge volume of raw multimedia data resulting in an excessive bandwidth consumption and energy utilisation. Without proper management, the transmission of multimedia data over WMSNs affects the performance of networks due to excessive packet drops. In recent years, various studies have been conducted for real-time multimedia data compression and transmission [160–162]. However, these studies do not

take into account the facts, such as real-time processing, QoS, and energy-efficient video transmission over WMSNs. Due to the limited available bandwidth, multimedia data, such as videos, need to be highly compressed. Higher compression makes it possible to encapsulate an entire video frame in one data packet [163]. Under this scenario, dropping of a single data packet results in a drop of an entire video frame. In sensor networks, burst packet losses frequently happen and hence produce significant quality degradation in real-time video transmission. As a result, there is a need to develop a real-time and an energy-efficient EC scheme to estimate lost or corrupted video frames.

In this chapter, we propose a novel framework to support QoS along with a lightweight EC scheme for scalable HD video streaming over WMSN. The EC scheme plays a vital role to enhance QoE by maintaining an acceptable quality at the receiving ends. Existing studies focus on QoS and end-to-end delay with limited number of data generating sources, and ignore the load balancing during transmission and packet drop effects on the transmitted data. In contrast, our approach addresses all these issues by adopting multi-hop scalable video transmission to efficiently utilise available network resources. The main objectives of the proposed framework are to maximise the network throughput and to release the effects produced by dropped video packets. To control the data rate, SVC is applied at multimedia sensor nodes with variable QPs. Multi-path routing is exploited to support real-time video transmission.

In Section 5.2, we discuss the proposed streaming system. In Section 5.3, we provide a detailed explanation of our proposed framework followed by the experimental set-up and performance evaluation in Section 5.4.

## 5.2 Streaming System

A WMSN usually consists of $K$ Multimedia Sensor Nodes (MSNs) and a BS which is used to monitor and collect data from these nodes for further processing and analysis. In our proposed approach, we have made the following assumptions.

1. MSNs and the BS have prior knowledge about the location of each other.
2. MSNs can cover a geographical range $R$ with an omni-directional antenna.

The density $d$ of MSNs in a particular geographical region $R$ with a radius $r$ can be calculated using Eq. 5.1.

$$d = \frac{\pi r^2}{RK}. \tag{5.1}$$

### 5.2.1 Video Encoding Model

In a dense geographical region, it is highly probable that a single event may be captured by more than one MSN at different angles which leads to the following drawbacks.

1. Energy consumption of MSNs for the same shot of a video.
2. Bandwidth consumption during the transmission of the same event.
3. Processing load, memory consumption and data redundancy on the BS.

To overcome these drawbacks, MSNs encode captured videos using different Quantisation Parameters (QPs). In our video encoding model, MSNs encode captured videos using an SHVC encoder with mixed configuration settings in the default layer mode. In this mode, SHVC generates two layers, i.e., a Base Layer (BL) and an Enhancement Layer (EL). A BL consists of Intra (I) frames while an EL consists of Predicted (P) frames. In the start, nodes will encode first Group of Pictures (GoP), i.e., first 10 video frames, as a BL and an EL, send it to the BS and wait for its response. As soon as the BS receives the first BL from one of the nodes in $R$, it stops receiving BLs from the remaining nodes in $R$ by broadcasting a stop message. After this step, BS starts receiving the BL from a single node while the remaining nodes switch to the processing and transmission of low-resolution EL frames. This step not only saves the bandwidth but also reduces the computational load on the BS and MSNs.

### 5.2.2 Video Streaming Model

In WMSNs, similar to any WSN, reliability, latency and bandwidth are always considered as highly critical factors. In our proposed approach, we adopt a multi-path and best-effort routing scheme to carry the video traffic. It is probable that a node may not directly approach the BS. As a result, intermediate nodes act as relay nodes to carry the video data. Since our approach uses a best-effort delivery, there is neither a fixed path nor an acknowledgement between senders and receivers. To ensure a best-effort delivery, the transmission region between sender nodes and the BS is divided into $n$ interference-free paths as shown in Fig. 5.2.

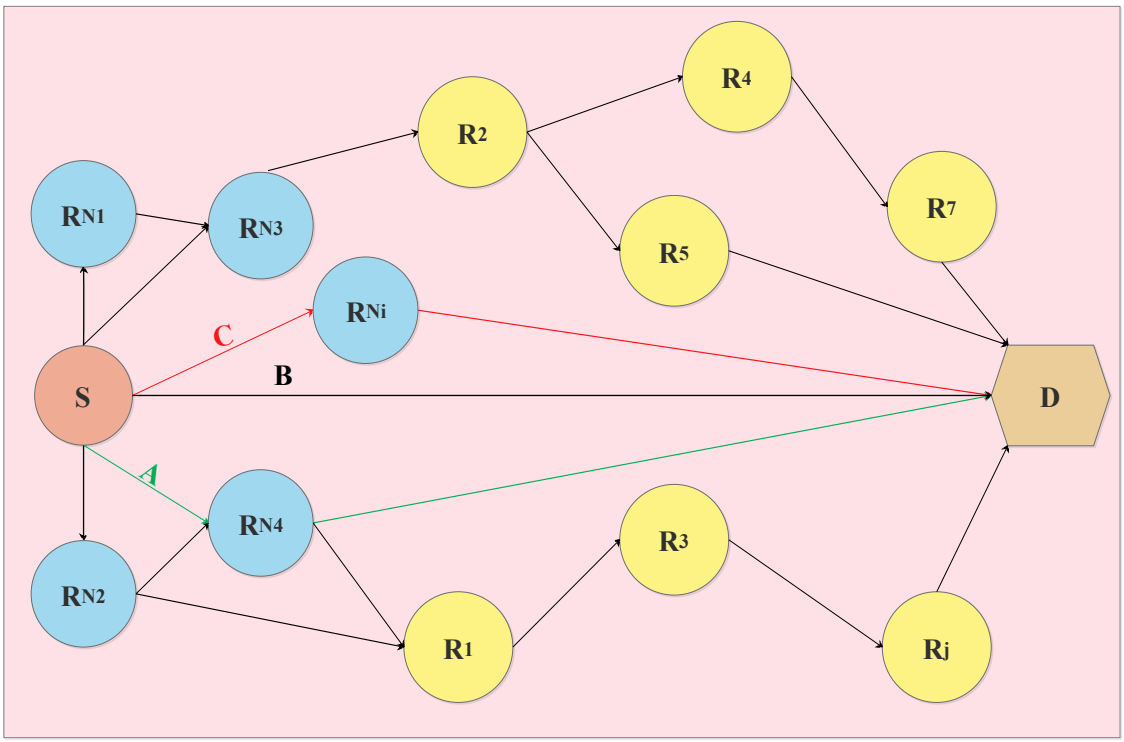



Figure 5.2: Multipath forwarding

The sender node is denoted by $S$, the BS is denoted by $D$, the nearest relay nodes are denoted by $R_{N_i}$ and the remaining relays are denoted by $R_j$. In an off-line multi-path routing, the paths are calculated first before initiating any video streaming. A drawback

of this approach is the amount of information travelling over the network which consumes a considerable amount of bandwidth. It also demands memory space to store information about every node in the network which is mostly useless and redundant. Our proposed multi-path scheme follows three rules.

1. Select the nearest neighbour nodes for relay-based transmission.
2. Follow the greedy transmission policy to balance the network load in a multi-path transmission.
3. Avoid network loops and holes.

In the greedy transmission policy, a source generates data at the maximum possible data rate and transmits at the earliest possible opportunity. In routing schemes, both data-rate and end-to-end delay play an important role for measuring reliability, latency and available bandwidth of the underlying transmission medium. In the case of best-effort delivery, per relay latency can help to estimate the end-to-end delay. This delay can be computed by summing up the latency of each relay utilised for transmission between the source nodes and the BS. To maintain QoS, enhanced channel distributed access is utilised in IEEE 802.11e standard to get the targeted reliability [164]. MSNs can obtain the available bandwidth information from the access points.

### 5.2.3 Network Impairment Model

Three types of distortions that affect the video quality in a transmission network are, QPs used during the encoding process, packet-loss during a transmission and false decoding. QPs not only control the video quality but the bit-rate as well. There are various reasons for packet-loss, such as buffer overloading, power failure and false routing. False decoding occurs when the decoder on the receiving side either receives incorrect decoding information in a video packet or a video packet has been modified during a transmission. Either of these factors can crash the decoder. In our proposed scheme, distortion occurs mainly due to a packet-loss. To maintain an acceptable level of received video quality, a threshold is

set on the receiver side. If a video packet does not arrive or arrives out of time, the decoder conceals that specific video packet by using a suitable EC technique rather than getting crashed.

### 5.2.4 Problem Description

Let $S$ be a set of selected MSNs and $k$ be the size of $S$, where $0 \leq k \leq K$. A node can play the role of a relay if it has an ample amount of available bandwidth. An ideal situation will be the availability of unlimited bandwidth to a node although it seems unrealistic. Let $S_i \in S$ be the $i$-th MSN which requests its nearest relay node for a data transfer. The data consists of a set of video frames denoted by $V$, encoded at a bit-rate of $b_{i,t}$ at time $t$. There can be two situations. The relay node is dealing with only one data transfer request at the time or more than one data transfer request at the same time. In the first situation, the relay node is always available and the data transfer request can be made any time. In the second situation, the relay node can be either busy or facing a bandwidth limitation issue. In this situation, the status of the relay node needs to be determined before making a data transfer request, Otherwise, the relay node can reject the data transfer request. Whenever the data transfer request of $S_i$ is rejected by the relay node at time $t$ due to a bandwidth limitation, $S_i$ follows two steps. First, it reduces the number of video frames in $V$ in order to reduce the bit rate. Second, it determines the current status of the relay node. If the current status of the relay node can support the reduced bit rate, $S_i$ sends the data transfer request again at time $t+1$, Otherwise, $S_i$ further reduces the bit rate by reducing the number of frames in $V$. If the bit-rate of a new set of video frames is denoted by $b_{i,t+1}$ at time $t+1$, the data delay by $q$, the packet-loss by $l$, the packet-loss threshold by $T$, and the relay node by $S_j$ ($\in S$), then the status of a relay node, denoted by $S_j^{sta}$, for a data transfer request can be determined using Eq. 5.2.

$$S_j^{sta} = \sum_{0 \le i \le k, 0 \le j \le K} (\, b_{i,t} - b_{i,t+1}) \; + w_i \\ -(\, b_{j,t+1}^{in} - b_{j,t+1}) \tag{5.2}$$

Here, $w_i$ is the bandwidth reserved by $S_i$, $b_{j,t+1}^{in}$ is the total bit-rate of an incoming data at $S_j$ and $b_{j,t+1}$ is the transmission bit-rate of $S_j$ at time $t+1$. This status of a relay node $S_j^{sta}$ is subject to the following constraints of Eq. 5.3-5.6.

$$g_i \le T_g, \quad where \quad i \in K, \tag{5.3}$$

$$(b_{j,t+1}^{in} + b_{i,t+1}) \le b_{j,t+1}, \tag{5.4}$$

$$q_j \le q_i, \tag{5.5}$$

$$(l_i + l_j) \le T. \tag{5.6}$$

where $g_i$ is the distortion faced by data produced by $S_i$, $T_g$ is the predefined threshold for data distortion, $q_j$ is the amount of delay happening at $S_j$, $q_i$ is the amount of delay set at $S_i$, $l_i$ is the amount of packet losses set at $S_i$ and $l_j$ is the amount of packet losses occurring at $S_j$.

