# DATA COMPRESSION TECHNIQUES IN WIRELESS SENSOR NETWORKS

BY

**MUHAMMAD MUSADDIQ**

A Thesis Presented to the
DEANSHIP OF GRADUATE STUDIES

**KING FAHD UNIVERSITY OF PETROLEUM & MINERALS**

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

## MASTER OF SCIENCE

**In**

## COMPUTER NETWORKS

**NOVEMBER 2015**

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN- 31261, SAUDI ARABIA

**DEANSHIP OF GRADUATE STUDIES**

This thesis, written by **MUHAMMAD MUSADDIQ** under the direction his thesis advisor and approved by his thesis committee, has been presented and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE IN COMPUTER NETWORKS.**

Dr. Tarek Sheltami
(Advisor)

Dr. Ahmad Al-Mulhem
Department Chairman

Dr. Ashraf Mahmoud
(Member)

Dr. Salam A. Zummo
Dean of Graduate Studies

Dr. Basem Almadani
(Member)

29/11/15
Date

عمادة الدراسات العليا
DEANSHIP OF GRADUATE STUDIES

*To my loving family, fiance and mentors*

# ACKNOWLEDGMENTS

*In the name of Allah, the most Gracious and ever-Merciful. All praise is due to Allah; we praise Him, seek His help and ask for forgiveness. Peace be upon the Prophet Muhammad, his family, his companions and all those who follow him until the Day of Judgement.*

*It is inevitable to extend my deepest gratitude and respect to my family, especially my parents, the ones who very kindly waived their rights on me to let me pursue my studies. There are no words that can show my respect and love for them. I will always be thankful to them for their continuous moral and emotional support and ever-needed prayers.*

*It has been privilege to work with Dr. Tarek Sheltami. I would like to admire his supervision, suggestions and guidance right from the beginning till the end of this research. His constant motivation helped me to produce quality work. I would like to thank my committee members: Dr. Basem Almadani and Dr. Ashraf Mahmoud for their useful response, advice and the time they spent reviewing this thesis.*

*I am very obliged to KFUPM for granting fully funded scholarship for my master's degree. Also, I would like to appreciate all the support that I received from the Computer Engineering Department particularly Wireless Sensor Laboratory in carrying out this research. I am thankful to Dr. Uthman Baroudi for his help in providing me the required hardware for my thesis.*

*I am also very thankful to my friends Muhammad Naseer Bajwa, Bilal Saeed, Muhammad Ijaz , Bilal Jehanzeb, Sheharyar Khan, Adil Qayyum, Tanvir Hussain, Awais Wahab and especially Hafiz Muhammad Ijaz for providing the moral support, pleasant atmosphere and never forgettable moments.*

*Last but not the least I would like to thank my fiancé who prayed a lot for me during this research. She has been a source of inspiration and she was always by my side.*

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# LIST OF ABBREVIATIONS

| | | |
|---|---|---|
| **ARQ** | : | Automatic Repeat Request |
| **DCT** | : | Discrete Cosine Transform |
| **DWT** | : | Discrete Wavelet Transform |
| **DoS** | : | Denial of Service |
| **DPCM** | : | Differential Pulse Code Modulation |
| **EEPROM** | : | Electrically Erasable Programmable Read Only Memory |
| **EZW** | : | Embedded Zerotrees of Wavelet Transform |
| **EBCOT** | : | Embedded Block Coding with Optimal Truncation |
| **EOB** | : | End of Block |
| **ETE** | : | End-to-End Delay |
| **FDCT** | : | Fast Discrete Cosine Transform |
| **JPEG** | : | Joint photographic experts group |
| **JSCCPC** | : | Joint Source Channel Coding and Power Control |
| **LSPs** | : | List of Significant Pixels |
| **LISs** | : | List of Insignificant Sets |

| | | |
|---|---|---|
| **LR-WPANs** | : | Low-rate Wireless Personal Area Networks |
| **MSE** | : | Mean Square Error |
| **MAC** | : | Media Access Layer |
| **OS** | : | Operating System |
| **PSNR** | : | Peak-to-Signal Noise Ratio |
| **QF** | : | Quantization Factor |
| **ROM** | : | Random Only Memory |
| **RLE** | : | Run-Length Encoding |
| **SPIHT** | : | Set partitioning in Hierarchal Trees |
| **SPECK** | : | Set partitioned Embedded Block Code |
| **S-SPECK** | : | Scalable SPECK |
| **UART** | : | Universal Asynchronous Receive and Transmit |
| **VQ** | : | Vector Quantization |
| **VLC** | : | Variable Length Coding |
| **WSN** | : | Wireless Sensor Network |
| **ZRL** | : | Zero Run Length |

# ABSTRACT

Full Name : Muhammad Musaddiq

Thesis Title : Data Compression Techniques in Wireless Sensor Networks

Major Field : Computer Networks

Date of Degree : November 2015

The advancement in the wireless technologies and digital integrated circuits led to the development of wireless sensor networks (WSN). WSN consists of various sensor nodes and relays capable of computing, sensing, and communicating wirelessly. However, nodes in WSNs have very limited resources i.e., memory, energy and processing capabilities. In WSN, many image compression techniques have been proposed to address these limitations but most of them are not applicable on sensor nodes due to memory limitation, energy consumption and processing speed.

To overcome this problem, we have selected discrete cosine transform (DCT) and discrete wavelet transform (DWT) image compression techniques as they can be implemented on sensor nodes. Both DCT and DWT allow an efficient trade-off between compression ratio and energy consumption. In this thesis, we have analyzed and implemented DCT and DWT using TinyOS on TelosB hardware platform. The metrics used for performance evaluation are peak signal-to-noise ratio (PSNR), compression ratio, throughput, end-to-end delay (ETE), and battery lifetime. Moreover, we evaluated DCT and DWT both in single-hop and multi-hop networks. Experimental results show that DWT outperforms DCT in terms of PSNR, throughput, ETE delay and battery lifetime. However, DCT provides better compression ratio than DWT. We have also calculated average MAC delay for both compression techniques.

# ملخص الرسالة

**الاسم الكامل: محمد مصدق**

**عنوان الرسالة: تقنيات ضغط البيانات في انظمة الاستشعار اللاسلكية**

**التخصص: شبكات حاسوبية**

**تاريخ الدرجة العلمية: اكتوبر 2015**


التطور في تقنيات الشبكات اللاسلكية والدوائر المنطقية الرقمية ادت الى التطور الكبير الذي حصل في انظمة الاستشعار اللاسلكية (WSN) , انظمة الاستشعار اللاسلكية تحتوي على العديد من وحدات الاستشعار ووحدات النقل للبيانات التي هي قادرة على معالجة البيانات والاستشعار والاتصال اللاسلكي. ولكن وحدات الاستشعار تحتوي على مصادر محدودة من الذاكرة والطاقة ووحدات المعالجة. في انظمة الاستشعار اللاسلكية تم اقتراح العديد من الطرق في عملية ضغط الصور ولكن العديد من هذة الطرق لا يمكن تطبيقة نظراً لمحدودية الذاكرة, وكمية الطاقة الكبيرة المستهلكة من قبل هذة الطرق والوقت الكبير التي تحتاجه وحدات الاستشعار خلال تطبيق هذة الطرق.

لحل هذة المشكلة, قمنا باختيار DCT و DWT في عملية ضغط الصور في وحدات الاستشعار. وهذين النظامين للمعالجة يمتازان باعطاء كمية ضغط مناسبة للصور مع الاخذ بعين الاعتبار عدم احتياجهما لطاقة كبيرة من وحدات الاستشعار خلال تنفيذ هاتين العمليتين. في هذة الرسالة قمنا بتحليل وتطبيق DCT و DWT باستخدام جهاز TelosB. المعايير التالية تم اخذها بعين الاعتبار عند عملية تقييمنا للنظام: PSNR, كمية ضغط البيانات,ETE بالاضافة الى كمية الصور التي يمكننا ارسالها خلال دورة كاملة للبطارية. كذلك تم تقييمDCT و DWT في الانظمة التي تحتوي على ناقل واحد بين المرسل والمستقبل والانظمة التي تحتوي على اكثر من ناقل بين المرسل والمستقبل. كنتيجة لهذة الدراسة وجدنا ان DWT تتفوق على DCT في المعايير التالية: ETE, PSNR بالاضافة الى كمية الصور التي يمكننا ارسالها خلال دورة كاملة للبطارية. ولكن DCT تعطي كمية ضغط اكثر للبيانات من DWT , كذلك قمنا بحساب معدل التاخير في ال MAC في كل من ال DCT و DWT.

# CHAPTER 1

# INTRODUCTION

Wireless Communication has a several advantages over typical wired network. It is able to develop small, low power, low-cost, and multi-functional sensing devices. These small devices have the capabilities to sense, compute and communicate. They are known as sensors. These sensors are used to sense the conditions of the surrounding, collect the data and process it. They can be grouped together to form a network communicating wirelessly using radio frequencies. The collection of these homogenous or heterogeneous sensor nodes are called Wireless Sensor Networks (WSNs).

## 1.1 Wireless Sensor Networks (WSNs)

A wireless sensor networks (WSNs) [1] is a group of dedicated sensors and actuators. They have wireless infrastructure for communication and are used to monitor environmental changes at different locations. A network has several nodes and each node is connected to other node. The nodes may have different functions to perform such as exchanging or relaying data and sensing etc. The node used for sensing the data is called a sensor node, and the one that relays the data is called a router. The node used for exchanging the data with other networks is called base station or sink node.

Each sensor node is equipped with a transducer, a radio transceiver, a microcontroller and a power supply. The transducer produces electrical signals depending upon the physical or environmental changes. The radio transceiver is used for data transmission and reception whereas microcontroller is used for processing the data.

In many applications, sensor nodes are deployed randomly to monitor the surroundings and they do not have deterministic location. They can communicate with each other through radio transceiver. The data is collected at the sensor node and then transmitted to the sink node which in turn is connected to the satellite network. The satellite network then transmits the data to the application as shown in Figure 1-1.

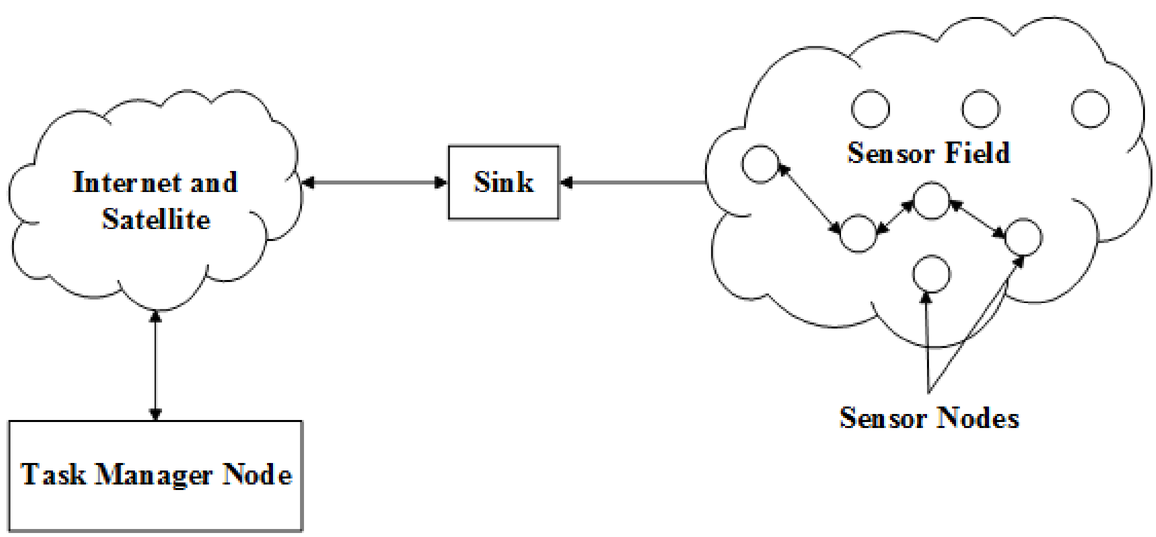



Figure 1-1 Structure of Typical Wireless Sensor Network [1]

### 1.1.1 Network Components of WSN

The major components of a general WSN are the sensor nodes, sink nodes and the events. The operation of these units is shown in Figure 1-1.

#### 1.1.1.1 Sensor Node and its Functional Units

In WSN, each and every sensor node has the capability of sensing, processing and sending data to the required receiver. The basic units in sensor nodes are sensing unit, memory unit, power unit, processing unit and communication unit.

**Sensing Unit**

Sensor is a device used to measure the change in the physical condition of an area of interest and gives response to that change. Sensors sense the environment, collect data and convert it to physical data (voltage or current) before sending it for further processing. There are different sensors which are available and can be used according to the required operation. The size of sensors and their energy consumptions are important factors that need to be considered before selecting the sensors.

**Memory Unit**

This unit of sensor node stores the program and data code. read-only memory (ROM) is used to store the data packets. While to store the program code, electrically erasable programmable read-only memory (EEPROM) or Flash memory is used.

**Power Unit**

A sensor node has a power unit to provide power to all other units because energy is required for computation and data transfer. The power is mostly consumed in computation and transmission. Of the two the transmission entity consumes the more power. Mostly the sensor nodes are battery operated.

**Processing Unit**

Sensor node has a microcontroller. This microcontroller has several other sub-components such as processing unit, memory, converters and universal asynchronous receive and transmit (UART). This processing unit is responsible for handling data gathering and incoming and outgoing transmission.

**Communication Unit**

The networking can be obtained by the sensor nodes using radio frequencies or optical communication. The radio units in the sensor nodes use electromagnetic spectrum to transfer the data to required destination.

#### 1.1.1.2 Base Station (Sink Node)

It is an interface between sensor network and management center. In a sensor network there can be single or multiple base station(s). The multiple base stations can perform better and decrease network delay. Base station can be stationary or dynamic.

### 1.1.2 Application Areas of WSN

WSN applications can be categorized in two fields: remote monitoring and mobile object location tracking. These fields can be further divided into indoor and outdoor applications. Some of the applications are presented in Figure 1-2.

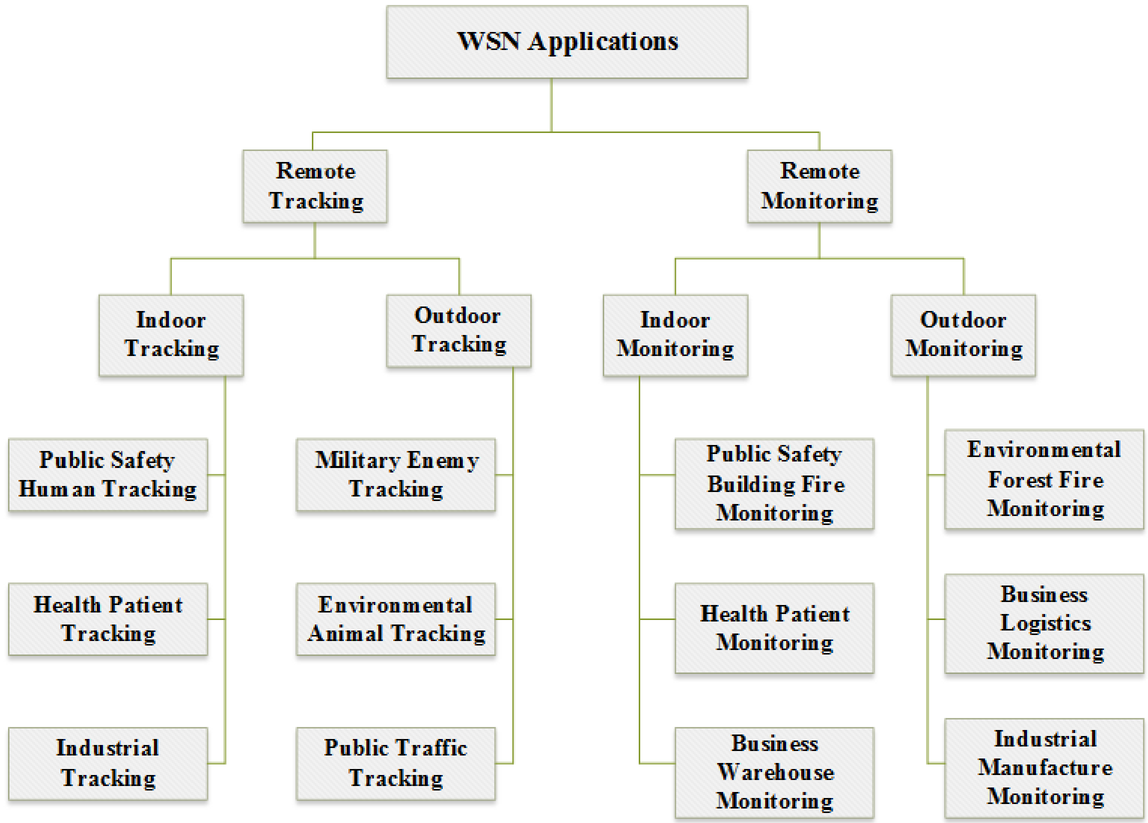



Figure 1-2 Applications of WSN [1]

Environmental applications include detecting forest fire, sensing or detecting chemical leakage and tracking the movement of animals. Military applications include tracking enemy movement, detecting nuclear attack, checking equipment status and monitoring forces. Inventory monitoring, vehicles and object tracking fall under commercial/logistic applications.

### 1.1.3 Challenges in Design and Implementation of WSN

There are some challenges in design and implementation of WSN. These challenges need to be addressed before implementing WSNs. Following are the major ones:

**Energy Efficiency**

The energy efficiency can be improved in several ways. One method is to optimize software and hardware design, which reduces the energy consumption and makes WSN power efficient. Another method is to optimize the power management at hardware and network levels.

**Interference**

The performance of WSNs can be severely affected by the other wireless system working in the same frequency spectrum and in the same area. Interference avoidance algorithms can be used in order to overcome the limitations of WSN such as slow computation.

**Security**

The wireless nature of WSN can cause some security issues, which are unavoidable. The data distributions must be safeguarded from unauthorized access. The system level security can be assured by focusing on remote Denial of Service (DoS) attacks.

**Data Management**

When large amount of data is sent over the network, the energy cost of such transmission is very high. So the data compression and aggregations techniques can be used to compress the data. Robust methods must be employed to reduce the amount of data while improving the in-network data processing.

## 1.2 Thesis Motivation

This research draws its inspiration from some of the unresolved issues presented in the related work in the domain of wireless sensor nodes. There are several emerging application of WSNs such as surveillance, recognition, tracking, localization and object detection; they all require vision capabilities [2] . As, the sensor nodes are mostly battery-powered and a number of WSN applications are used for long-term monitoring of environment so conserving the energy in order to increase the lifetime of sensors is the critical issue. The solution to this problem is to reduce the amount of data to be sent because in sensor nodes the most energy consuming operation is transmission [3]. The amount of data can be reduced with the help of compression. The advancement in compression algorithms has become an essential need in the multimedia field.

Image compression algorithms have been developed in order to reduce the size of the image [2]. The applications based on images such as medical imaging, cameras, video-on-demand systems …etc have large amount of data to transmit, therefore, radio transmitting nodes consume more data than receiving nodes. It has been observed from the literature that there are many compression algorithms but due to resource constraints, such as memory or processing speed, most of them become inapplicable in real-time environment.

To conclude, all the problems mentioned above show that the research in this area is very challenging yet encouraging. Therefore, to overcome the issues, efficient compression techniques are needed.

## 1.3 Research Objectives

The objective of this research is to implement data compression techniques such as DCT and DWT to compress the images and consequently decrease the energy consumption. The main objectives of this research are:

1. To implement both DCT and DWT compression techniques in wireless sensor networks using TinyOS on TelosB microcontroller.
2. To compare both the techniques DCT and DWT in order to evaluate their performance with respect to image quality, throughput, compression ratio, ETE delay and battery lifetime.
3. To make recommendations regarding the two techniques based on different scenarios.

## 1.4 Research Methodology

The following methodology has been used to achieve the objectives:

**Task 1: Implementation of DCT**

- Image transformation into suitable block format such as 8x8, 16x16 etc.
- Transformation of block into frequency domain by using discrete cosine transform (DCT).
- Quantization after DCT to reduce number of bits.
- Entropy coding and transmitting the data packets over wireless sensor network.

**Task 2: Implementation of DWT**

- Image transformation into suitable block format such as 8x8, 16x16 etc.
- Distribution of image into four sub-bands using filters.
- Reduction in number of bits, quantization and coding of sub-bands.
- Variable length coding to encode the message fully and transmit over wireless sensor network.

**Task 3: Comparison of DCT and DWT**

- Conducting comparison of both the techniques based on image quality, throughput, compression ratio, end-to-end delay and battery lifetime.

