Sudan University of Science & Technology

College of Graduate Studies

جامعة السودان للعلوم والتكنولوجيا

كلية الدراسات العليا

كلية الدراسات العليا

Ref: SUST/ CGS/A11

Approval Page

(To be completed after the college council approval)

Name of Candidate: Tawfeeg Suliman Nour Jebreel

Thesis title: Modding of Power Consumption of Video Transmission in Internet of Things devices with Acceptable Quality.

نمذجة استهلاك الطاقة في ارسال المرئيات عبر اجهزة انترنت الأشياء بجودة مقبولة

Degree Examined for: PhD Computer Science

Approved by:

1. External Examiner

Name: Abubakr elSiddig Mirghani elHussein

Signature: .... Date: Sept. 13th, '18

2. Internal Examiner

Name: Izzeldin Mohamed Osman

Signature: Izzeldin Osman Date: 13/9/2018

3. Supervisor

Name: C/ Dr Salah Elfaki (Dean)

Signature: S.Faki Date: 29/10/2018

Sudan University of Science & Technology

College of Graduate Studies

Modeling of Power Consumption of Video Transmission in Internet of Things Devices with Acceptable Quality of Experience

نمذجة استهلاك الطاقة في ارسال المرئيات عبر أجهزة إنترنت الأشياء بجودة مقبولة

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy in Computer Science

By:

Tawfeeg Suliman Noor Jebreel

Supervisor:

Dr. Is-Haka Mkwawa

Co. Supervisor:

Dr.Niemah Izzeldin osman

September 2018

DEDICATION

I dedicate this thesis to my beloved family without their support and prayers, I would never have completed my study.

.

DECLARATION

I declare that the work described in this thesis is original work undertaken by me for the degree of Doctor of Philosophy, at **Sudan University of Science and Technology, Khartoum, Sudan**

No part of the material described in this thesis has been submitted for any award of any other degree or qualification in this or any other university or college of advanced education.

I also declare that part of this thesis has been published in some of my following publications:

- T. S. Noor and N. I. Osman, “The Influnce of Gender on QoE Subjective Assessment for Video Services,” vol. 7, no. 4, pp. 198–203, 2018.
- T. S. Noor, N. I. Osman, and I.-H. M. Mkwawa, “Analysis and Modelling of Power Consumption in IOT with Video Quality Communication,” *Int. J. Multimed. Its Appl.*, vol. 10, no. 1/2/3, pp. 15–27, 2018.

ACKNOWLEDGMENTS

First of all, all thanks and praise would first go to **God (Allah)** for all the success.

My sincere thanks would go to my supervisor **Dr. Is-Haka M. Mkwawa**for all his support, time and guidance. This thesis would not have been completed without the in-depth discussions and comments from **Dr. Is-Haka**. I also would like to thank my Co-supervisor **Dr. Niemah I. Osman** for their insightful comments and advice.

I also would like to thank the other SUST staffs who have given me some of their precious time to comment on my work. I also would like to thank all my colleagues at the SUST for the valuable advice and discussions

Tawfeeg Suliman Noor Jebreel

ABSTRACT

Internet of Things applications such as environmental monitoring and healthcare may involve multimedia communications from IoT devices to humans for decision-making. Therefore, the delivered multimedia should be in good perceived quality. Higher video quality results into higher energy consumption due to encoding and decoding processes and as a result, will affect the performance of IoT devices due to their inherent energy constraints. This thesis presents the impact of video encoding parameters on the energy consumption of IoT devices. The experimental results from Cooja simulator show that the videos with high bitrates and low frame rates consume more power than videos with low bitrates and high frame rates. It was also found that videos with high movement consume more power than videos with medium and slow movement. Also this thesis proposes a power model that takes into account video parameters such as bit rate, frame rate and content types. The proposed model can play a vital role in video quality adaptation in multimedia communication over IoT devices. The experiment results from the subjective test show that the video with the same bitrate ,frame rate and resolution, slow and medium movement recorded better MOS values and hence better quality than fast movement video sequence. Finally, this thesis presents the impact of gender on the Quality of Experience for video services. Initial results from subjective tests suggested that male viewers request higher video quality compared to female viewers, while female viewers possibly concentrate more on the content of the video.

ABSTRACT (ARABIC)

المستخلص

تتضمن تطبيقات إنترنت الأشياء مثل المراقبة البيئية والرعاية الصحية اتصالات متعددة الوسائط من أجهزة إنترنت الأشياء إلى البشر من أجل اتخاذ القرار. لذلك ، يجب أن تكون الوسائط المتعددة التي يتم تسليمها جيدة في الجودة. تؤدي الجودة العالية للمرئيات إلى زيادة استهلاك الطاقة بسبب عمليات الترميز وفك الترميز ، ونتيجة لذلك تؤثر على أداء أجهزة إنترنت الأشياء نظرًا للقيود المتعلقة بالطاقة. تعرض هذه الرسالة تأثير عوامل ترميز المرئيات على استهلاك الطاقة لأجهزة انترنت الاشياء. تظهر النتائج التجريبية من المحاكي Cooja أن مقاطع المرئيات ذات معدل الثنائيات المرتفع ومعدل الإطارات المنخفض تستهلك طاقة أكبر من مقاطع المرئيات ذات معدل الثنائيات المنخفض ومعدل الإطارات المرتفع. وقد تبين أيضًا أن مقاطع المرئيات ذات الحركة العالية تستهلك طاقة أكبر من مقاطع المرئيات ذات الحركة المتوسطة والبطيئة. تقدم هذه الأطروحة كذلك نموذجًا للطاقة يأخذ في الاعتبار معاملات المرئيات مثل معدل الثنائيات ومعدل الإطارات ونوع المحتوى. يمكن للنموذج المقترح أن يلعب دوراً حيوياً في تكييف جودة المرئيات في الاتصالات متعددة الوسائط عبر أجهزة إنترنت الأشياء. تظهر نتائج تجربة الاختبار الذاتي أن المرئيات بنفس معدل الثنائيات ومعدل الإطارات ودرجة الوضوح ،سجلت المرئيات ذات الحركة البطيئة والمتوسطة قيم MOS افضل من المرئيات ذات الحركة السريعة ، وبالتالي جودة افضل. وأخيرًا ، تعرض هذه الأطروحة تأثير نوع المشاهد على جودة التجربة لخدمات المرئيات. تقترح النتائج الأولية للاختبارات الذاتية إلى أن المشاهدين الذكور يطلبون جودة مرئيات أعلى مقارنة بالمشاهدين الإناث ، بينما من المحتمل ان تركز المشاهدات للنساء بشكل أكبر على محتوى المرئيات.

Table of Contents

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

IoT	Internet of Things
CDN	Content Delivery Network
QoS	Quality of Service
QoE	Quality of Experience
PQOS	Perceive Quality of Service
VoIP	Voice/video over IP
MOS	Mean Opinion Score
ITU	International Telecommunication Union
VQOG	Video Quality Experts Group
DSIS	Double Stimulus Impairment Scale
SSCQE	Single Stimulus Continuous Quality Evaluation
DSCQS	Double Stimulus Continuous Quality Scale
ACR	Absolute Category Rating
DCR	Degradation Category Rating
DMOS	Degradation Mean Opinion Score
Pc	Pair Comparison Method
FR	full Reference
RR	Reduced Reference
IPTV	Internet Protocol television

NR	No-Reference
MPEG	Moving Picture Experts Group
ISO	International Organization for Standardization
IEC	International Electro technical Commission
AVC	Advanced Video Coding
HD	High Definition
NAL	Network Abstraction Layer
NALUs	Network Abstraction Layer Units
DP	Data Partitioning
VCL	Video Coding Layer
MB	Macroblocks
MIoT	Multimedia Internet of Things
QoT	Quality of things
CCTV	Closed-Circuit Television
IEEE	Institute of Electrical and Electronics Engineers
LAN	Local Area Network
AV	Audio and Video
SNR	Signal-to-Noise Ratio
PSNR	Peak Signal-to-Noise Ratio
SVC	Scalable Video Coding
RD	Rate-Distortion

MGS	Medium Grain Scalability
WSN	Wireless Sensor Network
RFID	Radio-Frequency Identification
LPWAN	Low Power Wireless Sensor Network
PLC	Power line communication
BSD	Berkeley Software Distribution
IETF	Internet Engineering Task Force
ROM	Read-Only Memory
RAM	Random-Access Memory
UDP	User Datagram Protocol
CoAP	Constrained Application Protocol
CSC	Cooja Simulation Configuration
XML	Extensible Markup Language
GUI	Graphic User Interface
CORE	Common Open Research Emulator
RPL	Routing Protocol
LLNs	Low-power and Lossy Networks
CTP	Collection Tree Protocol
PhY	Physical Layer
MAC	Media Access Control
QCIF	Quarter Common Intermediate Format

RTP	Real time protocol
UDGM	Unit Disk Graph Radio Medium
ANOVA	Analysis of Variance
SPSS	Statistical Package for Social Science
CPU	Central processing unit energy consumption
TX	transmission energy consumption for these cycles
RX	listen energy consumption for this cycle
RMSE	the Root Mean Squared Error
R^2	correlation coefficient
DODAG	Direction-Oriented Directed Acyclic Graph

CHAPTER I

1. Introduction

In recent years video services and Internet of Things (IoT) have generated growing interest in the computing /networking research community [1]–[8]. Video services is a mature topic that has developed into the most voluminous type of data currently traversing the Internet, with estimated percentage usage above 50% of all Internet traffic [2], [5], [6]. It is also expected that this ratio will increase, with the increase of the population socially and professionally interacting over the Internet. Video streaming applications involve a wide range of techniques, from appropriate video compression and encoding, Content Delivery Network (CDN) architecture and management, the data transmission/delivery strategy adoption, network Quality of Services (QoS) mechanisms and client-side media player techniques for overcoming delay, jitter and data loss.

IoT refers not only to mobile phones connected through the Internet but to the wireless interconnection of all of the billions of "things" and devices through the Internet or local area networks, to increase efficient utilization. With those billions of things come billions of batteries that must be purchased, maintained, and disposed of.

In the Internet of Things, users can access video services anywhere and anytime using smart devices. These smart devices connect to the Internet via one or more telecommunications operators. Users' expectations along with additional parameters such as cognitive and behavioral states, cost, and network parameters and non-network parameters may determine the acceptability of the service from the user's point of view. If users are not satisfied with the service, they may switch to different providers or may stop using a particular application or service. Therefore network and service providers have realized that the traditional parameters to evaluate video service quality such as delay, packet loss and jitter are not enough and therefore, non-network parameters such as devices power and user context should be taken into considerations.

This chapter presents the motivations behind the project, the fundamental research questions, and the aims and objectives of the project. Furthermore, the chapter highlights the main contributions of this thesis.

1.1 Research Motivations

For IoT devices, memory and power are one of the major constraints. Therefore, in order to deliver acceptable video quality, video encoding parameters should be carefully chosen in order to save devices' energy during video transmission without jeopardizing the QoE. Much effort has been put into energy serving techniques based on the transmission protocols. However, a little effort has been put into investigating the impact of non-network parameters such as video encoding parameters on energy consumption over IoT devices and their effect on the video quality of experience.

This research mainly focuses on developing a power model to predict power usage of IoT devices considering video encoding parameter as non-network and investigate the impact of power and gender on QoE for video services for the following important reasons:

- The battery life must be long enough to complete the communication session, moreover, the environment must have facilities to charge the battery. This is difficult to achieve and we have to find a solution for this situation because it will affect the user's QoE.

- Traditionally technology-centric approaches based on the Quality of Services (QoS) parameters and video related parameters are lacking sufficient consideration of user's actual perception.

- Due to the rapid development of video services, customers are enjoying more acceptable quality videos, and their perception of video service will directly affect the service provider’s performance, which makes it significant to develop this model.
- Ensuring acceptable quality of video services for the end user.

- Many QoE metrics are able to estimate the user perceived quality or acceptability of video services but may not be accurate enough for the overall user experience prediction due to the complexity of user experience.

1.2 Research Questions

The main questions that will be addressed in this research are:

Q1: How can video encoding parameters influence the power consumption over IoT devices?

This question has triggered an investigation on the impact of video encoding parameters on the power consumption over IoT devices. Subjective tests are used to investigate the quality of the delivered videos over the IoT devices.

Q2: How can acceptable video quality be delivered over IoT device?

Video encoding parameters should be carefully chosen in order to save device energy during video transmission without jeopardizing the QoE.

Q3: How can a power model that predicts the power usage of IoT devices be developed?

The development of the power model should take into account the bit rate, frame rate and non-linear regression is used to model the power consumption model.

Q4: How can the context parameters related to the user such as gender affect the quality of experience on video services?

This question has triggered an investigation on the impact of gender which is a context parameter on the video QoE. The investigation to answer this question was carried out by the use of subjective tests over the H.264 codec.

1.3 Aims and Objectives

The main goal of this research is to propose a novel power model based on non-network parameters such as bitrate and frame rate.

This goal can be achieved by the following objectives,

1. To investigate and evaluate the impact of non-network parameters such as bitrate and frame rate on power consumption for IoT devices.
2. To investigate and evaluate the impact of context parameters such as gender on QoE of video services.
3. To conduct subjective tests in order to assess and measure the quality of transmitted video sequences over IoT devices.
4. To develop a power model based on video coding parameter which is a non-network parameter.

1.4 Research Scope

The scope of this research is as follows:

1. Using Cooja simulator to transmit video traces and record the power consumption over IoT devices.
2. It is extremely difficult to use all parameters that affect the user experience of videos services. This research focuses mainly on the important non-network parameter (power) and video content parameters (bitrate and frame rate).
3. Compare the proposed power consumption based QoE model with the subjective tests' results.
4. In this research, we use only the H.264 video codec and encode each video with multi-bitrates and different frame rates.

1.5 Research Methodology

In order to achieve the research objectives, we should adapt a methodology that enables the researcher to obtain the research goals. The methodology is described as follows:

- **Sample Population:**

For the first experiment, a total of 29 people took part in the study on a volunteering basis. No participants had any experience working in video quality assessment or coding. There were 13 females and 16 male participants, all of them had normal eye vision.

For the second experiment, a total of 34 people took part in the study on a volunteering basis. Participants have been chosen with respect to their gender (17 male and 17 female) that we used in the subjective test. Also, no participants were working in video quality assessment and no money was paid to them

- **Analysis of Data :**

After the tests were completed, all the participants' data were collected and input into an Excel sheet for averaging to compute MOS for each video sequence.