Eq. 5.3 ensures that $g_i$ should not exceed $T_g$ to maintain an acceptable quality of received video data. Eq. 5.4 ensures that the bit-rate of an incoming data (i.e., $b_{j,t+1}^{in}$) aggregated with the data forwarded by $S_i$ (i.e., $b_{i,t+1}$) should not exceed the bit-rate of $S_j$ (i.e., $b_{j,t+1}$) at the time $t+1$. Eq. 5.5 is used to ensure that $q_j$ should not exceed $q_i$. Eq. 5.6 guarantees that $l_i$ and $l_j$ should not exceed a predefined threshold $T$.

## 5.3 Video Streaming and Error Concealment Framework

Our proposed Video Streaming and Error Concealment (VSEC) framework is a cross-layered framework as shown in Fig. 5.3. In the VSEC framework, videos are encoded using SHVC standard and are streamed in a multi-hop scenario. SHVC standard based video encoding helps in generating multiple video streams with variable bit-rates. In the multi-hop streaming, feedback messages are used to improve the end-to-end QoS and to provide a smooth video streaming. After receiving the video data, BS checks whether all the video frames of a video are arrived successfully or not. If some video frames are missing, then an efficient and light-weight Scalable EC (SEC) scheme is applied to recover the missing video frames.

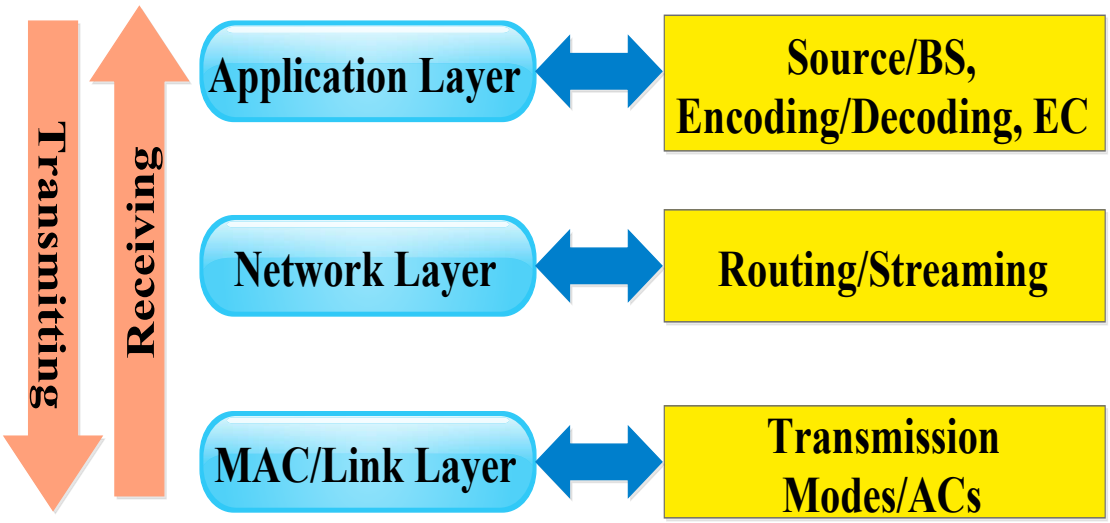



Figure 5.3: Cross-layer architecture

### 5.3.1 Scalable High Efficiency Video Coding

SHVC is adopted in the VSEC framework to perform encoding and decoding of HD videos. To control the bit-rate, each layer of an encoded video stream needs to be handled carefully according to the status of the available bandwidth.

#### 5.3.1.1 Reducing Frames per Retransmission

For a retransmission, we need to assume certain parameters to better describe the reduction of video frames in each retransmission. In any layer $x$ of a scalable encoded video, where $x$ can be either $intra$ or $inter$ representing the layer of intra- or inter-frames, if the new set of video frames is represented by $V^n$, the length of the new set of video frames by $L_{V^n,x}$, the video packet size in bits by $p$, then the total number of packets ($T_p$) required to transmit $V^n$ is calculated using Eq. 5.7.

$$T_p = \frac{L_{V^n,x}}{p}. \tag{5.7}$$

Each time a transfer request is rejected by a relay node, a counter is incremented on the sender node. By default, this counter is set to zero. A threshold is set for the retransmission requests to reduce the number of frames in $V$. In the VSEC framework, this threshold is set to 3. When the value of the counter is equal to the value of the threshold, the sender node selects another relay for a transmission along with signalling the BS about the status of the relay node through an error message. The signalling is done once the data packet is successfully arrived on BS and a guaranteed transmission path is established. The signal packet contains the status of the relay nodes involved during a transmission. These error messages help in troubleshooting the overloaded nodes. The probability ($w^s$) of the current status of the available bandwidth of a relay node can be estimated using Eq. 5.8.

$$w^s = \frac{1}{T_p} \sum_{i=1}^{T_p} \Upsilon(i|s_c). \tag{5.8}$$

Here, $\Upsilon(i|s_c)$ represents the probability of the status of a transmission channel for a video packet $i$.

Based on the approach presented in [165], the total bit-rate of an entire $V$ or $V^n$ at time $t$ can be computed using Eq. 5.9.

$$b_{i,t} = L_{V,x} \times p. \tag{5.9}$$

#### 5.3.1.2 Controlling Coding Complexity

Based on the available bandwidth and a predefined bit-rate, the coding complexity (i.e., $\theta$) of any layer $x$ in a scalable coded video can be computed using Eq. 5.10-5.12 [166].

$$\theta = \alpha_x \times \beta_x \times \gamma_x, \tag{5.10}$$

$$\begin{aligned} \beta_x = & \sum_{i=1}^{R_Y} \sum_{j=1}^{C_Y} \frac{(|Y_{i,j} - Y_{i,j+1}| + |Y_{i,j} - Y_{i+1,j}|)}{C_Y \times R_Y} \\ & + \sum_{i=1}^{R_U} \sum_{j=1}^{C_U} \frac{|U_{i,j} - U_{i,j+1}| + |U_{i,j} - U_{i+1,j}|}{C_U \times R_U} \\ & + \sum_{i=1}^{R_V} \sum_{j=1}^{C_V} \frac{|V_{i,j} - V_{i,j+1}| + |V_{i,j} - V_{i+1,j}|}{C_V \times R_V}, \end{aligned} \tag{5.11}$$

$$\gamma_x = \sum_{i=0}^{255} (log_2 \gamma_Y[i] + log_2 \gamma_U[i] + log_2 \gamma_V[i]). \tag{5.12}$$

Here, $x$ represents the layer ID, $\alpha_x$ represents the coding complexity of current/reference I-frame, $\beta_x$ and $\gamma_x$ represent the gradient and the histogram information of a video frame, $Y_{i,j}$ represents the luminance and $U_{i,j}$ and $V_{i,j}$ represent the chrominance values at any pixel $(i, j)$. Furthermore, $(R_Y, C_Y)$, $(R_U, C_U)$ and $(R_V, C_V)$ are the total numbers of rows and columns in $Y = [Y_{i,j}]$, $U = [U_{i,j}]$ and $V = [V_{i,j}]$ components, respectively, $\gamma_Y[i]$ represents the histogram of a luminance level $i$ and $\gamma_U[i]$ and $\gamma_V[i]$ represent the histograms of chrominance level $i$, correspondingly.

#### 5.3.1.3 Selection of Quantisation Parameter for Intra Frames

Based on the total bit-rate of relay nodes, the weighted average bit count ($b_{intra}$) for I-frames can be computed using Eq. 5.13-5.14.

$$b_{intra} = T_b \times \frac{I_w}{I_w + P_w \times P_n}, \tag{5.13}$$

$$T_b = \frac{(1 + P_n) \times T_{tb}}{r_f}. \tag{5.14}$$

Here, $P_n$ indicates the total number of P-frames dependent on an I-frame, $I_w$ and $P_w$ are weights set by a scalable encoder for I and P-frames, respectively, $T_b$ is the total bit-rate for I-frames, $T_{tb}$ is the total targeted bit-rate for I-frames and $r_f$ is the frame rate.

To avoid buffer overloading during a retransmission, the total number of retransmission bits needs to be carefully calculated. Using Eq. 5.9 and Eq. 5.13, the total number of retransmission bits ($T_{rb}$) can be computed as shown in Eq. 5.15.

$$T_{rb} = b_{intra} - b_{i,t}. \tag{5.15}$$

The Quantisation Parameter for I-frames ($QP_{intra}$) can be obtained using Eq. 5.16.

$$QP_{intra} = ln(\frac{T_{rb}}{\alpha_{intra} \times \beta_{intra} \times \gamma_{intra}}). \tag{5.16}$$

#### 5.3.1.4 Selection of Quantisation Parameter for Inter Frames

If a retransmission is performed, then the bit-rate needs to be reduced for each P-frame. Here, the QP plays an important role in reducing the bit-rate of each P-frame. The QP for P-frames (i.e., $QP_{inter}$) can be computed by modifying Eq. 5.10-5.16, i.e., by replacing I-frames with P-frames. The weighted average bit count ($b_{inter}$) for P-frames can be

computed using Eq. 5.17-5.18.

$$b_{inter} = \bar{T}_b \times \frac{I_w + P_w}{I_w + P_w \times P_n}, \tag{5.17}$$

$$\bar{T}_b = \frac{(1 + P_n) \times \bar{T}_{tb}}{r_f}. \tag{5.18}$$

Here, $\bar{T}_b$ is the total bit-rate for P-frames and $\bar{T}_{tb}$ is the total targeted bit-rate for P-frames.