**Task 4: Experimental Evaluation of DCT and DWT**

- Experimental evaluation of DCT and DWT using TelosB in different scenarios such as single-hop and multi-hop network.

- Use of TinyOS for the implementation of wireless sensor networks.

## 1.5 Thesis Breakdown

Having presented the basis for the research problem and laid down the foundation of the problem area in the first chapter, the literature review of different compression techniques is presented in second chapter. Chapter 3 and chapter 4 describe the DCT and DWT compression techniques respectively in detail. Chapter 5 presents the experimental setup of DCT and DWT. Chapter 6 describes the performance evaluation and comparison of both the compression techniques. Finally, Chapter 7 concludes the findings and suggests the future work.

# CHAPTER 2

# LITERATURE REVIEW

Recently, with the emergence of wireless technologies, WSNs have gained research interests [4][5]. Moreover, there is strong interest in deploying WSNs for multimedia applications [5].

Wireless sensor networks in multimedia industry has different applications in various fields such as, traffic surveillance, health care, security monitoring etc. The multimedia data such as, audio, video and images is normally delay sensitive and bandwidth limited. These characteristics of multimedia data requires enough resources to transmit data from source to sink. As, many applications are resource constraints such has limited energy supply, limited memory and low processing speed. These constraints gives challenges to get desired results of applications [6]. Hence, in order to tackle these resource constrained problems, a number of techniques has been proposed. Which are as follows:

## 2.1 Discrete Cosine Transform (DCT)

The researchers have done a lot of work in the data compression based on DCT. The basic idea of DCT is to convert a signal into basic frequency components. For compression, image is divided into different blocks and a sum of cosine functions on different frequencies can be mathematically used to express each block of image. Joint photographic experts group (JPEG) [7] is a renowned compression scheme based on DCT.

This technique has been analyzed in different ways such as, reducing the computation complexity, increasing the compression ratio, and minimizing power consumption [8].The power consumption of DCT-based methods are more within DCT-based technique [4]. Many researchers have made attempts to decrease it computation complexity. Some of them are mentioned in this related work. In [9], they have used parallel a pipelined implementation of multidimensional DCT. In order to reduce computational complexity both arithmetic units and processing elements operates in parallel in their proposed method. In [10], researchers used the same concept but with the integer cosine transform. The integer cosine transform has less computational complexity. This has been used to reduce the complexity that what achieved in [9].

The authors in [11] have done their evaluation on video surveillance and analyzed the trade-off between power consumption and image quality. They have used integer DCT rather than floating-point DCT and showed that how image compression has put impact on image delay using automatic repeat request (ARQ). Moreover, [12] has developed image sensor network platform to transmit compressed images using multi-hop sensor network. They have done their compression on JPEG and its variant JPEG2000. The researchers shows that JPEG2000 is more appropriate than JPEG, as they have compared the two on the basis of packet loss and bit errors. But as their evaluation doesn't consider power consumption, so it would be non-practical for all multimedia sensor network applications. In [13], the purpose of their research is to find an algorithm which is energy efficient for the image compression. So, the authors have addressed the problem of reducing the energy consumption. The trade-off between image quality and energy consumption has also been considered using JPEG compression method. But they have done the evaluation on simulation instead on real-time experiments.

## 2.2 Discrete Wavelet Transform (DWT)

Discrete wavelet transform was used to overcome the weakness of DCT-based method and also to increase the features of DCT i.e., frequency and localization in time [14]. Most of the related work in DWT is based on 2D-DWT. Firstly, the implementation of DWT can be done by using 1D-DWT in row wise to get L and H bands. Secondly, 1D-DWT can apply in column wise to get four sub-bands such as LL, LH, HL, and HH. Furthermore, each of these four bands can then be divided into four sub-bands. Researchers have proposed some schemes based on wavelet transform that are discussed below.

Embedded zerotrees of wavelet transform (EZW) image compression technique was introduced in [15]. It was designed for 2D dimension but it can also be used in other dimensions. In EZW, encoder compress the image into a bit stream and it is based on progressive encoding [16]. The input image decomposes into wavelet coefficients. It is multi-pass process in which it has two pass such as dominant pass and subordinate pass and more details can be found in [17] . The authors in [17] has proposed the method based on EZW for image compression. In [17], EZW give two type of resolution such as high and low resolution. High resolution can be used for the regions where interference has been detected while low resolution can be used for other regions. They have concluded that this scheme is better for saving power and bandwidth. It is an efficient algorithm due to progressive encoding and multi resolution wavelet transform. However, the image quality can be degraded by the number of passes. Also, if the number of coefficients increase then due to less storage it is difficult to store those significant coefficients. Furthermore, packet loss is also a factor because EZW is vulnerable against packet losses [18].

EZW is improved by another scheme named as set partitioning in hierarchal trees (SPIHT). It has achieved high compression ratio because of setting partition algorithm in an image wavelet transform. It also has three linked lists as discussed in [18] . In SPIHT, an input image is decomposed into coefficients and then these coefficients group together to form sets called as spatial orientation trees. Then these coefficients are encoded separately. SPIHT has two coding passes i.e., sorting pass (looks for zerotrees) and refinement pass (sends precision bits) [18].

Many researchers proposed methods in order to minimize its limitations and increase SPIHT features. The researchers in [19] gave the idea of network conscious compression to advance its performance in lossy conditions. While in [20], the internal memory usage has been reduced and tried to enhance the speed of SPIHT process. This has been achieved by having a new coding process and improved zerotree structure. This held in getting better image quality after compression.

Moreover, the researchers of [21] used a strip-based method for the image compression based on SPIHT. This technique divides the image intro strips and these strips can be encoded separately. According to technique, the DWT module decompose the initial few lines. Then the coefficients of wavelets are computed and stored in buffer. Lastly, the stream is transmitted. SPIHT still has the problem of internal memory usage in power-limited applications as it uses three lists to save coding information which requires large memory. This requirement of memory increases the computational complexity. But SPIHT is better than EZW in terms of power consumption.

In [22] researchers proposed another algorithm known as embedded block coding with optimal truncation (EBCOT). It is a block-based encoding algorithm. In this scheme, each block is divided into blocks that are non-overlapping and having DWT coefficients. These blocks are called as code blocks. It generates separate bit stream instead of one single bit stream for each block because each block is coded independently. Moreover, EBCOT shows the basic and important functioning of JPEG2000. It has two tiers i.e., Tier 1 and Tier 2. It is an efficient compression algorithm but it also requires large memory and dissipate more power, it has been reported in [22] that tier 1 consumes more power in the processing follows by other components. In [23], the authors merge the three coding passes into single pass in order to reduce the memory and increase the performance of system but they have not addressed the issue of packet loss.

The framework in [24] explains about the compression by using JPEG2000. To increase the network lifetime the visuals sensors are released. The sensors are organized in a cluster except the camera sensor. The visual sensor creates its own cluster and send to cluster instead of joining the cluster straight away. This technique can increase the network lifetime but it consumes large amount of power. An energy efficient scheme has been proposed in [25] for compression. The proposed scheme named as joint source channel coding and power control (JSCCPC). In this scheme, the number of layers in which an input image is encoded depends on three factors such as, image content, end-to-end image constraint and channel condition. It finds the multi resolution property of bit steam but the authors has not discussed the importance between magnitude information and structured information. Furthermore, releasing camera sensors may not be the better option all the time as it increases power consumption and may reduce the network lifetime. This power consumption is due to DWT and become the source of major power consumption.

In [26], set partitioned embedded block code (SPECK) compression technique was developed. In this scheme, the researchers used set of pixels rather than using trees. The set of pixels were used to form blocks. The algorithms works in a manner that it takes input image and starts sub-band transformation usually DWT. then sorting pass and refinement pass phase has been repeated. These were the same phases discussed in SPIHT. It also has two link lists i.e., list of significant pixels (LSPs) and list of insignificant sets (LISs). The detailed procedure of this techniques is discussed in [27]. SPECK has some advantages over other compression techniques. It has better compression ratio and less memory requirements. Moreover, it has low computational complexity as compared to above techniques discussed.

However, it has some shortcomings for which few attempts has been made by authors to overcome them.  One of the variant of SPECK is listless SPECK names as LSK [27]. In LSK, there is no need of lists as was in [27] because of breadth first search. Furthermore, it use policies of block-partitioning. While in [28] another variant of SPECK has been proposed called as scalable SPECK (S-SPECK). It has very low complexity as it makes

SPECK scalable. In the initial research, LIS and LSP requires large memory as these lists increase with the increase in encoding. Which in turn made this technique inappropriate for real-time hardware implementations. SPECK is also in the need of error correction scheme due to its vulnerability against packet loss. However, this technique can be preferred over EZW due to its high compression, low complexity and low power consumption.

## 2.3 Vector Quantization Compression (VQ)

Vector Quantization (VQ) is different from typical data compression technique as presented in [28]. The method known as codebook is the mapping of large vectors into small subsets of code.

Several schemes has been proposed based on vector quantization. Some of them are discussed here. Image processing VQ method has been used in [29] to reduce the power dissipation. This has been done by reducing the LSB of the image and codeword. Moreover, in [30] memory reduction has been considered. The authors have reduced the memory size of the codeword. Also, they minimize the number of access of memory. But this research is based on low-power image coding. In [31], researchers proposed a technique low-power pyramid VQ.  The advantage of VQ over other compression techniques is that it has simple decoder. Also, as there is no transformation block, so it reduce the computation complexity. While, considering the overall complexity of VQ, it increases due to the increase in vector dimension. Which in turn increase the power consumption. Moreover, with the presence of codeword for the large images, it requires more memory.

## 2.4 Fractal Compression

This technique is based on fractal theory. Which states that, image can be defined by a set of fractals. It is a lossy compression technique like VQ. It is also different from typical compression techniques such as JPEG and JPEG2000. Moreover, it has no transformation block as was in DCT and DWT [8]. In this technique, fractal encoder converts image into fractal codes while decoder works other way around. The fractal-based image coder was first introduced in 1990 by Jaquin. The researchers are proposed various variations of fractal coders. The focus of research of most authors was on improvement of encoding process [32]. Moreover, attempts have been made to joining transformations with the fractals [33]. In [34], attempt of wavelet transformation with the fractal has been made. Fractal image compression has high compression ratio and has simple decoding procedure. However, the encoding process is time consuming and it is slows down due to the computation. Also, it is not implemented on sensors nodes [34].

Though, a lot of research has been done on image compression in wireless sensor networks as discussed above. The characteristics of all the compression techniques are summarized in Table 2-1 [35][36].

Table 2-1 Summary of Compression Algorithms

| Features | JPEG | JPEG 2000 | EZW | SPIHT | EBCOT | VQ | Fractal |
|---|---|---|---|---|---|---|---|
| Transform | DCT | DCT | DWT | DWT | DWT | None | None |
| Computational Load | Low | High | Low | Low | High | Very High | Very High |
| Memory Requirements | Average | High | High | Average | High | High | Low |
| Complexity | Low | Average | High | Low | High | High | High |
| Compression Ratio | Very High | Very High | Low | High | High | Low | Very High |
| Processing Speed | Average | Low | Average | High | Low | Low | Low |
| Image Quality | Low | Low | Low | High | High | Average | High |
| Power Consumption | Average | Very High | High | Low | High | High | Very High |
| Real-Time Implementation | Yes | Yes | Yes | Yes | Yes | Yes | No |
| Most Suitable for WSN | ✓ | X | X | ✓ | X | X | X |

It is clear from the Table 2-1 that the JPEG and SPIHT libraries of DCT and DWT respectively are the most suitable for WSN. The non-transformed techniques such as vector quantization and fractal compression are better in terms of computational complexity and memory requirement respectively. However, vector quantization lacks in memory consumption and fractal compression has slow encoding process. Also, the fractal compression cannot be implemented in real-time environment due to its large processing time. On the other hand, DCT and DWT can be implemented in real-time environment. In this study, two compression techniques DCT and DWT have been selected for real-time implementation on hardware platform. Both DCT and DWT have several advantages over other compression techniques such as DCT provides high compression ratio. Whereas, DWT has less energy and memory consumption, fast processing and better image quality.

Table 2-2 Literature Summary for DCT and DWT

| **Parameters** | **DCT** | **DWT** |
| --- | --- | --- |
| PSNR | Reported [4] [6] [9] [11][25][36] [39] | Reported [21][27][30] [36] [39] [42] [43] [45] [47] |
| Throughput | Not Reported | Not Reported |
| Compression Ratio | Reported [35] [36] | Reported [35] [36] [45] |
| ETE Delay | Not Reported | Not Reported |
| MAC Delay | Not Reported | Not Reported |
| Battery Lifetime | Not Reported | Not Reported |

It is evident from literature review that a lot of work has been done on DCT and DWT but the potential of DCT and DWT has not been fully investigated. As shown in Table 2-2, a number of performance evaluation parameters such as throughput, end-to-end (ETE) delay, medium access layer (MAC) delay and battery lifetime has not been reported in literature so far to the best of our knowledge. It cannot be justified to evaluate the performance of DCT and DWT on the basis of just two reported parameters i.e., peak signal-to-noise ratio (PSNR) and compression ratio. So, a comprehensive analysis of these two techniques is needful.

# CHAPTER 3

# DISCRETE COSINE TRANSFORM

## 3.1 Overview

The discrete cosine transform (DCT) image compression technique has a number of steps [38]. The image is first transformed into an appropriate format for image compression. All the image components are divided into 8x8 blocks. The next step is encoding a block but encoding requires a block to be transformed into frequency plane. This transformation can be achieved using discrete cosine transform. It is used to exploit the spatial correlation between the pixels. After the transformation, most of the information is intense to a few low-frequency components. To decrease the number of bits required to represent the image, these components of the image will then be quantized. In this step, the image quality will be lower due to decreasing the precision of components. The quantization matrix controlled the tradeoff between produced bits and image quality. The ZigZag scanned to the components will also be performed to place the most likely non-zero components first and the most likely zero components last in the bit stream. In the next step, entropy encoding is performed to achieve better compression. The common encoders are Huffman encoder, arithmetic encoder and simple run-length encoder. In this research, we have used a combination of Huffman encoding and variable length encoding. Lastly, data packets are created for transmission over the wireless sensor network. The DCT image compression is shown in Figure 3-1.

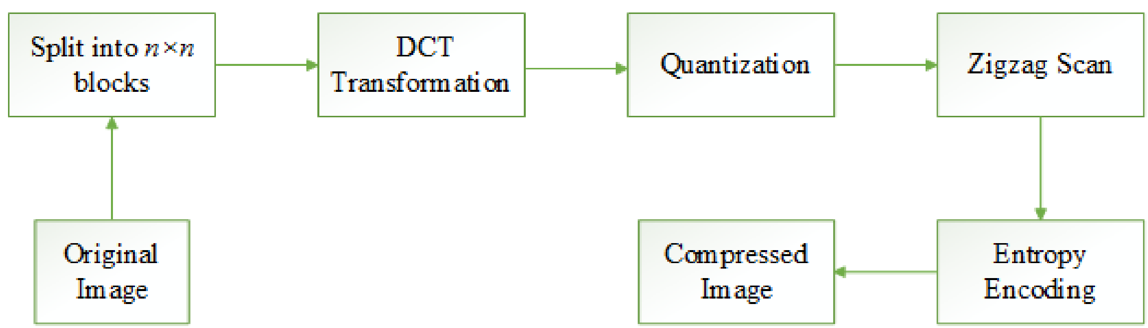



Figure 3-1 DCT Compression Method

## 3.2 Discrete Cosine Transform

In DCT, the transformation from spatial domain to frequency domain is performed on each block of the image [39]. This will produce a matrix of 8x8 having different frequency coefficients. These coefficients are divided into lowest frequency coefficients and highest frequency coefficients. The lowest frequency coefficient are known as DC coefficient, represents the average value of a signal. While the remaining coefficients are known as AC coefficients, represents higher frequencies. Most of the information in the block will be concentrated to a few low-frequency coefficients. It is due to the intensity values of an image that vary slowly and there will be only high frequency coefficients around the corners. The DCT transform is given in Eq. (3.1) [39][37].

$$F(i,j)=\frac{1}{\sqrt{2N}}C(i)C(j)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)\cos\left[\frac{(2x+1)i\pi}{2N}\right]\cos\left[\frac{(2y+1)i\pi}{2N}\right] \tag{3.1}$$

$$C(n)=\begin{cases}\frac{1}{\sqrt{2}} & if\ n=0\\ 1 & if\ n>0\end{cases} \tag{3.2}$$

The Eq. (3.1) computes the i,j$^{th}$ entry of the DCT of an image and *f*(*x,y*) is the x,y$^{th}$ element of the image represented by matrix *p*. The size of the block is *N*. The standard 8x8 block has values *N*=8, *x* and *y* ranges from 0 to 7. Thus, *F*(*i,j*) would be known in Eq. (3.3)

$$F(i,j)=\frac{1}{4}C(i)C(j)\sum_{x=0}^{7}\sum_{y=0}^{7} f(x,y)\cos\left[\frac{(2x+1)i\pi}{16}\right]\cos\left[\frac{(2y+1)i\pi}{16}\right] \tag{3.3}$$

The resultant matrix depends on the horizontal, vertical and diagonal frequencies due to DCT that uses cosine functions. So, the resulting matrix of one color image has a large value for the first value and zeros for the other elements. However, with a lot of different frequencies in the black image produces a very random looking matrix.

### 3.2.1 DCT Matrix

The DCT matrix of the Eq. (3.1) can be obtained using the following Eq. (3.4).

$$U_{i,j} = \begin{cases} \dfrac{1}{\sqrt{N}} & if\ i = 0 \\ \sqrt{\dfrac{2}{N}} \cos\left[\dfrac{(2j+1)i\pi}{2N}\right] & if\ i > 0 \end{cases} \tag{3.4}$$

$$U = \begin{bmatrix} .3536 & .3536 & .3536 & .3536 & .3536 & .3536 & .3536 & .3536 \\ .4904 & .4157 & .2778 & .0975 & -.0975 & -.2778 & -.4157 & -.4904 \\ .4619 & .1913 & -.1913 & -.4619 & -.4619 & -.1913 & .1913 & .4619 \\ .4157 & -.0975 & -.4904 & -.2778 & .2778 & .4904 & .0975 & -.4157 \\ .3536 & -.3536 & -.3536 & .3536 & .3536 & -.3536 & -.3536 & .3536 \\ .2778 & -.4904 & .0975 & .4157 & -.4157 & -.0975 & .4904 & -.2778 \\ .1913 & -.4619 & .4619 & -.1913 & -.1913 & .4619 & -.4619 & .1913 \\ .0975 & -.2778 & .4157 & -.4904 & .4904 & -.4157 & .2778 & -.0975 \end{bmatrix} \tag{3.5}$$

The $U$ matrix is of block 8x8 meaning that $N$=8. When $i$=$0$, the first row of the matrix is same $(1/\sqrt{8})$ as expected from Eq. (3.4). This is a standard 8x8 matrix for DCT.

### 3.2.2 Applying DCT on 8x8 Block

Before applying DCT on 8x8 block, we should know the pixel values of a grayscale image. The pixel values ranges from 0 to 255. The 0 value represents the pure black while pure white is represented by 255.