- **Tools & Methods**:

Non-Linear regression technique was used to develop the proposed model. Evalvid,x264, mp4container , FFMPEG, etmp4 and Cooja simulator.

- **Validation of Results**:

ANOVA was analyzed, R2 ,RMSE.

1.6 Research Contributions

The contributions of this thesis are outlined as follows:

- Gender as a context parameter has an impact on quality of experience for video services. Therefore video and network service providers should consider this on their video quality control and optimizations schemes. Paper [9] reflects this contribution.
- The non-network parameter (power) has an impact on the quality of video service over IoT devices. The proposed new model which takes bitrate and framerate consideration can be used by the video service and network providers for video quality control and optimization of IoT devices. This contribution has been published in [10].
- The video sequence transmission and power recording modules developed in C language for Cooja platform can be shared with other researchers for further evaluation and improvement.
- Subjective tests datasets and Cooja results can be used by other researchers in their studies for comparison and validation.

1.7 Outline of Thesis

2 Chapter 2 discusses the literature review related to this research, techniques, tools and software used in this study. It also reviews video quality assessment such as subjective and objective methods, video coding techniques and quality of experience.

3 In Chapter 3, the basic knowledge about the Internet of Things will be introduced firstly. Then details of IoT including IoT architecture, IoT challenges, IoT hardware and software , IoT simulation , RPL and LPWAN , QoE in internet of things ,QoE for MIoT will be explained.

4 Research methodology which includes approaches to methods, source of data, sample population, data collection, data analysis, validation of results and experimentation tools are discussed in Chapter 4.

5 Chapter 5 explains the experiment setup for the subjective tests for the gender parameter and the experiment setup for the videos transmission over Cooja simulation and experiment setup for the subjective tests for the power parameter.

6 Chapter 6 discusses the impact of gender on the quality of experience for the videos services and the impact of the non network parameter on the quality of experience for the videos on IoT devices.

7 Chapter 7 concludes the thesis and proposes future work. This chapter includes contributions to knowledge and limitations of the current research.

CHAPTER II

2. Literature Review

2.1 Quality of Experience (QoE) Measurement for Videos Services

The term quality of experience is used to represent a measurement of user experience with service, especially in communication and video delivery.

Quality of experience is defined as "the overall acceptability of an application or service as perceived subjectively by the end-user " according to ITU-T Rec.p.10G.100 Amendment 2[11].QoE is normally regarded as perceive quality of service (PQOS) in differentiation with the quality of service (QoS) which is generally regarded as network quality of service or related metrics for network quality such as packet loss, delay and jitter. The most important metric in QoE for videos services applications is mean opinion score which is used to represent an overall quality of speech quality provided during VoIP calls and it is also used to represent perceived video quality for a video call or video streaming application. the mean opinion score (MOS) is obtained by giving the average opinion of quality based on asking people to grade the quality for voice or video on a five-point scale: excellent 5, good 4, fair 3, poor 2 and bad 1.the mean opinion score is most widely used QoE metric for videos services application.

2.2 Video Quality Assessment

Video quality can be assessed using either subjective or objective method. Subjective quality is the users' perception of quality (ITU-T P.910) [12]. Mean Opinion Score is the most widely used metric for subjective testing. The most reliable method of measuring video quality is through subjective test approach. On the other hand, objective measurement can be performed in an intrusive or non-intrusive manner

2.2.1 Subjective Video Quality Assessment

International Telecommunication Union (ITU) and the Video Quality Experts Group (VQEG) have both defined the subjective methods as a testing method whereby a number of viewers are selected to watch video clips under test in a controlled environment. These viewers are asked to grade the quality of the video clips on a five-point Mean Opinion Score (MOS) scale which may range from ‘bad’ (1) to ‘excellent’ (5). Subjective testing can be time-consuming and expensive because a large sample of participants is needed to obtain results that are statistically meaningful. Considering the costs and time demands by this testing method, recently, uncontrolled testing environments such as crowdsourcing have emerged as a cheaper and quicker alternative to traditional laboratory-based quality evaluation for video streaming services. Subjective test methods are described in ITU-R T.500-13 (2012) [13] and ITU-T Rec. P.910 (1999)[12] . These methods guide subjective test coordinators on the type of viewing conditions, the benchmark for viewers and the selection of test materials, the procedure for assessment and methods to statistically analyze testing results. ITU-R Rec. BT.500-13 described subjective methods that are specialized for television applications, while ITU-T Rec. P.910 is proposed for multimedia applications.

The most popular and broadly used subjective methods are:

2.2.1.1 Double Stimulus Impairment Scale (DSIS):

The subjects in this method are shown pairs of the degraded video clips along with many reference clips. The reference clip is shown before any degraded pair. Subjects score according to a scale of impairment as, imperceptible, perceptible but not annoying, slightly annoying, annoying, and very annoying. This scale is also known as the 5point scale. 5 are referred to as being imperceptible and 1 is very annoying.

2.2.1.2 Single Stimulus Methods:

The subject in this method, as opposed to the previous one, is shown multiple separate scenes. This method can be run in one of two ways:

The first one is the single stimulus when the test scenes are not repeated and the second is the single stimulus with repetition of many times of the test scenes. As opposed to the previous one, three different types of methods for the scoring are used:

1. Adjectival: This is the same as described in Double Stimulus Impairment Scale with the difference that half scales are allowed.
2. Numerical: In this one, an 11-grade numerical scale is used. This is useful if a reference is not available.
3. Non-categorical: In this one, a continuous scale is used with no numbers.

Alternately a large range of numbers can be used, e.g. 0 - 100.

2.2.1.3 Stimulus Comparison Method:

This method is used when two well-matched monitors are available. The differences between pairs of scenes are scored in one of two ways:

1. Adjectival: This is a 7-grade scale labeled from +3 to - 3: The scale is translated as, .much better, better, slightly better, the same, slightly worse, worse, and much worse.
2. Non-categorical: This is very similar to the previous one where a continuous the scale is used without numbers or a relation number used either in absolute terms or related to a standard pair.

2.2.1.4 Single Stimulus Continuous Quality Evaluation (SSCQE):

According to this method, the subjects watch a program of typically 20-30 minutes with no reference to the original video clip. The subjects rate using a slider continuously perceived quality at that instant in time on a scale range from ‘bad’ To ‘excellent’ this corresponds to an equivalent numerical scale from 0 to 100.

2.2.1.5 Double Stimulus Continuous Quality Scale (DSCQS):

At DSCQS the viewers watch multiple pairs of quite short (i.e. 10 seconds) reference and test sequences. Each pair appears twice, with random order of the reference and the test sequence. The viewers do not know of the reference video clip order and are asked to rate each of the two separately on a continuous quality scale. This scale range from ‘bad’ to ‘excellent’. This is equivalent to a numerical scale from 0 to 100. The aforementioned methods are described in detail in the ITU-R Rec. T.500-11 document and are mainly intended for television signals. Based on slight modifications and adaptations of these methods, some other subjective evaluation methods like the Absolute Category Rating (ACR) and Degradation Category Rating (DCR) for multimedia services are described in ITU-T Rec. P.910 [12] and are listed below.

Absolute Category Rating (ACR) method is the most commonly used method and it gives the Mean Opinion Score (MOS) as a metric of measurement. The Degradation Category Rating (DCR) method is also used in some situations and it gives the Degradation Mean Opinion Score (DMOS) as a metric.

2.2.1.6 Absolute Category Rating (ACR) method:

In the ACR method, the viewers watch a video clip without watching the original reference clip. Once they have watched the clip, they are asked to give a quality rating. The quality rating is based on an opinion scale as shown in (Table 2.1) The Mean Opinion Score (MOS) is then calculated as an average of all opinion scores of the subjects.

Table 2.1: Opinion scale for ACR test

Category	Video quality
5	Excellent
4	Good
3	Fair
2	Poor
1	Bad

Figure 2.1 shows the time pattern for the presentation of the video. The time of voting is equal to or less than 10s.

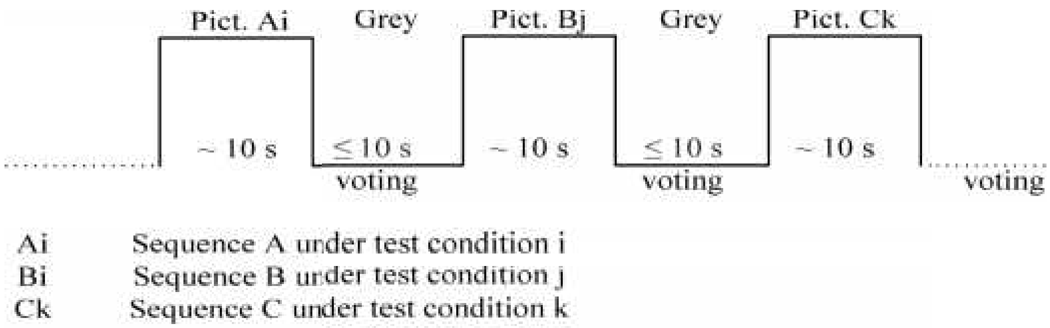

Figure2.1: Stimulus presentation in the ACR method

2.2.1.7 Degradation Category Rating (DCR) Method:

The DCR method is more effective when the quality difference in a video clip is minimum. The ACR method struggles to pick up the slight differences I quality (e.g. between 3 and 4). The DCR method is then recorded in an annoyance scale and a quality reference. The annoyance or degradation level is rated by the viewers by comparing the degraded video sequence to the original (reference). The rating scales or the degradation levels are shown in (Table 2.2)

Table 2.2: Stimulus presentation in the DCR method

Category	Video quality
5	Imperceptible
4	Perceptible but not annoying
3	Slightly annoying
2	Annoying
1	Very annoying

Figure 2.2 shows the time pattern for the presentation of the video clips.

The time of voting should be less than or equal to 10 seconds.

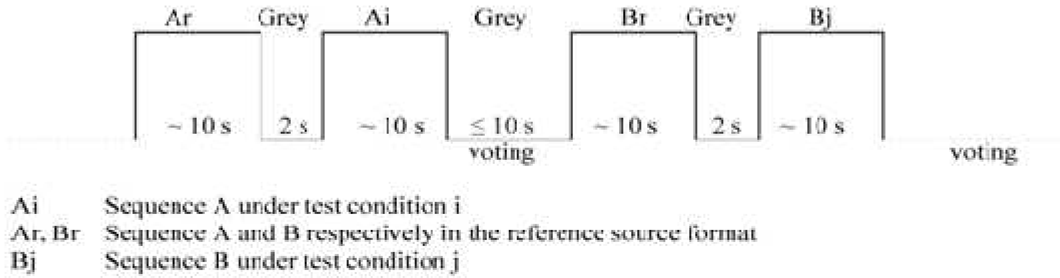

Figure **2.2**: Stimulus presentation in the DCR method

2.2.1.8 Pair Comparison Method (PC):

In pair Comparison method, pairwise comparison of test sequences is repeatedly conducted. In comparison to other methods such as single stimulus and double stimulus, the test sequences are combined in all possible combinations, all pair of sequences are presented in both possible orders and subjects only need to provide preference between each pair instead of assigning a discrete or continuous score. The time pattern for the presentation of video clips can be shown in Figure 2.3. As previously, the voting time should be less than or equal to 10 s.

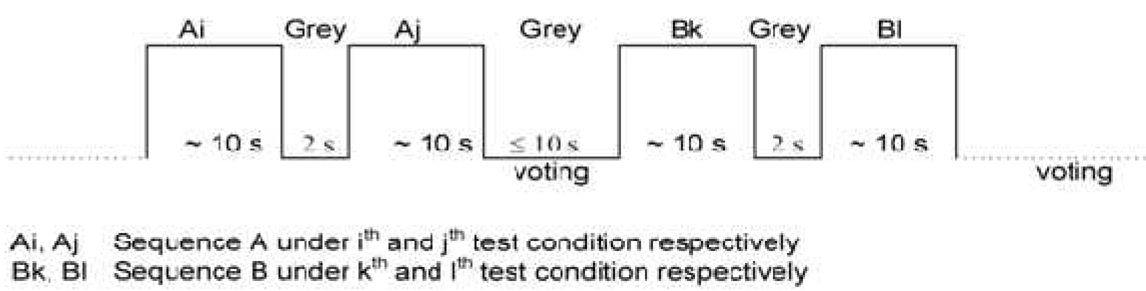

Figure 2.3: Stimulus presentation in the PC method.

2.2.2 Experimental Design of Subjective Tests

Once the data has been collected, several aspects must be taken into account in the experimental design of subjective tests. A description of these aspects is done in the next paragraphs. Subjective tests designed for this project were based on existing VoIP test in [14] and took into account all of the aspects described below

1. Scene Characteristics

It is important to select the test sequences representative of the data to be collected. Also, the choice of scenes should be different so the subjects are not bored. The test sequences should be the same for all subjects (scene characteristics is the same across the test) .It The short sequences should be less than 10 s and long sequences more than 10 s.

2. Replications

The replication of the video sequences is required by ITU-T P.910. At least two, otherwise three or four repetitions of the same test sequence should be shown to the viewers. This is important as it validates both subjects and the results produced by them.

3. Presentation Order

The presentation order of the video sequences should be randomized. It should be different to the number of subjects in the same test. However, when analyzing the results, presentation order should be taken into account. This is because if the viewers viewed a .bad. (degraded) the clip then viewed .afair, the, they may rate is as ,good.

4. Viewers

The number of viewers recommended by ITU-T standard is from 4-40. In practice, a minimum of 15-18 non-expert viewers should be used for obvious reasons. Ideally, they should be native viewers with little or no experience of these types of tests.

5. Viewing Conditions

The viewing conditions should be uniform for all viewers, e.g. display equipment, seating position, etc.

6. Instructions to Viewers

The subjects should be briefed about the intended application of the test to be undertaken. These instructions must be in writing to explain fully what is required from the viewers.

7. Training Session

A training session should be included before any real testing is carried out. The purpose of this session is to familiarize the users with the type of test.

8. Evaluation

As seen before in the test methods description part, different kinds of evaluative scales are used depending on the test method used. Grading scales (five-graded, seven-graded for comparison or even with more points) or continuous scales are possible, and the scale must be clearly illustrated during the training phase or before. In addition, the viewers can be given a specially designed questionnaire to rate the test.