The total number of retransmission bits ($\bar{T}_{rb}$) for P-frames can be computed as shown in Eq. 5.19.

$$\bar{T}_{rb} = b_{inter} - b_{i,t}. \tag{5.19}$$

The Quantisation Parameter for P-frames ($QP_{inter}$) can be obtained using Eq. 5.20.

$$QP_{inter} = ln( \frac{\bar{T}_{rb}}{\alpha_{inter} \times \beta_{inter} \times \gamma_{inter}}). \tag{5.20}$$

### 5.3.2 Scalable Video Streaming

In the VSEC framework, we have proposed a multi-path routing concept for an efficient video streaming over WMSNs. In this scheme, a sender node determines $n$ disjointed paths towards BS as shown in Fig. 5.2. The path $B$ is a direct and shortest path between the sender and the BS. In the case of unavailability of a direct path, an indirect path is selected based on the selection of next hop at a distance $A$. This indirect path is known as Indirect Path from Sender (IPS). In the case of IPS, paths $A$ and $C$ are the two nearest paths. Next, we explain the criterion for selecting the next hop in order to form an optimum IPS along with counting total number of hops in an IPS, traffic control and probability of packet-drop to maintain the QoS.

#### 5.3.2.1 Selection of Next Hop

Due to a random deployment of MSNs, it is not necessary that a relay node must be at a shorter distance. It is more likely that a sender node plays a dual role, i.e., a sender and a relay for another sender. A sender node $S_i$ calculates distances to all the possible nearby relay nodes and selects the one ($S_j$) with a minimum distance $\Delta d_{i,j}$ from $S_i$ and $S_j$ is the nearest relay node. Let us assume that the positions of the sender, the relay and the BS are $(x_i, y_i)$, $(x_j, y_j)$ and $(x_k, y_k)$, respectively. $\Delta d_{i,j}$ between $S_i$ and $S_j$ is computed using Equations 5.21-5.23 [167].

$$\Delta d_{i,j} = \sqrt{(x - x_j)^2 + (y - y_j)^2}, \tag{5.21}$$

$$x = x_i + c(x_k - x_i), \quad y = y_i + c(y_k - y_i), \tag{5.22}$$

$$c = \frac{(x_j - x_i)(x_k - x_i) + (y_j - y_i)(y_k - y_i)}{(x_k - x_i)^2 + (y_k - y_i)^2}. \tag{5.23}$$

A node $S_j$ with a minimum value of $\Delta d_{i,j}$ is selected as a relay node. Thus, a sender node exploits the distance information of its neighbouring nodes to perform an efficient and shortest distance routing.

#### 5.3.2.2 Path Length/Maximum Hop Count

Our proposed routing scheme ensures a minimum hop count between senders and the BS in a densely deployed environment to support QoS. Let us assume that a video packet travels an $h$ unit distance from source node $S_i$ to the base station $B$. In Eq. 5.18, $\Delta d_{(i,j)}$ represents the optimum displacement from source node $S_i$ to next hop $S_j$. Similarly, the optimum displacement between source node $S_i$ and the base station (BS) can be defined by $\Delta d_{i,k}$ that is obtained by simply replacing $j$ with $k$ in Eq. 5.18. The total number of

hops between source $S_i$ and the base station (BS) can be obtained by $\frac{\Delta d_{i,k}}{h}$. This condition only applies to indirect paths between sources and the base station (BS). In the case of an indirect path and a densely deployed environment, a shortest path is the one where an $h$ unit distance covers only one hop between senders and the base station. The expected route length (i.e., $E$) between sender $S_i$ and the base station (BS) under constraints, such as node density $d$, multiple hops along the path with a minimum $h$ unit distance and accuracy probability $\eta$ under uniform nodes distribution, can be estimated using Eq. 5.24-5.25.

$$E = \eta \frac{\Delta d_{i,k}}{h}, \tag{5.24}$$

$$\eta = 1 - \frac{4\Delta d_{i,k}^2}{d(\Delta d_{i,k}^2 - h^2)}. \tag{5.25}$$

where $E$ and $\eta$ are based on the model proposed in [167] and $d$ is computed from Eq. 5.1.

#### 5.3.2.3 Traffic Control

To maintain a smooth QoS between senders and BS, data traffic constraint needs to be taken into account. The data traffic between a sender $S_i$ and a relay node $S_j$ at time $t$ can be controlled using Eq. 5.26-5.27 (assuming that there are three paths $A$, $B$ and $C$ from $S_i$ and $S_j$ as shown in Fig. 5.2).

$$b_{j,t}^{in} = b_{ij,t}^{A} + b_{ij,t}^{B} + b_{ij,t}^{C}, \tag{5.26}$$

$$b_{j,t+1}^{in} = \vartheta_{i,j}(b_{j,t}^{in}). \tag{5.27}$$

Here, $b_{j,t}^{in}$ is the total incoming bit-rates on relay node $S_j$ from sender $S_i$; $b_{ij,t}^{A}$, $b_{ij,t}^{B}$ and $b_{ij,t}^{C}$ are the maximum amounts of data traffic sent from sender node $S_i$ to the relay node $S_j$ through transmission paths $A$, $B$ and $C$, respectively, at time; $t$. $b_{j,t+1}^{in}$ is the estimated

incoming bit-rate on relay node $S_j$ at time $t+1$. $\vartheta_{i,j}$ is a control function which helps in estimating the amount of incoming video traffic. It enables a relay node to avoid buffer overflow and congestion, and is calculated using Eq. 5.28.

$$\vartheta_{i,j}(b^{in}_{j,t}) = \begin{cases} 1, & \text{if} \quad b^{in}_{j,t} \leq b_{j,t} - b_{i,t} \\ 0, & \text{otherwise} \end{cases} \tag{5.28}$$

where $b^{in}_{j,t}$ is the total incoming bit-rate at node $S_j$, $b_{i,t}$ is the transmitted bit-rate of sender node $S_i$ at time $t$ and $b_{j,t}$ is the supported bit-rate of relay node $S_j$ at time $t$.

#### 5.3.2.4 Packet Drop

The Packet Loss Rate (PLR) is calculated by a Markov chain-based model, proposed in [168]. The Markov chains are characterised by a set of probabilities and states, such as probability $\pi_a$ for each steady state $a$, probability $\pi_b$ for each non-steady state $b$ and probability $p_{a,b}$ for transition between states $a$ and $b$. In the VSEC framework, CF also maintains a counter and a state flag for each transmitted video packet from the sender nodes. Here, the state flag can be either 1 or 0. By default, both the counter and the flag are set to 0. If a video packet is sent and arrived successfully on the destination, the flag is set to 1, and this state is known as success steady state $a$. On the other hand, if a video packet is sent successfully but it does not arrive on the targeted destination, then the flag is set to 0, and this state is known as failure steady state $b$. For each failure, the counter is incremented by 1. The steady state probabilities can be computed by using Eq. 5.29.

$$0 \leq \pi_b \leq 1, \quad \sum \pi_a + \sum \pi_b = 1, \quad \pi_b = 1 - \sum \pi_a. \tag{5.29}$$

The overall PLR of the entire video streaming can be computed by summing up the failure state probabilities at different times, i.e.,

$$PLR = \sum \pi_b. \tag{5.30}$$

### 5.3.3 Scalable Error Concealment

In the VSEC framework, missing video frames are recovered using a light-weight and quality-driven SEC scheme, proposed in Chapter 4. One ideal solution is the retransmission technique to recover the missing frames. However, it is infeasible for real-time video streaming over a bandwidth-limited sensor network. In practice, the EC techniques are always executed at the receiver side. In our case, the SEC scheme is executed on the BS to maintain a certain level of QoE. If one EL frame is lost during transmission, it can be recovered through the corresponding BL frame easily. On the other side, the BL frames are not dependent on any other frame(s). If one BL frame is lost during transmission, it brings down the dependent EL frames too. To maintain the link between the dependent EL frames and BL frames, the lost BL frames need to be recovered carefully. If the lost BL frames are improperly recovered, it might bring down the dependent EL frames, and as a result, the QoE of received videos will be compromised. The missing BL video frames and EL video frames are estimated by following the steps, explained in Section 4.2.

## 5.4 Experimental Set-up and Performance Evaluation

In this section, we evaluate the performance of our proposed framework. We explain our framework in the following three subsections, i.e., video processing, video streaming and error concealment.

### 5.4.1 Video Processing

For video encoding and decoding, we use SHVC [40]. Test video sequences are chosen from the SHVC standard documentation [152]. Two different video sequences, BQTerrace and Kimono, are used for simulation purposes in our experiments. The resolution, total

number of frames and frame rate of test video sequences are listed in Table 5.2.

| Sequence | Algorithm | 20% PLR | | | 30% PLR | | |
|---|---|---|---|---|---|---|---|
| | | QP (20,22) | QP (28,30) | QP (36,38) | QP (20,22) | QP (28,30) | QP (36,38) |
| BQTerrace | FC | 23.23 | 24.64 | 25 | 25.99 | 24.09 | 23.77 |
| | BMA | 27.73 | 28.88 | 29.99 | 27.6 | 26.57 | 25.59 |
| | Proposed | 32.34 | 31.71 | 32.23 | 32.78 | 32.23 | 31.99 |
| Kimono | FC | 22.77 | 21.69 | 22.69 | 21.89 | 22.55 | 20.4 |
| | BMA | 23.58 | 24.67 | 24.45 | 23.12 | 24.64 | 23.67 |
| | Proposed | 30.48 | 29.45 | 31.23 | 30.46 | 31.34 | 32.34 |

Table 5.2: Average PSNR for test video sequences

The videos are encoded with a default layer setting, i.e., one BL and one EL. The encoding details in terms of QPs and the corresponding bit-rates, are tabulated in Table 5.3. Although QPs are chosen from SHVC standard documentation [152], they can be changed according to the requirements of the underlying architecture. Videos are encoded in an Unsliced mode, i.e., a data packet contains an entire video frame. In the start, GoP is set to 10, which can gradually be reduced to 1 due to network impairments.

| Sequence | Algorithm | 20% PLR | | | 30% PLR | | |
|---|---|---|---|---|---|---|---|
| | | QP (20,22) | QP (28,30) | QP (36,38) | QP (20,22) | QP (28,30) | QP (36,38) |
| BQTerrace | BMA | 108.89 | 122.25 | 121.53 | 177.71 | 165.39 | 127.72 |
| | Proposed | 0.2057 | 0.2109 | 0.2000 | 0.2148 | 0.2000 | 0.2148 |
| Kimono | BMA | 108.15 | 123.69 | 125.05 | 199.72 | 164.45 | 132.05 |
| | Proposed | 0.2092 | 0.2037 | 0.2093 | 0.2116 | 0.2057 | 0.2037 |

Table 5.3: Average computational time (minutes) of test video sequences

#### 5.4.1.1 Video Streaming

For video streaming and network simulation, we use NS-3 [169]. In our experiments, 200 MSNs are randomly deployed in an area of 300m $\times$300m. Network topology is based on IEEE 802.11e standard. The BS is located outside the sensor field to handle the traffic

generated by MSNs. To avoid the energy hole problem [170], multiple paths for data traffic exist within the network. The existence of multiple paths not only avoids energy holes but also reduces the burden on the gateway nodes, i.e., one-hop neighbours of the BS. The default value for a processing delay is set to 2ms. Total number of sources generating the data can be either fixed or variable. In our simulations, they are fixed. Each intermediate node may act as a stand-alone relay or data-oriented relay. A stand-alone relay does not generate its own data but forwards the data of downstream neighbours. A data-oriented relay aggregates its own data with that of the data from downstream neighbours and forwards it to the BS. The performance of our proposed routing algorithm is tested for both fixed/constant and variable bit-rates.