An image can be composed of many 8x8 blocks but here for an example purpose only one 8x8 block is taken to describe that what happens to one 8x8 block. One thing should be noted that the following 8x8 block is taken from an exemplary image. This particular block is chosen from the very-upper left corner of the image.

$$
Original\ Block = \begin{bmatrix} 154 & 123 & 123 & 123 & 123 & 123 & 123 & 136 \\ 192 & 180 & 136 & 154 & 154 & 154 & 136 & 110 \\ 254 & 198 & 154 & 154 & 180 & 154 & 123 & 123 \\ 239 & 180 & 136 & 180 & 180 & 166 & 123 & 123 \\ 180 & 154 & 136 & 167 & 166 & 149 & 136 & 136 \\ 128 & 136 & 123 & 136 & 154 & 180 & 198 & 154 \\ 123 & 105 & 110 & 149 & 136 & 136 & 180 & 166 \\ 110 & 136 & 123 & 123 & 123 & 136 & 154 & 136 \end{bmatrix} \tag{3.6}
$$

The design of DCT is such that it works on the pixel values ranging from -128 to 127. The each entry of original block is subtracted from 128 to “leveled-off”. The resultant matrix is in Eq. (3.7).

$$
N = \begin{bmatrix} 26 & -5 & -5 & -5 & -5 & -5 & -5 & 8 \\ 64 & 52 & 8 & 26 & 26 & 26 & 8 & -18 \\ 126 & 70 & 26 & 26 & 52 & 26 & -5 & -5 \\ 111 & 52 & 8 & 52 & 52 & 38 & -5 & -5 \\ 52 & 26 & 8 & 39 & 38 & 21 & 8 & 8 \\ 0 & 8 & -5 & 8 & 26 & 52 & 70 & 26 \\ -5 & -23 & -18 & 21 & 8 & 8 & 52 & 38 \\ -18 & 8 & -5 & -5 & -5 & 8 & 26 & 8 \end{bmatrix} \tag{3.7}
$$

As, we now have all the components to perform DCT, so the matrix multiplication is used as given in Eq. (3.8)

$$
F = UNU^t \tag{3.8}
$$

To transform the rows, initially matrix $N$ is multiplied to DCT matrix $U$. Then to transform the columns, it is multiplied on the right by the transpose of the DCT matrix. This transformation of rows and columns results the matrix $F$ in Eq. (3.9).

$$F = \begin{bmatrix} 162.3 & 40.6 & 20.0 & 72.3 & 30.3 & 12.5 & -19.7 & -11.5 \\ 30.5 & 108.4 & 10.5 & 32.3 & 27.7 & -15.5 & 18.4 & -2.0 \\ -94.1 & -60.1 & 12.3 & -43.4 & -31.3 & 6.1 & -3.3 & 7.1 \\ -38.6 & -83.4 & -5.4 & -22.2 & -13.5 & 15.5 & -1.3 & 3.5 \\ -31.3 & 17.9 & -5.5 & -12.4 & 14.3 & -6.0 & 11.5 & -6.0 \\ -0.9 & -11.8 & 12.8 & 0.2 & 28.1 & 12.6 & 8.4 & 2.9 \\ 4.6 & -2.4 & 12.2 & 6.6 & -18.7 & -12.8 & 7.7 & 12.0 \\ -10.0 & 11.2 & 7.8 & -16.3 & 21.5 & 0.0 & 5.9 & 10.7 \end{bmatrix} \qquad (3.9)$$

Now, $F$ matrix has 64 coefficients. These coefficients can be represents as $c_{ij}$, where $i,j$ ranges from 0 to 7. The $c_{00}$ is the lowest frequency component while all other components are high frequency components, $c_{77}$ correspond to highest frequency component. The low frequency components has their significance as human eye is sensitive to them and results from quantization step will elaborate this fact.

## 3.3 Quantization

This quantization step is very important in DCT image compression because selecting the particular quantization metrics gives the different quality and compression levels. The quality levels has a range from 1 to 100. Where 1 provides the worst quality and highest compression while 100 gives the best quality and lowest compression. The experiments performed previously considering human visual system gives the standard quantization matrix with the quality level of 50. This standard matrix provides both very good image quality and high compression.

$$
Q_{50} = \begin{bmatrix}
16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\
12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\
14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\
14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\
18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\
24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\
49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\
72 & 92 & 95 & 98 & 112 & 100 & 103 & 99
\end{bmatrix} \tag{3.10}
$$

The quantization can be obtained by using Eq. (3.11). Each entry of transformed image matrix $F$ in Eq.(3.9) is divided by each element of the quantization matrix $Q_{50}$ in Eq. (3.10). The result of each entry will then be rounded to nearest integer.

$$
C_{i,j} = round\left(\frac{F_{i,j}}{Q_{i,j}}\right) \tag{3.11}
$$

$$
C = \begin{bmatrix}
10 & 4 & 2 & 5 & 1 & 0 & 0 & 0 \\
3 & 9 & 1 & 2 & 1 & 0 & 0 & 0 \\
-7 & -5 & 1 & -2 & -1 & 0 & 0 & 0 \\
-3 & -5 & 0 & -1 & 0 & 0 & 0 & 0 \\
-2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix} \tag{3.12}
$$

The coefficients on the upper-left corner of the matrix $C$ is the low frequency components which are sensitive to human eye. Moreover, the zero components represents higher frequencies that are less important and will be discarded. So, only the non-zero components can be used to reconstruct the image.

## 3.4 ZigZag Reordering

After the DCT coefficients have been quantized, DCT is divided into DC coefficient and AC coefficients as shown in Figure 3-2.

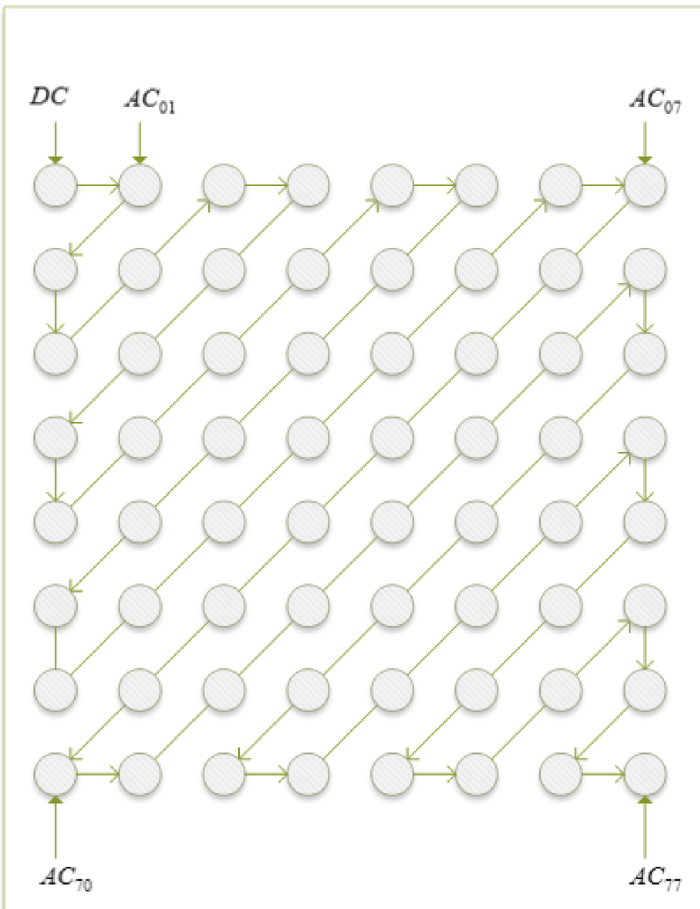



Figure 3-2 ZigZag Scan

In our case, the DC coefficient is of value equals to 10. While all the other coefficients are AC coefficients as mentioned in Eq. (3.12).

The DC coefficient in every 8x8 block is usually large and varies across the blocks. Moreover, DC coefficients are differential pulse code modulation (DPCM) coded and then they are entropy encoded with AC coefficients. Both DC and AC values are encoded in same manner but due to long runs in AC coefficients, additional run length process is applied to decrease their redundancy.

To use the entropy encoding the order of the quantized frequency coefficients is changed. The reason for doing this is to move the coefficients which are more likely to be zero to the end. This is known as ZigZag scan. All the quantized coefficients are reordered as presented in Figure 3-2.

The 8x8 matrix $C$ is mapped into a 1-D format i.e., 8x8 to a 1x64 vector using ZigZag as shown in Eq. (3.13).

$$\begin{aligned} ZigZag = [&4 \quad 3 \quad -7 \quad 9 \quad 2 \quad 5 \quad 1 \quad -5 \quad -3 \quad -2 \quad -5 \quad 1 \quad 2 \quad 1 \\ &0 \quad 1 \quad -2 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad -1 \quad -1 \quad EOB] \end{aligned} \tag{3.13}$$

A size of zero in the AC coefficient is to show that the rest of 8x8 block is zeros called as end of block (EOB).

## 3.5 Entropy Encoding

In this last step, the number of bits required to characterize the quantized coefficients that are encoded using both the run-length encoding (RLE) and Huffman encoding [40][41][38]. In RLE sequences a value that comes in repeated data elements are substituted with data value and run length. However, Huffman encoding is a variable length coding (VLC). In Huffman encoding, the term codeword is used as each symbol is represented by codeword. The length of these codewords varies. A symbol that comes rarely provided with a long codeword while the symbols that appears frequently assigned with a short codeword. Entropy encoding applies to both the DC and AC coefficients which are as follows:

### 3.5.1 Entropy Encoding: DC Components

The DC coefficient is initially predicted from the DC coefficient of the previous block. There are two parts of the DC coefficients such as symbol for the category and the value within that category. The value in the category is used to define the additional bits. There are different categories that are used for encoding is presented in Table 3-1.

Table 3-1 Categories for the DC Coefficients [41]

| SIZE | Value | Codeword | Length |
|---|---|---|---|
| 0 | 0 | 00 | 2 |
| 1 | -1, 1 | 010 | 3 |
| 2 | -3, -2, 2,3 | 011 | 3 |
| 3 | -7,...,-4,4,...,7 | 100 | 3 |
| 4 | -15,...,-8,8,...,15 | 101 | 3 |
| 5 | -31,...,-16,16,...,31 | 110 | 3 |
| 6 | -63,...,-32,32,...,63 | 1110 | 4 |
| 7 | -127,...,-64,64,...,127 | 11110 | 5 |
| 8 | -255,...,-128,128,...,255 | 111110 | 6 |
| 9 | -511,...,-256,256,...,511 | 1111110 | 7 |
| 10 | -1023,...,-512,512,...,1023 | 11111110 | 8 |
| 11 | -2047,...,-1024,1024,...,2047 | 111111110 | 9 |

For example, the DC coefficient taken from Eq. (3.12) is 10. Now we will see that how we can encode this DC coefficient. We have to see that where this value 10 placed in Table 3-1. It corresponds to SIZE=4, which has codeword 101 and the binary value of 10 is 1010 so the new representation is **1011010**. The difference between DC coefficients of adjacent blocks varies but this difference is small.

### 3.5.2 Entropy Encoding: AC Components

The AC components are encoded using both the RLE and Huffman encoding. In quantized matrix, the coefficients has numerous zero values. To encode these coefficients every Huffman symbol has divided into two parts such as run-length and the category for the non-zero coefficient.

Run-length can be define as number of zero coefficients before the first non-zero coefficient. In Huffman table, these two parts are known as run and size correspondingly. In encoding of AC coefficients if the number of successive zeros are more than maximum run-length i.e., 15, a special codeword is used names as zero run length (ZRL) code. This code represents 16 consecutive zero coefficients. While all the other coefficients having value zero named as end of block (EOB). The Huffman encoding table for AC coefficients for different categories is given in Table 3-2. While the Huffman Run/Size pairs table is presented in Table 3-3.

Table 3-2 Categories for the AC Coefficients

| SIZE | Value |
|---|---|
| 1 | -1, 1 |
| 2 | -3, -2, 2,3 |
| 3 | -7,...,-4,4,...,7 |
| 4 | -15,...,-8,8,...,15 |
| 5 | -31,...,-16,16,...,31 |
| 6 | -63,...,-32,32,...,63 |
| 7 | -127,...,-64,64,...,127 |
| 8 | -255,...,-128,128,...,255 |
| 9 | -511,...,-256,256,...,511 |
| 10 | -1023,...,-512,512,...,1023 |

Table 3-3 Huffman Table for Run/Size

| **Run/ SIZE** | **Length** | **Codeword** | **Run/SIZE** | **Length** | **Codeword** |
|---|---|---|---|---|---|
| 0/0 | 4 | 1010 | 1/1 | 4 | 1100 |
| 0/1 | 2 | 00 | 1/2 | 5 | 11011 |
| 0/2 | 2 | 01 | 1/3 | 7 | 1111001 |
| 0/3 | 3 | 100 | 1/4 | 9 | 111110110 |
| 0/4 | 4 | 1011 | 1/5 | 11 | 11111110110 |
| 0/5 | 5 | 11010 | 1/6 | 16 | 11111111100 00100 |
| 0/6 | 7 | 1111000 | 1/7 | 16 | 11111111100 00101 |
| 0/7 | 8 | 11111000 | 1/8 | 16 | 11111111100 00110 |
| 0/8 | 10 | 1111110110 | 1/9 | 16 | 11111111100 00111 |
| 0/9 | 16 | 11111111100 00010 | 1/A | | 11111111100 01000 |
| 0/A | 16 | 11111111100 00011 | …15/A | More | Such rows |

AC components are encoded uses Eq. (3.13). For instance, the value of the AC coefficient is 4. If we see the value in Table 3-2, it belongs to SIZE=3. Now, see the SIZE=3 in the Table 3-3 which gives the run/size pair is 0/3. So, the value 4 can be read as (0/3)4. Whereas the code for 0/3 is 100 and the binary for 4 is 0100. So the new representation is **1000100**.

# CHAPTER 4

# DISCRETE WAVELET TRANSFORM

## 4.1 Overview

The discrete wavelet transform (DWT) technique is a method in which two filters have been used to decompose a signal i.e., sequence of digital samples [42]. These filters are low-pass filter *L* and high-pass filter *H.* As, image is a two-dimensional signal, so 2-D equivalent of DWT is performed. This can be done by applying both the filters to the lines of samples, row-by-row, then again passing the output through the same filters to the columns. Thus, the image is distributed into non-overlapping multi-resolution sub-bands such as *LL*, *LH*, *HL* and *HH*. The division of an image into sub-bands is on first level and these four sub-bands are further divided into two type of DWT coefficients. *LL* represents the coarse-scale DWT coefficients while *LH*, *HL*, *HH* are fine-scale DWT coefficients. The *LL* contains the low-pass information and the other sub-bands contain the high-pass information of horizontal, vertical and diagonal orientation. In the next step, to reduce the number of bits that are required to show an image, quantization has been used. This step will surely lower the image quality but unlike the DCT because the quality will depend on the value of quantization used. After that, entropy encoding is performed [43]. In DWT, arithmetic encoder is used to perform encoding. This coder scheme is different from others because it encodes source message fully and take it as a single number. Overview of DWT technique is shown in Figure 4-1 .

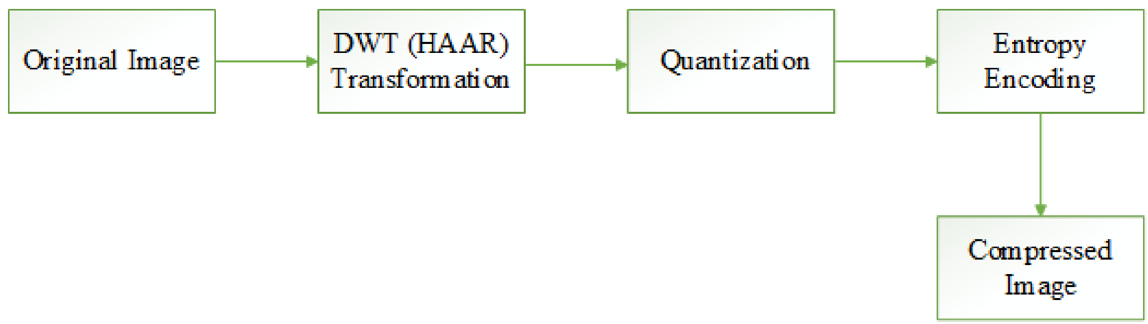



Figure 4-1 DWT Compression Method

## 4.2 Discrete Wavelet Transform

Discrete wavelet transform (DWT) technique exploits both the spatial and frequency correlation of the data using contractions of the parent wavelet on an input image. DWT provides support to the analysis of multiresolution data meaning that it can be applied to different level according to the requirement. This characteristic allows the data to transmit without the need of extra storage.

In DWT, an input image is passed through both the filters i.e., low pass and high pass in both the directions, rows and columns. After that, the outputs are down sampled by 2. By using these filters in one stage, an image is breakdown into four sub-bands i.e., *LL*, *LH*, *HL* and *HH* as shown in Figure 4-2.

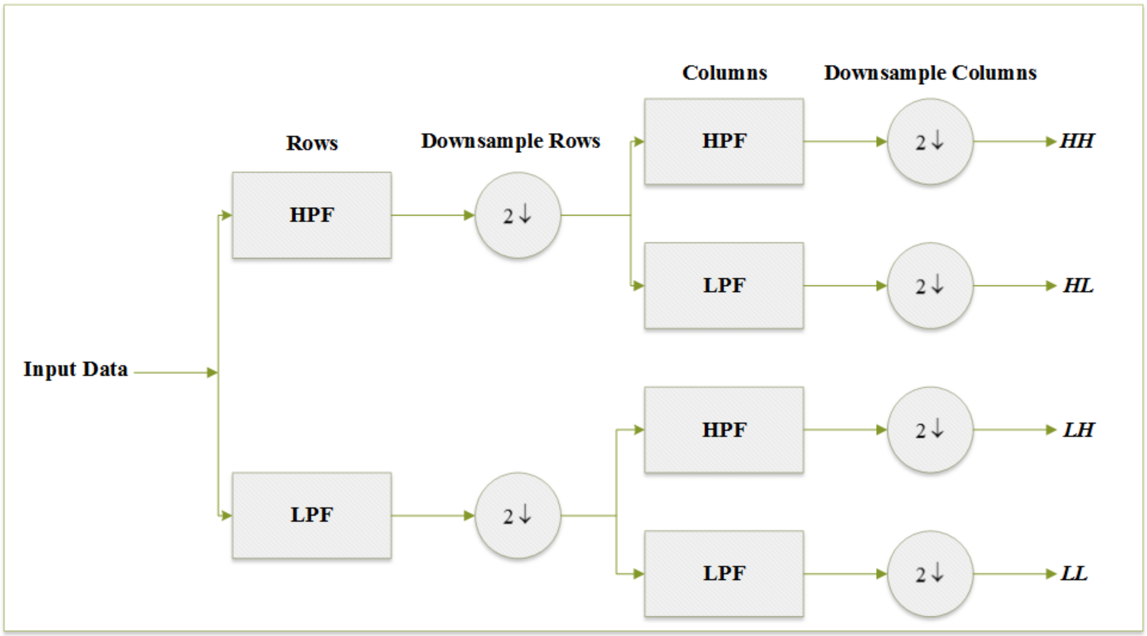



Figure 4-2 Block Diagram of 2D-DWT

Each resolution has three types of detail images such as horizontal *(HL)*, vertical *(LH)* and diagonal *(HH)*. The operation can be repeatedly performed on *LL* band using the same filter stage. Therefore, a typical 2D-DWT structure used in image compression is presented in Figure 4-3. *LL* means that the first letter is transform in row while the second letter represents the transform in column.

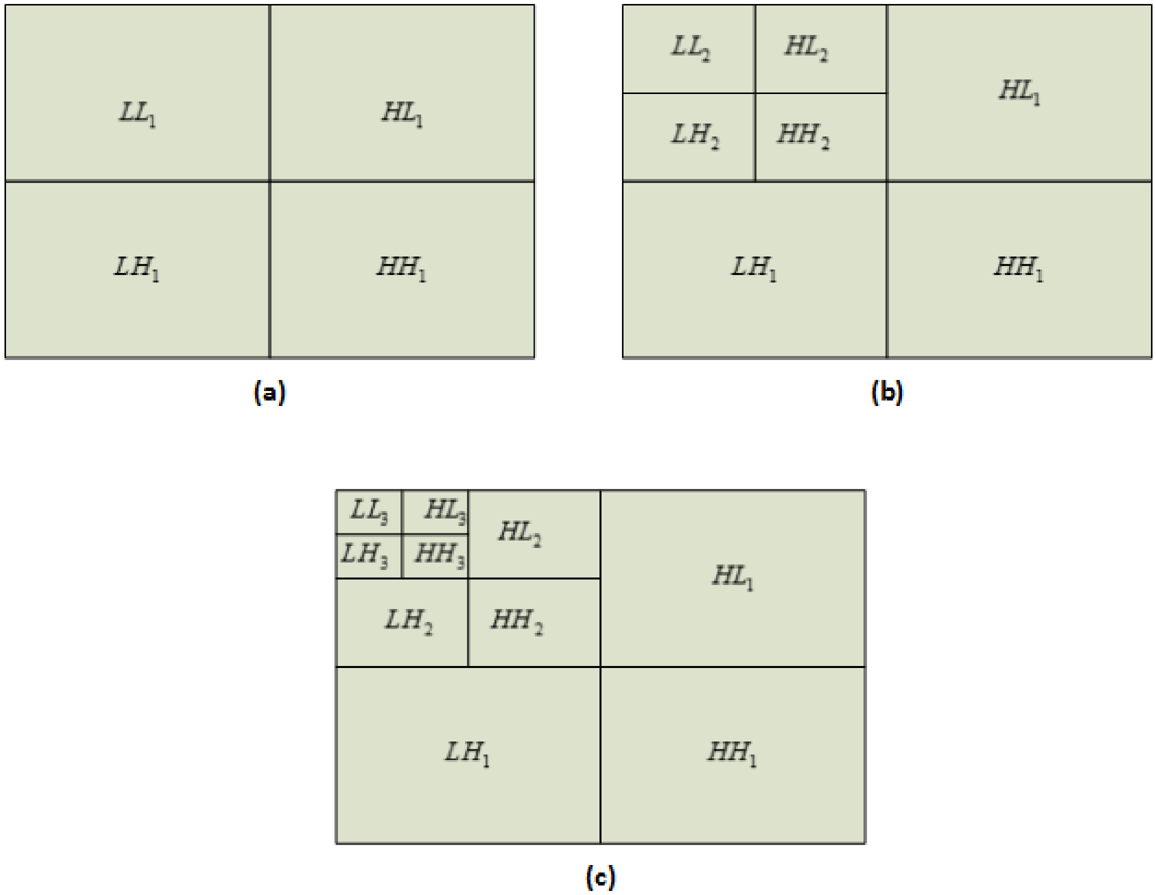



Figure 4-3 2D-DWT Structure of Wavelet Decomposition [42]

As to implement DWT technique in detail, following steps are used:

1. HAAR wavelet transform
    a. Thresholding
2. Quantization
3. Encoding

### 4.2.1 HAAR Wavelet Transform

Alfred Haar was a Hungarian mathematician. The wavelet theory was introduced by him. In discrete form, the Haar wavelets are related to the mathematical operation known as Haar transform [44][45]. This is one of the simplest method to transform from the spatial domain to frequency domain. Each signal in Haar transform is divided into two components such as averaging (approximation) and differencing (detail) [46]. To perform the Haar wavelet transform, let us take the 8x8 matrix from the part of an image as presented in Figure 4-4 and the matrix of 8x8 image is given in Eq. (4.1).