Apart from that, it is important to decide if the evaluation of the video is concentrated on the whole video or in concrete objects. In the case of short video tests, the assessment of quality can be applied to consider the impact of the quality of single objects on the overall quality of the video clip, contrary to long videos that should be evaluated in general perspective. Subjective tests are composed of two phases: initial phase (instructions and training phase) and test sessions. A session should last less than half an hour. The sessions must be split up to provide for breaks so that the viewers are not tired or fatigued from the experience

2.2.3 Objective Quality Assessment

The subjective tests are expensive and time-consuming because a large sample of evaluators is needed to obtain results that are statistically meaningful. These challenges have limited the implementation of subjective test assessment methods, especially for research purposes. Additionally, the subjective test cannot be used in real-time video quality evaluation. Objective testing methods on the hand are quick and easy to set up, thus making them highly desirable for video quality evaluation. Study groups such as VQEG SG9 [15] is dedicated to finding effective objective methods that can be used to obtain results that are comparable to those of subjective video quality evaluation. Objective video quality measurements are divided into three main areas, this includes, full reference, Reduced-reference, and no-reference. Both full-reference and reduced reference approaches of quality measurement are intrusive while the no-reference approach is a non-intrusive measurement.

2.2.3.1 Intrusive Video Quality Assessment Methods

The intrusive approach to video quality measurement can be defined as a full reference and reduced reference.

1. Full Reference Methods

The full reference (FR) methods require access to the source or original video. This approach of quality measurement works by comparing the original video to the received degraded video. The comparison between the original and the source video provide an objective value which is used as an indicator of quality. They are impractical for real-time monitoring and estimation where access to the original video is not possible. However, in a lab environment, FR can be used to evaluate reference free prediction models. Examples of full reference video quality evaluation metrics include PSNR, SSIM, and VQM.

- PSNR

Peak signal-to-nosie ratio (PSNR) is widely used as an objective video metric. The objective value is found by comparing each video frame of the original sequence against the one in the degraded sequence. PSNR is defined as the Mean Squared Error (MSE) of two compared frames (1). The higher the PSNR the lower the MSE and hence, the higher the quality. PSNR values are given in (2).

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j - K(i,j)]^2 \qquad (1)$$

$$PSNR = 20\log\left(\frac{\max_i}{\sqrt{MSE}}\right) \qquad (2)$$

where **I** and **K** two arrays of size mn monocrhrome image, respectively. $\max_i$ denotes the maximum pixel value of the frame I

- SSIM

Structure Similarity Index Measurement (SSIM) provides the index Measure of the similarity between two frames .It was developed by wang [16]. The measure between two frames x and y of size mn is given by Equation (3).

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \qquad (3)$$

μx,μy σx,σy.,σxy be the mean of x, the mean of y, the variance of x, the variance of y, and the covariance of x and y, respectively.c1,c2 is constant

SSIM in Equation (3) is applied luma and the maximum value of 1 will denote that the video is of excellent quality. Structure Dissimilarity (DSSIM) is given in Equation (4).

$$DSSIM(x, y) = \frac{1}{1 - SSIM(x, y)} \qquad (4)$$

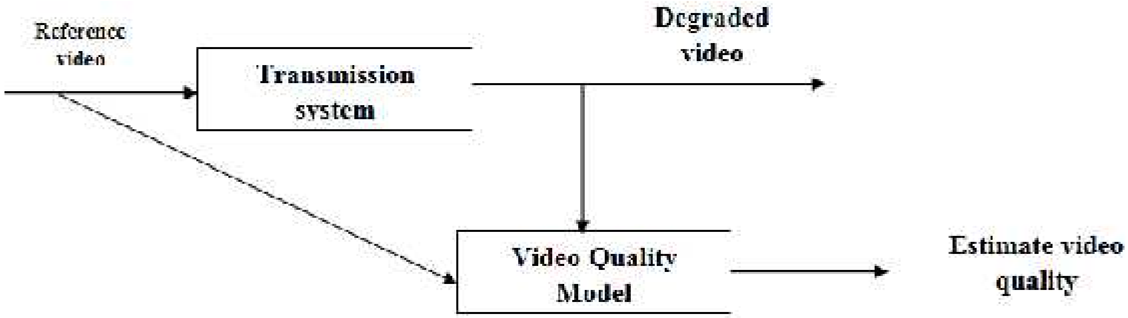

Figure 2.4: Full reference method

2. Reduced Reference Methods

Unlike the full reference quality measurement method, reduced reference method uses only some features extracted from the original video sequence. Therefore, if a reduced-reference method is to be used in an Internet Protocol television (IPTV) system, the user requirement should specify the side channel, through which the feature data are transmitted.

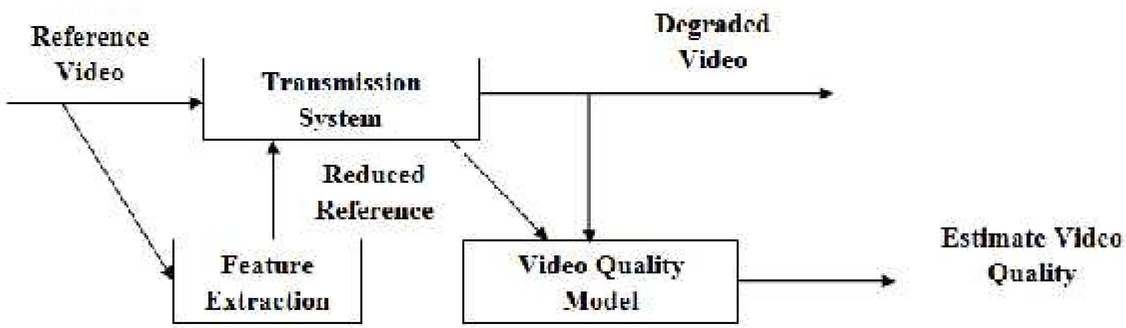

Figure 2.5:Reduced Reference Methods

2.2.3.2 Non-Intrusive Video Quality Assessment

In contrast to both full and reduced reference quality measurement methods, reference-free video quality measurement methods do not require access to the original video sequence. The video quality is estimated using information extracted from either the degraded bitstream or from the decoded video clip. In order to monitor the perceptual video quality at the receiver side, it is difficult to use the full-reference methods as they require access to the original video which may not always be available. This makes reference-free methods an attractive option for quality estimation, especially on end users' side as they do not have access to the source video sequence In general, objective quality metrics are classified into five main categories by ITU standardization activities [17], these include:

1. Media-layer models: This category of models uses the video signal to compute QoE without requiring any information about the system under test. This type of objective measurement is suitable for codec comparison and optimization scenarios

2. Parametric packet-layer models: These models use packet-header information for QoE prediction without having access to the media signals. This is a lightweight solution for QoE prediction considering that the models do not have to process the media signals.

3. Parametric planning models: QoE prediction using this type of models is based on quality planning parameters for networks and terminal devices. As a result, prior knowledge about the system under test is required.
4. Bitstream-layer models: The type of models uses information from the encoded bitstream and packet layer to predict QoE.
5. Hybrid models: These models are based on a combination of two or more models mentioned above.

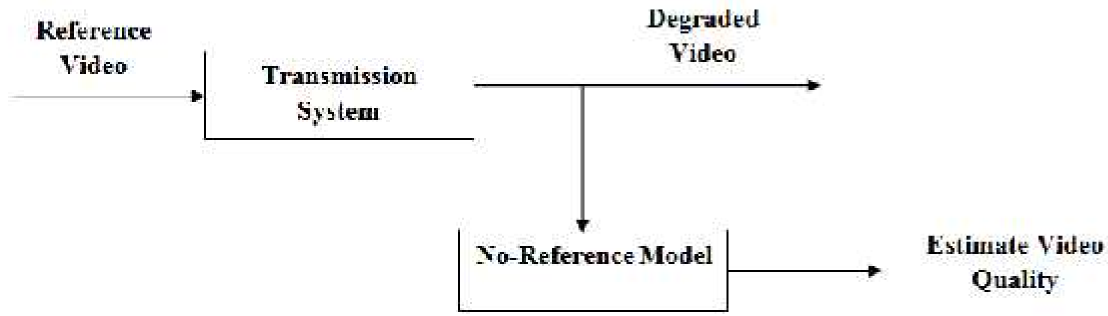

Figure 2.6: No-reference (NR) models

2.3 Video Coding Techniques

2.3.1 Coding Basic Concept

Video compression algorithms manipulate video signals to dramatically reduce the storage and bandwidth required while maximizing perceived video quality. In general, video coding techniques are divided into two categories as lossless and lossy.

- Lossless:

In this category will always make sure that the original data can be recovered exactly in its original format.

- Lossy:

In this category states that the original data cannot be retrieved fully. The theory of coding is that small quantization errors in the high-frequency components of the image are not apparent by the human visual system.

The progress of video coding research can be well represented via the development of video coding standards by ITU-T Video Coding Experts Group (VCEG) and ISO/IEC Moving Picture Experts Group (MPEG) [14][18].

2.3.2 Sampling and YUV Format

Both MPEG4 and H.264 use YUV format. YUV format separates a pixel into luma and chroma components. There are one luma (Y) component and two chroma components (U, V) per pixel. The luma component is associated with the brightness of the pixel while the other two chroma components are associated with the color of the pixel. In this thesis YUV 4:2:0 format has been used. In YUV 4:2:0 format, every pixel has its distinct luma component while every 2x2 block of pixels shares the same chroma (U, V) components. Since the human visual system is more sensitive to brightness (luma) than color (chroma), so the chroma components are undersampled [14] .

2.3.3 MPEG-4

ISO/IEC Moving Picture Experts Group defines MPEG4 video compression standard for encoding and decoding video storage and transmission over IP networks. Parts 2 and 10 out of 23 parts specify how the uncompressed video is compressed ready for transmission over the IP network. Part 10 defines Advanced Video Coding (AVC) most commonly known as H.264. In MPEG-4, a series of consecutive frames are divided in a Group of Pictures (GOP). Within a GOP, frames are of three different types, normally starting with I-frame followed by a number of P-frames and B-frames.

1. Intra coded frames (I-frame):

I-frames are coded as a single picture. They do not have references to either P-frame or B-frame and are relatively bigger than other frame types.

2. Predictive coded frames (P-frame):

P-frames are coded from previous I or P frames and are normally Smaller than the I frames, but bigger than the B-frames.

3. Bi-directional coded frames (B-frame):

B-frames are coded from preceding and succeeding I or P frames in the GOP structure. B-frames are normally the smallest.

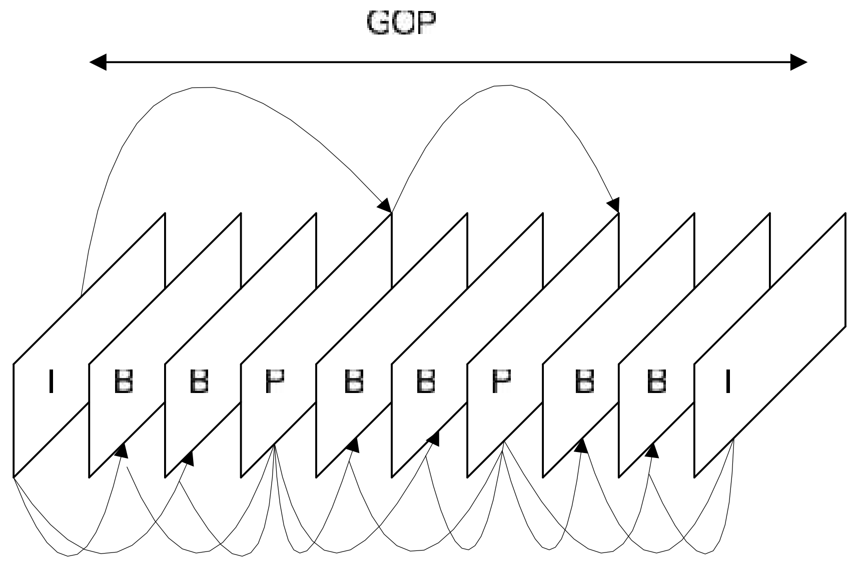

Figure **2.7**:MPEG-4 GOP structure

A sample of MPEG-4 GOP structure is depicted in Figure 2.7. GOP(N,M) denoted N as the I to I frame distance and M is the I to P frame distance. Figure 2.7 can be presented as GOP(9,3), this means that one I-frame, 6 Bframes and 2 P-frames are in the GOP. The second I-frame marks the start of the next GOP. Arrows in Figure 2.7 indicate that the B-frames and P-frames are decoded depending on the succeeding and preceding I-frames or P-frames.

2.3.4 H.264/AVC Advanced Video Coding

H.264/AVC is the latest codec and is also known as the next generation in the video compression technology in the MPEG4 standard. It is also known as MPEG4 part 10 and as Advanced Video Coding (AVC). The H.264 coded can produce the same quality as MPEG2 but at up to half the data rate. H.264 is the recommended codec for 3G to HD (High Definition) i.e. everything in between 40kbps to upwards of 10Mbps.

In this thesis H.264/AVC baseline profile (there are seven profiles recommended specific targeting classes of applications) has been used as it is designed for low complexity and low rate applications: (the target study of this thesis is mobile applications). The baseline profile does not support B slices. It only supports I and P slices. The Network Abstraction Layer (NAL) configures the video Coding Layer (VCL) illustration of the video and procures header information in a way suitable for movement by a collection of transport layers or storage media. The NAL units (NALUs) are the output of NAL. The optional part is Data Partitioning (DP).

A hybrid of temporal and spatial prediction is found in the VCL of H.264/AVC together with transform coding. Each frame is divided into no overlapping areas. These areas are called macroblocks (MB). They consist of 16×16 samples of the luma and 8×8 samples of each of the two chroma components. The macroblocks are arranged in slices which represent subsets of macroblocks that can be decoded independently. The macroblocks can be further divided into smaller blocks of up to 4 x 4 pixels.

2.3.5 Video Codec

It converts or encodes raw (uncompressed) image or video sequence into a compressed format and decompresses or decodes this to get a copy of the source sequence. The device which is used for compressing is called "encoder". While the other device which is used for decompressing is called "decoder". Figure 2.8 shows the block structure codec of H.264/AVC [19].