We execute the simulation for three times to monitor the performance of the network. It is observed that the constant bit-rate produces a much higher PLR compared to the variable bit-rate as shown in Fig. 5.4. This figure shows a comparison in terms of packet-drop between the constant and variable bit-rates for 50 data generating sources. The higher peaks point out higher bit-rates which are due to the transmission of I-frames. Constant bit-rate is the first reason for packet-drop resulting in a much higher drop as compared to a variable bit-rate. An increasing number of relays between the source and the BS increases the end-to-end delay as shown in Fig. 5.5.

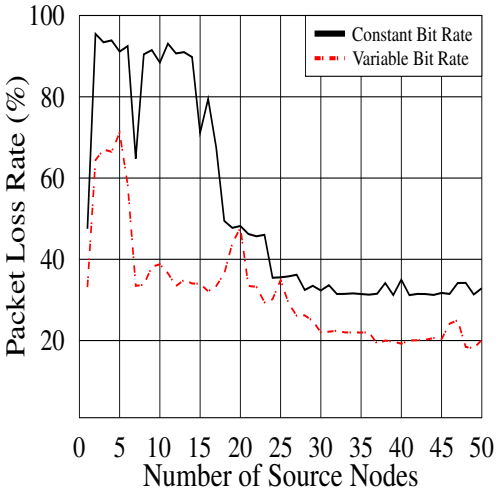



Figure 5.4: Packet loss rate

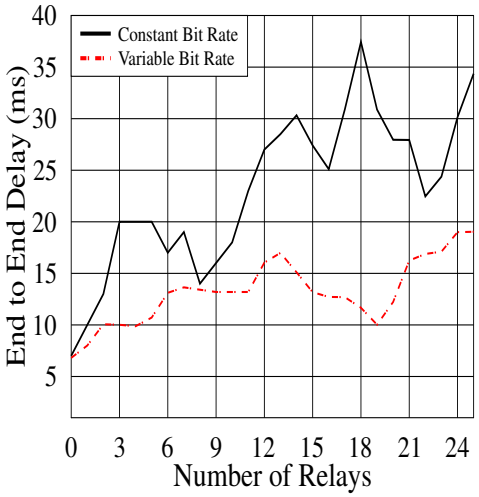



Figure 5.5: End-to-end delay

For most of the cases, the delay increases with an increase in the number of relays for a constant bit-rate. However, in the case of a variable bit-rate, there is little fluctuation in the delay with an increase in the number of relays. This shows that a variable bit-rate scenario is highly suitable for a scalable large-scale MWSN. Higher PLR has a direct relationship with end-to-end delay as shown in Fig. 5.6. In the case of a variable bit-rate, the fluctuation in the bit-rate results in a lower PLR as compared to the fixed rate transmission of the constant bit-rate. End-to-end delay can be considered as a second reason for packet-drop. Higher PLRs and end-to-end delays are the main reasons behind the deterioration of a video quality in any transmission network.

Our proposed scheme follows a multi-path transmission mode, therefore its performance is compared with a single path transmission as shown in Fig. 5.7. The single path transmission misuses the available network resources, increases the end-to-end delay and the PLR and overloads the intermediate nodes during an upstream traffic flow towards the BS. Furthermore, a single path transmission interrupts the network connectivity, i.e., energy hole problem, if one or more intermediate nodes die. This interruption affects the scalability and fault tolerance of the network, and may result in the failure of the whole

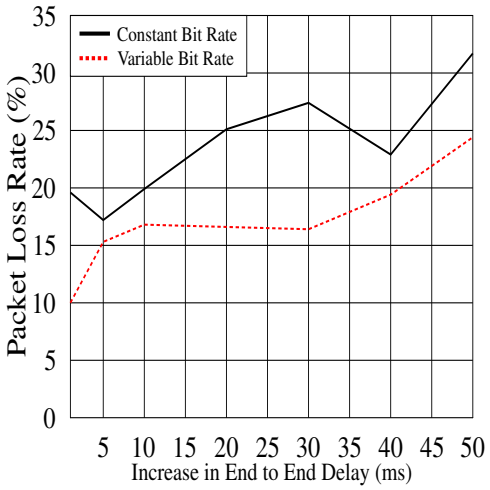



Figure 5.6: End-to-end delay Vs. PLR

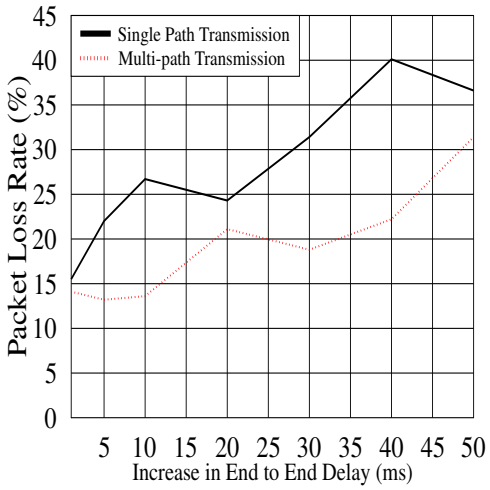



Figure 5.7: Single path Vs. multi-path transmission

network, ultimately causes the degradation of QoS. As a result, single path transmission becomes the third reason for a packet-drop.

Finally, a comparison between the constant and the variable bit-rate scenarios is performed for overhead packets as shown in Fig. 5.8. In the case of variable bit-rate, total number of overhead packets increases with an increase in the number of relay nodes. When the total number of relay nodes is 40 or less, the total number of overhead packets is relatively low. However, there is a sudden jump when the relay nodes increase beyond 40. An increasing number of relay nodes between sources and destinations in a variable bit-rate scenario generates more overhead packets due to packet-drop and service denial packets from relay nodes. To control the amount of overhead packets, sender nodes not only adjust their own bit-rates but also inform the neighbouring nodes about the situation to adjust their bit-rates through update packets. In the case of a constant bit-rate, increasing total number of relays has little effect in terms of overhead packets, i.e., the constant bit-rate incurs fewer overhead packets with an increase in the number of relay nodes. In the case of packet-drop and denial of service from relay nodes, source nodes do not inform their

neighbor nodes and retransmit the data packets on either the same or an alternate path.

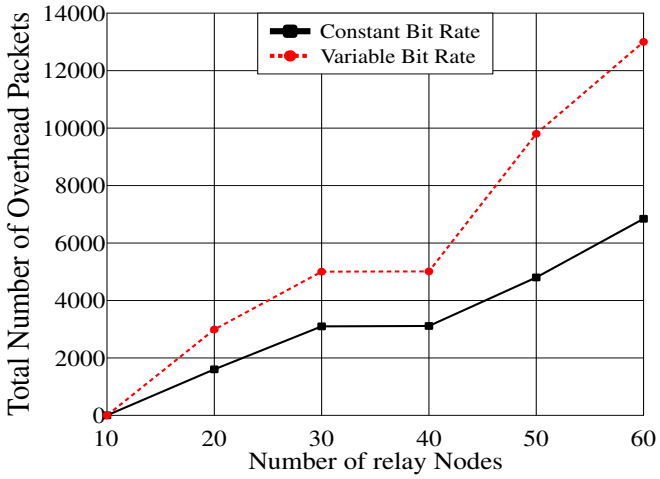



Figure 5.8: Overhead packets Vs. relay nodes

#### 5.4.1.2 Error Concealment

The EC algorithm is implemented in Matlab2016a [171]. To compare the performance of the proposed EC algorithm, we implement popular algorithms, such as the Frame Copy (FC) [153] and the classic Block Matching Algorithm (BMA) [154] under the same experimental settings. The simulation results under different values of QPs, in terms of average PSNR are shown in Table 5.2. Mean square error metric is used to estimate the PSNR. Average computational time for proposed and the BMA techniques under different QPs is tabulated in Table 5.3. Our proposed EC algorithm outperforms the FC and the classic BMA techniques in average PSNR, as shown in Table 5.2. On average, our proposed EC algorithm requires approximately 99% less computational time when compared to the classic BMA as shown in Table 5.3. Both, the concealed video quality and the computational time, are always given importance in real-time processing. Although MSNs are faster and powerful than simple sensor nodes, still we cannot expect high computational speed and unlimited memory from these nodes. Therefore, simple processing and less

hardware resources demands are recommended. In order to meet QoE constraint, visual comparison under 30% PLR and $(36, 38)$ QPs between the referenced and the proposed techniques is depicted in Fig. 5.9. In Fig. 5.9, column (a) represents the original video frames from test video sequences, while columns (b), (c) and (d) represent outputs of the FC, BMA and the proposed techniques, respectively. There is not much visible difference between the original video frame and the outputs of the FC and the proposed techniques. The blurred lines are quite visible in the outputs of the BMA technique. However, under higher PLRs, FC does not perform well and produces visible fluctuations and video freezing effects. The average PSNR under different QP pairs, is plotted in Fig. 5.10. As shown, our proposed EC technique outperforms both the FC and the BMA techniques in terms of average PSNRs.

Figure 5.9: Visual comparisons

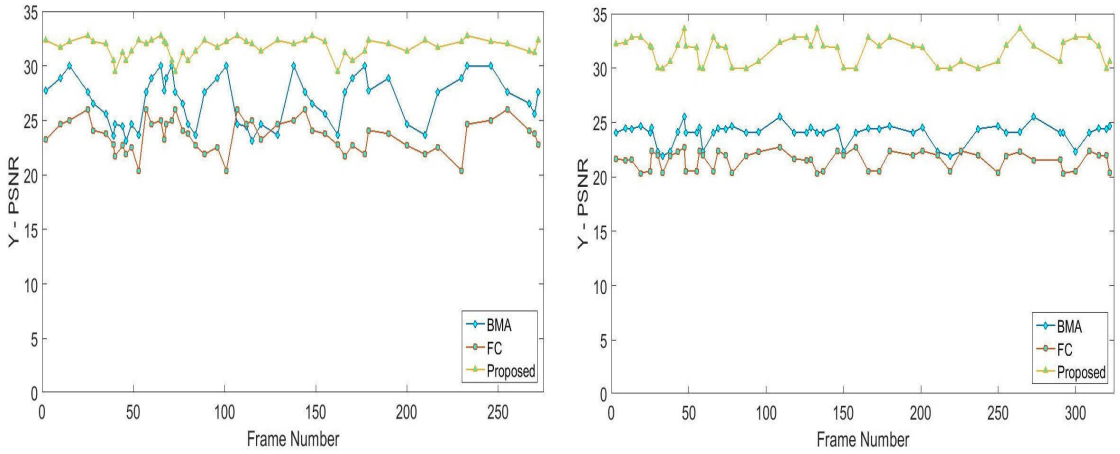



Figure 5.10: PSNR comparisons

## 5.5 Summary

In this chapter, we studied the problem of real-time video streaming over WMSNs and proposed a cross-layer design for video processing, streaming, and error concealment over WMSN. First, a streaming system model has been presented that covers scalable video coding, streaming and network impairments along with problem statement. The scalable high efficiency video coding has been used to encode the videos. Unlike the traditional video streaming architectures, an adaptive style has been adopted in our proposed scheme, which changes the total number of frames for transmission with respect to different network impairments, e.g., packets drop, buffer overloading, bandwidth consumption, etc. For end-to-end streaming, a scalable video streaming scheme has been proposed. A feedback-based on acknowledgement has also been introduced to update the multimedia sender node. The QoS have been maintained through hop counts, adaptive streaming, variable bit-rate, and BS's feedback. On the other hand, the QoE has been maintained through a lightweight and quality oriented video processing and EC algorithm.