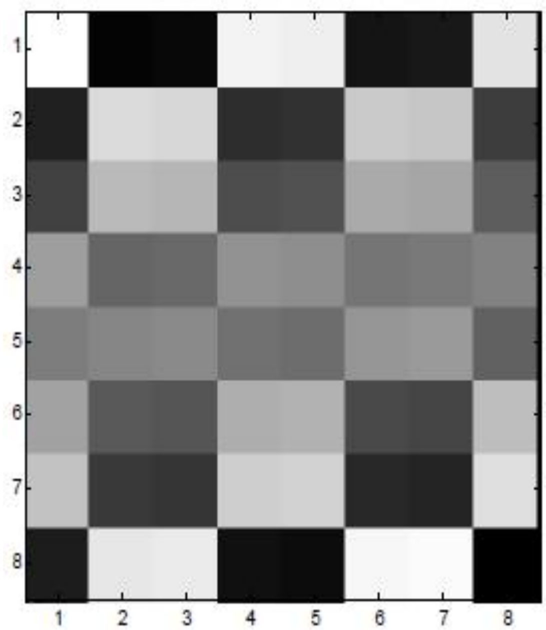



Figure 4-4 Original Image

$$Original = \begin{bmatrix} 64 & 2 & 3 & 61 & 60 & 6 & 7 & 57 \\ 9 & 55 & 54 & 12 & 13 & 51 & 50 & 16 \\ 17 & 47 & 46 & 20 & 21 & 41 & 42 & 24 \\ 40 & 26 & 27 & 37 & 36 & 30 & 31 & 33 \\ 32 & 34 & 35 & 29 & 28 & 38 & 39 & 25 \\ 41 & 23 & 22 & 44 & 45 & 19 & 18 & 48 \\ 49 & 15 & 14 & 52 & 53 & 11 & 10 & 56 \\ 8 & 58 & 59 & 5 & 4 & 62 & 63 & 1 \end{bmatrix} \tag{4.1}$$

Now, the operation of averaging and differencing is performed. This operation is used to get the new matrix that represent the same image but in a more concise way. Here, for simplicity only the first row of an original matrix is taken to describe this operation. The first row is given below in Eq. (4.2). As, original matrix is 8x8 so the process will have three steps $(2^3 = 8)$.

$$First\ Row = \begin{bmatrix} 64 & 2 & 3 & 61 & 60 & 6 & 7 & 57 \end{bmatrix} \tag{4.2}$$

- **Step 1:**

  In this step, take the average of each pair in Eq. (4.2) and place the resultant values in the first four places of the new row. While the rest of the four numbers are differences that can be computed by taking the difference of the first entry in each pair and its corresponding average.

  Averaging= (64+2)/2 =33, (3+61)/2= 32, (60+6)/2= 33, (7+57)/2= 32

  Differencing= 64-33= 31, 3-32= -29, 60-33= 27, 7-32= -25

  Thus, the transformed row is in Eq. (4.3).

$$Transformed\ Row = [33 \quad 32 \quad 33 \quad 32 \quad 31 \quad -29 \quad 27 \quad -25] \tag{4.3}$$

The coefficients that comes from differencing are also known as detailed coefficients.

- **Step 2:**

  In this step, the same procedure is applied but to the first four components of the transformed row i.e., Eq. (4.3).

  Averaging= (33+32)/2= 32.5, (33+32)/2= 32.5

  Differencing= 33-32.5= 0.5, 33-32.5= 0.5

  These four coefficients (two averaging and two detail) are placed on the first four places of the resulting transformed row. But remaining four detailed coefficients are carried down as it is from the step 1. The new transformed row is presented in Eq. (4.4)

$$Transformed\ Row = [32.5 \quad 32.5 \quad .5 \quad .5 \quad 31 \quad -29 \quad 27 \quad -25] \tag{4.4}$$

- **Step 3:**

  In the last step, the averaging and differencing applies only on the first pair of the Eq. (4.4), while remaining all six coefficients have been carried down again in the new transformed row.

  Averaging= (32.5+32.5)/2= 32.5

  Differencing= 32.5-32.5= 0

  The final transformed row is given below:

$$Transformed\ Row = [32.5 \quad 0 \quad .5 \quad .5 \quad 31 \quad -29 \quad 27 \quad -25] \tag{4.5}$$

Here, we have one average coefficient at first place of the new row and seven detail coefficients.

After performing these steps to first row, we can similarly apply averaging and differencing to all rows of an entire matrix. If the averaging and differencing have been applied to all rows, we can have the matrix given in Eq. (4.6).

$$T_r = \begin{bmatrix} 32.5 & 0 & .5 & .5 & 31 & -29 & 27 & -25 \\ 32.5 & 0 & -.5 & -.5 & -23 & 21 & -19 & 17 \\ 32.5 & 0 & -.5 & -.5 & -15 & 13 & -11 & 9 \\ 32.5 & 0 & .5 & .5 & 7 & -5 & 3 & -1 \\ 32.5 & 0 & .5 & .5 & -1 & 3 & -5 & 7 \\ 32.5 & 0 & -.5 & -.5 & 9 & -11 & 13 & -15 \\ 32.5 & 0 & -.5 & -.5 & 17 & -19 & 21 & -23 \\ 32.5 & 0 & .5 & .5 & -25 & 27 & -29 & 31 \end{bmatrix} \tag{4.6}$$

Similarly, if the same operation is performed on the columns of the above matrix i.e., Eq. (4.6), we will get the final matrix presented in Eq. (4.7)

$$T_c = \begin{bmatrix} 32.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4 & -4 & 4 & -4 \\ 0 & 0 & 0 & 0 & 4 & -4 & 4 & -4 \\ 0 & 0 & .5 & .5 & 27 & -25 & 23 & -21 \\ 0 & 0 & -.5 & -.5 & -11 & 9 & -7 & 5 \\ 0 & 0 & .5 & .5 & -5 & 7 & -9 & 11 \\ 0 & 0 & -.5 & -.5 & 21 & -23 & 25 & -27 \end{bmatrix} \tag{4.7}$$

In the above matrix, the top left corner coefficient i.e., 32.5 is overall average of the all the elements in the matrix. However, all the remaining values are detailed coefficients.

#### 4.2.1.1 Thresholding

Threshold can be defined as a non-negative value ε. In the transformed matrix given in Eq. (4.7) any detail coefficient whose value is less than or equal to threshold value is set to zero. This will increase the number of zeros in the matrix which increase the compression. As we know that, we can remove some information and still able to achieve a good approximation of our matrix. For instance, we set a threshold value ε=5. That means the detail coefficients of matrix in Eq. (4.7) whose values that are less or equal to 5 will be reset to zero. So, we will get the matrix presented below in Eq. (4.8).

$$T = \begin{bmatrix} 32.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 27 & -25 & 23 & -21 \\ 0 & 0 & 0 & 0 & -11 & 9 & -7 & 0 \\ 0 & 0 & 0 & 0 & 0 & 7 & -9 & 11 \\ 0 & 0 & 0 & 0 & 21 & -23 & 25 & -27 \end{bmatrix} \tag{4.8}$$

### 4.2.2 Quantization

Quantization usually converts the sequence of floating number to integer sequence. The simplest form of quantization is to round to the nearest integer. Another way is to multiply with some *K* constant and then round to nearest integer [47]. The low frequency sub-bands have more information. On the other hand, the high frequency sub-bands have relatively lesser information. However, the quantization in image compression is used in low frequency components rather than high frequency components because low frequency components has more information.

### 4.2.3 Arithmetic Coding

Arithmetic coding scheme is also variable-length encoding similar to Huffman encoding [48]. It is used in data compression without the loss of data. Unlike other encoding schemes, it encodes the full message and represents by a single floating number. While, other encoding schemes break down the message into symbols and encode each symbol separately using codeword.

# CHAPTER 5

# EXPERIMENTAL IMPLEMENTATION OF DCT & DWT

In this chapter, real-time implementation of DCT and DWT has been described. Then, the flow charts of developed code for both techniques has been presented. We also explained the different scenarios and network topologies for our experiments. Finally, the performance metrics for experimentation are explained.

## 5.1 Description of DCT and DWT for Real-Time Implementation

There are many libraries available for DCT and DWT compression techniques. We have selected JPEG and SPIHT for both DCT and DWT respectively. These two libraries are selected on the basis of the literature review as described in Table 2-1. The JPEG and SPIHT are the most suitable libraries for the implementation in wireless sensor networks. In our implementation, initially we have implemented basic DCT and DWT in WSN and then we have made the modifications in DCT and DWT to implement JPEG and SPIHT respectively. The modification we have done in DCT to implement JPEG is that we have changed the encoding scheme. The basic encoding scheme for DCT is arithmetic encoding. But we have changed it to the combination of Huffman and Run-length encoding scheme. We have selected both the Huffman and Run-length encoder to achieve higher compression ratio because JPEG is considered the compression scheme with higher compression ratio. In order to implement SPIHT in WSN we have add SPIHT module in DWT process. This SPIHT module allows us to implement SPIHT in WSN. Generally, the DWT is considered the fast compression scheme but SPIHT library in DWT has higher processing speed and that is why we have made the modification in DWT to implement this library. The implementation of above mentioned two libraries are very much fulfil the requirement of WSN, as they both use reasonable amount of power and memory consumption.

## 5.2 Flow Charts of DCT on WSN

In this section, we have explained the developed code with the help of flow charts of both compression and decompression as shown in Figure 5-1 and Figure 5-2 respectively.

### 5.2.1 DCT Compression

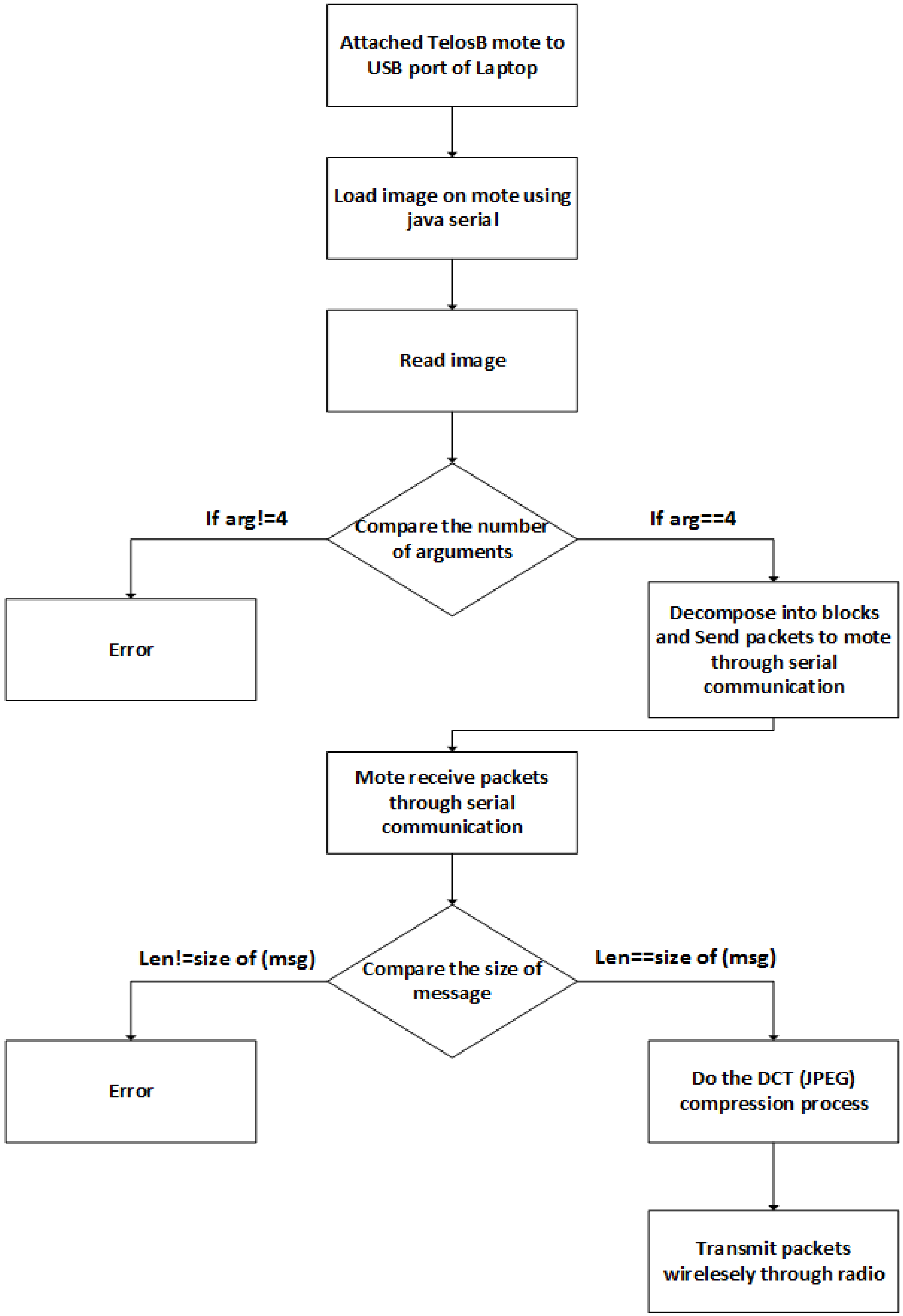



Figure 5-1 DCT Compression Developed Code

### 5.2.2 DCT Decompression

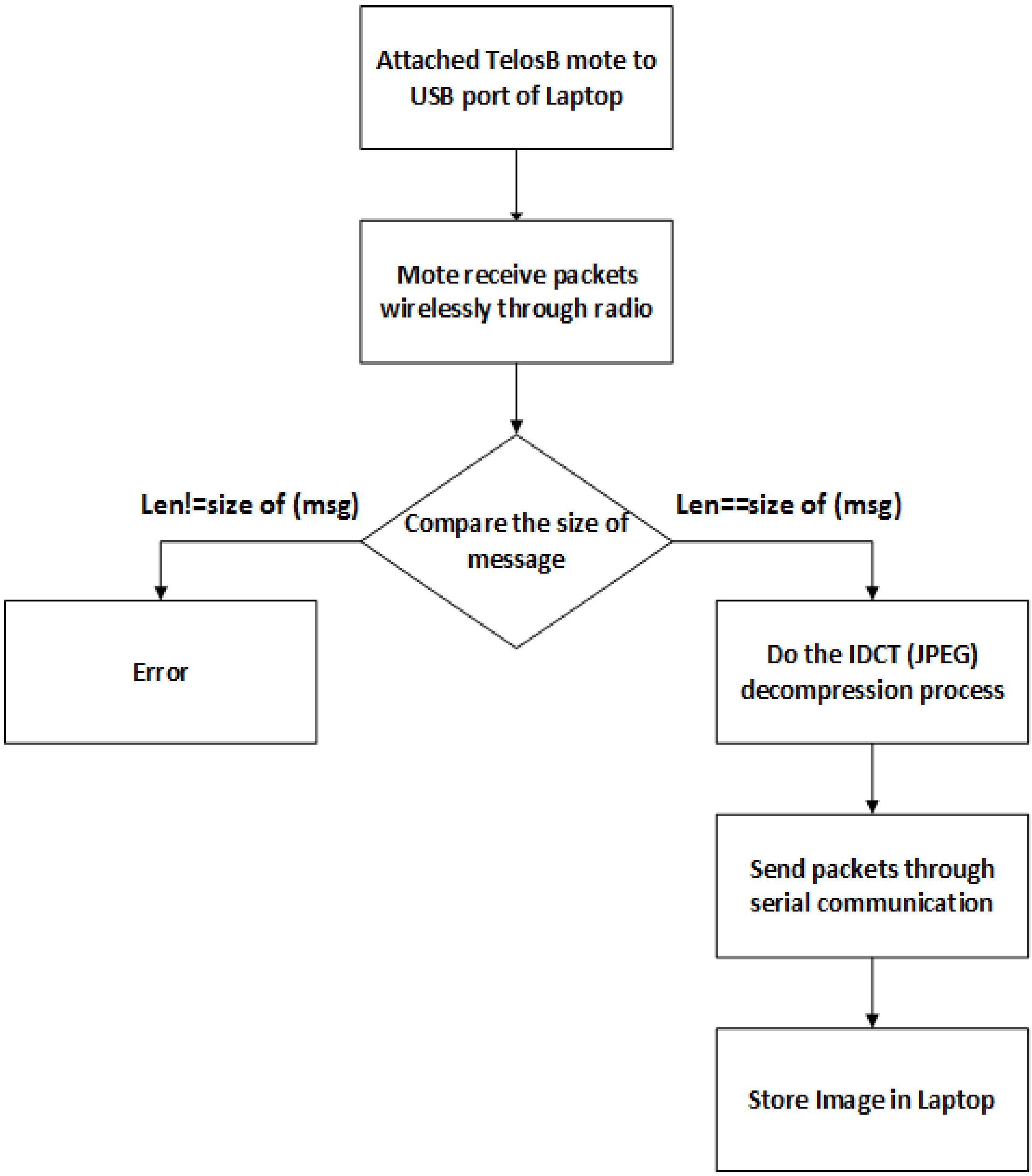



Figure 5-2 DCT Decompression Developed Code

## 5.3 Flow Charts of DWT on WSN

The flow charts of compression and decompression of DWT developed code are presented in Figure 5-3 and Figure 5-4.