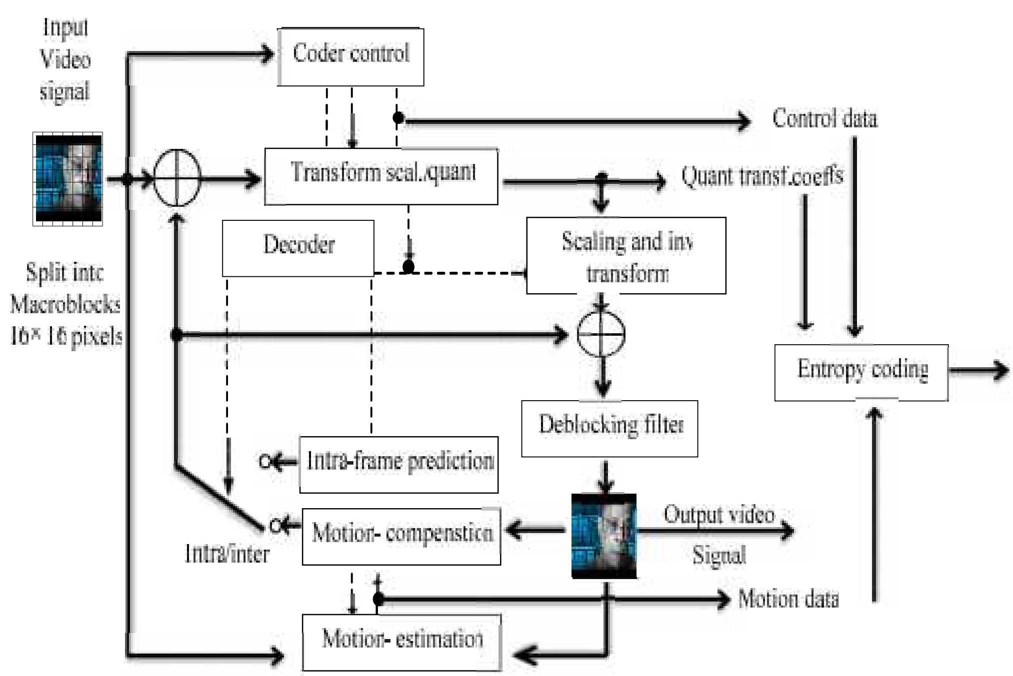

Figure 2.8: The block structure codec of H.264/AVC

2.3.6 Video Structure

The structure of a video is formed by splitting video sequences into frames. Frames are partitioned into slices, the following explains the slice and macroblock structures.

2.3.6.1 Slices Structure

Slice is known as a group of macroblocks that are merged together. Each slice is decoded independently of other slices in a frame. Typically, one encoded slice is packed in one NAL unit and transmitted in one packet. The purpose of using slices is that if a packet is lost then only a part of a video frame is lost instead of the whole frame. Since slices reduces compression efficiency, the inter prediction between macroblocks of different slices is not allowed. H.264/AVC standard defines the following slice types:

I slice: I or Intra slices include macroblocks that are encoded using macroblocks in the same slice of the same frame. All the slices in the first frame of a video sequence are encoded as I slice.

P slice: P or Predicted slices contain macroblocks that are encoded using macroblocks in a previously encoded and decoded frame. Some macroblocks in a P slice may be encoded in intra mode.

B slice: B or Bi Directional Predicted slices contain macroblocks that are encoded using macroblocks in the past and future I or P slices (in playback order). The decoding order of B slice is after the past and future I or P reference slices.

2.3.6.2 Macroblock Structure

The macroblock (MB) forms a block of 16x16 pixels. MB can be separated into different forms such as sub-macroblock of 16x8, 8x16 and 8x8. Sub-macroblock in H.264/AVC extends to 8x4, 4x8 and 4x4 blocks. Figure 2.9 illustrates the macroblock and sub-macroblock partitions [20].

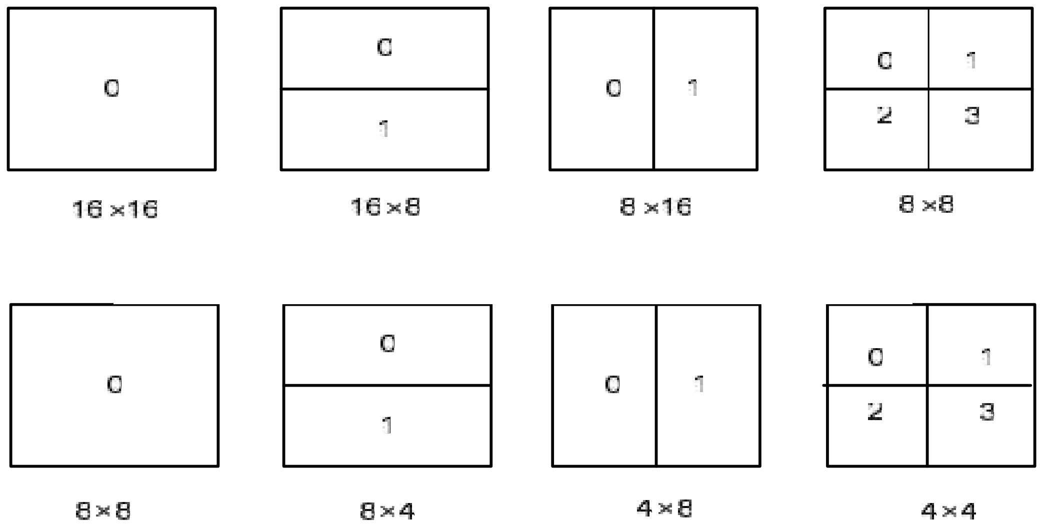

Figure 2.9:Macroblock/sub-macroblock partitions

2.3.7 H.264/AVC Encoder Process

In Figure 2.10, the flowchart explains the encoder process which includes three tools addressed as a prediction tool, spatial tool and entropy encoder. The input raw (uncompressed) video sequence is fed into the prediction model which exploits the similarity between neighbor frames to decrease the redundancy. The prediction includes data in the current frame or one or more previous and/or future frame which are generated by spatial extrapolation from neighboring image samples or by compensating for the difference between the frames called inter or Motion Compensation Prediction (MCP). The result of this difference is named as (residual frame) which forms as an input to the spatial model where the encoder transforms the residual samples into the other domains called transform coefficients. Quantization is used to decrease the number of coefficients and gives a set of quantized transform coefficients. These are then used as input to the entropy encoder. The entropy encoder removes statistical redundancy by creating a compressed bit stream which includes coded prediction parameters, coded residual coefficients and header information .The bit stream is transmitted and/or stored [21], [22].

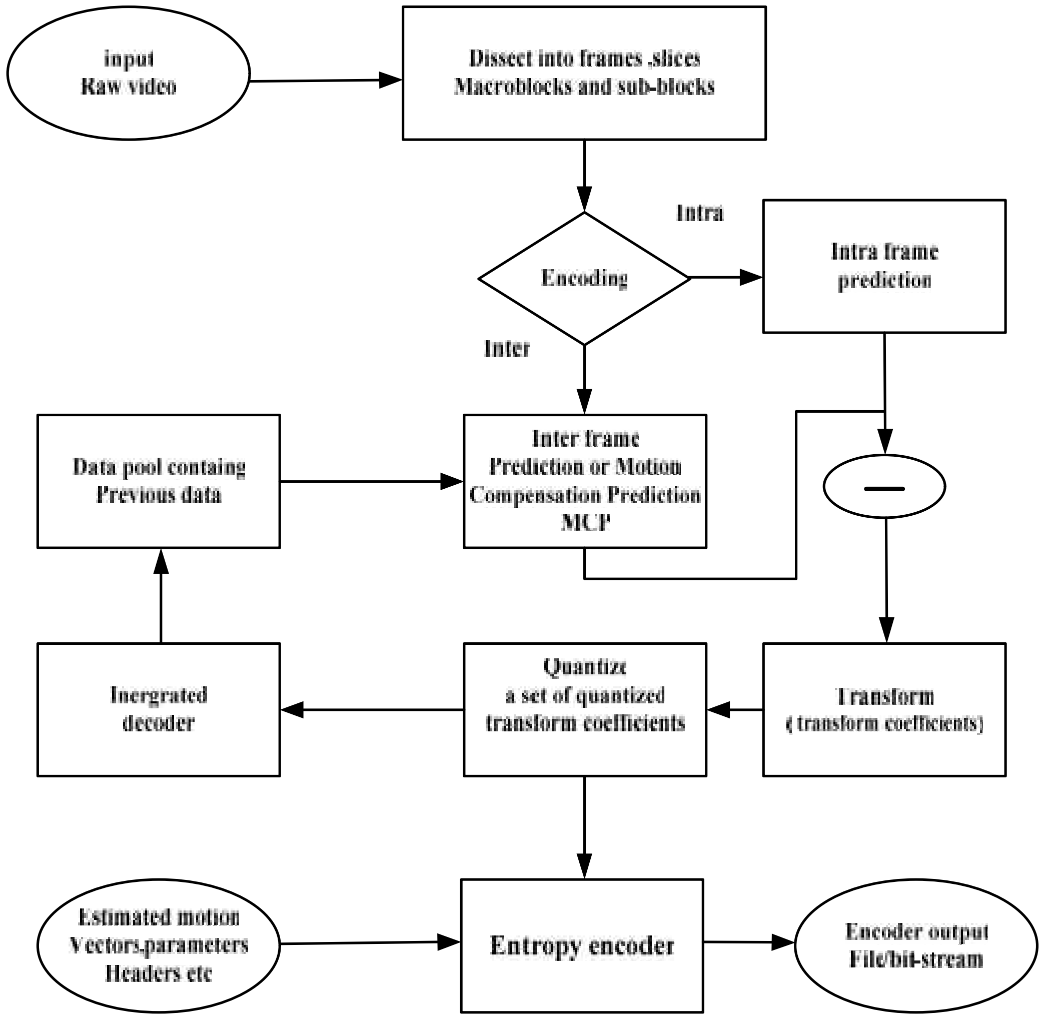

Figure 2.10:The flowchart of the encoder process

2.3.8 H.264/AVC Decoder Process

After the encoder has transmitted compressed bit stream, the entropy coding at the decoder side calculates the same prediction value of the frame by using the entropy coding based on the information and motion data. The decoder also transforms the quantized transform coefficients to suitable form of residual which were added to the prediction. They will then be inserted into de-blocking filter which provides the decoder video as its output. The de-blocking filter is mandatory for theH.264/AVC decoder because it differentiates the previous codecs by enhancing the block artifacts at 4x4 sub block boundaries to achieve the quality. Figure 2.11 explains the flowchart of the decoder process after the encoder has compressed the raw video, where the result is the decompressed bit-stream to get video data [21], [22].

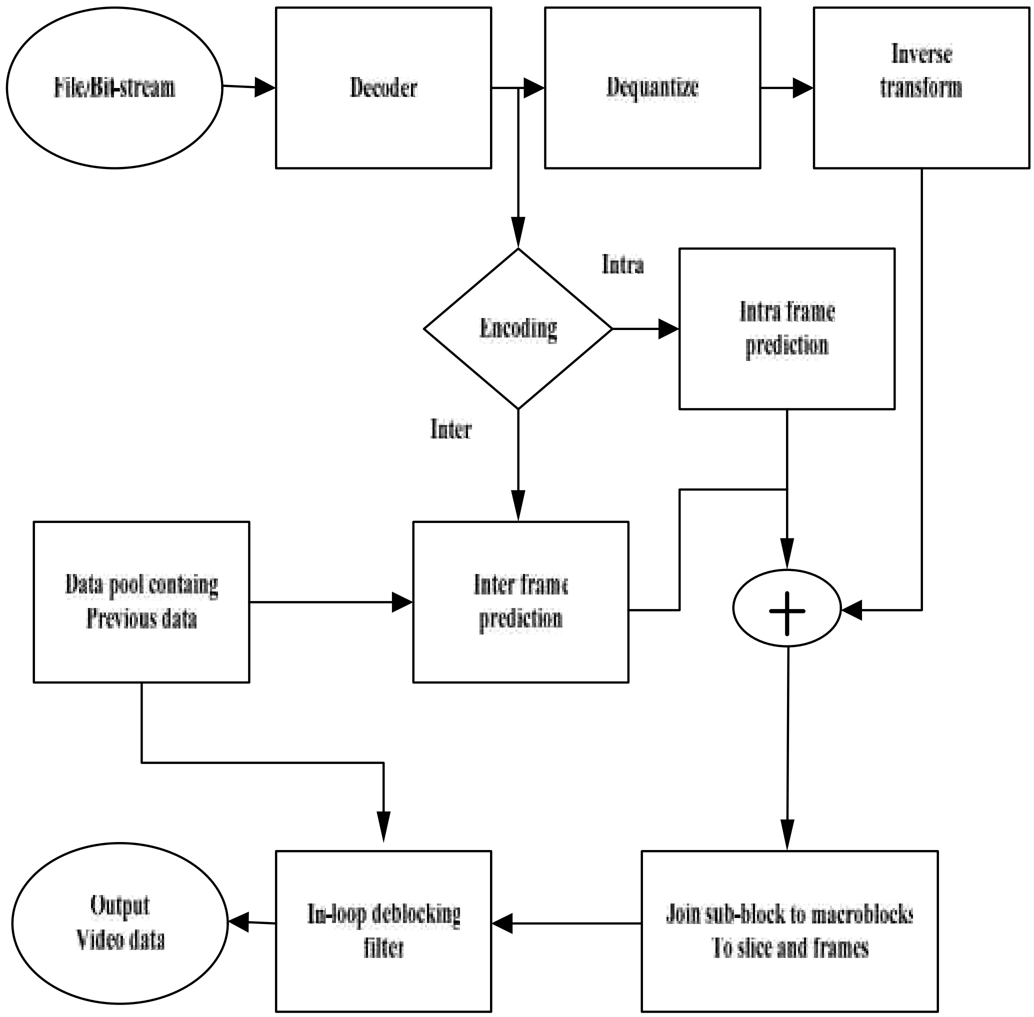

Figure 2.11:The flowchart of the decoder process

2.4 The Impact of Gender on the Quality of Experience for Video Services

2.4.1 QoE Research Challenges

Researchers considering the problem of QoE modeling, measurement and prediction face a number of challenges. These include:

1. QoE Modeling

QoE measurement and prediction may involve a large parameter space comprising of several QoE and context parameters as shown in Figure 2.12 [16]. There can be N context parameters affecting M QoE parameters. Further M QoE parameters can affect each other. Thus, selecting relevant parameters and finding relationships between these parameters can be challenging. The relationships between these parameters are usually non-linear and hard to quantify. This necessitates the development of novel QoE modeling techniques to model all these parameters efficiently. The QoE models should not only be conceptual but should also transcend to solving the challenges associated with QoE measurement and prediction. For example, rather than simply classifying and representing the parameters, QoE models should directly be used for QoE measurement and prediction.