Chapter 6

# Conclusion and Future Work

The growth of HD and UHD video streaming over the Internet has increased tremendously in recent years. The streaming of HD and UHD has become possible with HEVC and SHVC standards. These standards are becoming popular due to their coding efficiency, minimum bitrate, and better reconstruction quality. The emerging cloud computing platforms facilitate the designing of next-generation video streaming systems by providing cost-effective, reliable and scalable computing and storage services. Designing a multimedia transmission system based on heterogeneous transmission networks faces many challenges, such as fluctuating bandwidth, variable PLR in the transmission channels, and limited user's device capabilities (e.g., processing speed, screen resolution, memory, etc.). Smooth video streaming can be achieved through smart routing protocols while the effects of packet drops on the received video quality can be minimised by applying efficient EC schemes. In this thesis, we have focused on these challenges by designing an efficient and lightweight EC scheme for HEVC and SHVC encoded HD and UHD videos. This thesis has also investigated and proposed a feedback-based routing scheme for video streaming over a multi-hop WMSN. The feedback helps in improving the QoS by reducing the PLR. The higher/stable QoS helps in maintaining the received video quality at a receiving end. It also helps in utilising the network resources, e.g., bandwidth, efficiently. The wireless and

multi-hop networks can be a combination of fixed and mobile nodes. An efficient routing scheme helps in saving the energy of mobile nodes, thus increasing the lifetime of these networks. This section summarises the research conducted in this thesis followed by the description of potential future works.

In Chapter 2, a detailed description of HEVC, SHVC, video error concealment, video streaming over WMSN, and multimedia cloud computing has been presented. A compact description of HEVC and SHVC architectures has been presented to summarise different phases involved during coding and decoding processes. The drawbacks of conventional error concealment algorithms have been highlighted and they have motivated us to design a lightweight and efficient error concealment scheme for our research. A brief overview of video streaming with layered and cross-layered architectures over WMSNs has been presented and their shortcomings have also been highlighted. We have concluded this chapter with a detailed discussion on multimedia cloud computing and its use for video streaming services.

Chapter 3 has proposed an efficient and lightweight EC algorithm( FI) for cloud-hosted HD video streaming. Our proposed algorithm is designed for HEVC encoded HD and UHD videos and is an extension of classic BMA technique. The FI algorithm is based on the concept of threshold-based parallel processing and uses two video frames only to estimate a missing one. As a result, it eliminates the need for a large buffer at the receiving side. Moreover, the FI algorithm achieves better reconstruction results as compared to the standard EC algorithms, such as the FC and the BMA. Unlike the conventional BMA, the parallel processing style of our proposed approach reduces the computational time, thus helps in saving the energy of mobile terminals.

Chapter 4 has proposed an EC algorithm (SFI) which is an extension of the FI algorithm proposed in Chapter 3. The SFI algorithm is designed for HD and UHD videos encoded by SHVC. During encoding, the SHVC generates multiple video bitstreams, known as BL and EL(s). The SFI first finds out whether the missing video frame belongs to BL or EL. After

finding out the link, it utilises two video frames of the corresponding layer to conceal the missing one. During concealment process, it uses a combination of threshold-based parallel processing and TZ search. The TZ search helps in locating the best match of missing block of pixels in the reference video frames. The use of threshold helps in terminating the searching process quickly with the nearest best matching result which enhances the battery life of the mobile terminal. The SFI is an optimum choice for multimedia applications based on scalable video streaming.

Chapter 5 proposed a feedback-based routing scheme for scalable video streaming over multi-hop WMSN. Without proper management, the transmission of multimedia data over WMSNs affects the performance of networks due to excessive packet drops. The main objectives of the proposed video streaming framework are to maximise the network throughput and to cover-up the effects produced by dropped video packets. The packet-drop effects are covered by SFI proposed in Chapter 4. The HD videos are encoded by SHVC which changes the total number of frames for transmission with respect to different network impairments, e.g., packet drops, buffer overloading, bandwidth consumption, etc. A feedback-based acknowledgement technique has been introduced to update a multimedia sender node. The proposed routing scheme first determines multiple uncorrelated paths between a multimedia sensor node and the corresponding BS, and, later on, schedules the video data packets based on their importance.

## 6.1 Future Work

Due to the rapid development in electronics and telecommunication technologies, HD and UHD video streaming have become a ubiquitous application in our daily lives. From terrestrial TV broadcasting to mobile video streaming, live video streaming is everywhere. The compressed video bitstreams are sensitive to bit errors and a small change during transmission can corrupt the entire video bitstream. The bit errors are very common in transmission networks, especially wireless networks. In wireless networks, transmission

channels are open to various network impairments, such as channels fading, noises, and natural disasters. Furthermore, wireless devices can be either fixed or mobile. Mobile devices are low with respect to various resources, such as battery power, storage, computational power, data rate and transmission range. Due to these limitations, smooth transmission of HD and UHD over unreliable wireless networks becomes a challenging task and has attracted a significant amount of attentions and researches both from industry and academia. In the past years, researchers were focusing on simple video transmission over transmission networks. The rise of technologies like Internet of Things (IoT), Mobile Ad-hoc Networks (MANETs), Vehicular Ad-hoc Networks (VANETs), Wi-Fi networks, and 5G cellular communication, has given a variety of transmission networks and challenges to the researchers from multimedia field. Streaming of HD/UHD videos over these technologies is not an easy job due to fluctuating bandwidths and limited computing resources. Cloud computing can only offer facilities like video coding and storage but cannot deal with transmission errors, thus putting a need for a reliable multimedia transmission system. Most of the multimedia applications, especially video streaming based applications, demand a real-time processing. QoS and QoE provisioning has a significant impact on the business of such applications.

Our contributions in Chapter 3 and Chapter 4 include the lightweight error concealment techniques to maintain the received video quality at the receiving end. Although, these two chapters address the challenges of improving the received video quality in real-time, but their ultimate goals have been to optimise the usage of computing resources. In other words, the algorithms proposed in both of these chapters have enhanced the received video quality of HEVC and SHVC encoded HD/UHD videos. However, these algorithms have their own shortcomings. In Chapter 3 and Chapter 4, the EC techniques have been developed based on integer-pixel-based motions. It would be interesting to see the outcomes of proposed EC techniques based on half- and quarter-pixel-motions. Moreover, the performance of the proposed EC techniques needs to be tested and analysed in a real-time based multimedia transmission system where the nature of the traffic on transmission

channels exhibits a bursty and uncorrelated behaviour. Furthermore, adaptivity features can be introduced in the proposed EC techniques for better utilisation of computing resources. The mobile terminal can be a resource-limited device, e.g., a smartphone with limited battery power. The EC technique should be able to rearrange its computing parameters by determining the features of the underlying computing machines, e.g., screen resolution, processing power, memory, and current status of battery power, etc.

In Chapter 5, we have proposed a feedback-based routing scheme for scalable video streaming over a multi-hop WMSN. The proposed system uses a Wi-Fi network for video transmission. However, it is yet to be determined how efficient and robust the scheme is for a long distance communication. Furthermore, the proposed approach is based on fixed wireless nodes, thus ignoring the complexity of the proposed routing scheme by assuming sufficient computational resources and battery power at end nodes. The efficiency can be analysed by testing its performance on mobile nodes with limited computational resources and battery power. Moreover, performance can be investigated and evaluated in the presence of both fixed and mobile nodes together.

Currently, we are focusing on the integration of the proposed video streaming scheme over MANET. A MANET is a combination of low and high power mobile devices, connected through the Internet. We aim to propose a routing scheme over MANET for smooth HD/UHD video streaming. We will compare the performance of the scheme against both fixed and variable bitrate scenarios. Furthermore, the scheme will be based on latest Wi-Fi standards. Another fascinating area for future research work is the integration of the proposed video streaming scheme over VANETs and IoT. Our proposed routing scheme reduces the network congestion and helps in improving the QoS, which makes it ideal for real-time multimedia applications in networks with limited resources.

# Bibliography


[1] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (hevc) standard," *IEEE Transactions on circuits and systems for video technology*, vol. 22, no. 12, pp. 1649–1668, 2012.

[2] J. Nightingale, Q. Wang, C. Grecos, and S. Goma, "The impact of network impairment on quality of experience (qoe) in h. 265/hevc video streaming," *IEEE Transactions on Consumer Electronics*, vol. 60, no. 2, pp. 242–250, 2014.

[3] B. Bross, W.-J. Han, J.-R. Ohm, G. J. Sullivan, and T. Wiegand, "High efficiency video coding (hevc) text specification draft 8," *JCTVC-J1003, July*, 2012.

[4] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the h. 264/avc video coding standard," *IEEE Transactions on circuits and systems for video technology*, vol. 13, no. 7, pp. 560–576, 2003.

[5] J.-R. Ohm, G. J. Sullivan, H. Schwarz, T. K. Tan, and T. Wiegand, "Comparison of the coding efficiency of video coding standardsincluding high efficiency video coding (hevc)," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1669–1684, 2012.

[6] J. Wu, C. Yuen, M. Wang, and J. Chen, "Content-aware concurrent multipath transfer for high-definition video streaming over heterogeneous wireless networks,"

*IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 3, pp. 710–723, 2016.

[7] P. Kyasanur and N. H. Vaidya, "Capacity of multi-channel wireless networks: impact of number of channels and interfaces," in *Proceedings of the 11th annual international conference on Mobile computing and networking*, pp. 43–57, ACM, 2005.

[8] A. J. Paulraj, D. A. Gore, R. U. Nabar, and H. Bolcskei, "An overview of mimo communications-a key to gigabit wireless," *Proceedings of the IEEE*, vol. 92, no. 2, pp. 198–218, 2004.