### 5.3.1 DWT Compression

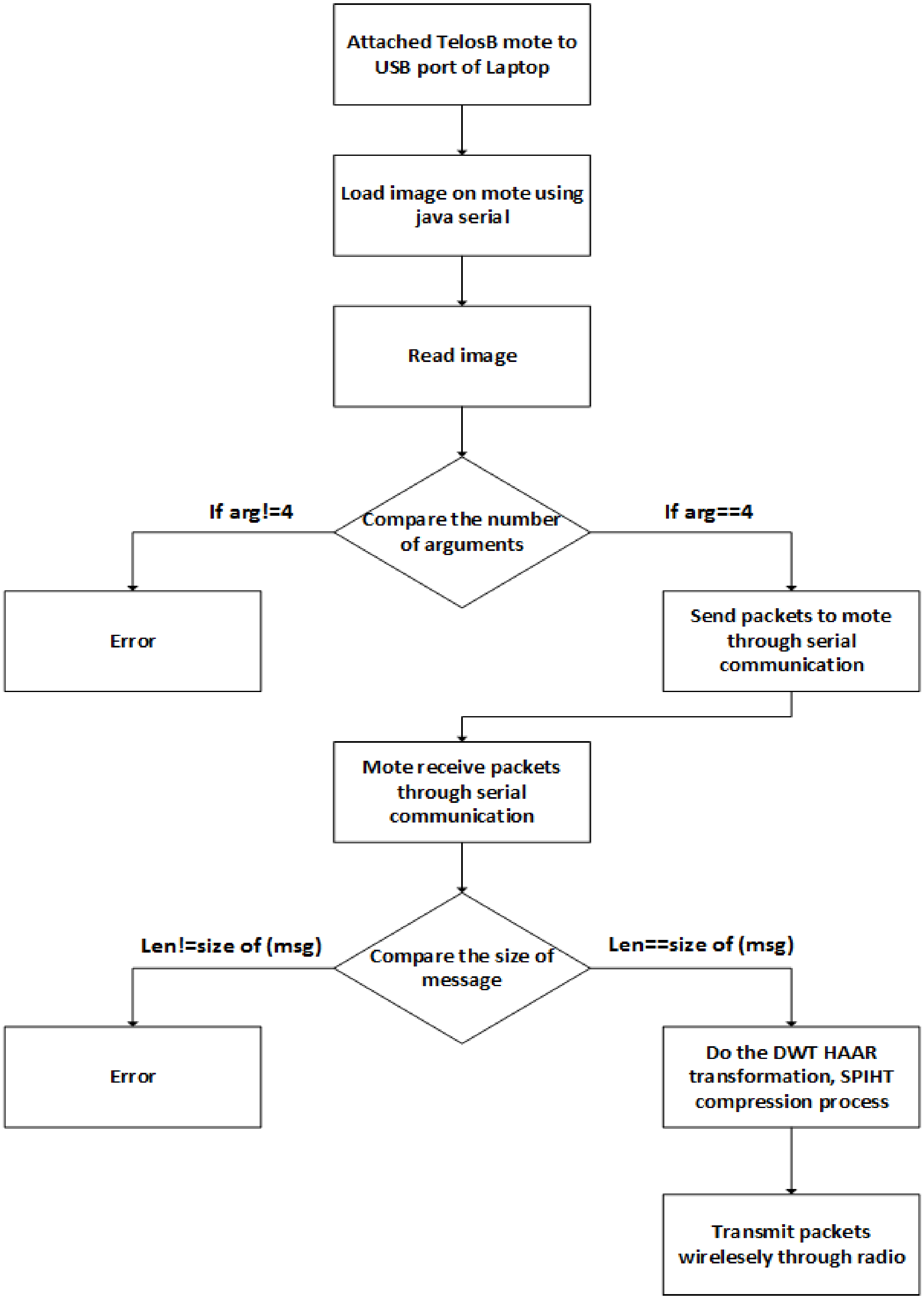



Figure 5-3 DWT Compression Developed Code

### 5.3.2 DWT Decompression

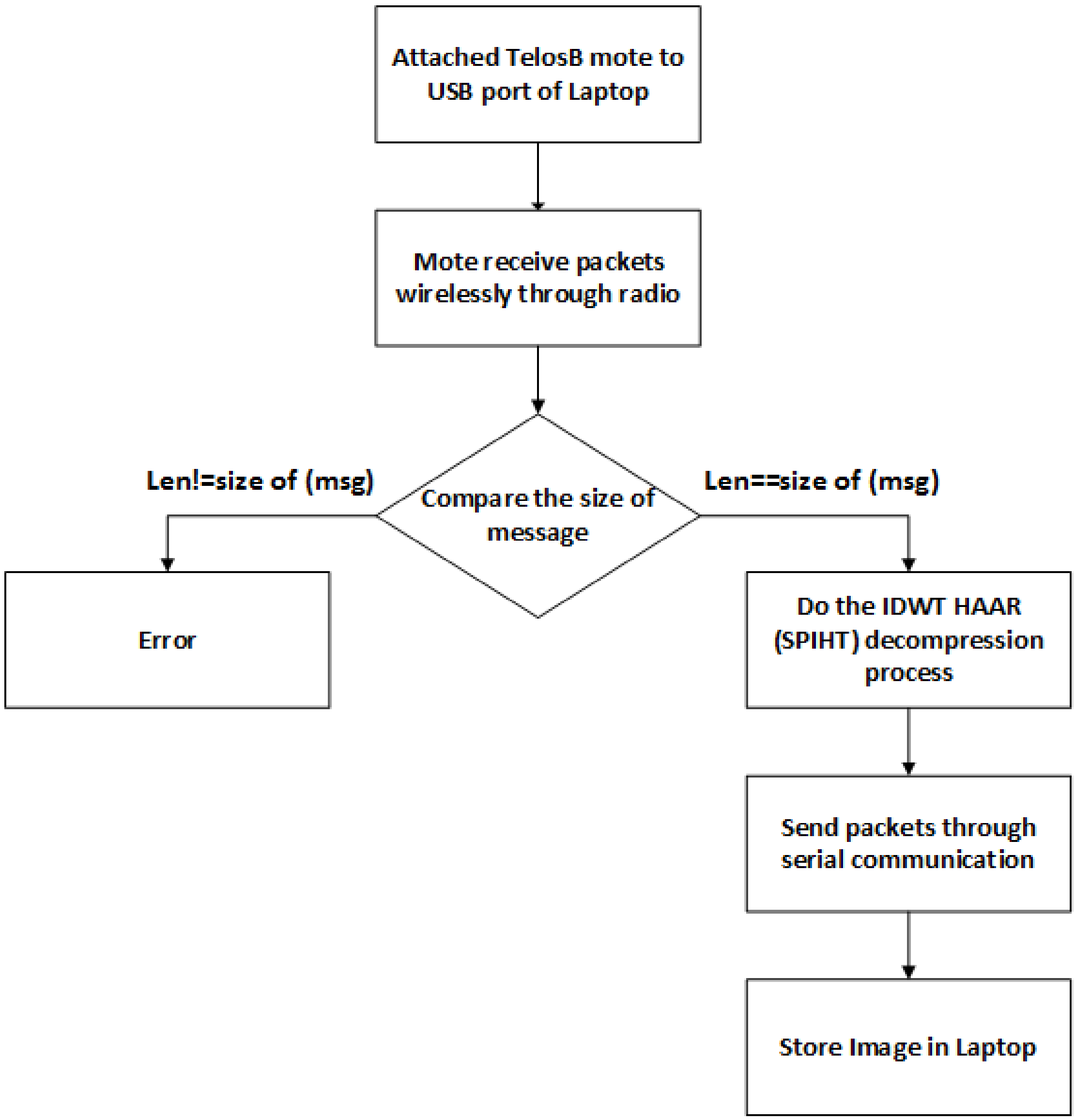



Figure 5-4 DWT Decompression Developed Code

## 5.4 Experimental Scenarios of DCT and DWT

Two different scenarios are considered: single-hop network and multi-hop network. In a single-hop network, there is no intermediate node between the sender and the receiver. However, in a multi-hop network there are intermediate nodes. The compression and decompression is applied on the sender and the receiver respectively. In both scenarios, the techniques DCT and DWT have been analyzed on PSNR, compression ratio, throughput, delay and battery lifetime. The average number of iterations per scenario is 10.

### 5.4.1 Single-Hop Network

In this scenario, we have two TelosB motes: a sender and a receiver. Each TelosB mote is attached to USB port of the computer. The compression is applied on the sender side while decompression is applied on the receiver side. The communication between motes is based on standard protocols 802.15.4 physical (PHY) and media access layers (MAC) for low-rate wireless personal area networks (LR-WPANs). The network topology is shown in Figure 5-5.

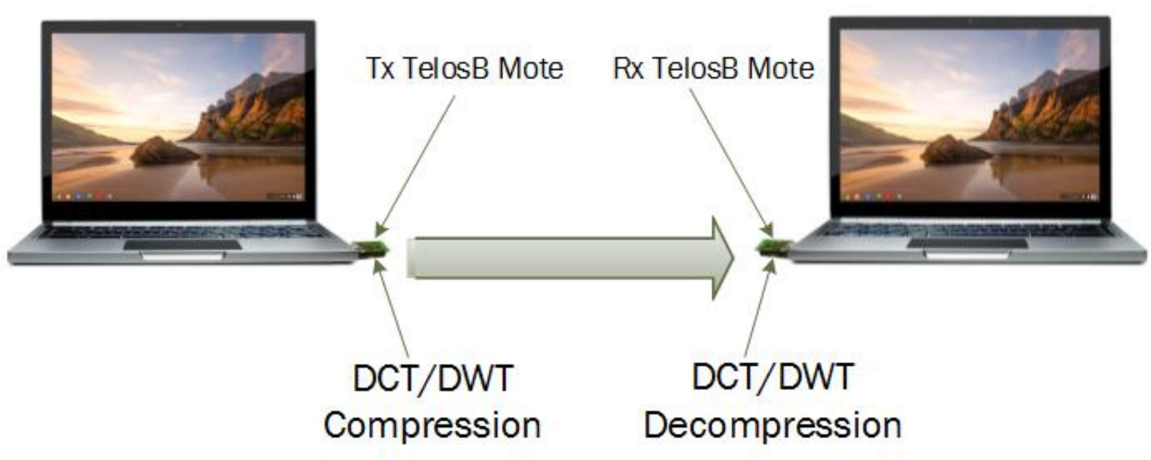



Figure 5-5 Single-Hop Network

The parameters we have set to analyze the performance are summarized in Table 5-1.

Table 5-1 Parameters for Single-Hop Scenario

| Parameters | Values |
|---|---|
| **Compression Techniques** | DCT, DWT |
| **Size of Network** | 2 nodes |
| **Image Resolution** | 32x32, 64x64, 128x128 |
| **Quantization Factor** | 10, 20, 30, 40 |
| **Hardware Mote** | TelosB |

In this research, we have used lena image for our experimentation as presented in Figure 5-6.

Figure 5-6 Lena Image for Experimentation

### 5.4.2 Multi-Hop Network

In this scenario, a network topology has different intermediate nodes between sender and receiver. The intermediate nodes are powered by battery. In multi-hop network, we have evaluated the performance of DCT and DWT having two and four intermediate nodes. For instance, network topology of multi-hop network with four intermediate nodes is illustrated in Figure 5-7.

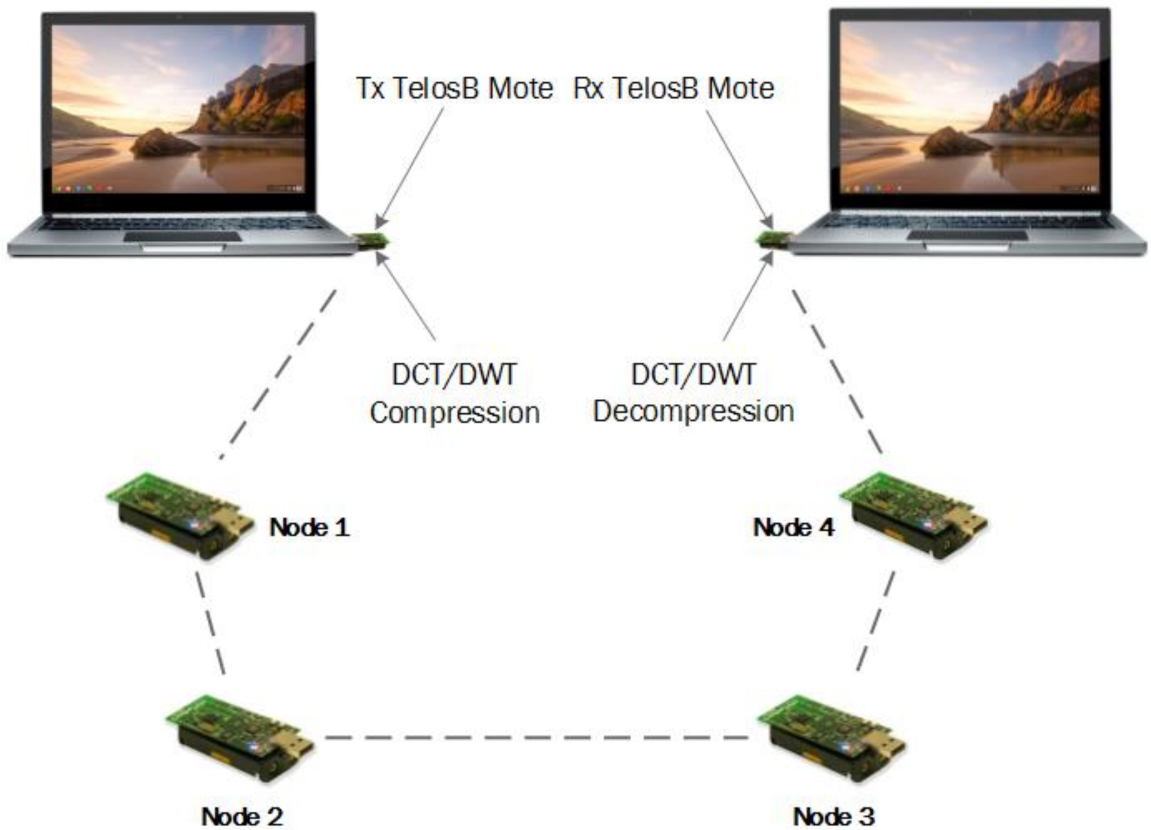



Figure 5-7 Multi-Hop Network

The parameters for this scenario are given in Table 5-2.

Table 5-2 Parameters for Multi-Hop Network

| Parameters | Values |
|---|---|
| **Compression Techniques** | DCT, DWT |
| **Size of Network** | 6 nodes |
| **Image Resolution** | 32x32, 64x64, 128x128 |
| **Quantization Factor** | 10, 20, 30, 40 |
| **Hardware Mote** | TelosB |

In this topology, the intermediate nodes are placed at a distance of 5m from one another. Compression/decompression is performed at sender/receiver side respectively and intermediate nodes are relaying the packets. But intermediate nodes have the conditions in the code to check the message size and the data type to make sure that the packets are same and the data type is same. If they are not the same then the intermediate nodes will discard the packets and will not forward it to other intermediate nodes. However, If the packets are same the routing of the packets are as follows: Sender sends the packets to node 1; node 2 forwards them to the next node and so on until it gets to the receiver as depicted in Figure 5-7.

## 5.5 Performance Evaluation Parameters

To evaluate the performance of both the compression techniques i.e., DCT and DWT, several performance metrics have been used in this research that are as follows:

- Peak Signal-to-Noise Ratio (PSNR)
- Throughput
- End-to-End (ETE) Delay
- Compression Ratio (CR)
- Battery Lifetime

### 5.5.1 Peak Signal-to-Noise Ratio (PSNR)

The term peak signal-to-noise ratio (PSNR) is an expression for the ratio between the maximum possible value (power) of a signal and the power of distorting noise that affects the quality of its representation. PSNR is typically used to measure the image quality. It describes the difference in the image quality of the decompressed image to the original image. The higher the ratio the better the technique. The mathematical expression of the PSNR is given in Eq. (5.1).

$$PSNR = 10\log_{10}\left(\frac{255^2}{MSE}\right) \tag{5.1}$$

Where,

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\left|A_0(i,j) - A_c(i,j)\right|^2$$

$$m, n = Image\,size > 0$$

$$A_0 = Original\,value$$

$$A_c = Compressed\,value$$

Here, mean square error (MSE) is the cumulative squared error between the compressed and the original image. If the value of MSE is low that means the error is low while PSNR will be higher because the relation between MSE and PSNR is inversely proportional. So the compression scheme having a higher PSNR is a good scheme because it gives a better image quality after reconstruction of the image.

### 5.5.2 Compression Ratio (CR)

It can be defined as the ratio of the number of bytes of the original image to the compressed image. It can also be described as ratio of the size of original image to the size of compressed image. The expression for compression ratio (CR) is given below. The higher the ratio the better the compression technique.

$$CR = \frac{\text{Number of bytes of Original Image}}{\text{Number of bytes of Compressed Image}} \tag{5.2}$$

### 5.5.3 Throughput

Generally, throughput is the amount of data that is successfully received over the communication channel. The higher the throughput the better the performance. To compute the throughput for DCT and DWT we use Eq.(5.3).

$$Throughput = 1 - \frac{\text{Sent Packets - Received Packets}}{\text{Sent Packets}} \tag{5.3}$$

### 5.5.4 End-to-End Delay (ETE)

End-to-End delay (ETE) delay is a time taken by a packet from source to destination over the network. ETE delay has many components such as compression delay, decompression delay, propagation delay, processing delay and MAC delay. ETE delay is presented in Eq. (5.4).

$$ETE\ Delay = [\mathrm{d}_{\mathrm{comp}} + \mathrm{d}_{\mathrm{decomp}} + \mathrm{d}_{\mathrm{prop}} + \mathrm{d}_{\mathrm{proc}} + \mathrm{d}_{\mathrm{mac}}] \tag{5.4}$$

Where,

$d_{comp}$= Compression delay

$d_{decomp}$= Decompression delay

$d_{prop}$= Propagation delay

$d_{proc}$= Procession delay

$d_{mac}$= MAC delay

### 5.5.5 Battery Lifetime

This metric is used to check how many images can be sent continuously for a full battery without recharging.

# CHAPTER 6

# RESULTS AND DISCUSSION

In this chapter, we discuss the results of DCT and DWT image compression techniques. We will first talk about the performance metrics and then we will explain the comparative analysis of both techniques.

## 6.1 PSNR for Single-Hop Network

DCT compression scheme is applied to the image shown in Figure 5-6 with different image resolutions along with different quantization factors. The image quality is visually illustrated in Table 6-1. Quantization factor (QF) has an impact on image quality.

As we increase the QF, it increases the compression ratio, which degrade the image quality as shown in Table 6-1. When QF=10, the image quality is better than that of higher QF's regardless of the image resolution. When QF=40 the reconstructed image is more distorted because of the information loss associated with compression. Similarly, DWT is applied to the input image as presented in Table 6-2.

Table 6-1 DCT Compression Technique in Single-Hop Network

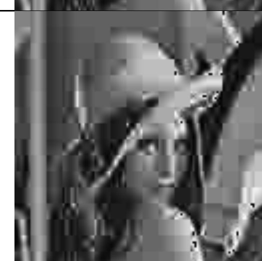
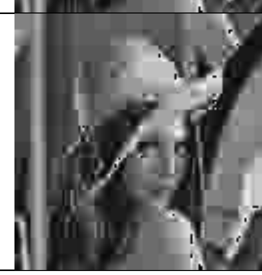
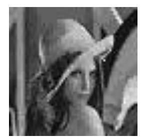
| **Image Resolution/ Quantization Factor** | **32x32** | **64x64** | **128x128** |
|---|---|---|---|
| **10** |  |  |  |
| **20** |  |  |  |
| **30** |  |  |  |
| **40** |  |  |  |

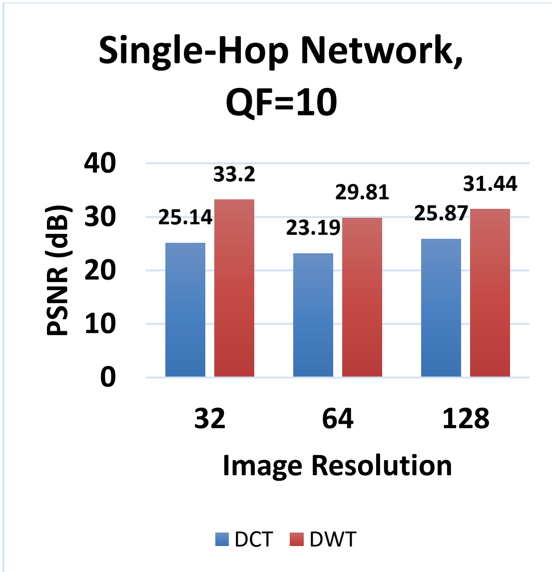
Table 6-2 DWT Compression Technique in Single-Hop Network

| **Image Resolution / Quantization Factor** | **32x32** | **64x64** | **128x128** |
|---|---|---|---|
| **10** |  |  |  |
| **20** |  |  |  |
| **30** |  |  |  |
| **40** |  |  |  |

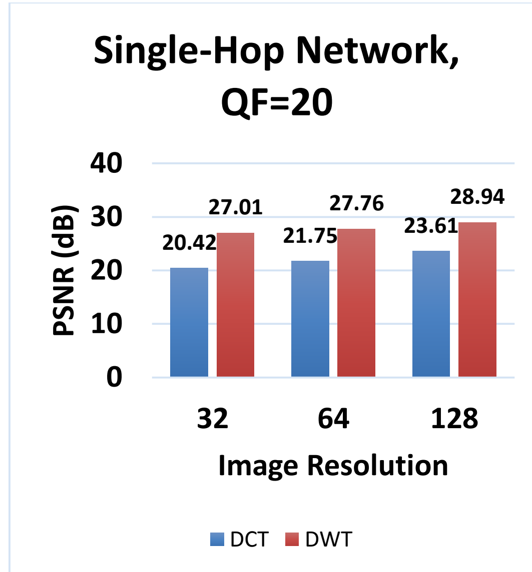
By visually analyzing both the DCT and DWT, we can infer that compression ratio and image quality is inversely proportional to each other. However, DWT is better than DCT in terms of image quality because even at QF=40, the image quality in DWT is far better than that of DCT. On the other hand, considering compression ratio DCT is better than DWT.

Both the techniques can also be analyzed in terms of PSNR as illustrated in Figure 6-1. We have computed the PSNR by using the Eq. (5.1). It is calculated using MATLAB. PSNR is signal to noise ratio. Here, “signal” is original input image and “noise” is error in reconstructed image. PSNR and image quality are directly proportional to each other.