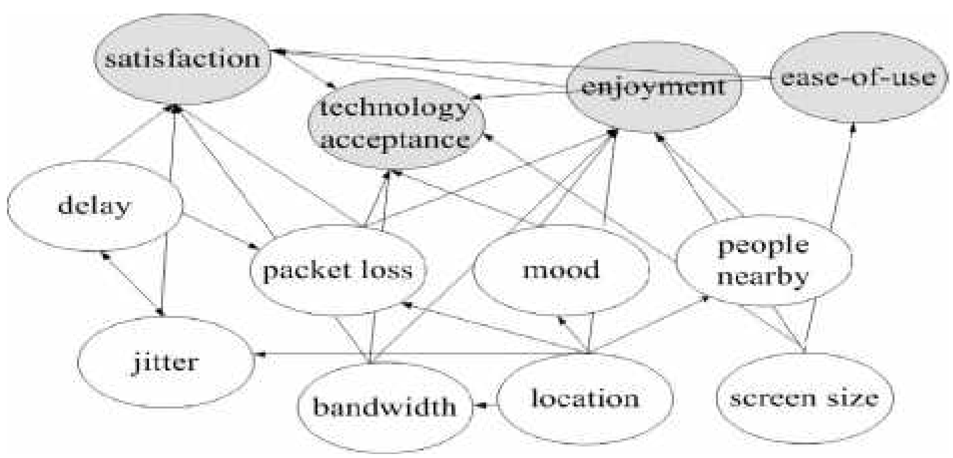

Figure 2.12: Parameter relationships between context and QoE parameters. Grey ovals depict QoE parameters and white ovals depict context parameters

2. QoE Measurement and Prediction

The challenge of QoE measurement and prediction involving multiple QoE and context parameters is not well addressed. Considering Figure2.9, each QoE parameter can be measured on a different scale and may involve different units of measurement [23], [24]. These scales can be qualitative Figure 2.13 (a) or quantitative Figure 2.13 (b).

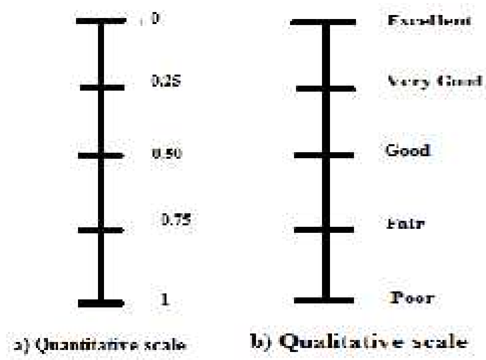

Figure 2.13: Typical scales for QoE measurement

2.4.2 QoE Modeling

QoE is considered by video service providers to reflect service quality from users' perspective. However, the estimation of QoE is hard due to multiple factors involved in the complicated service context and the divergence in users' perception. In order to evaluate the QoE from a holistic and unified view, it is necessary to understand the communication ecosystem where various factors interactively affect users' experience [23].

A user-centric communication ecosystem incorporates different domains such as technology, business, context, and human. The technology domain is concerned with the service itself, which is provided by the equipment manufacturers, the networked operators, and the service providers. The business domain provides the metric to regulate the utility functions of the actors in the ecosystem, which directly influences the final intention of purchasing a service and the price at which a provider can offer the service.

The context in a communication ecosystem represents the circumstances and situations at the time of interaction among human, technology, and business entities [25], which includes both the natural factors and social factors. For instance, the natural factors consist of noise, illumination, temperature, etc, while social factors include the policy, custom, relationship and so on. The human domain focuses on users' needs, feelings, performance, and intentions for services, and includes the psychological, physiological, and cognitive factors. This domain interacts with the other domains, and its influence in the communication ecosystem directly forms the QoE [26]. To evaluate the QoE accurately, the influences of these domains should be taken into consideration.

A number of mathematical QoE metrics have been developed and used for quality management in video services in order to achieve a high user satisfaction. However, these metrics are limited as they take only a few aspects of user experience into consideration.

In [27] the authors proposed a model of QoE evaluation for networked services regarding the communication ecosystem. In this model, both the technology and the human domain are considered and incorporated to evaluate the users' (QoE) of a realistic context in an IPTV service. The QoE influential factors of the human domain including watching duration, the frequency, and duration of fast forward, are integrated with those in the technology domain such as video quality, to establish the objective model for accurate QoE prediction. The proposed model is well consistent with the subjective QoE.This model needs enhancing and improvement by adding more influential factors.In addition, the model does not define contextual parameters.

A mobile video environment model is described in [28]. The proposed framework is simple but encompasses many factors and clearly states each factor’s contribution to the overall user experience. It may benefit from the user-centered design of mobile video delivery and relative research.

Mobile video vendors may develop effective strategies to improve user experience by taking into consideration the factors in different components of the framework. This model does not propose exact formulae for QoE calculation. It organized QoE influential factors into three components: user, system, and context and mapped their impacts upon four elements of the mobile video delivery framework, namely, mobile user, mobile device, mobile network, and mobile video service.

Authors of [29] presented a QoE model for measuring user experience of videos services. Their model produced very interesting categorization of QoE, QoS, and business aspects based on measurable and non-measurable parameters. They considered technical parameters as measurable parameters and subjective user parameters such as satisfaction and attitude as non-measurable parameters. However, in our view, subjective context factors can also be quantified using some empirical approaches

The work in [30] presented a simple and intuitive interaction between a person, technology, and business. However, it neither provides a classification of QoE factors into subcategories nor any details on the taxonomy. More importantly, Kilkki's model does not define contextual parameters in any way.

The Authors of [31] presented an initial conception of a QoE framework with a special focus on human behavior, technology, and business. They demonstrated its application through a use case based on service delivery of composed services. The initial QoE conceptualization needs further enhancement and improvement in terms of considering more concepts, taxonomy, and interdomain mapping using the Template.

2.5 QoE Evaluation and Power-Driven Video Quality in the IoT

IoT is defined as a network of interconnected objects which are able of acquiring some information from the physical world and make this information available on the Internet [32].Consequently, Multimedia IoT (MIoT) can be considered as a "network of interconnected objects capable to acquire multimedia contents from the real world and/or present information in a multimedia way.

The emerging categories of IoT objects tend to be mobile, multi-sensorial and smart, such as wearable sensors, Smartphone's, and smart vehicles, bringing also to an increase of multimedia content in the IoT. Multimedia content refers to a combination of two or more different media contents such as text, audio, image, video, etc.

A few studies that address the overall QoE and power-driven video quality of IoT are as follows.

The author [33] presents a comprehensive model for the power consumption of wireless sensor nodes. The model takes a system-level perspective to account for all energy expenditures: communications, acquisition, and processing. Furthermore, it is based only on parameters that can empirically be quantified once the platform (i.e., technology) and the application (i.e., operating conditions) are defined. This results in a new framework for studying and analyzing the energy life-cycles in applications, and it is suitable for determining in advance the specific weight of application parameters, as well as for understanding the tolerance margins and tradeoffs in the system. In our views, This model needs enhancing and improvement by considering the end user's Quality of Experience (QoE) in the context of video services which is very important when we came to reduce power consumption for the IoT devices.

To increase the battery life to complete the VoIP session the author in [34] proposed VoIP quality adaptation scheme whereby an acceptable quality is maintained by changing video send bitrate in order to conserve power and hence, prolong VoIP communication session .the results have shown the effectiveness of the proposed scheme in terms of power saving while maintaining acceptable QoE. The power saving was between 10-30% of the total system power .this model it is only for mobile device and we can enhance this model by including the IoT devices and adding more influential factor such as the frame rates and the context parameter such as gender to offer an acceptable quality for the end users.

In [35] the author presents a framework that reduces the energy consumption of wireless mobile devices during streaming video content over Wireless LAN networks by utilizing the power saving mechanisms defined in the IEEE 802.11e standard. The framework allows video applications to directly optimize the overall energy efficiency by controlling the sleep cycles of wireless network adapters based on video QoE. The framework employs a simple QoE estimation algorithm that estimates the user level video quality and adjusts the sleep intervals of the wireless adapter to maintain video quality while maximizing power efficiency. Although the framework was claimed to save 20-30% of total system power but did not answer some key concerns of practical interests in the real world implementation of VoIP communication. Under the presence of VoIP signaling traffic the sleep cycles technique is not realistic. The ideal technique is to reduce power consumption of the AV application while keeping the VoIP quality at an acceptable QoE level.changing the frame rates and bitrates for the videos will reduce the power consumption for the IoT devices and offering an acceptable quality for the end users.

In [36] The author proposed a simple QoE model and several power models relevant to mobile video service are formulated by considering that user's QoE is constrained by factors like devices and the ambient environment, a QoE-driven adaptive streaming scheme to help save radio resources and reduce power consumption. Based on these models, a QoE-driven energy-efficient resource allocation algorithm is introduced. Once QoE driven adaptive streaming scheme and QoE driven energy-efficient resource allocation algorithm are adopted the power consumption can be saved at both base station side and user equipment side while not decreasing user’s QoE at the same time. A non-network parameter such as video coding parameter and context parameter such as gender is important to ensure an acceptable quality of videos services on IoT devices for end users.

In [37] the authors conducted subjective experiments and they focused on the perceived quality in actuators connected to the IoT. They developed a testbed that consisted of an electro-mechanical arm controlled over a packet switched unreliable link. The experiment required users to direct a fixed laser attached to the fixed arm’s grabber towards a set of targets. The experimental factors were the average one-way delay, the packet loss and the number of degrees of freedom of the arm. From the subjective quality results, in terms of the Mean Opinion Score (MOS), the authors defined a network QoS metrics which could not reflect the characteristics of subjective perception on video service. They ignored the other quality metrics such as non networks parameter and context parameters which are important for the end users moreover, using objective test would have made the result more accurate.

CHAPTER III

3. Internet of Things

3.1 Introduction

The Internet of Things (IoT) is a new paradigm that is increasing in popularity. It is defined as "an interconnection of uniquely identifiable embedded computing things within the existing Internet infrastructure, offering advanced connectivity of things , systems, and services that goes beyond machine-to-machine communications and covers a variety of protocols, domains, and applications"[38]. The number of Internet of Things devices will grow exponentially in the coming years (Table 3.1). Looking to the future, Cisco estimates that IoT will consist of 50 billion devices connected to the Internet by 2020[39].

The main idea of IoT has also evolved since its creation, from offering status information to automating whole systems. The evolution of hardware has also helped in the expansion of this paradigm, it is cheaper, consumes less power, and nowadays nearly everybody has a mobile phone with the capability to use a whole lot of connection types, so people can interact with objects anywhere anytime. Connecting the devices to each other allows them to share data among themselves and make intelligent decisions based on the data. Any physical device which is connected to some sort of network can be called an IoT device.

Table 3.1:Internet of things devices

World population	6.3 Billion	6.8 Billion	7.2 Billion	7.6 Billion
Connected Devices	500 Million	12.5 Billion	25 Billion	50 Billion
Connected devices per person	0.08	1.84	3.47	6.58
Year	2003	2010	2015	2020

3.2 IoT Architecture

There is no single consensus on architecture for IoT, which is agreed universally. Different architectures have been proposed by different researchers.

The traditional IoT architecture proposed in [40], is based on a layered model which has 3 layers:

1. **Physical level layer**:

The main function of this layer is to acquire the physical information of an environment with different scalar devices and then transmit the data to the network layer via the Wireless Sensor Network (WSN) gateway.

2. **Network Layer:**

This layer is responsible for connecting to other smart things, network devices, and servers. Its features are also used for transmitting and processing sensor data.

3. **Application Layer:**

This layer is a combination of service layer and application layer. The service layer stores the information from the network layer, it also provides decision making, data analysis, and information management. It defines various applications in which the Internet of Things can be deployed, for example smart homes, smart cities, and smart health.

3.3 IoT Application Domain

The IoT is becoming more and more used in several areas of activity since it enables the creation of a multitude of applications and services. By means of the use of large volume and variety of data produced by networked devices, the IoT fosters the generation of new applications [40].
Internet of Things examples extend from smart connected homes to the smart city to healthcare. In fact, IoT is slowly becoming part of every aspect of our lives. Not only is the Internet of Things application enhancing our comfort, but they also give us more control to simplify routine work, life and personal tasks. Here are Internet of Things application areas that have the potential for exponential growth:

1. **Healthcare:**

Remote patient health real-time monitoring, patient-flow monitoring, identification and authentication of patients and staff and automatic medical inventory management are examples of healthcare applications.

2. **Personal and Social:**

The user is enabled to interact with other people to maintain and build social relationships. Applications concern the automatic update of information about social activities in social networks and search engines for things with radio frequency identification (RFID) to prevent losses and thefts.

3. **Smart Home/Smart Building:**

Instrumenting buildings with advanced IoT technologies brings to adaptable rooms heating,cooling and lighting, monitoring and alarm systems, smart household appliances and the optimization of power consumption costs (together with the smart grid).

4. **Smart City:**

IoT applications concerned with the optimization of physical city infrastructure, smart parking and pollution monitoring.

5. **Smart grid and Smart Metering:**

The energy consumption can be efficiently monitored in a smart home or in a small office or building.

6. **Smart Environment:**

Monitoring and transmission of critical parameters of the environment, access to critical areas that are dangerous for human operators (volcanoes, etc.) and natural disaster monitoring and detection.

7. **Transportation:**

Smart vehicles equipped with sensors and actuators, together with WSN produce information for assisted driving and traffic monitoring applications; identification and monitoring of objects, people, and animals in motion with tracking applications.

8. **Smart Business and Logistics:**

RFID technology used for inventory management and monitoring throughout supply/ delivery chain, monitoring of real-time product availability and retrieval of products' data after purchase.

9. **Security and Surveillance:**

Smart video cameras with automatic behavior analysis and event detection, ambient sensors, alarms, and RFID for personal identification.

3.4 Challenges for Using IoT

Here are four IoT challenges facing IoT solutions:

1. **Battery Life**

Most IoT things IoT sensors, smart meters, location trackers, and so on, use batteries that might need to keep running for months or years. These things are typically small, with space constraints that limit the size of batteries. As a result, preserving battery life becomes critical.