[9] A. Vetro, J. Xin, and H. Sun, "Error resilience video transcoding for wireless communications," *IEEE Wireless Communications*, vol. 12, no. 4, pp. 14–21, 2005.

[10] Z. Chang and S.-H. G. Chan, "Video management and resource allocation for a large-scale vod cloud," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 12, no. 5s, p. 72, 2016.

[11] H. Omar, K. Abboud, N. Cheng, K. Malekshan, A. Gamage, and W. Zhuang, "A survey on high efficiency wireless local area networks: Next generation wifi,"

[12] S. Pejhan, M. Schwartz, and D. Anastassiou, "Error control using retransmission schemes in multicast transport protocols for real-time media," *IEEE/ACM Transactions on Networking (TON)*, vol. 4, no. 3, pp. 413–427, 1996.

[13] A. Nafaa, T. Taleb, and L. Murphy, "Forward error correction strategies for media streaming over wireless networks," *IEEE Communications Magazine*, vol. 46, no. 1, pp. 72–79, 2008.

[14] S. Mao, S. Lin, S. S. Panwar, and Y. Wang, "Reliable transmission of video over ad-hoc networks using automatic repeat request and multipath transport," in *Vehicular Technology Conference, 2001. VTC 2001 Fall. IEEE VTS 54th*, vol. 2, pp. 615–619, IEEE, 2001.

[15] I. Recommendation, "E. 800: Terms and definitions related to quality of service and network performance including dependability," *ITU-T August 1994*, 1994.

[16] C. Inc., "The zettabyte era: Trends and analysis," *Cisco VNI: Forecast and Methodology, 2015-2020*, 2016.

[17] Netflix, "Internet connection speed recommendations," 2016.

[18] V. Srivastava and M. Motani, "Cross-layer design: a survey and the road ahead," *IEEE Communications Magazine*, vol. 43, no. 12, pp. 112–119, 2005.

[19] G. Miao, N. Himayat, Y. G. Li, and A. Swami, "Cross-layer optimization for energy-efficient wireless communications: a survey," *Wireless Communications and Mobile Computing*, vol. 9, no. 4, pp. 529–542, 2009.

[20] Q. Liu, S. Zhou, and G. B. Giannakis, "Queuing with adaptive modulation and coding over wireless links: cross-layer analysis and design," *IEEE transactions on wireless communications*, vol. 4, no. 3, pp. 1142–1153, 2005.

[21] J. Tian, H. Zhang, D. Wu, and D. Yuan, "Interference-aware cross-layer design for distributed video transmission in wireless networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 5, pp. 978–991, 2016.

[22] N.-S. Vo, T. Q. Duong, H.-J. Zepernick, and M. Fiedler, "A cross-layer optimized scheme and its application in mobile multimedia networks with q o s provision," *IEEE Systems Journal*, vol. 10, no. 2, pp. 817–830, 2016.

[23] M. Zhao, X. Gong, J. Liang, W. Wang, X. Que, and S. Cheng, "Qoe-driven cross-layer optimization for wireless dynamic adaptive streaming of scalable videos over http," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 3, pp. 451–465, 2015.

[24] C. Xu, J. Zhao, and G.-M. Muntean, "Congestion control design for multipath transport protocols: A survey,"

[25] F. Pescador, M. Chavarrias, M. J. Garrido, E. Juarez, and C. Sanz, "Complexity analysis of an hevc decoder based on a digital signal processor," *IEEE Transactions on Consumer Electronics*, vol. 59, no. 2, pp. 391–399, 2013.

[26] CODE:Sequoia, "Hevc-what are ctu, cu, ctb, cb, pb and tb?," 2016.

[27] X. Xu, R. Cohen, A. Vetro, and H. Sun, "Predictive coding of intra prediction modes for high efficiency video coding," in *Picture Coding Symposium (PCS), 2012*, pp. 457–460, IEEE, 2012.

[28] R. Lins, D. Henriques, E. Lima, and S. Melo, "A faster pixel-decimation method for block motion estimation in h. 264/avc," *TEMA (São Carlos)*, vol. 15, no. 1, pp. 119–130, 2014.

[29] Wikipedia, "Wikipedia block matching algorithms," 2016.

[30] M. P. Sharabayko and O. G. Ponomarev, "Fast rate estimation for rdo mode decision in hevc," *Entropy*, vol. 16, no. 12, pp. 6667–6685, 2014.

[31]

[32] H. Fraunhofer Institute for Telecommunications, Heinrich-Hertz-Institut, "Hm-16.14," 2017.

[33] Wikipedia, "Rate-distortion optimisation," 2017.

[34] Multimedia and V. Group, "Efficient and scalable video coding," 2016.

[35] H. Fraunhofer, "The scalable video coding amendment of the h. 264/avc standard," *Instititue Nachrichtentechnik Heinrich-Hertz-Institute*, 2008.

[36] J. M. Boyce, Y. Ye, J. Chen, and A. K. Ramasubramonian, "Overview of shvc: Scalable extensions of the high efficiency video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 1, pp. 20–34, 2016.

[37] Y. Ye and P. Andrivon, "The scalable extensions of hevc for ultra-high-definition video delivery," *IEEE MultiMedia*, vol. 21, no. 3, pp. 58–64, 2014.

[38] H. Lee, J. W. Kang, J. Lee, J. S. Choi, J. Kim, and D. Sim, "Scalable extension of hevc for flexible high-quality digital video content services," *ETRI Journal*, vol. 35, no. 6, pp. 990–1000, 2013.

[39] G. J. Sullivan, J. M. Boyce, Y. Chen, J.-R. Ohm, C. A. Segall, and A. Vetro, "Standardized extensions of high efficiency video coding (hevc)," *IEEE Journal of selected topics in Signal Processing*, vol. 7, no. 6, pp. 1001–1016, 2013.

[40] H. Fraunhofer Institute for Telecommunications, Heinrich-Hertz-Institut, "Shm-10.0. dev," 2017.

[41] A. Midya, J. Chakraborty, and R. Ranjan, "Video error concealment through 3-d face model," *Multimedia Tools and Applications*, pp. 1–25, 2016.

[42] J. Koloda, A. M. Peinado, and V. Sánchez, "Kernel-based mmse multimedia signal reconstruction and its application to spatial error concealment," *IEEE Transactions on Multimedia*, vol. 16, no. 6, pp. 1729–1738, 2014.

[43] J. Liu, G. Zhai, X. Yang, B. Yang, and L. Chen, "Spatial error concealment with an adaptive linear predictor," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 3, pp. 353–366, 2015.

[44] H. Asheri, H. R. Rabiee, N. Pourdamghani, and M. Ghanbari, "Multi-directional spatial error concealment using adaptive edge thresholding," *IEEE Transactions on Consumer Electronics*, vol. 58, no. 3, pp. 880–885, 2012.

[45] M. M. Hoque, M. L. Q. Pimentel, M. M. Hasan, K. Ahn, J. Kim, and O. Chae, "Edge-based spatial concealment of digital dropout error in degraded archived media," *Electronics Letters*, vol. 50, no. 14, pp. 996–997, 2014.

[46] J. Koloda, V. Sánchez, and A. M. Peinado, "Spatial error concealment based on edge visual clearness for image/video communication," *Circuits, Systems, and Signal Processing*, vol. 32, no. 2, pp. 815–824, 2013.

[47] D. Xu and R. Wang, "Two-dimensional reversible data hiding-based approach for intra-frame error concealment in h. 264/avc," *Signal Processing: Image Communication*, vol. 47, pp. 369–379, 2016.

[48] E. Yaacoub, F. Filali, and A. Abu-Dayya, "Qoe enhancement of svc video streaming over vehicular networks using cooperative lte/802.11 p communications," *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 1, pp. 37–49, 2015.

[49] J. Zhang and T. Tan, "Brief review of invariant texture analysis methods," *Pattern recognition*, vol. 35, no. 3, pp. 735–747, 2002.

[50] Wikipedia, "Edge detection," 2017.

[51] S.-H. Yang, C.-W. Chang, and C.-C. Chan, "An object-based error concealment technique for h. 264 coded video," *Multimedia Tools and Applications*, vol. 74, no. 23, pp. 10785–10800, 2015.

[52] W.-N. Lie, C.-M. Lee, C.-H. Yeh, and Z.-W. Gao, "Motion vector recovery for video error concealment by using iterative dynamic-programming optimization," *IEEE Transactions on Multimedia*, vol. 16, no. 1, pp. 216–227, 2014.

[53] X. Liu, W. Yang, and Z. Shen, "H. 264/avc video error concealment algorithm by employing motion vector recovery under cloud computing environment," *The Journal of Supercomputing*, vol. 70, no. 3, pp. 1180–1199, 2014.

[54] S. Cui, H. Cui, and K. Tang, "Error concealment via kalman filter for heavily corrupted videos in h. 264/avc," *Signal Processing: Image Communication*, vol. 28, no. 5, pp. 430–440, 2013.

[55] Z. Zhou, M. Dai, R. Zhao, B. Li, H. Zhong, and Y. Wen, "Video error concealment scheme based on tensor model," *Multimedia Tools and Applications*, pp. 1–17, 2016.

[56] T.-L. Lin, T.-L. Ding, N.-C. Yang, P.-Y. Wu, K.-H. Tung, C.-K. Lai, and T.-E. Chang, "Video motion vector recovery method using decoding partition information," *Journal of Display Technology*, vol. 12, no. 11, pp. 1451–1463, 2016.

[57] M. Ebdelli, O. Le Meur, and C. Guillemot, "Video inpainting with short-term windows: application to object removal and error concealment," *IEEE Transactions on Image Processing*, vol. 24, no. 10, pp. 3034–3047, 2015.

[58] M. Yang, X. Lan, N. Zheng, and P. Cosman, "Depth-assisted temporal error concealment for intra frame slices in 3-d video," *IEEE Transactions on Broadcasting*, vol. 60, no. 2, pp. 385–393, 2014.

[59] P.-J. Lee, K.-T. Kuo, and C.-Y. Chi, "An adaptive error concealment method based on fuzzy reasoning for multi-view video coding," *Journal of Display Technology*, vol. 10, no. 7, pp. 560–567, 2014.

[60] Y. Zhou, W. Xiang, and G. Wang, "Frame loss concealment for multiview video transmission over wireless multimedia sensor networks," *IEEE Sensors Journal*, vol. 15, no. 3, pp. 1892–1901, 2015.

[61] Y.-K. Kuan, G.-L. Li, M.-J. Chen, K.-H. Tai, and P.-C. Huang, "Error concealment algorithm using inter-view correlation for multi-view video," *EURASIP Journal on Image and Video Processing*, vol. 2014, no. 1, p. 1, 2014.