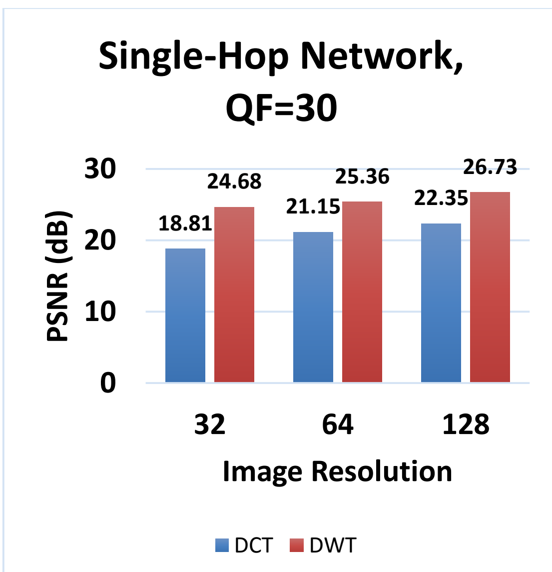
Consider a case study in which QF=10 and image resolution is 32x32. In Figure 6-1 (a), the PSNR of DCT is 25.14dB. However, PSNR of DWT is 33.2dB which indicates that the image quality for DWT is better than that of DCT.



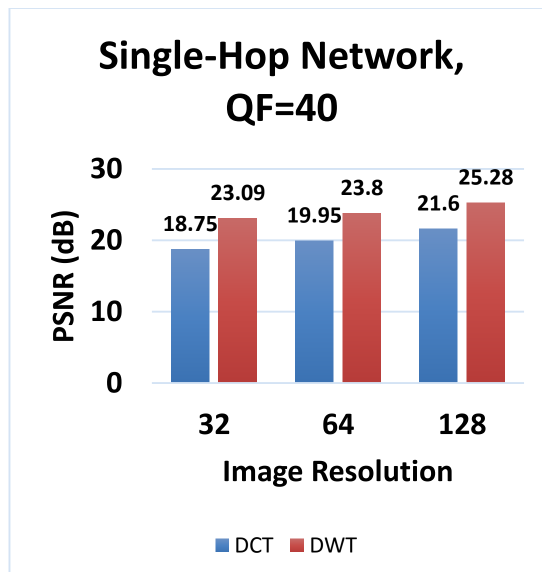

(a)




(b)




(c)




(d)

Figure 6-1 PSNR vs Image Resolution of Single-Hop Network

Likewise, if we analyze the performance of DCT and DWT for different quantization factors with different image resolutions as presented in Figure 6-1 (b), (c) and (d), the PSNR for DWT in all cases is higher than the

DCT. The higher PSNR results in better image quality. Though, DCT image quality is degraded due to having higher compression ratio than DWT which results in a poor image quality.

## 6.2 PSNR for Multi-Hop Network

To show the effect of intermediate nodes on the image quality, we have considered the scenario with four intermediate nodes. DCT and DWT compression schemes have been applied on Figure 5-6. The visual illustration of the image quality for both DCT and DWT is presented in Table 6-3 and Table 6-4 respectively. It can be clearly visualized from Table 6-4 and Table 6-3 that the image quality for DWT is better than that of DCT for all values of quantization factor or image resolution.

Table 6-3 DCT Compression Technique in Multi-Hop Network

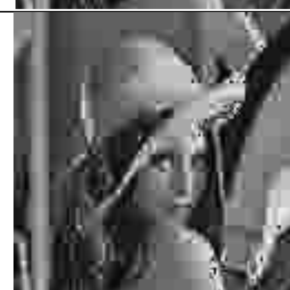
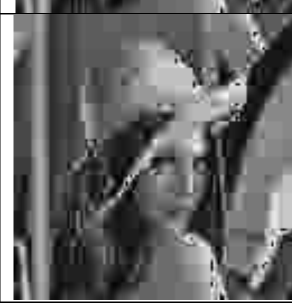
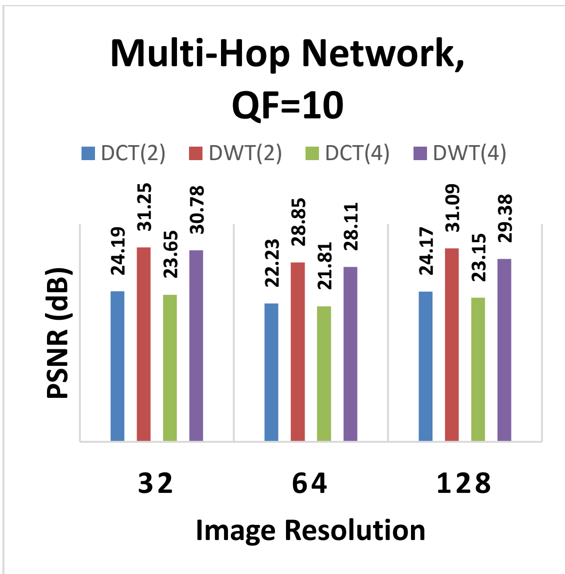
| **Image Resolution/ Quantization Factor** | **32x32** | **64x64** | **128x128** |
|---|---|---|---|
| **10** |  |  |  |
| **20** |  |  |  |
| **30** |  |  |  |
| **40** |  |  |  |

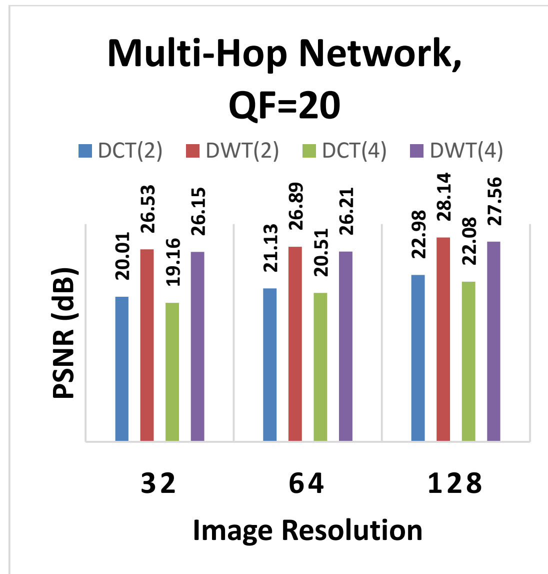
Table 6-4 DWT Compression Technique in Multi-Hop Network

| Image Resolution/ Quantization Factor | 32x32 | 64x64 | 128x128 |
|---|---|---|---|
| **10** |  |  |  |
| **20** |  |  |  |
| **30** |  |  |  |
| **40** |  |  |  |

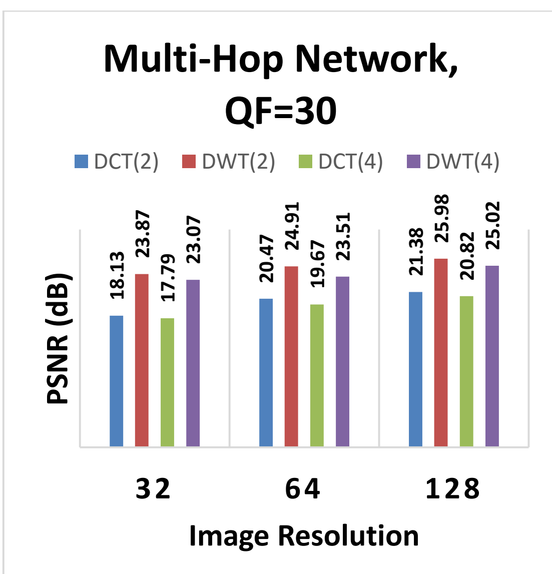
Moreover, if we compare the image quality of both techniques in multi-hop network with the single-hop network, we can see that there is more distortion in the images in multi-hop than in single-hop network. It is due to the fact that there are four intermediate nodes in former paradigm that are effecting the image quality. The image quality is analyzed on the basis of PSNR as presented in Figure 6-2.



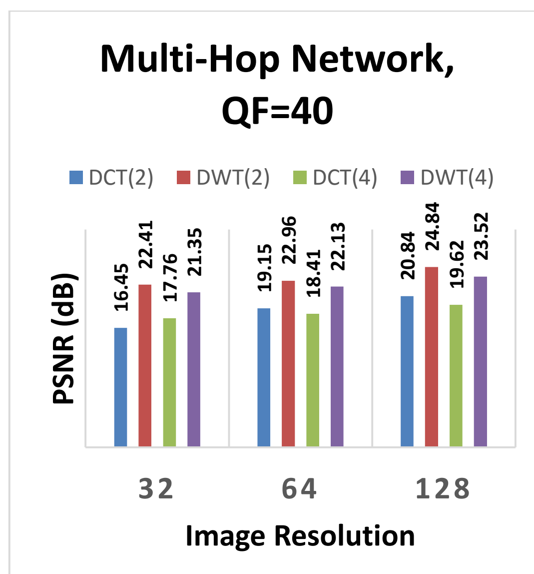

(a)




(b)




(c)




(d)

Figure 6-2 PSNR vs Image Resolution of Multi-Hop Network

In Figure 6-2, DCT(2)/DWT(2) is representing DCT/DWT used in multi-hop network with two intermediate nodes while, DCT(4)/DWT(4) is representing DCT/DWT used in corresponding network with four intermediate nodes. Now, we compare the PSNR of DCT and DWT with different intermediate nodes. Consider a case study in which QF=10, image resolution is 32x32 and intermediate nodes are two. The PSNR for DCT in Figure 6-2 (a) is 24.19dB while PSNR of DWT is 31.25dB. Again, the PSNR of DWT is higher than that of DCT. If we compare these PSNR values with the ones we achieved in single-hop network as shown in Figure 6-1 (a), we can see that intermediate nodes are degrading the image quality.

Take another case study where all the parameters are same as above except the intermediate nodes are increased to four. Here, the PSNR in Figure 6-2 (a) for DCT and DWT is 23.65dB and 30.78dB respectively. Clearly, these values are lesser than the values we got with two intermediate nodes. As the number of nodes are increasing the image quality is decreasing, so the number of nodes are affecting the image quality. In the same way, we can compare the results of all the QFs with any image resolution.

## 6.3 Comparison of DCT and DWT

### 6.3.1 Comparative Analysis of DCT and DWT for Single-Hop and Multi-Hop Network

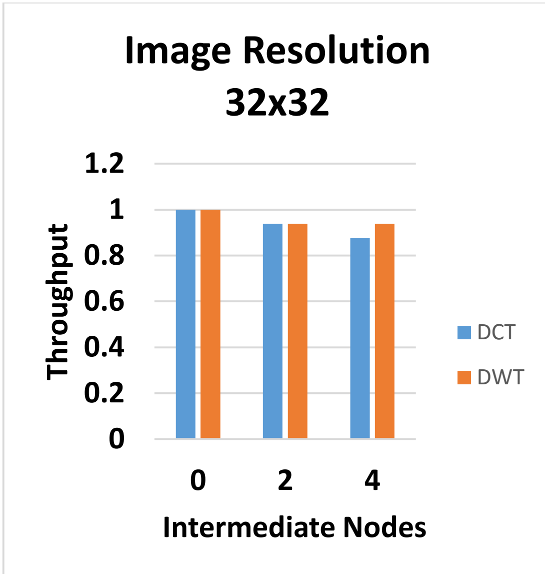
The following section presents the comparative analysis of the techniques for throughput, ETE delay, compression ratio and battery lifetime. In this section, following scenarios are considered:

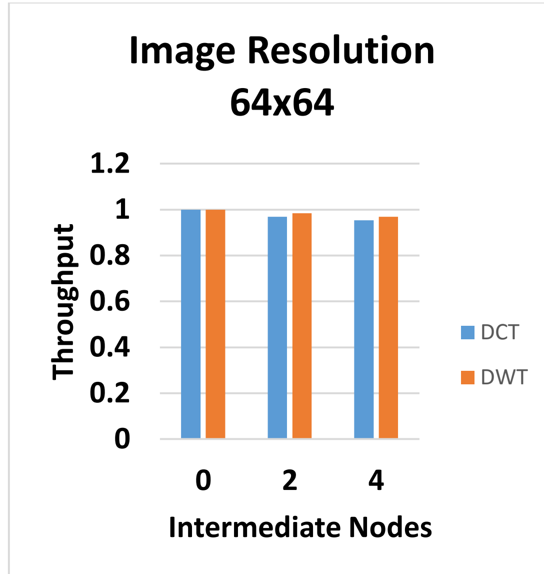
- Single-Hop Network
- Multi-Hop Network
  - Two Intermediate Nodes
  - Four Intermediate Nodes

#### 6.3.1.1 Throughput

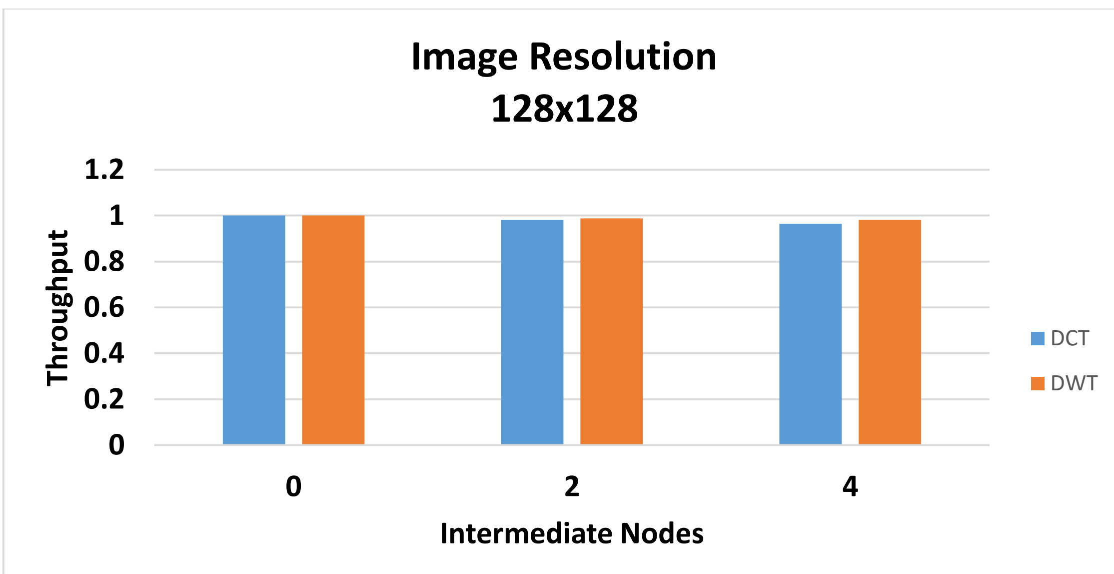
Throughput is computed to study the performance of the techniques in single-hop and multi-hop networks. It is calculated using Eq.(5.3). The results of throughput are presented in Figure 6-3.

In Figure 6-3 (a), the image size is 32x32. It is evident that the throughput of single-hop network is 1 for both techniques, but it is slightly degraded in multi-hop network. Moreover, if we increase the image size as shown in Figure 6-3 (b) and (c), the throughput will remain close to 1. However, Figure 6-3 shows that overall throughput of DWT is better than DCT in both single-hop and multi-hop scenarios.




(a) (b)




(c)

Figure 6-3 Throughput vs Intermediate Nodes

It can be concluded that if we increase the intermediate nodes, the throughput of both techniques will decrease as compared to network with no intermediate nodes. But this degradation in throughput is negligible because in our experiments we have maximum of four intermediate nodes only. This small-scale network has a little impact on throughout but if we have large-scale network then there would be more decrease in throughput. Another reason for this slight degradation is that there is no obstacle in line of sight or external interference in the network. However, DWT is better than DCT because there are less packet drops in former than the later technique which results in better performance.

#### 6.3.1.2 End-to-End Delay

End-to-End delay (ETE) delay is overall time taken by a packet from source to destination over the network. To compute the experimental ETE delay we have considered the multi-hop scenario as shown in Figure 6-4.

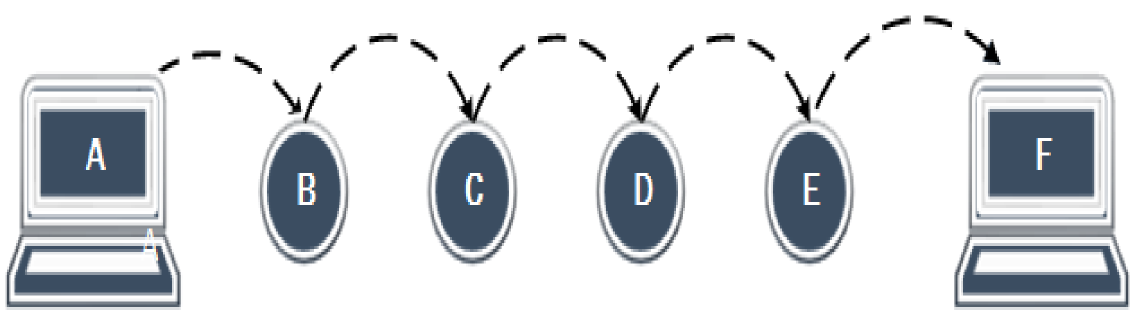



Figure 6-4 Multi-Hop Network Topology for ETE Delay

The parameters used to determine ETE delay are summarized in Table 6-5.

Table 6-5 Parameters for ETE delay

| **Parameters** | **Values** |
|---|---|
| **Compression Technique** | DCT, DWT |
| **Image Resolution** | 32x32 |
| **Packet Size** | 67 bytes |
| **Data Rate** | 250 Kbps |
| **Number of Nodes** | 6 |
| **Number of Hops** | 5 |
| **Total Packets** | 16 |
| **Intermediate Nodes Distance** | 5m |

The ETE delay consists of several delays such as compression, decompression, processing, propagation and MAC delay. The mathematical expression to compute ETE delay is given in Eq. (5.1).

$$\begin{aligned}\text{ETE Delay}_{AF} = {} & \text{Compression delay + Processing delay + Decompression...} \\ & \text{delay} + \text{Propogation delay + MAC delay}\end{aligned} \tag{5.1}$$

To determine the ETE delay for $n$ number of intermediate nodes the above Eq. (5.1) can be written as:

$$\begin{aligned}\text{ETE Delay}_{AF} = {} & \text{Compression delay + (Processing delay} \times n) + \text{Decompression...} \\ & \text{delay + (Propogation delay} \times (n+1)) + \text{(MAC delay} \times (n+1))\end{aligned} \tag{5.2}$$

The compression and decompression delays mentioned in Eq. (5.2) are calculated experimentally at sender and receiver node respectively. Consider a case in which a 32x32 image is transmitted over a multi-hop network consisting of four intermediate nodes using DWT. The following values are obtained experimentally:

Total ETE delay = 235.5ms

Compression delay = 75.9ms

Decompression delay = 83.9ms

The processing and propagation delays can be calculated as follows:

$$\text{Processing Delay} = \text{Packet Size / Data Rate}$$
$$\text{Processing Delay} = 67\ bytes / 250Kbps = 67 \times 8 / 250\text{x}10^3 = 536 / 250 \times 10^3 = 2.14ms$$

$$\text{Propagation Delay} = \text{Distance / Velocity}$$
$$\text{Propagation Delay} = 5m / 2 \times 10^8 m/\sec = 2.5 \times 10^{-8} \sec = 2.5 \times 10^{-5} ms$$

We can find the MAC delay by substituting compression, decompression, processing and propagation delays.

$$\text{ETE Delay}_{AF} = 75.9 + ((2.14 \times 4) + 83.9 + (5 \times 2.5 \times 10^{-5}) + (\text{MAC delay} \times 5)$$
$$235.5 = 168.37 + (5 \times \text{MAC delay})$$
$$\text{MAC Delay} = (235.5 - 168.37) / 5 = 14.68ms$$

The average MAC delay is given below:

$$\text{Avg. MAC Delay=MAC Delay / Number of Hops}$$
$$\text{Avg. MAC Delay} = 14.68 / 5 = 2.93ms$$

Similarly, we can compute the MAC delay for DCT. The following values are obtained experimentally.

Total ETE delay = 874.6ms

Compression delay = 514.8ms

Decompression delay = 275.8ms

The propagation and processing delays for both techniques are the same. So, we can substitutes all the delays in Eq. (5.2) to calculate the MAC delay.

$$\begin{aligned}
&\text{ETE Delay}_{AF} = 514.8 + ((2.14 \times 4) + 275.8 + (5 \times 2.5 \times 10^{-5}) + (\text{MAC delay} \times 5) \\
&874.6 = 799.17 + (5 \times \text{MAC delay}) \\
&\text{MAC Delay} = (874.6 - 799.17) / 5 = 15.08 ms
\end{aligned}$$

The average MAC delay is given by:

$$\begin{aligned}
&\text{Avg. MAC Delay=MAC Delay / Number of Hops} \\
&\text{Avg. MAC Delay} = 15.08 / 5 = 3.01 ms
\end{aligned}$$

The average MAC delays of DCT and DWT are 3.01ms and 2.93ms respectively. Average MAC delay of DCT is slightly higher than DWT because of the experimental error.