2. Data Costs

The amount of data that will be generated by billions of IoT things will dwarf anything we've seen on the Internet to date. And the types of networks becoming prevalent in IoT implementations have stringent limitations on the amount of data that can be sent. Data networks are expensive, and the frequent, bursty messages produced by IoT things can make data costs spiral quickly out of control.

3. Operational Efficiency

IoT solutions often communicate over slower wireless networks. This means that more actionable data needs to be extracted faster.

4. Low-Power Networks

Today's wireless networks, especially the low-power WANs (LPWANs) increasingly used in the IoT, run at extremely low data rates. These low data rates impose severe limits on the amount of data that can be communicated over a wireless network.

3.5 The IoT Hardware and Software

3.5.1 Hardware for Smart Things:

Smart things contain a piece of hardware, which is a set of electrical circuits. The hardware consists of four main components, as shown in Figure 3.1 .

1. Communication Device

This gives the smart things its communication capabilities. For wireless smart things, the communication device typically is a radio transceiver. For wired smart things, the communication device connects to a wired network connection such as Ethernet or Power line communication (PLC).

2. Microcontroller

It gives smart things their intelligence. It runs the software of the smart things and is also responsible for connecting the radio with the sensors and actuators.

3. Set of Sensors or Actuators

Smart things interact with the physical environment in which they are deployed by using sensors and actuators. Sensors are used to sense the environment and actuators are used to affect or change the environment.

4. Power Source

A smart thing is driven by electronics, and electronics need power. Therefore, every smart thing needs a power source. Today, the most common power source is a battery, but there are several other possibilities for power, such as solar cells, piezoelectricity, radio-transmitted energy, and other forms of power scavenging.

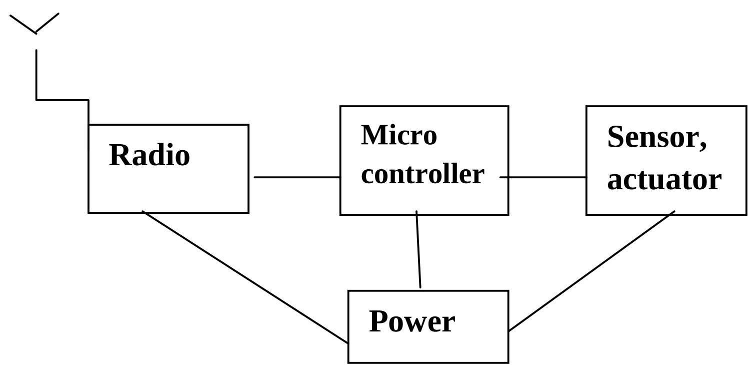

Figure 3.1: Hardware architecture of smart things with radio-equipped

3.5.2 Software for Smart Things

The behavior of smart things is defined by the software running on the microcontroller inside the smart device. The software inside the smart device is usually written similar to software for general purpose computers. The programs are written in a programming language, such as C, and compiled with a compiler to machine code for the microcontroller. The machine code is written to the Read Only Memory (ROM) of the microcontroller when the smart device is manufactured. When the smart device is switched on, the microcontroller runs the software.

3.5.3 Operating Systems for Smart Things

Like general purpose computers, smart things use operating systems. These operating systems are very different from general purpose operating systems used on PCs and mobile phones. The severe resource constraints regarding memory and processing power make a large-scale operating system such as Microsoft Windows, Mac OS X, or Linux impossible to use. Even scaled-down versions such as Microsoft Windows Mobile or the Linux-based Google Android are too large.

Operating systems for smart things are tailored to the specific requirements of smart things and to the specific constraints imposed by the hardware. The memory constraints make the programming model different from general purpose operating systems. The processing speed constraints require the use of low-level programming languages, such as the C programming language.

3.5.4 Operating System Examples

1. Contiki Operating System

The Contiki operating system is an open source operating system for networked embedded systems in general, and smart things in particular [41]. Extant uses for Contiki include systems for street lighting, sound monitoring for smart cities, radiation monitoring, and alarms. It is open-source software released under a Berkeley Software Distribution (BSD) license. Contiki provides multitasking and a built-in Internet Protocol Suite Transmission Control Protocol/Internet Protocol (TCP/IP stack), yet needs only about 10 kilobytes of Random-Access Memory (RAM) and 30 kilobytes of Read-Only Memory (ROM). A full system, including a graphical user interface, needs about 30 kilobytes of RAM. Contiki is designed to run on types of hardware devices that are severely constrained in memory, power, processing power, and communication bandwidth. A typical Contiki system has memory in the order of kilobytes, a power budget in the order of milli-watts, processing speed measured in megahertz, and communication bandwidth in the order of hundreds of kilobits/second. Such systems include many types of embedded systems and old 8-bit computers[42].

Contiki is an open source operating system for the Internet of Things. Contiki connects tiny low-cost, low-power microcontrollers to the Internet. Contiki is a powerful toolbox for building complex wireless systems, and provides powerful low-power Internet communication. it supports fully standard IPv6 and IPv4, along with the recent low-power wireless standards: 6lowpan, RPL, CoAP. With Contiki's ContikiMAC and sleepy routers, even wireless routers can be battery-operated. The development is easy and fast and applications are written in standard C. With the Cooja simulator Contiki networks can be emulated before burned into hardware, and Instant Contiki provides an entire development environment in a single download. It can be freely used both in commercial and non-commercial systems and the full source code is available.

The Internet of Things (IoT) is all about interconnecting of embedded devices to form a local mesh network or connect directly to the internet. The existing networking stacks are not optimal for IoT devices due to constraints like low power (and in most cases, human intervention to replace the battery is not available) and low processing power. For this type of requirements, the IETF (Internet Engineering Task Force) has drafted a few protocols to accommodate the new network requirements. To test and debug these new protocols, in addition to considering the functionality of the protocol, the properties of the device like battery performance and memory usage should be tested. For such requirements the traditional network simulators won’t suffice, so there are simulators especially designed for these devices. One such simulator for IoT devices is Cooja simulator.

2. TinyOS Operating System

Like Contiki, TinyOS is an open source operating system for smart objects and sensor networks. It was originally created at the University of California, Berkeley, but is currently being worked on by a team from Stanford University [43]. The initial versions of TinyOS were released in 2000. It is primarily used for research into WSNs and has a large user base from academia. TinyOS focuses on networking and communication mechanisms for WSN.

Programs in TinyOS are written to resemble the way hardware is designed. Programs are event-driven and consist of callback functions invoked in response to events, both external and internal.

3. The FreeRTOS Operating System

FreeRTOS is a small, open source operating system designed for embedded systems. Unlike Contiki and TinyOS, FreeRTOS provides real-time guarantees to applications. This means that applications running on top of FreeRTOS can schedule exactly when they want events in the system to occur.

This is important, for instance, in control applications where timing is of the essence. For example, an application that controls a robotic arm must be able to specify exactly when to turn the robot motor on and off or else the arm movements will be incorrect. FreeRTOS uses a preemptive, multi-threaded programming model.

4. Embedded Linux

This OS is a lightweight version of the Linux kernel that is intended for use on hardware with clear limitations; however, it is not suitable for use on resource-constrained things because it requires approximately 1 MB of RAM and 1 MB of ROM, specifications that a low-power resource-constrained device cannot meet. It can run any programming language, including Java and Python. Moreover, it can use most of the available programs for desktop versions of Linux [44].

5. OpenWSN

It is not an OS by itself, but it must be included in this list [45]. OpenWSN is an open-source implementation that provides a complete protocol stack based on IoT standards, supporting both User Datagram Protocol (UDP) and TCP connections. It runs on top of OpenOS, FreeRTOS, and RIOT. A simple application requires approximately 14 KB of RAM and 50 kB of ROM. The programming language is C.

6. RIOT

The newest OS on this list, it was designed to improve real-time operation, modularity, and multithreading. It focuses on the use of Constrained Application Protocol (CoAP) and Concise Binary Object Representation (CBOR), thereby reducing memory usage and allowing simple applications to require less than 2 kB of RAM and less than 6 kB of ROM. RIOT supports the C and C++ programming languages [46].

3.5.5 Survey of Different IoT Simulators

Simulators are a very important component of any prototype development, architecture exploration and testing of threat models. IoT projects usually have a very high density of sensor and actuator nodes. Hence it is not feasible to physically deploy an entire network and test new ideas. A simulator is required which can both simulate IoT functionality and the network properties of the network.

1. Cooja

Cooja is an emulator built over the Contiki OS. Cooja emulates different functional properties of IoT things such as emulating temperature sensing, pressure sensing etc. It can also emulate nearby peer-to-peer communication among the nodes and nearby broadcast of messages based on distance. It emulates all the functionalities by building over the abstraction layer of Contiki OS. Since it emulates a real OS it can do time-based profiling of different functions.

Cooja is designed for simulating sensor networks running the Contiki sensor network operating system. It is a Java-based simulator implemented in Java but allows sensor node software to be written in C. All the interactions with the simulated nodes are performed via plugins like simulation visualizer , Timeline, and Radio logger. It stores the simulation in an Extendable Markup language (XML) file with extension 'CSC' (Cooja simulation configuration). This file contains information about the simulation environment, plugins, the nodes and its positions, random seed and radio medium etc.

2. IoTSim

IoTSim is another simulator for analyzing IoT applications [47]. It uses the cloudsim simulation toolkit, which allows the simulation of cloud computing environments.

It also simulates the map-reduce framework for the purpose of handling the big data from the IoT applications on the cloud. However, it only simulates the end-user level IoT applications and its data processing in the cloud but it doesn't simulate the IoT nodes individually and the communications between different nodes in the network.

3. OMNeT++

OMNeT++ is adiscrete event-based simulator which is written in C++ and is used for simulation of communication networks, distributed systems, and another multiprocessor applications [48]. It is a generic simulator which allows the development of various simulation models and frameworks on top of it. It has an open source 6LoWPAN integration project associated with Contiki-2.6, but it isn't straightforward integrating it as a wrapper over Contiki. Moreover, CoAP support still trails behind and with the introduction of Contiki 3.0, the code structure was changed significantly enhancing reliability and optimizations in terms of memory.

4. NS3

NS3 is a Network Simulator which is the open-source, research-oriented, community-supported simulator for testing networking protocols [49]. It has been developed over the years and it supports almost all communication protocols such as MQTT, Zigbee, Bluetooth etc. It also supports different modules which allow parallel simulation, distributed simulation etc. It can be extended easily to support and test different applications. The disadvantage of this simulator is that there is no power model for IoT and it lacks a graphic user interface (GUI) and performance studies and obtaining results would have been more cumbersome by writing a loT of C++ code.

5. OPNET

OPNET simulator is a tool to simulate the behavior and performance of any type of network. The main difference with other simulators lies in its power and versatility. This simulator makes possible working with OSI model, from layer 7 to the modification of the essential physical parameters.

OPNET is the expensive commercial network simulator which is maintained by Riverbed technologies and supports only windows platform [50].

6. NetSim

NetSim is another commercial simulator and emulator which is used to test IoT networks and applications [47]. NetSim is a discrete event simulator covering a broad range of wired, wireless, mobile and sensor networks that comes with a user-friendly GUI.

7. CORE

The CORE is the Common Open Research Emulator which is used to emulate PCs and networks on one or multiple PCs [51]. It uses FreeBSD's kernel network stack for the emulation. However, it does not support many of the protocols and network stacks used in IoT communication.

3.6 RPL: IPv6 Routing Protocol for Low Power and Lossy Networks

RPL is the IPv6 routing protocol for low-power and lossy networks (LLNs) and was designed to be suitable for resource-constrained devices in industrial, home, and urban environments [52]. RPL is the industry standard for IoT Wireless Sensor Networks.The main goal of RPL is to provide IPv6 connectivity to a large number of battery-operated embedded wireless devices that use low-power radios to communicate and deliver their data over multiple hops. From the initial design phase, RPL builds upon widely-used routing protocols and research prototypes in the WSN domain such as the collection tree protocol (CTP) [53] but is extended and re-designed to be part of, and ready for, IPv6. To compute the optimal path by building up a graph of the nodes in the network based on dynamic metrics and constraints like minimizing energy consumption or latency we can use RPL.

LLNs are a class of network in which both the routers and their interconnect are constrained. LLN routers typically operate with constraints on processing power, memory, and energy. RPL provides a mechanism whereby multipoint-to-point traffic from devices inside the LLN towards a central control point as well as point-to-multipoint traffic from the central control point to the devices inside the LLN is supported. Support for point-to-point traffic is also available.

3.7 6LowPAN

Wireless sensors are being actively used in mission-critical applications such as battlefield control, medical assistance and natural disaster forecasting. Sensors have distinguishing characteristic of limited resources (memory and processing) and autonomous but with limited power supply. ZigBee 802.15.4 protocol stack [54] is a non-IP protocol used for WSN. Zigbee has found incompatible with IP and introduces many other constraints like resource usage, limited bandwidth, energy consumption etc. Due to the urgent need of connecting WSN and the Internet, a new IETF working group namely 6LowPAN was established [55].

6LoWPAN is basically IPv6 plus low power RF [56], It has several benefits, as it is lightweight, energy efficient, scalable, has reliable standards and provides end-to-end data flow. It would allows extending the already existing IP network. Again the network that needs a full fledged TCP/IP stack along with operating system and a powerful processor is the major bottleneck for IoT things. This is because embedded things used in IoT have several challenges to overcome that include power management and duty cycles. A multicast feature of IP causes unnecessary power drain in multi-hop networks; limited bandwidth restricts the frame sizes [56]. Thus in order to make sure that the above problems are sorted to a certain extent 6LoWPAN must be used.
The stack shown in (Table 3.2) is the 6LowPAN protocol stack along with the TCP/IP protocol stack to illustrate the comparison of where exactly the 6LoWPAN resides. It can be seen clearly that instead of IP as in TCP/IP the LoWPAN resides at the network layer with IPv6. This can be shown as a LoWPAN below IPv6 but practically they are implemented as a single section. 6LoWPAN adopts the Physical Layer (PHY) and Media Access Control **(** MAC) layer protocols of IEEE 802.15.4 and along with IPv6 protocol, it is used as network layer protocol in 6LoWPAN.