[62] T.-L. Lin, T.-E. Chang, G.-X. Huang, C.-C. Chou, and U. S. Thakur, "Improved interview video error concealment on whole frame packet loss," *Journal of Visual Communication and Image Representation*, vol. 25, no. 8, pp. 1811–1822, 2014.

[63] S.-C. Tai, C.-C. Wang, C.-S. Hong, and Y.-C. Luo, "An effiicient full frame algorithm for object-based error concealment in 3d depth-based video," *Multimedia Tools and Applications*, pp. 1–21, 2015.

[64] W.-N. Lie and G.-H. Lin, "Error concealment for the transmission of h. 264/avc-compressed 3d video in color plus depth format," *Journal of Visual Communication and Image Representation*, vol. 32, pp. 237–245, 2015.

[65] P. A. A. Assunçao, S. Marcelino, S. Soares, and S. M. de Faria, "Spatial error concealment for intra-coded depth maps in multiview video-plus-depth," *Multimedia Tools and Applications*, pp. 1–24.

[66] S. Khattak, T. Maugey, R. Hamzaoui, S. Ahmad, and P. Frossard, "Temporal and inter-view consistent error concealment technique for multiview plus depth video," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 5, pp. 829–840, 2016.

[67] WiseGEEK, "What is the difference between 3d and 2d?," 2017.

[68] Y. Zhao, C. Zhu, L. Yu, and M. Tanimoto, "An overview of 3d-tv system using depth-image-based rendering," in *3D-TV System with Depth-Image-Based Rendering*, pp. 3–35, Springer, 2013.

[69] J. Seiler, M. Schoberi, and A. Kaup, "Spatio-temporal error concealment in video by denoised temporal extrapolation refinement," in *2013 IEEE International Conference on Image Processing*, pp. 1613–1616, IEEE, 2013.

[70] A. Midya and S. Sengupta, "Switchable video error concealment using encoder driven scene transition detection and edge preserving sec," *Multimedia Tools and Applications*, vol. 74, no. 6, pp. 2033–2054, 2015.

[71] M. Yang, N. Gadgil, M. L. Comer, and E. J. Delp, "Adaptive error concealment for temporal–spatial multiple description video coding," *Signal Processing: Image Communication*, vol. 47, pp. 313–331, 2016.

[72] B. Chung and C. Yim, "Hybrid error concealment method combining exemplar-based image inpainting and spatial interpolation," *Signal Processing: Image Communication*, vol. 29, no. 10, pp. 1121–1137, 2014.

[73] T.-L. Lin, T.-L. Ding, C.-Y. Fan, and W.-C. Chen, "Error concealment algorithm based on sparse optimization," *Multimedia Tools and Applications*, pp. 1–17, 2015.

[74] H. Hadizadeh, I. V. Bajić, and G. Cheung, "Video error concealment using a computation-efficient low saliency prior," *IEEE Transactions on Multimedia*, vol. 15, no. 8, pp. 2099–2113, 2013.

[75] T.-L. Lin, W.-C. Chen, C.-Y. Fan, H.-C. Lee, T.-L. Ding, C.-J. Wang, S.-L. Chen, and C.-H. Hsia, "Switching error concealment algorithm based on optimal decisions for performance and complexity," *Multimedia Tools and Applications*, pp. 1–21, 2015.

[76] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, "A survey on wireless multimedia sensor networks," *Computer networks*, vol. 51, no. 4, pp. 921–960, 2007.

[77] J. F. Kurose, *Computer Networking: A Top-Down Approach Featuring the Internet, 3/E*. Pearson Education India, 2005.

[78] D. P. Agrawal and Q.-A. Zeng, *Introduction to wireless and mobile systems*. Cengage Learning, 2015.

[79] M. Yang, Y. Li, D. Jin, L. Zeng, X. Wu, and A. V. Vasilakos, "Software-defined and virtualized future mobile and wireless networks: A survey," *Mobile Networks and Applications*, vol. 20, no. 1, pp. 4–18, 2015.

[80] B. Fu, Y. Xiao, H. J. Deng, and H. Zeng, "A survey of cross-layer designs in wireless networks," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 110–126, 2014.

[81] D. Feng, C. Jiang, G. Lim, L. J. Cimini, G. Feng, and G. Y. Li, "A survey of energy-efficient wireless communications," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 1, pp. 167–178, 2013.

[82] I. T. Almalkawi, M. Guerrero Zapata, J. N. Al-Karaki, and J. Morillo-Pozo, "Wireless multimedia sensor networks: current trends and future directions," *Sensors*, vol. 10, no. 7, pp. 6662–6717, 2010.

[83] T. O. Olwal, K. Djouani, and A. M. Kurien, "A survey of resource management towards 5g radio access networks," 2016.

[84] X. Chen, J. Wu, Y. Cai, H. Zhang, and T. Chen, "Energy-efficiency oriented traffic offloading in wireless networks: a brief survey and a learning approach for heterogeneous cellular networks," *Selected Areas in Communications, IEEE Journal on*, vol. 33, no. 4, pp. 627–640, 2015.

[85] A. Ahmad, S. Ahmad, M. H. Rehmani, and N. U. Hassan, "A survey on radio resource allocation in cognitive radio sensor networks," *Communications Surveys & Tutorials, IEEE*, vol. 17, no. 2, pp. 888–917, 2015.

[86] I. Al-Anbagi, M. Erol Kantarci, and H. T. Mouftah, "A survey on cross-layer quality of service approaches in wsns for delay and reliability aware applications," 2014.

[87] A. F. Molisch, K. Balakrishnan, D. Cassioli, C.-C. Chong, S. Emami, A. Fort, J. Karedal, J. Kunisch, H. Schantz, U. Schuster, *et al.*, "Ieee 802.15. 4a channel model-final report," *IEEE P802*, vol. 15, no. 04, p. 0662, 2004.

[88] Z. Alliance *et al.*, "Zigbee specification," 2006.

[89] A. Chehri, P. Fortier, and P. M. Tardif, "Uwb-based sensor networks for localization in mining environments," *Ad Hoc Networks*, vol. 7, no. 5, pp. 987–1000, 2009.

[90] T. Adame, A. Bel, B. Bellalta, J. Barcelo, and M. Oliver, "Ieee 802.11 ah: the wifi approach for m2m communications," *Wireless Communications, IEEE*, vol. 21, no. 6, pp. 144–152, 2014.

[91] E. Khorov, A. Lyakhov, A. Krotov, and A. Guschin, "A survey on ieee 802.11 ah: An enabling networking technology for smart cities," *Computer Communications*, vol. 58, pp. 53–69, 2015.

[92] M. Al-Jemeli and F. A. Hussin, "An energy efficient cross-layer network operation model for ieee 802.15. 4-based mobile wireless sensor networks," *Sensors Journal, IEEE*, vol. 15, no. 2, pp. 684–692, 2015.

[93] F. Dobslaw, T. Zhang, and M. Gidlund, "Qos-aware cross-layer configuration for industrial wireless sensor networks," 2015.

[94] Y.-W. Kuo and K.-J. Liu, "Enhanced sensor medium access control protocol for wireless sensor networks in the ns-2 simulator," *Systems Journal, IEEE*, vol. 9, no. 4, pp. 1311–1321, 2015.

[95] H. Yetgin, K. T. K. Cheung, M. El-Hajjar, and L. Hanzo, "Cross-layer network lifetime maximization in interference-limited wsns," *Vehicular Technology, IEEE Transactions on*, vol. 64, no. 8, pp. 3795–3803, 2015.

[96] A. Platonov and I. Zaitsev, "New approach to improvement and measurement of the performance of phy layer links of wsn," *Instrumentation and Measurement, IEEE Transactions on*, vol. 63, no. 11, pp. 2539–2547, 2014.

[97] P. Di Marco, C. Fischione, F. Santucci, and K. H. Johansson, "Modeling ieee 802.15. 4 networks over fading channels," *Wireless Communications, IEEE Transactions on*, vol. 13, no. 10, pp. 5366–5381, 2014.

[98] S. Aust, R. V. Prasad, and I. G. Niemegeers, "Outdoor long-range wlans: a lesson for ieee 802.11 ah," *Communications Surveys & Tutorials, IEEE*, vol. 17, no. 3, pp. 1761–1775, 2015.

[99] T. Kim, D. J. Love, M. Skoglund, and Z.-Y. Jin, "An approach to sensor network throughput enhancement by phy-aided mac," *Wireless Communications, IEEE Transactions on*, vol. 14, no. 2, pp. 670–684, 2015.

[100] Q. Ma, K. Liu, X. Miao, and Y. Liu, "Opportunistic concurrency: A mac protocol for wireless sensor networks," in *Distributed Computing in Sensor Systems and Workshops (DCOSS), 2011 International Conference on*, pp. 1–8, IEEE, 2011.

[101] D. Torrieri, S. Talarico, and M. C. Valenti, "Performance comparisons of geographic routing protocols in mobile ad hoc networks," *Communications, IEEE Transactions on*, vol. 63, no. 11, pp. 4276–4286, 2015.

[102] H. Huang, H. Yin, Y. Luo, X. Zhang, G. Min, and Q. Fan, "Three-dimensional geographic routing in wireless mobile ad hoc and sensor networks," *IEEE Network*, vol. 30, no. 2, pp. 82–90, 2016.

[103] C. H. Lima, P. H. Nardelli, H. Alves, and M. Latva-Aho, "Contention-based geographic forwarding strategies for wireless sensors networks," 2015.

[104] J. Wang, Y. Zhang, J. Wang, Y. Ma, and M. Chen, "Pwdgr: pair-wise directional geographical routing based on wireless sensor network," *Internet of Things Journal, IEEE*, vol. 2, no. 1, pp. 14–22, 2015.

[105] H.-H. Liu, J.-J. Su, and C.-F. Chou, "On energy-efficient straight-line routing protocol for wireless sensor networks,"

[106] L. Cheng, J. Niu, J. Cao, S. K. Das, and Y. Gu, "Qos aware geographic opportunistic routing in wireless sensor networks," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 25, no. 7, pp. 1864–1875, 2014.

[107] M. Won and R. Stoleru, "A hybrid multicast routing for large scale sensor networks with holes," *Computers, IEEE Transactions on*, vol. 64, no. 12, pp. 3362–3375, 2015.

[108] L. Cheng, J. Niu, M. Di Francesco, S. K. Das, C. Luo, and Y. Gu, "Seamless streaming data delivery in cluster-based wireless sensor networks with mobile elements,"

[109] G. A. Shah, F. Alagoz, E. A. Fadel, and O. B. Akan, "A spectrum-aware clustering for efficient multimedia routing in cognitive radio sensor networks," *Vehicular Technology, IEEE Transactions on*, vol. 63, no. 7, pp. 3369–3380, 2014.

[110] D. Zhang, G. Li, K. Zheng, X. Ming, and Z.-H. Pan, "An energy-balanced routing method based on forward-aware factor for wireless sensor networks," *Industrial Informatics, IEEE Transactions on*, vol. 10, no. 1, pp. 766–773, 2014.