The experimental ETE delay and delay per hop for the topology shown in Figure 6-4 are presented in Figure 6-5 and Figure 6-6 respectively.

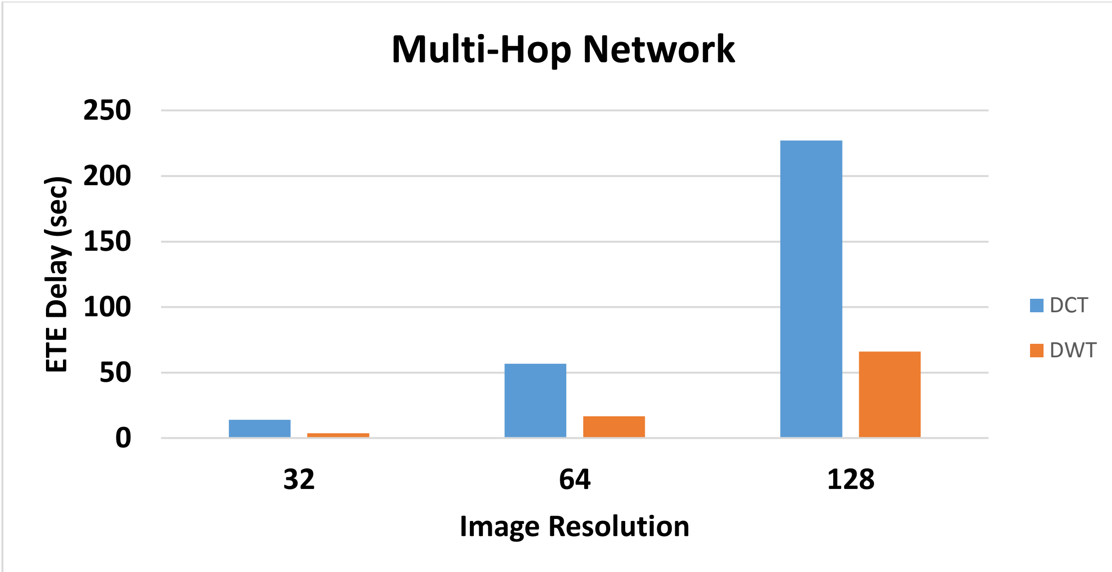



Figure 6-5 ETE Delay vs Image Resolution

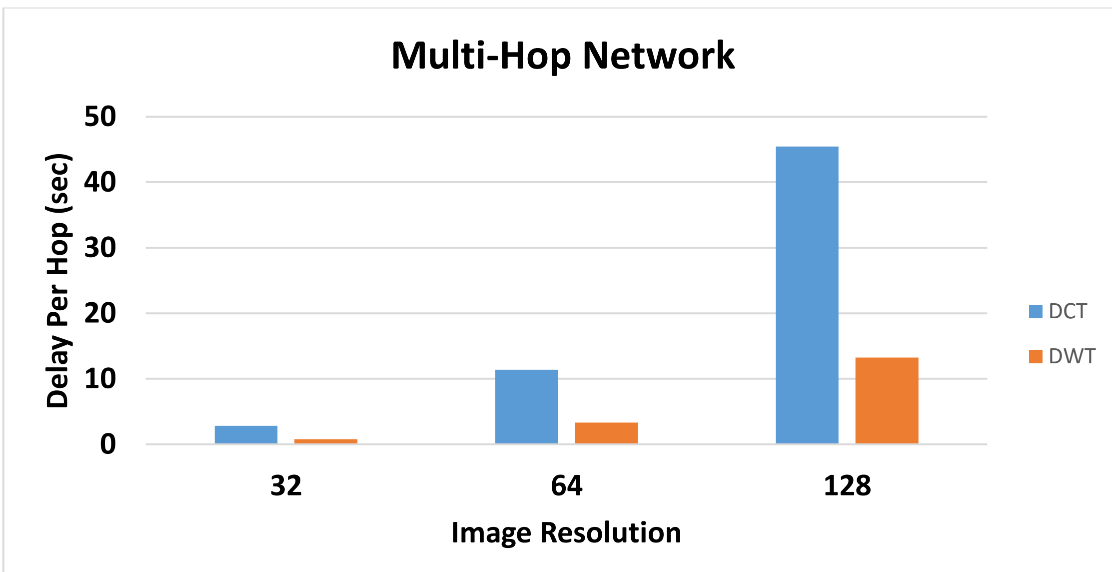
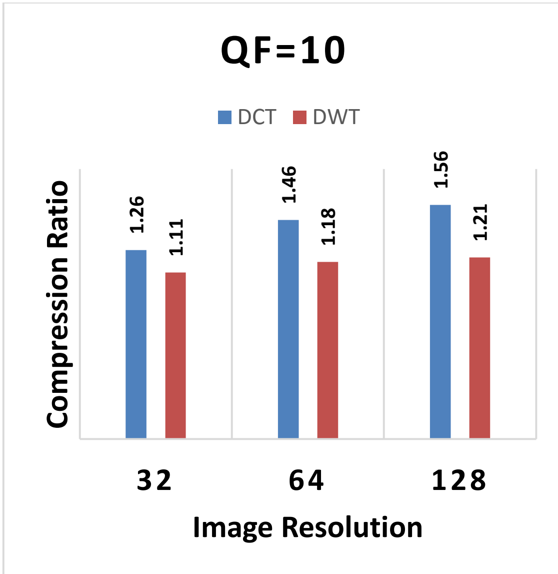



Figure 6-6 Delay per Hop vs Image Resolution

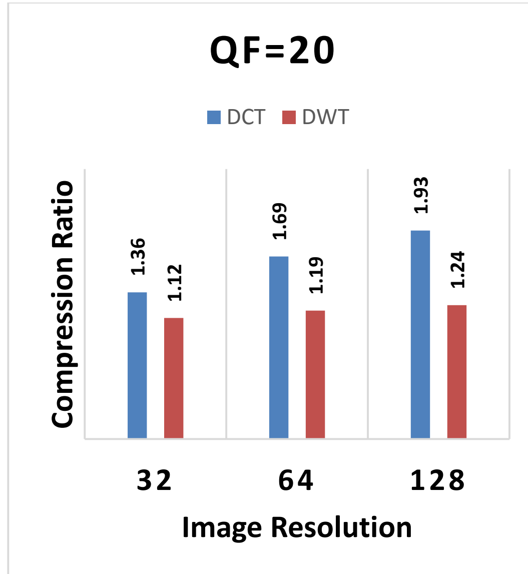
ETE delay and delay per hop for DWT of 32x32 image are 3.80 sec and 0.76 sec respectively. Similarly, we can determine ETE delay for other sizes of image. Naturally, the delay will be higher for higher image sizes because there will be more number of packets to be transmit. Furthermore, if we compare ETE delay and delay per hop for DCT and DWT, we can analyze that DWT is better than DCT. This is due to the fact that DWT is a fast yet less complex compression technique which results in better performance.

#### 6.3.1.3 Compression Ratio

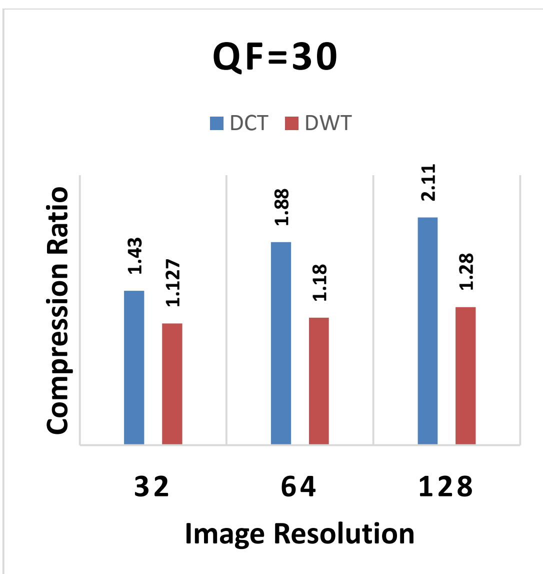
Compression Ratio (CR) is used to assess the performance of both the techniques. CR is calculated using Eq. (5.2). We are using the same parameters as presented in Table 5-1. The compression ratio for DCT and DWT is illustrated in Figure 6-7.



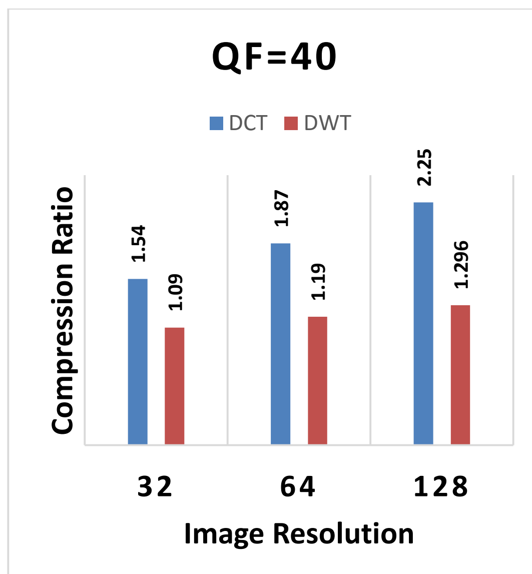

(a)




(b)




(c)




(d)

Figure 6-7 Compression Ratio vs Image Resolution

Figure 6-7 shows that DCT compression ratio for all image resolutions is higher than that of DWT. For instance, if we analyze Figure 6-7 (a) where QF=10 and image resolution is 128x128, DCT has CR=1.56 while DWT has CR=1.21. As shown in Figure 6-7 (b), (c) and (d), the CR is gradually increasing with increase in QF and at QF=40 CR eventually reaches to 2.25 and 1.29 for DCT and DWT respectively. The factor that is affecting the compression ratio is quantization factor. We conclude that the information loss in DWT is less as compared to DCT.

#### 6.3.1.4 Battery Lifetime

The power source for sensor motes is battery. Normally, mote requires 3V or two AA batteries. There are different brands in the market for different batteries but considering the low environmental impact on the network, rechargeable batteries are mostly used for economy reasons. The rechargeable batteries have different technologies. Previously batteries were based on NiCd, but life cycle of these batteries were reduced because of the memory effect.

Nowadays, a new technology is normally used based on NiMH. This technology is better in terms of memory effect and life of batteries [49]. The battery characteristics are given in Table 6-6.

Table 6-6 Battery Characteristics [49]

| Characteristics | Value |
|---|---|
| **Maximum charge voltage** | 1.5V |
| **Nominal voltage** | 1.2V |
| **Nominal Capacity** | 2850mAh |
| **Standard charge** | 270mA/16h |
| **Fast Charge** | 2700mA/1.1h |

The battery lifetime calculated in [49] is 245.19h. We have used this battery lifetime value in our results to determine that how much number of images can be transmitted in this battery lifetime.

We have considered the following parameters to do our calculations for both compression techniques.

Table 6-7 Parameters used for Computation

| Parameters | Value |
|---|---|
| **Image Resolution** | 32x32 |
| **Image Compression Technique** | DCT, DWT |
| **Execution Time for DCT (32x32)** | 14.20sec ≈ 15.0sec |
| **Execution Time for DWT (32x32)** | 4.17sec ≈ 5.0sec |
| **Battery Lifetime** | 245.19h ≈ 10d, 5h, 12m |

The execution time is the time taken by 32x32 image for transmission. We have round it off to the next integer to simplify the calculations. First, we do our calculation on DCT as shown below.

$$\begin{aligned}
&\text{Number of Images in 1min} = 4\\
&\text{Number of Images in 60min} = 1\text{hr} = 240\\
&\text{Number of Images in 24hr} = 1\text{day} = 5760\\
&\text{Number of Images in 10days} = 57600\\
&\text{Total Images in } 245.20\text{h} = 10\text{d},5\text{h},12\text{m} = 57600 + 1200 + 48 = 58848
\end{aligned}$$

Now, we do our calculations for DWT, so that we can compare performance of both the techniques.

Number of Images in 1min =12

Number of Images in 60min =1hr = 720

Number of Images in 24hr =1day = 17280

Number of Images in 10days = 172800

Total Images in 249.20h =10d,5h,12m = 172800 + 3600 + 144 = 176544

Number of images for DCT and DWT are 58848 and 176544 respectively. It can be seen from the above calculations that DWT transmit large number of images for given battery lifetime as compared to DCT. For other image resolutions, the number of images is given in Table 6-8.

Table 6-8 Number of Images for DCT and DWT

| Image Resolution | DCT | DWT |
|---|---|---|
| **32x32** | 58848 | 176544 |
| **64x64** | 15486 | 51924 |
| **128x128** | 3888 | 13374 |

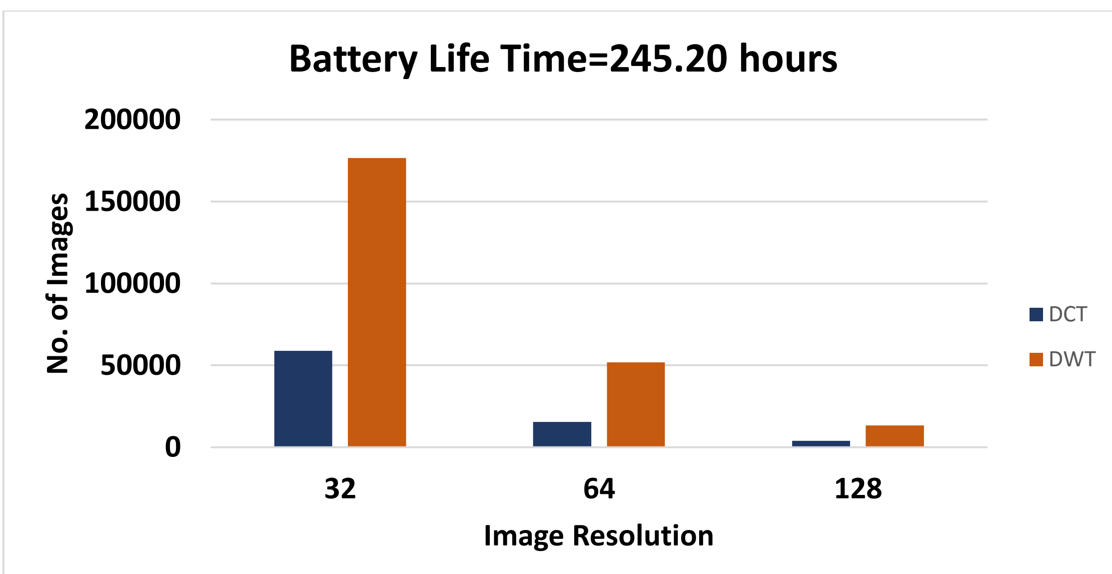



Figure 6-8 Number of Images vs Image Resolution

The above values are obtained analytically because practically it is not plausible to transmit images continuously with no delay in between the transmission. The performance of DWT is better than DCT as it is capable of transmitting more data and it has less execution time as shown in Figure 6-8. It is approximately three times faster than the DCT. Also, to save battery lifetime DWT is suitable technique as it consumes less battery for a given data.

# CHAPTER 7

# CONCLUSIONS AND FUTURE WORK

## 7.1 Conclusion

In this study, a comprehensive literature review for data compression techniques has been conducted. Two schemes have been implemented for image compression. DCT and DWT image compression techniques have been implemented using TinyOS on a hardware platform TelosB. We analyzed DCT and DWT on the basis of different performance metrics, mainly: PSNR, compression ratio, throughput, ETE delay and battery lifetime. The experimental results show that the overall performance of DWT is better than that of DCT. DWT has a higher PSNR, which gives a better image quality irrespective of the image resolution and quantization factor. In addition, DWT is a faster compression technique than DCT, because of the less execution time and delay. Moreover, DWT consumes less energy than that of DCT, because for a specific battery lifetime DWT transmits almost three times more images than DCT. Also, DCT has a higher compression ratio than that of DWT, which results in a poor image quality.

Changing the topology affects the performance of both techniques. For instance, as we increase the number of intermediate nodes, the performance of both the techniques degrades.

Although, DCT is considered a good compression technique due to its higher compression ratio, however, DWT has been widely accepted as a better compression technique in wireless sensor network.

## 7.2 Future Work

For a future work, the following points can be considered as extension to the work done in this thesis.

- In this thesis, the implementation is done on TelosB platform but it can also be done on other platforms, which will allow us to study the effect of hardware platforms on the performance of compression techniques.
- The experiments have been performed on a small-scale multi-hop network due to limited resources such as TelosB, geographical area and laptops. However, the experiments can be performed in a more realistic environment to analyze the performance on a bigger scale.

# REFERENCES


[1] S. Yang, *Wireless Sensor Networks Principles, Design and Applications*. 2014.

[2] F. Marcelloni and M. Vecchio, "An Efficient Lossless Compression Algorithm for Tiny Nodes of Monitoring Wireless Sensor Networks," *The Computer Journal*, vol. 52, no. 8, pp. 969–987, Apr. 2009.

[3] Y. Wang, "Data Compression Techniques in Wireless Sensor Networks," *Pervasive Computing*, pp. 1–12, 2012.

[4] D. M. Pham and S. M. Aziz, "An Energy Efficient Image Compression Scheme for Wireless Sensor Networks," in *IEEE 8th International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, 2013, pp. 260–264.

[5] M. Mowafi, F. Awad, E. Taqieddin, and O. Banimelhem, "Experimental Evaluation of Image Compression and Transmission for TinyOS-based Imote2 Platform," in *IEEE International Conference on Innovations in Information Technology*, 2011, pp. 173–178.

[6] O. Ghorbel, W. Ayedi, M. Wasim, and M. Abid, "DCT & DWT Images Compression Algorithms In Wireless Sensors Networks: Comparative Study and Performance Analysis," *International Journal of Wireless & Mobile Networks*, vol. 4, no. 6, pp. 45–59, 2012.

[7] G. K. Wallace, "The JPEG Still Picture Compression Standard," *IEEE Transactions on Consumer Electronics*, vol. 38, no. 1, pp. 18–34, 1992.

[8] A. Mammeri, B. Hadjou, and A. Khoumsi, "A Survey of Image Compression Algorithms for Visual Sensor Networks," *International Scholarly Research Network*, vol. 2012, pp. 1–19, 2012.

[9] G. a. Ruiz, J. a. Michell, and A. Burón, "High Throughput Parallel-Pipeline 2-D DCT/IDCT Processor Chip," *The Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology*, vol. 45, no. 3, pp. 161–175, Dec. 2006.

[10] G. A. Ruiz, J. A. Michell, and A. M. Burón, "Parallel-Pipeline 8x8 Forward 2-D ICT Processor Chip for Image Coding," *IEEE Transactions on Signal Processing*, vol. 53, no. 2, pp. 714–723, 2005.

[11] C. F. Chiasserini and E. Magli, “Energy Consumption and Image Quality in Wireless Video-Surveillance Networks,” *The 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, vol. 5, pp. 2357–2361, 2002.

[12] G. Pekhteryev, Z. Sahinoglu, P. Orlik, and G. Bhatti, “Image Transmission over IEEE 802.15.4 and ZigBee Networks,” *IEEE International Symposium on Circuits and Systems*, vol. 4, pp. 3539–3542, 2005.

[13] L. Ferrigno, S. Marano, V. Paciello, and A. Pietrosanto, “Balancing Computational and Transmission Power Consumption in Wireless Image Sensor Networks,” in *IEEE International Conference on Virtual Environments, Human-Computer Interfaces and Measurement Systems*, 2005, no. July, pp. 18–20.

[14] G. Mallat, “A Theory for Multiresolution Signal Decomposition : The Wavelet Representation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674–693, 1989.

[15] Y. Mu, B. Murali, and A. L. Ali, “Embedded Image Coding Using Zerotrees of Wavelet Coefficients for Visible Human Dataset,” in *Conference Record of the Thirty-Ninth Asilomar Conference on Signals, Systems and Computers*, 2005, pp. 276–280.

[16] V. S. Shingate, T. R. Sontakke, and S. N. Talbar, “Still Image Compression using Embedded Zerotree Wavelet Encoding,” *International Journal of Computer Applications*, vol. 1, no. 7, pp. 65–75, Feb. 2010.

[17] M. Hamdi, N. Boudriga, and M. S. Obaidat, “Bandwidth-Effective Design of a Satellite-Based Hybrid Wireless Sensor Network for Mobile Target Detection and Tracking,” *IEEE Systems Journal*, vol. 2, no. 1, pp. 74–82, Mar. 2008.

[18] P. Chithra and P. Thangavel, “A Fast and Efficient Memory Image Codec (encoding/decoding) Based on All level Curvelet Transform Co-efficients with SPIHT and Run Length Encoding,” in *Recent Advances in Space Technology Services and Climate Change*, 2010, pp. 174–178.