Table 3.2:TCP/IP and 6lowpan protocol stack

HTTP	RTP	Application	Application
TCP/UDP		Transport	TCP/UDP
IP		Network	IPv6
Ethernet MAC		Data link	IEEE802.15.4 MAC
Ethernet PHY		Physical	IEEE 802.15.4 PHY

TCP/IP Protocol Sack 6LoWPAN Protocol Stack

3.8 QoE in Internet of Things

Multimedia communications play a major role in Internet of Things (IoT) applications in both commercial and military domains. Some examples are: real-time multimedia based security/monitory systems in smart homes, remote patients monitored with multimedia based telemedicine services in smart hospitals, environmental monitoring, traffic control and management intelligent multimedia surveillance systems deployed in smart cities, transportation management optimized using smart video cameras, remote multimedia based monitoring of an ecological system, etc.

Multimedia communications in real time IoT applications may experience network delay and congestions due to bandwidth constraints and packet loss, which have an adverse impact on the delivered multimedia quality. QoE/QoS aware Multimedia communications in IoTs (MIoT) have been proposed in recent years [57], [58] to address the above issues. However, according to Qualinet QoE is the overall acceptability of an application or service perceived by the end user, and is more user-driven [59]. Accordingly, the concept of IoT is the connection of number of devices/objects that communicate in a smart way without human involvement. For example, CCTV monitoring in real time IoT application, the quality of images (e.g., car number plate, accidents and traffic jam) should be detectable by detection software. In this context we do not need to consider QoE. So, are the current QoE models for multimedia communications in IoTs still appropriate? In general, QoE is a term to represent the end user experience for a multimedia service (e.g., video streaming or on-line video gaming). It is used for services with customers as the consumption entity. However, in IoT concepts, machine to machine (M2M) communications will be the dominant applications. With the billions of things/objects connected by the internet, the QoT concept is more appropriate than the concept of QoE because in most cases, human beings are not the ultimate target [60].

Authors in [32] have defined MIoT in 3 scenarios based on the use of multimedia content such as multimedia as IoT Input and output, multimedia as IoT input and multimedia as IoT output, and proposed a QoE layered model for MIoT applications, which has 5 layers:

1) Real-World Layer: Real world objects such as physical multimedia objects or devices acquire the multimedia content from the environment and send it to the virtual object layer through the network layer.

2) Network Layer: this layer considers traditional QoS parameters such as delay, packet loss and jitter can measure the performance of data transmission.

3) Virtualization Layer: in this layer, virtual objects are created to virtualize the functionalities of the real-world objects or devices.

4) Combination Layer: in this layer, composite virtual objects are created. It is composed of different virtual objects to provide determined service, that a single virtual object cant accomplish.

5) Application Layer: this layer considers the needs of an application and is mainly focused on QoE evaluation and control in terms of control, interactivity and presentation.

3.9 Quality of Experience for MIoT

To ensure the quality of multimedia content such as audio, video, and image to be collected, processed and delivered in MIoT applications, it is necessary to design and develop a quality aware IoT architecture. Few studies have discussed the concept of quality aware IoTs. In [40] , [32], and [61] the authors focused on QoS/QoE metrics to increase the customer satisfaction for MIoTs applications. However, M2M communications will be the dominant applications in IoT [62][63]. The basic concept of IoT is to connect the number of objects/devices in a smart way without human involvement, in such cases QoE is not applicable. Traditional QoS parameters only considered network parameters such as delay and jitter, packet loss and ignore other factors such as environmental factors and device factors. Here the concept of QoT comes into consideration for M2M communications in MIoTs. In general, QoT means Quality of Things, which is a successful operation of IoT system. It focuses on the quality of multimedia data to be collected, processed and delivered between two or more devices/objects in an IoT environment. The goal of the QoT is to meet the minimum quality that an IoT object can meet as requirement of an IoT application. By meeting the minimum requirements of an IoT object on an IoT application, the system will avoid over provisioning of multimedia quality and hence, use less bandwidth. It is also envisaged that the system will save the energy in IoT objects.

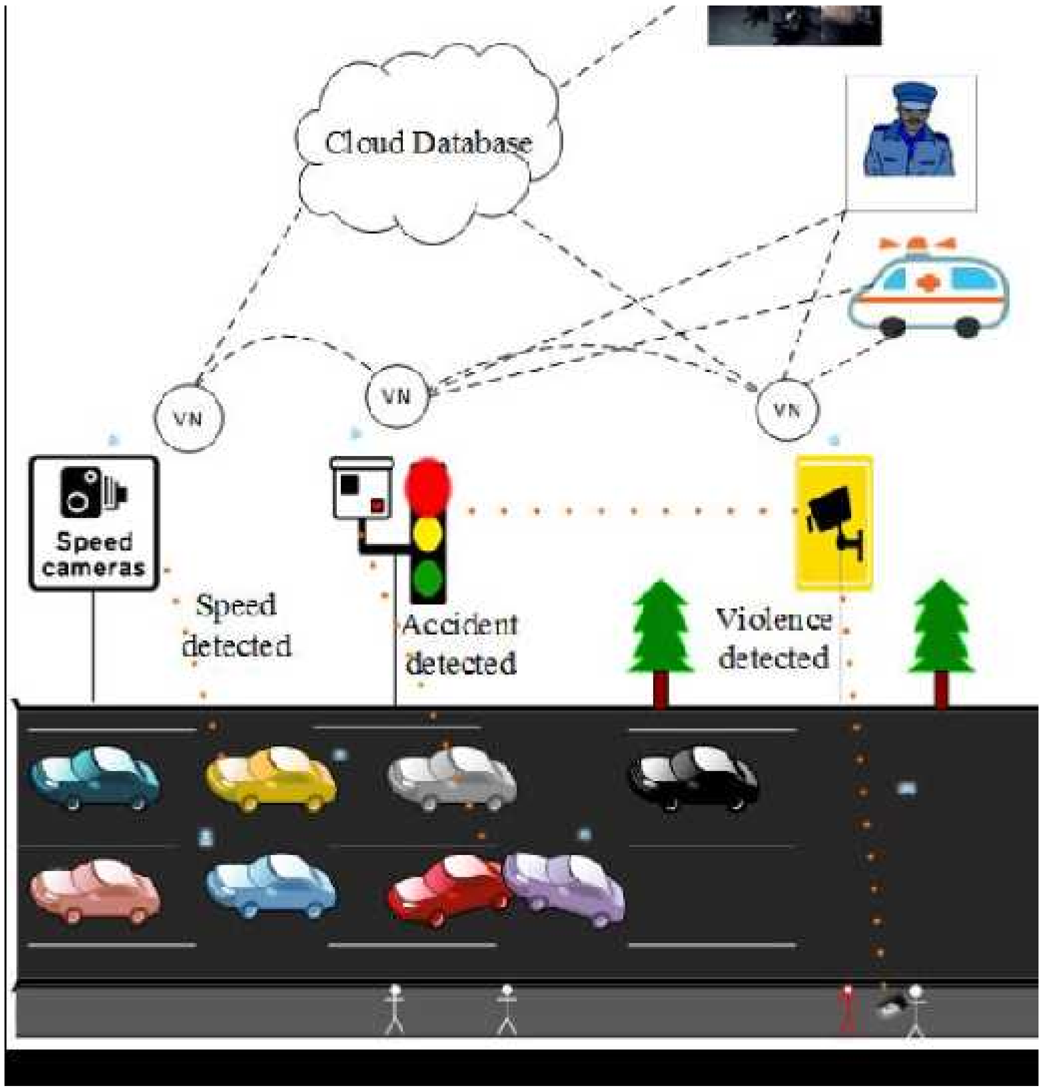

Figure 3.2: QoE in IoT

Figure 3.2 The scenario of QoE for MIoTs with several IoT system devices such as camera sensors, scalar sensors are deployed in an IoT environment to monitor real-time traffic environment [60]. Conventionally, the attacks or accidents are discovered by multimedia streaming to remote server or pre-processing video tasks at the camera nodes. However, the above two concepts would lead to various delays; processing video tasks at the camera node may lead to computational delays due to limited computational resources, and delivering video tasks to the remote cloud database may lead to network delays and traffic congestions due to limited bandwidth. Conversely, in QoE for MIoT applications the process can be done at virtual nodes to interpret the accidents or human violence and appropriate approaches can be followed such as reporting to health service department or the police department.

The scenario in Figure 3.2 is a real-time traffic monitoring system. A speed camera deployed on a road detects the car registration number and sends the information to a virtual object to check the owners' details such as name and home address for further processing. If an accident occurs in an IoT environment, the camera sensor can detect the event and send the information to virtual nodes. The virtual nodes process the information and report to a nearby police station.

If the person is critically injured it sends an alert message to ambulance centre. Similarly, a gun shot fired by an attacker can be detected by the camera sensor which can send a report to virtual nodes to process the information and report to ambulance service and police service, and at the same time the information will be sent to other traffic signals to control the traffic or alert the crowd on the street. The attacker would be identified by his/her behavior amongst the crowd. Virtual nodes send this information to the cloud to check the attackers' previous records such as criminal history. The quality of images and videos captured by the camera sensors should meet the requirement to be able to be detectable by detection software for further investigation. Here the QoT metrics play an important role to improve the connection between device to device or machine to machine. For M2H communications such as E-health monitoring and navigation systems, it takes the QoE metrics into consideration such as end-user devices, preferences, satisfaction, and background[60].

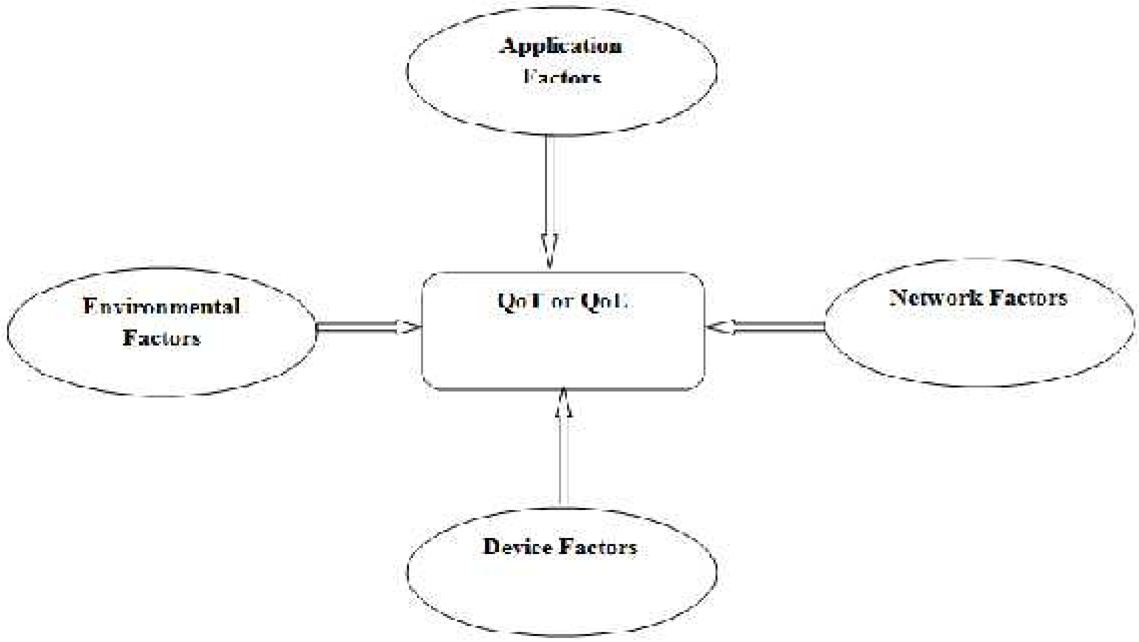

Figure 3.3The main factors of QoT for MIoTs applications

Figure3.3, describes the main QoT factors for MIoTs applications that can affect service/application performance which includes monitoring, data collection, processing and delivery. These factors are described as follows,

1) Environmental Factors: The factors that can affect the QoE requirements on environment such as physical location, temperature and accuracy of time for an IoT application. For example, the information or data captured by the multimedia devices sent to the network gateway needs to measure the accuracy of the information such as physical location, time and temperature for further processing.

2) Device Factors: Real-time multimedia monitoring in an IoT platform may lead to high energy. Thus, device factors such as device type, device transmission, battery can affect the device life time according to the current device state.

3) Network Factors: Network factors such as packet loss, and jitter have an impact on the network performance. For instance, packet loss will cause data loss and thus, affect the quality of data transmission.

4) Application Factors: Application factors such as application and codec types are focused to meet the requirement of an MIoT device. For example, codec type such as H265 could be used instead of H264 to minimize network bandwidth at the same time providing high quality video transmission.

CHAPTER IV

4. Research methodology

In order to accomplish the goals mentioned in chapter 1, this chapter starts by describing approaches to methods of data collection conducted in this study and comparing advantages and disadvantages of the uncontrolled (public) and controlled environment. The sources of data and the reasons for sample population are explained. The methods of data collection and analysis of the data are also outlined. It also justifies the selected research methodology including reliability and validity issues.

4.1 Approached Methods

The subjective test can be conducted in two kinds of environment: controlled environment and uncontrolled environment. In this experiment the participants evaluated the experience of video service in an uncontrolled environment where they had the freedom of completing the evaluation at home or in their office. The uncontrolled environment is more close to the users' real viewing experience. However the controlled environment is often used in subjective tests, which is not the usual place where the common viewers watch video. The results may not be an accurate reflection of viewers' true viewing experience in the wild, where other factors, such as delay, may also have an influence on QoE. Table 4.1 shows the requirements of both environments specified by the International Telecommunication Union Recommendation ITU-R BT. 500-11 [13].

Table 4.1:Test environment requirements

Requirement	Laboratory	Home
Ratio of inactive screen luminance to peak luminance	≤ 0.02	≤ 0.02
Ratio of background luminance to picture's peak luminance	≈ 0.15	N/A
Peak luminance	N/A	$200cd/m^2$
Ratio of screen only black level luminance to peak white luminance	≈ 0.01	N/A
Display brightness and contrast	PLUGE [58], [59]	PLUGE
Background chromaticity	D_{65}	N/A
Environmental illuminance on the screen	N/A	$200lux$
Other room illumination	Low	N/A
Maximum observation angle relative to the normal	$30°$	$30°$
Screen size	N/A	Meet rules of PVD
Monitor	N/A	No digital processing; Meet resolution requirement

In this experiment instant-Contiki was used to transmit the video sequences over IoT devices with varying bitrate , frame rates, and content type and displaying power consumption for each mote.