[111] P. Mell and T. Grance, "The nist definition of cloud computing," 2011.

[112] Wikipedia, "Cloud platforms," 2017.

[113] Wikipedia, "Cloud computing," 2017.

[114] Z. Huang, C. Mei, L. E. Li, and T. Woo, "Cloudstream: Delivering high-quality streaming videos through a cloud-based svc proxy," in *INFOCOM, 2011 Proceedings IEEE*, pp. 201–205, IEEE, 2011.

[115] R. N. Calheiros, R. Ranjan, and R. Buyya, "Virtual machine provisioning based on analytical performance and qos in cloud computing environments," in *2011 International Conference on Parallel Processing*, pp. 295–304, IEEE, 2011.

[116] T. Hobfeld, R. Schatz, M. Varela, and C. Timmerer, "Challenges of qoe management for cloud applications," *IEEE Communications Magazine*, vol. 50, no. 4, pp. 28–36, 2012.

[117] H. Jiang, F. Shen, S. Chen, K.-C. Li, and Y.-S. Jeong, "A secure and scalable storage system for aggregate data in iot," *Future Generation Computer Systems*, vol. 49, pp. 133–141, 2015.

[118] I. A. T. Hashem, I. Yaqoob, N. B. Anuar, S. Mokhtar, A. Gani, and S. U. Khan, "The rise of big data on cloud computing: Review and open research issues," *Information Systems*, vol. 47, pp. 98–115, 2015.

[119] Z. Chang, J. Gong, T. Ristaniemi, and Z. Niu, "Energy efficient resource allocation and user scheduling for collaborative mobile clouds with hybrid receivers," 2014.

[120] Z. Xiao, W. Song, and Q. Chen, "Dynamic resource allocation using virtual machines for cloud computing environment," *IEEE Transactions on parallel and distributed systems*, vol. 24, no. 6, pp. 1107–1117, 2013.

[121] I. F. Akyildiz, T. Melodia, and K. R. Chowdury, "Wireless multimedia sensor networks: A survey," *IEEE Wireless Communications*, vol. 14, no. 6, pp. 32–39, 2007.

[122] J. He, Y. Wen, J. Huang, and D. Wu, "On the cost–qoe tradeoff for cloud-based video streaming under amazon ec2's pricing models," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 4, pp. 669–680, 2014.

[123] A. Alasaad, K. Shafiee, H. M. Behairy, and V. C. Leung, "Innovative schemes for resource allocation in the cloud for media streaming applications," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 4, pp. 1021–1033, 2015.

[124] Y. Nie and K.-K. Ma, "Adaptive rood pattern search for fast block-matching motion estimation," *IEEE Transactions on Image Processing*, vol. 11, no. 12, pp. 1442–1449, 2002.

[125] S. Metkar and S. Talbar, *Motion estimation techniques for digital video coding*. Springer, 2013.

[126] M. Usman, X. He, K.-M. Lam, M. Xu, S. M. M. Bokhari, and J. Chen, "Frame interpolation for cloud-based mobile video streaming," *IEEE Transactions on Multimedia*, vol. 18, no. 5, pp. 831–839, 2016.

[127] W. I. Choi, B. Jeon, and J. Jeong, "Fast motion estimation with modified diamond search for variable motion block sizes," in *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, vol. 2, pp. II–371, IEEE, 2003.

[128] Wikipedia, "Block matching algorithms," 2017.

[129] I.-K. Kim, J. Min, T. Lee, W.-J. Han, and J. Park, "Block partitioning structure in the hevc standard," *IEEE transactions on circuits and systems for video technology*, vol. 22, no. 12, pp. 1697–1706, 2012.

[130] Z. Pan, Y. Zhang, and S. Kwong, "Efficient motion and disparity estimation optimization for low complexity multiview video coding," *IEEE Transactions on Broadcasting*, vol. 61, no. 2, pp. 166–176, 2015.

[131] S. Goel, Y. Ismail, and M. A. Bayoumi, "Adaptive search window size algorithm for fast motion estimation in h. 264/avc standard," in *48th Midwest Symposium on Circuits and Systems, 2005.*, pp. 1557–1560, IEEE, 2005.

[132] Y. Ismail, J. B. McNeely, M. Shaaban, H. Mahmoud, and M. A. Bayoumi, "Fast motion estimation system using dynamic models for h. 264/avc video coding," *IEEE*

*Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 1, pp. 28–42, 2012.

[133] R. P. Brent and P. Zimmermann, *Modern computer arithmetic*, vol. 18. Cambridge University Press, 2010.

[134] LG, "Lg audio video," 2017.

[135] Mitsubishi, "Mitsubishi audio video," 2017.

[136] Panasonic, "Panasonic audio video," 2017.

[137] Philips, "Philips audio video," 2017.

[138] Samsung, "Samsung audio video," 2017.

[139] Sony, "Sony audio video," 2017.

[140] Toshiba, "Toshiba audio video," 2017.

[141] Sharp, "Sharp audio video," 2017.

[142] O. Baar, "Motion interpolation," 2017.

[143] Y.-L. Zhang, W.-Y. Liu, I. E. Magnin, and Y.-M. Zhu, "Feature-preserving smoothing of diffusion weighted images using nonstationarity adaptive filtering," *IEEE Transactions on Biomedical Engineering*, vol. 60, no. 6, pp. 1693–1701, 2013.

[144] B. Smolka, K. Malik, and D. Malik, "Adaptive rank weighted switching filter for impulsive noise removal in color images," *Journal of Real-Time Image Processing*, vol. 10, no. 2, pp. 289–311, 2015.

[145] S. Wenger, "Nal unit loss software," *JCT-VC Document, JCTVCH0072*, 2012.

[146] J. Zhou, B. Yan, and H. Gharavi, "Efficient motion vector interpolation for error concealment of h. 264/avc," *IEEE Transactions on Broadcasting*, vol. 57, no. 1, pp. 75–80, 2011.

[147] A. Seetharam, P. Dutta, V. Arya, J. Kurose, M. Chetlur, and S. Kalyanaraman, "On managing quality of experience of multiple video streams in wireless networks," *IEEE Transactions on Mobile Computing*, vol. 14, pp. 619–631, March 2015.

[148] J. He, D. Wu, Y. Zeng, X. Hei, and Y. Wen, "Toward optimal deployment of cloud-assisted video distribution services," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, pp. 1717–1728, October 2013.

[149] H. Dinh, C. Lee, D. Niyato, and P. Wang, "A survey of mobile cloud computing: Architecture, applications and approaches," *Wireless Communications and Mobile Computing*, vol. 13, pp. 1587–1611, December 2013.

[150] J. Boyce, Y. Ye, J. Chen, and A. Ramasubramonian, "Overview of shvc: Scalable extensions of the high efficiency video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, pp. 20–34, January 2016.

[151] "Motion interpolation," 2010.

[152] G. Sullivan and V. Seregin, "Shvc draft," 2015.

[153] E. Yaacoub, F. Filali, and A. Dayya, "Qoe enhancement of svc video streaming over vehicular networks using cooperative lte/802.11p communications," *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, pp. 37–49, February 2015.

[154] J. Zhou, B. Yan, and H. Gharavi, "Efficient motion vector interpolation for error concealment of h.264/avc," *IEEE Transactions on Broadcasting*, vol. 57, pp. 75–80, March 2011.

[155] M. Corporation, "Matlab parallel computing," 2015.

[156] M. U. Shahid, A. Ahmed, M. Martina, G. Masera, and E. Magli, "Parallel h. 264/avc fast rate-distortion optimized motion estimation by using a graphics processing unit and dedicated hardware," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 4, pp. 701–715, 2015.

[157] A. Aysu, G. Sayilar, and I. Hamzaoglu, "A low energy adaptive hardware for h. 264 multiple reference frame motion estimation," *IEEE Transactions on Consumer Electronics*, vol. 57, no. 3, 2011.

[158] M. Al-Mualla, C. N. Canagarajah, and D. R. Bull, *Video coding for mobile communications: efficiency, complexity and resilience*. Academic Press, 2002.

[159] S. Ehsan and B. Hamdaoui, "A survey on energy-efficient routing techniques with qos assurances for wireless multimedia sensor networks," *Communications Surveys & Tutorials, IEEE*, vol. 14, no. 2, pp. 265–278, 2012.

[160] Z. Zou, Y. Bao, F. Deng, and H. Li, "An approach of reliable data transmission with random redundancy for wireless sensors in structural health monitoring," *Sensors Journal, IEEE*, vol. 15, no. 2, pp. 809–818, 2015.

[161] C. Tapparello, O. Simeone, and M. Rossi, "Dynamic compression-transmission for energy-harvesting multihop networks with correlated sources," *IEEE/ACM Transactions on Networking (TON)*, vol. 22, no. 6, pp. 1729–1741, 2014.

[162] X. Wu, Y. Xiong, P. Yang, S. Wan, and W. Huang, "Sparsest random scheduling for compressive data gathering in wireless sensor networks," *Wireless Communications, IEEE Transactions on*, vol. 13, no. 10, pp. 5867–5877, 2014.

[163] J. Lainema, F. Bossen, W.-J. Han, J. Min, and K. Ugur, "Intra coding of the hevc standard," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 22, no. 12, pp. 1792–1801, 2012.

[164] X. Pérez-Costa and D. Camps-Mur, "Ieee 802.11 e qos and power saving features overview and analysis of combined performance [accepted from open call]," *Wireless Communications, IEEE*, vol. 17, no. 4, pp. 88–96, 2010.

[165] C. Li, H. Che, and S. Li, "A wireless channel capacity model for quality of service," *Wireless Communications, IEEE Transactions on*, vol. 6, no. 1, pp. 356–366, 2007.

[166] J. Yang, Y. Sun, Y. Wu, and S. Sun, "Joint h. 264/scalable video coding-multiple input multiple output rate control for wireless video applications," *IET image processing*, vol. 6, no. 1, pp. 43–52, 2012.

[167] G. A. Shah, W. Liang, and O. B. Akan, "Cross-layer framework for qos support in wireless multimedia sensor networks," *IEEE Transactions on Multimedia*, vol. 14, no. 5, pp. 1442–1455, 2012.

[168] S. Tang and P. R. Alface, "Impact of random and burst packet losses on h. 264 scalable video coding," *IEEE Transactions on Multimedia*, vol. 16, no. 8, pp. 2256–2269, 2014.

[169] "Network simulator (ns-3)," 2016.

[170] M. A. Jan, P. Nanda, X. He, and R. P. Liu, "Pasccc: Priority-based application-specific congestion control clustering protocol," *Computer Networks*, vol. 74, pp. 92–102, 2014.

[171] M. Corporation, "Matlab 2016a," 2016.