[19] S. Iren and P. D. Amer, “Application Level Framing Applied to Image Compression,” *Annales des télécommunications*, vol. 57, no. 5–6, pp. 502–519, 2002.

[20] Y. Sun, H. Zhang, and G. Hu, “Real-Time Implementation of A New Low-Memory SPIHT Image Coding Algorithm using DSP Chip.,” *IEEE transactions on Image Processing*, vol. 11, no. 9, pp. 1112–1116, Jan. 2002.

[21] L. Chew, W. Chia, L. Ang, and K. Seng, “Very Low-Memory Wavelet Compression Architecture Using Strip-Based Processing for Implementation in Wireless Sensor

Networks," *EURASIP Journal on Embedded Systems*, vol. 2009, no. 1, pp. 1–16, 2009.

[22] D. Taubman, "High Performance Scalable Image Compression with EBCOT.," *IEEE transactions on Image Processing*, vol. 9, no. 7, pp. 1158–1170, Jan. 2000.

[23] C. Lian, K. Chen, H. Chen, and L. Chen, "Analysis and Architecture Design of Block-Coding Engine for EBCOT in JPEG 2000," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 3, pp. 219–230, 2003.

[24] Q. Lu, X. Ye, and L. Du, "An Architecture for Energy Efficient Image Transmission in WSNs," in *IEEE International Conference on Networks Security, Wireless Communications and Trusted Computing*, 2009, vol. 1, pp. 296–299.

[25] W. Yu, Z. Sahinoglu, A. Vetro, and M. Electric, "Energy Efficient JPEG 2000 Image Transmission over Wireless Sensor Networks," in *IEEE Global Telecommunications Conference*, 2004, vol. 5, pp. 2738–2743.

[26] W. a. Pearlman, a. Islam, N. Nagaraj, and a. Said, "Efficient, Low-Complexity Image Coding With a Set-Partitioning Embedded Block Coder," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 11, pp. 1219–1235, Nov. 2004.

[27] M. V. Latte, N. H. Ayachit, and D. K. Deshpande, "Reduced Memory Listless Speck Image Compression," *Digital Signal Processing*, vol. 16, no. 6, pp. 817–824, Nov. 2006.

[28] H. Matsumoto, S. Member, K. Sasazaki, and Y. Suzuki, "Color Image Compression with Vector Quantization," in *IEEE Conference on Soft Computing in Industrial Applications*, 2008, pp. 84–88.

[29] K. Masselos, P. Merakos, and C. E. Goutis, "Power Efficient Vector Quantization Design Using Pixel Truncation," *Integrated Circuit Design. Power and Timing Modeling, Optimization and Simulation*, pp. 409–418, 2002.

[30] A. Moinuddin, E. Khan, and M. Ghanbari, "Low Complexity, Efficient and Embedded Color Image Coding Technique," *IEEE Transactions on Consumer Electronics*, vol. 54, no. 2, pp. 787–794, May 2008.

[31] S. Kasaei, M. Deriche, and B. Boashash, "A Novel Fingerprint Image Compression Technique using Wavelets Packets and Pyramid Lattice Vector Quantization," *IEEE transactions on image processing*, vol. 11, no. 12, pp. 1365–1378, Jan. 2002.

[32] A. Selim, M. M. Hadhoud, M. I. Dessouky, and F. E. Abd El-Samie, "A Simplified Fractal Image Compression Algorithm," in *IEEE International Conference on Computer Engineering & Systems*, 2008, pp. 53–58.

[33] U. Nandi and J. K. Mandal, “Fractal Image Compression by using Loss-less Encoding on the Parameters of Affine Transforms,” in *1st International Conference on Automation, Control, Energy and Systems*, 2014, pp. 1–6.

[34] U. Nandi and J. K. Mandal, “Fractal Image Compression using Fast Context Independent HV Partitioning Scheme and MRBH, MRBHM Coding,” in *1st International Conference on Automation, Control, Energy and Systems*, 2014, pp. 1–5.

[35] M. Shwetha, P. Ashwini, and B. M. Sujatha, “Analysis of Image Compression Algorithms in WSN: A Review,” *International Journal of Science, Engineering and Technology Research*, vol. 3, no. 4, pp. 1029–1032, 2014.

[36] H. ZainEldin, M. a. Elhosseini, and H. a. Ali, “Image Compression Algorithms in Wireless Multimedia Sensor Networks: A Survey,” *Ain Shams Engineering Journal*, vol. 6, no. 2, pp. 481–490, 2015.

[37] A. Katharotiya, S. Patel, and M. Goyani, “Comparative Analysis between DCT & DWT Techniques of Image Compression,” *Journal of Information Engineering and Applications*, vol. 1, no. 2, pp. 9–18, 2011.

[38] J. Karlsson, “Image Compression for Wireless Sensor Networks .,” 2007.

[39] K. Cabeen and P. Gent, “Image Compression and the Discrete Cosine Transform,” in *College of the Redwoods*, 1999, pp. 1–11.

[40] M. I. Khalil, “Image Compression Using New Entropy Coder,” *International Journal of Computer Theory and Engineering*, vol. 2, no. 1, pp. 39–41, 2010.

[41] D. S. Taubman, M. W. Marcellin, and M. Rabbani, *JPEG2000: Image Compression Fundamentals, Standards and Practice*, vol. 11, no. 2. Boston, MA: Springer US, 2002.

[42] V. Lecuire, C. Duran-Faundez, and N. Krommenacker, “Energy-Efficient Transmission of Wavelet-based Images in Wireless Sensor Networks,” *Journal on Image and Video Processing*, vol. 1, pp. 1–19, 2007.

[43] B. . Deokate and P. . Patil, “An Algorithm for Image Compression Using 2D Wavelet Transform,” *International Journal of Modern Engineering Research*, vol. 2, no. 4, pp. 2311–2315, 2012.

[44] C. Mulcahy, “Image Compression using the Haar Wavelet Transform,” *Journal of Spelman Science and Mathematics*, vol. 1, pp. 22–31, 1988.

[45] S. . Tamboli and V. . Udupi, "Image Compression Using HAAR Wavelet Transform," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 2, no. 8, pp. 3166–3170, 2013.

[46] P. Morton, "Image Compression Using the Haar Wavelet Transform," *Math 45*, pp. 1–9, 1997.

[47] M. Rathee and A. Vij, "Image Compression Using Discrete HAAR Wavelet Transform," *International Journal of Engineering and Innovative Technology*, vol. 3, no. 12, pp. 47–51, 2014.

[48] G. Coq, C. Olivier, O. Alata, and M. Arnaudon, "Information Criteria and Arithmetic Codings: An Illustration on Raw Images," in *European Signal Processing Conference*, 2007, pp. 634–638.

[49] A. de Escol Pablo, "Master Thesis in Development of a wireless sensor network with 6LoWPAN support," 2009.

# APPENDIX

# OPERATING SYSTEM AND HARDWARE

In order to implement compression techniques in WSN, one of the main elements is a suitable working environment that will fulfil hardware and software requirements. In this research, the implementation of DCT and DWT in WSNs is performed using TinyOS and TelosB as software and hardware respectively.

## A.1 Operating System

The WSN applications use a particular operating system (OS) such as TinyOS. This OS can be installed in different environments i.e., Windows XP, Windows Vista and Ubuntu. But mostly Ubuntu is used because it is most suitable.

### A.1.1 TinyOS

TinyOS is a free open source operating system designed by University of California Berkeley. It is designed for low-power wireless devices such as sensors. Four major properties make this operating system most suitable to sensors such as event driven, non-preemptive, non-real time, and energy consumption. In this operating system, NesC language is used as set of cooperating tasks and processes. It is a component-based event driven programming language. Moreover, it is an extension to the C language to optimize the memory limits. NesC has interfaces and modules that can be used to develop TinyOS applications. TinyOS and NesC have a documentation feature called nesdoc which generates documentation automatically from the source code. Two versions of TinyOS have been developed such as TinyOS 1.0 and 2.0. In this research we have used the latest release of version 2.0 i.e., TinyOS 2.1.2. This TinyOS version is used to work with the network devices. The architecture of TinyOS is presented in Figure A-1.

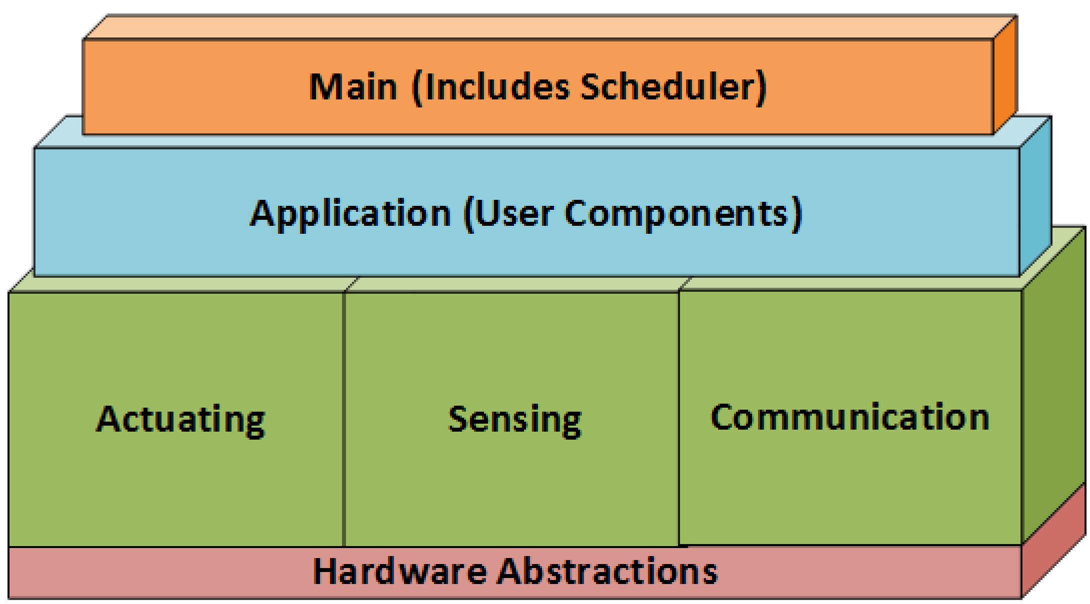



Figure A-1 TinyOS Architecture

## A.2 Hardware

In WSN, the devices used for the communication are called as nodes or motes. The typical architecture of a mote consist of power unit, communicating and processing units and a series of sensors as shown in Figure A-2.

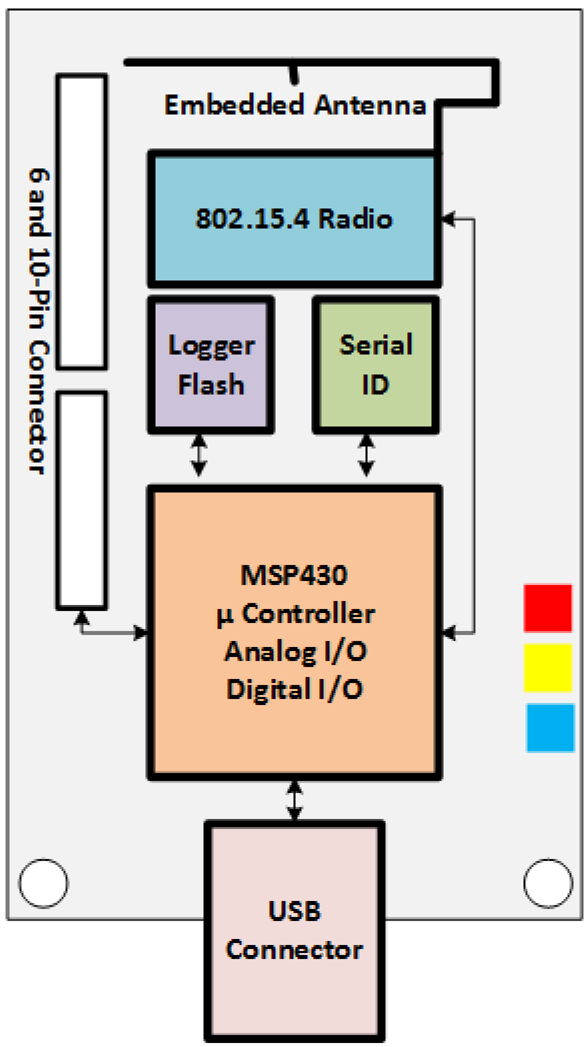



Figure A-2 Sensor Architecture

These units are included into the motes to define a series of platform. TinyOS support different platforms. The most common hardware platform are Imote2, MicaZ, TelosB and EPIC. In this research, we have used TelosB motes for our experimentation.

### A.2.1 TelosB

TelosB mote was developed by UC Berkeley and license was given to different manufacturers such as Crossbow, Arch rock and Moteiv. Crossbow’s TelosB mote (TPR2400) is an open source platform designed to enable cutting-edge experimentation for the research community.

Figure A- 3 Crossbow's TelosB

The TPR2400 has all the fundamentals for lab studies into a single platform such as IEEE 802.15.4 radio having integrated antenna, USB programming capability, a low power MCU with extended memory. This platform has several features given in Table A-1:

Table A-1 TelosB Features

| | |
|---|---|
| **Size** | 65x31x6 mm |
| **Air Temperature** | -40 to $85^0$ C |
| **Transmission Range** | 50m indoor, 125m outdoor |
| **I/O** | 16-pin connector |
| **Antenna** | Integrated |
| **Power** | Two AA batteries/USB Port |
| **Radio** | IEEE 802.15.4 |
| **Data Rate** | 250kbps |
| **External Flash** | 1MB |
| **T1 MSP430 Microcontroller** | 8MHz |
| **Operating System** | TinyOS 1.1.10 or higher |
| **Optional Sensors** | Light, Temperature and Humidity |

Two AA batteries is used to provide power to the TPR2400. If TelosB mote is connected through USB port, power is delivered through the computer. It allows users to communicate or interface with different devices.

# VITAE

## MUHAMMAD MUSADDIQ


Date of Birth: 02 October, 1989
Passport #: CA0877831
KFUPM Student Housing 806-310 Dhahran, Saudi Arabia
E-mail:muhammad_musaddiq38@yahoo.com
Cell No. : +966-535484562
Skype ID: muhammad.musaddiq3


## OBJECTIVE

As a prolific learner and zealous workaholic, I would like to dedicate the best of my efforts, skills and time for such a firm where I can find maximum opportunities to learn and employ the best of my capabilities.

## EDUCATIONAL HISTORY

- **MS Computer Networks** **Sep 2013-Aug 2015**
  *King Fahd University of Petroleum & Minerals, Dhahran Saudi Arabia*

- **BS Computer Engineering** **Sep 2008-Sep 2012**
  *COMSATS Institute of Information Technology, Lahore, Pakistan*
  GOT Campus **Silver Medal**

## PUBLICATIONS

- **Journals**
  - Tarek Sheltami, ***Muhammad Musaddiq***, Elhadi Shakshuki, "Data Compression Techniques in Wireless Sensor Networks" (Submitted in ISI indexed journal)
  - Basmem Almadani, ***Muhammad Musaddiq***, Sadiq M. Sait, "Real-Time Publish/Subscribe Model in NS-2" (Submitted in ISI indexed journal).
  - Basem Almadani, Bilal Saeed, ***Muhammad Musaddiq***, Shehryar Khan, "Healthcare Systems Integration Using Real Time Publish Subscribe (RTPS) Middleware" (Submitted in ISI Indexed journal).

- **Conferences**
  - ***Muhammad Musaddiq***, Shehryar Khan, Uthman Baroudi, and Bilal Saeed, "Performance Evaluation of IEC 61850 under wireless Communication Networks", ACM 6th International Conference on Management of Emergent Digital EcoSystems (MEDES), pp: 90-94, September 2014.

- Basem Almadani, Shehryar Khan, Tarek R. Sheltami, Elhadi M. Shakshuki, ***Muhammad Musaddiq***, and Bilal Saeed, “Automatic Vehicle Location and Monitoring System”, *Procedia Computer Science*, 5th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN), pp:127-134, 2014.

- Rao Naveed bin Rais**,** ***Muhammad Musaddiq***, Mayyda Mukhtar, Anees Shafiq, Hafiz Muhammad Imran, and Muhammad Najam-Ul-Islam, "Analysis of TCP under Wireless Circumstances – A Performance Evaluation ", IEEE International Conference on Frontiers of Information Technology (FIT), pp: 371-376, December 2012.

## RESEARCH EXPERIENCE

- **Data Compression Techniques in Wireless Sensor Networks**
  - It was **MS Thesis** in which I have implemented two image compression techniques i.e., DCT and DWT. These compression techniques were implemented on real platform Telosb using TinyOS. The performance of both the techniques have been analyzed on several parameters.
    ***Supervisor: Dr. Tarek Sheltami***

- **Automatic Vehicle Location and Monitoring System**
  - Proposed a Real-Time Automatic Vehicle Location (AVL) and Monitoring system for traffic control of pilgrims coming towards the holy city of Makkah in Saudi Arabia based on data distribution service (DDS) specified by the Object Management Group (OMG), which is a Real-Time Publish/Subscribe Middleware.
    ***Supervisor: Dr. Basem Almadani***

- **Performance Evaluation of IEC 61850 under wireless Communication Networks**
  - The standard IEC 61850 is considered as a candidate for communication standard for smart grid applications. The delay performance is one of the critical issues that were specified by IEC 61850. This project involved the modeling and simulation of IEC 61850 substation automation system (SAS) under wireless and hybrid networks. OPNET is used to model and simulate the Intelligent Electronic Device (IEDs).
    ***Supervisor: Dr. Uthman Baroudi***

- **Real-Time Publish/Subscribe Model in NS-2**
  - The Real-Time Publish/subscribe (RTPS) paradigm is a powerful data dissemination model. The design implementation of the RTPS model in NS-2 Simulator is described. This implementation has been done on a Peer-to-Peer (P2P) communication model. The performance of RTPS is analyzed using different parameters
    ***Supervisor: Dr. Basem Almadani***

## PROFESSIONAL EXPERIENCE

- **Resident Engineer** **Nov 2012-Aug 2013**

  *Interactive Group of Companies (IACGRP), Islamabad, Pakistan*

  Responsibilities include:
  - Configuring, troubleshooting & management of Cisco/Catalyst switches, PoE switches.
  - Configuration of switch ports for implementing project wise VLANs, Trunk configuration (ISL & IEEE 802.1Q)
  - Installation of network devices such as switches, routers etc.
  - Provided Network Support and Application Support.
  - Conducted and delivered training lecturers regarding application.
  - Reporting and documentation on daily/weekly basis.
  - Managed Open Ticket Resource System.

## CERTIFICATIONS/WORKSHOP

- **Cisco Certified Network Associate** (CCNA) Routing and Switching

  **Dated: June 6, 2015**
- **Cisco Certified Entry Networking Technician** (CCENT)

  **Dated: April 22, 2015**
- **MS Office Certified** (MS – Word, MS – Excel, MS – Power Point)
- **Intelligent Control Systems** offered by Power & Control Computing Research Group

## TECHNICAL SKILLS

- **Networking:** TCP/IP, OSI, OSPF, EIGRP, RIP, VLANs, VPN, LAN, WAN, DNS, IPv6, DHCP, cabling
- **Networking Tools:** GNS3, Packet Tracer, Putty
- **Network Simulator:** Network Simulator 2 (NS2), OPNET Modeler, TinyOS
- **Academic Software's:** Model Sim, MP Lab, Emulator, Microwind, MATLAB
- **Applications:** MS Office, Visio, Adobe Reader
- **Languages:** C, TCL, MatLab, C#, Python
- **Operating Systems:** Windows (8.1, 7, XP), Linux (Ubuntu)

## ACHIEVMENTS & AWARDS

- **Fully Funded Merit Scholarship** from King Fahd University of Petroleum & Minerals.
- **Performance Appreciation Award** from Interactive Group of Companies.
- Stood **2nd** among 65 students in undergraduate program of BS Computer Engineering and received **Campus Silver Medal.**
- **Performance Scholarships** from COMSATS throughout the engineering.
- Awarded **Merit Scholarships** at school level.

## REFERENCES

**Dr. Tarek Sheltami**
Associate Professor
KFUPM University
Dharan, Saudi Arabia
Tel: +966-3-860-4678
Email: *tarek@kfupm.edu.sa*

**Dr. Uthman Baroudi**
Assistant Professor
KFUPM University
Dharan, Saudi Arabia
Tel: +966-3-860-4283
Email: *ubaroudi@kfupm.edu.sa*

**Dr. Rao Naveed Bin Rais**
Associate Professor
MAJU University
Islamabad, Pakistan
Tel: +92-51-111-878-787-131
Email: *naveedbinrais@jinnah.edu.pk*