4.2 Source of Data

In this research the data collected from two sources as we follows:

1. Data were collected from the Cooja simulator which is the fraction of the time that a node remains in a particular power state. The power consumption is calculated by using the following formula [64].

$$Power\ consumption = (Energest_{value} * pre - measure * voltage)/RTIMER_{second}/Runtime \quad (5)$$

Energest_Value is the difference between the number of ticks in two time intervals (for example difference between TX in time interval 20 and 10). Pre-measure from data sheet The voltage for EXP430F5438 mote is 3V. RTIMER_SECOND is 4096.Runtime is the time interval in which we perform measurements (2 in our example).

2. Data were collected from participants of the subjective tests through questionnaires after viewing each videos sequence. All participants were allowed to view the videos as many times as they wished evaluation the decision, but they had to see them all at least once.

4.3 Sample Population

In the experiment, a total of 34 people took part in the study on a volunteering basis and no participants were working in video quality assessment and no money was paid to them.

Judgmental sampling [65] was used to select the sample of this research. In this sampling, we select participants who are presumed to be representing the population in qualitative research.

In the experiment also , a total of 29 people took part in the study on a volunteering basis and no participants were working in video quality assessment and no money was paid to them.

4.4 Participants Tasks

Participants were given the following tasks that should be executed in order:

1. Participate in a training session of no longer than 5 minutes.
2. Watch video played back on the laptop.
3. Grade the video quality in the range of 1 (bad) to 5 (Excellent).
4. Have a break from watching a video before watching the next video

4.5 Experimental Setting

The experiment setting for recording power consumption in IoT devices instant-Contiki 7 is used and two EXP430F5438 motes are selected, one for sending the trace information and the other mote is the receiver which is receives the trace information .

Two subjective tests were conducted in Gadarif, Sudan. The experimental design for the subjective video tests is based on ITU recommendations [13],[66]. In the first subjective test the video sequences were presented to 34 different people that included 17 males and 17 females. And for the second subjective test the video sequences were presented to 29 different people .All video sequences were displayed in a laptop and participants were asked to hold it and put it anywhere because in daily life the participants are not constrained when watching video content on their laptop. The laptop had calibrated Pentium(R) 3 CPU 1.90GHz, memory 8GB and 14-inch monitor to show the video sequence. Participants were supervised by the experimenter. The experimenter remained silent and unobtrusive during the course of subjective tests.

4.6 Test Procedures

The following procedures were followed to get the power consumption:

1. Decode original videos into h264 with varying frame rates, bitrates, and content types.
2. Generate the videos trace files by using Evalvid.
3. Transmit video trace files via Cooja simulator.
4. Record power consumption for sender motes.

The following procedures were followed in the subjective tests:

1. On arrival, participants were briefed about the aims, objectives and the importance of the tests in which they were about to participate.
2. Participants watched video sequences at least once.
3. Once each video sequence has ended, participants fill in the MOS value scores in the range of 1 (bad) to 5 (excellent). The voting was conducted by using paper ballots.

Participants continue to watch video sequences in a laptop until all sequences are completed.

4.7 Analysis of Data

After the tests were completed, all the participants' data are collected and the data are input into an Excel sheet for averaging to compute MOS for each video sequence.

4.7.1 Removing Participant Bias

To remove participant bias, the technique introduced by [67] was used with the following steps.

1. Estimate MOS for each video sequence,

$$MOS_j = \frac{1}{N_j} \sum_{i=1}^{N_j} p_{ij} \qquad \textbf{(6)}$$

where p_{ij} denotes observer rating for participant i of video sequence j and N_j is the total number of subjects that rated video sequence j.

2. Estimate participant bias,

$$MOS_{\Delta_i} = \sum_{j=1}^{J_i} (p_{ij} - MOS_j) \qquad \textbf{(7)}$$

where MOSδi estimates the overall shift between the i participant's scores and the true values (opinion bias) and Ji is the number of video sequences rated by participant i.

3. Compute the normalized ratings by removing participant bias from each rating,

$$\alpha_{ij} = p_i j - MOS_{\delta_i} \qquad (8)$$

where α_{ij} is the normalized rating for participant i and video sequence j. MOS is then calculated normally. This normalization does not impact MOS,

$$MOS_j = \frac{1}{N_j}\sum_{i=1}^{N_j} \alpha_{ij} \qquad (9)$$

4.7.2 Subjective Test Performance Verification

To verify the performance of the first subjective test, 3 videos with video codec H264 were employed in the experiment with a variety of content (arranged in random order according to ITU-T standard [68] .The duration of each video was less than one minute with frame rate ranging from 25-30fps and a resolution of 176 x 144. The age of the participants varied from 18 to 56 years old. Participants had no knowledge of video signaling and processing. They were 50% men and 50% women; all had normal vision and clear understanding of the test.

To verify the performance of the second subjective test, 3 videos with video codec H264 were employed in the experiment with a variety of content (arranged in random order according to ITU-T standard [68] .The duration of each video was less than one minute with frame rate ranging from 25-30fps and a resolution of 176 x 144. The age of the participants varied from 18 to 40 years old. Participants had no knowledge of video signaling and processing. All participants had normal vision and clear understanding of the test.

4.8 Validation of Results

Analysis of Variance (ANOVA) is a statistical method that is used to test the differences of at least more than two means. The t-test (Independent Samples) has been used in this study to compare two groups of data, especially in the comparison of MOS values amongst a group of participants [69], [70]. In this research, values that are at most 0.05 would mean that there is the statistically significant difference in the values under the experiments.

4.9 Data Analysis Tools

SPSS is used for statistical data analysis and all the video quality assessment data were analyzed by using SPSS. SPSS help to organize large amounts of data, and provides statistical results from descriptive statistical analysis to complicate multi-factors statistical analysis.

4.10 Video Coding

To verify the performance of the subjective test for the gender, FFMPEG was used to encode YUV videos to H.264/AVC with different bitrates and frame rates. Frame rate measures how many still images appear on the screen over the span of one second, indicating how smooth the video looks. bitrate describes how much data the video contains, measured in megabits per second. The bitrate depends partially on video resolution not on frame rate , because higher resolution video files contain more information. All experiments to handle multimedia data under FFmpeg were conducted in Ubuntu 16.04.1 LTS trusty Intel(R) Pentium(R) 3 CPU 1.90GHz, memory 8GB, and 32 bits OS.

To verify the performance of the subjective test for the non network parameter, FFMPEG was used to encode YUV videos to H.264/AVC with different bitrates and frame rates. All experiments to handle multimedia data under FFmpeg were conducted in Ubuntu 16.04.1 LTS trusty Intel(R) Pentium(R) 3 CPU 1.90GHz, memory 8GB, and 32 bits OS.

4.11 Videos Sequences

To verify the performance of the subjective test for the gender three videos were used, FFmpeg was used to encode YUV videos to H.264 the frame rate ranging from 25-30fps and a resolution of 176 x 144. All videos were less than ten seconed long.
To transmit the video sequences over IoT devices with varying bitrate, frame rates, and content type, three videos were used, x264 was used to encode YUV videos to H.264/AVC at different bit rates and frame rates as shown in table 4.2.

To verify the performance of the subjective test for the non-network parameter three videos were used, x264 was used to encode YUV videos to H.264 at different bitrate , frame rate and fixed resolution of as shown in table 4.2. All videos were less than ten second long.

Table 4.2 : Videos frame rate and bitrate for the three videos

Video sequence	Bitrate(kbps)	resolution	Frame rate(fps)	
Bus,crew and suzie	30	176x144	25	30
	40	176x144	25	30
	50	176x144	25	30
	60	176x144	25	30
	70	176x144	25	30
	80	176x144	25	30
	90	176x144	25	30
	100	176x144	25	30

CHAPTER V

5. Experiment Setup

The aim of this chapter is to provide through details about the experiment setup for video transmission and the QoE subjective test.

5.1 Generation and Transmission of Video Trace Files

This section explains how to generate the video trace files and transmit them over Cooja simulator.

5.1.1 Generating the Video Trace Files

Video trace files are generated by the following step:

1. Video Source

Three original videos(raw videos) which are bus, crew and Suzie in quarter common intermediate formate(QCIF) format from, e.g., https://media.xiph.org/video/derf/ and each videos has different content type .

2. Codec

X264 is used to decode the original videos into H.264 with varying frame rates and bitrates.

3. MP4-Container

The mp4 container is used in the experiment to create ISO MP4 files containing the video samples (frames) and a hint track which describes how to packetize the frames for the transport with Real time protocol (RTP).

4. EvalVid

Evalvid is used in the experiment which is a framework and tool-set for evaluation of the quality of video transmitted over a real or simulated communication network.
After creating the hinted mp4 files the mp4trace tool from EvalVid is able to send a hinted mp4-file per RTP/UDP to a specified destination host.the command:
mp4trace -f -s 192.168.0.2 12346 a01.mp4
sends the H.264 track of a01.mp4 to the UDP port 12346 of host 192.168.0.2

5. Trace Files

When mp4trace finishes the transmission of the video, the corresponding video trace files is generating. The relevant data contained in the video trace file is the frame number, the frame type and size and the number of segments in case of (optional) frame segmentation. Table (5.1)shows the video trace file format the time in the last column is only informative when transmitting the video over UDP so that it can be seen during transmission if all runs as expected (The time should reflect the frame rate of the video, e.g. 40 ms at 25 Hz).

Table 5.1:Format of video trace file

Frame Number	Frame Type	Frame size	Number of UDP-packets	Sender Time
1	I	180	1	0.5500 ms
2	B	90	1	0.3330 ms
3	B	77	1	0.2213 ms
4	I	500	1	.5901ms
5	B	1300	2	.6223mS
6	B	1020	1	.5761ms

5.1.2 Transmission of Video over Cooja Simulated Network

In our scenario, We used Instant Contiki 2.7 development environment to transmit the video trace file. It contains all the tools and compilers needed for Contiki development and debugging. VMWare Workstation 10 Virtual Machine Player is used to load and run

Instant Contiki image. In the Home directory, we can see two folders: Contiki and Contiki 2.7 that contains all tools, libraries, platforms, documentation, or examples.

With Contiki build system we can easily compile Contiki applications for different hardware platforms by simply supplying different parameters to the make command, without editing Makefile or application code. The makefile contains the instructions for the compilation tools.

In the following the simulation setup is clarify by explaining the initial setting, creating , running and saving simulation and find the power consumption.

5.1.2.1 Start-up and Initial Settings

In "Contiki 2.7/tools" directory there is tools such as power trace (used to estimate the power consumption in our experiment), release tools; collect view, Cygwin, turn slip and Cooja.

To start the simulation software open a terminal window is open and the following commands are entered:

cd Contiki/tools/cooja

ant run

In the event that large, memory heavy simulations are to be run, the command **'ant run_bigmem'** can alternatively be used. The Cooja software should start, resulting in the screen as shown in Figure 5.1.This chapter demonstrates how to set up a simple UDP-rpl network utilizing unicast sender mote and unicast receiver motes, and it is possible to observe radio communication, messages output from motes within the network and to alter the transmission and interference range of motes.

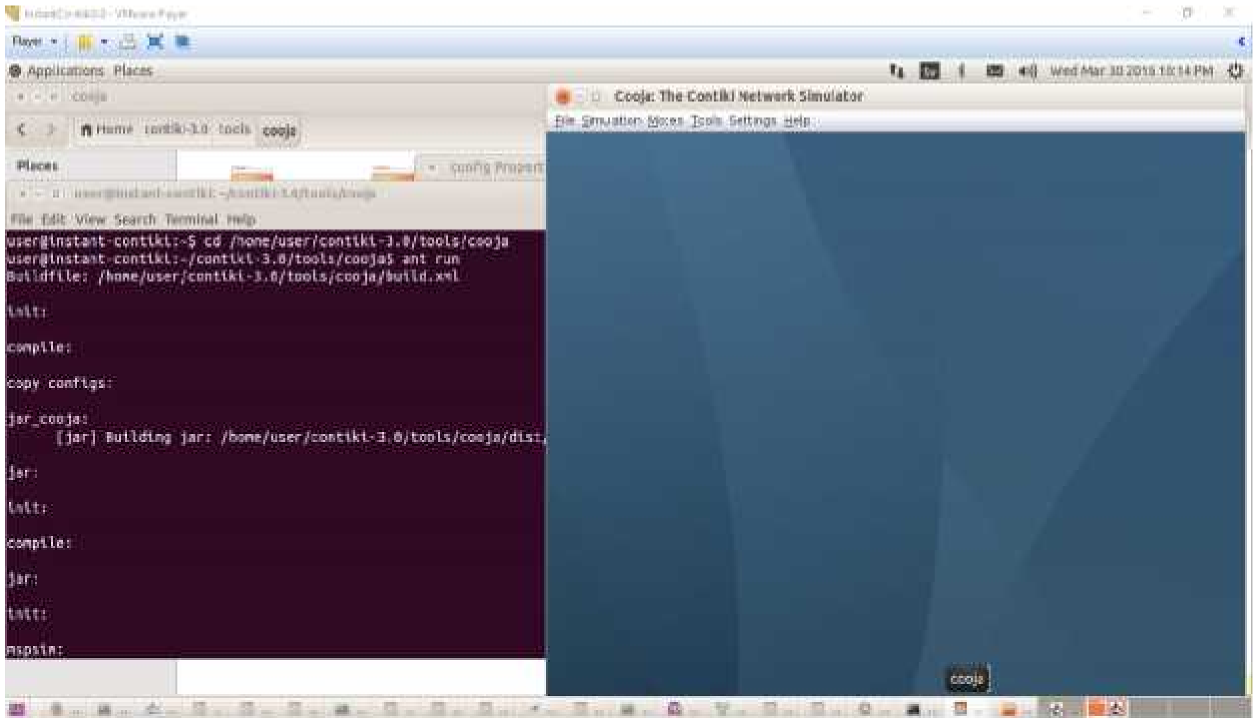

Figure 5.1:Initial Cooja Screen

5.1.2.2 Creating a new Simulation

To create a new simulation for the experiment the following main menu command is used:

File->New Simulation

A “Create new simulation” window is displayed. The name of the simulation, and the advanced settings such as Radio Medium, Mote startup delay and Random seed are entered. After creating the simulation, Cooja displays all simulation tools as can be seen in (Figure5.2) such as: Network (show all nodes in the network), Simulation control (panel used to Start, Pause, Reload or Executing Steps of the simulation), Mote output (Show output for the nodes), Timeline (Simulation Timeline that shows channel change, LED change, log outputs) and Notes (Notes about simulation) .

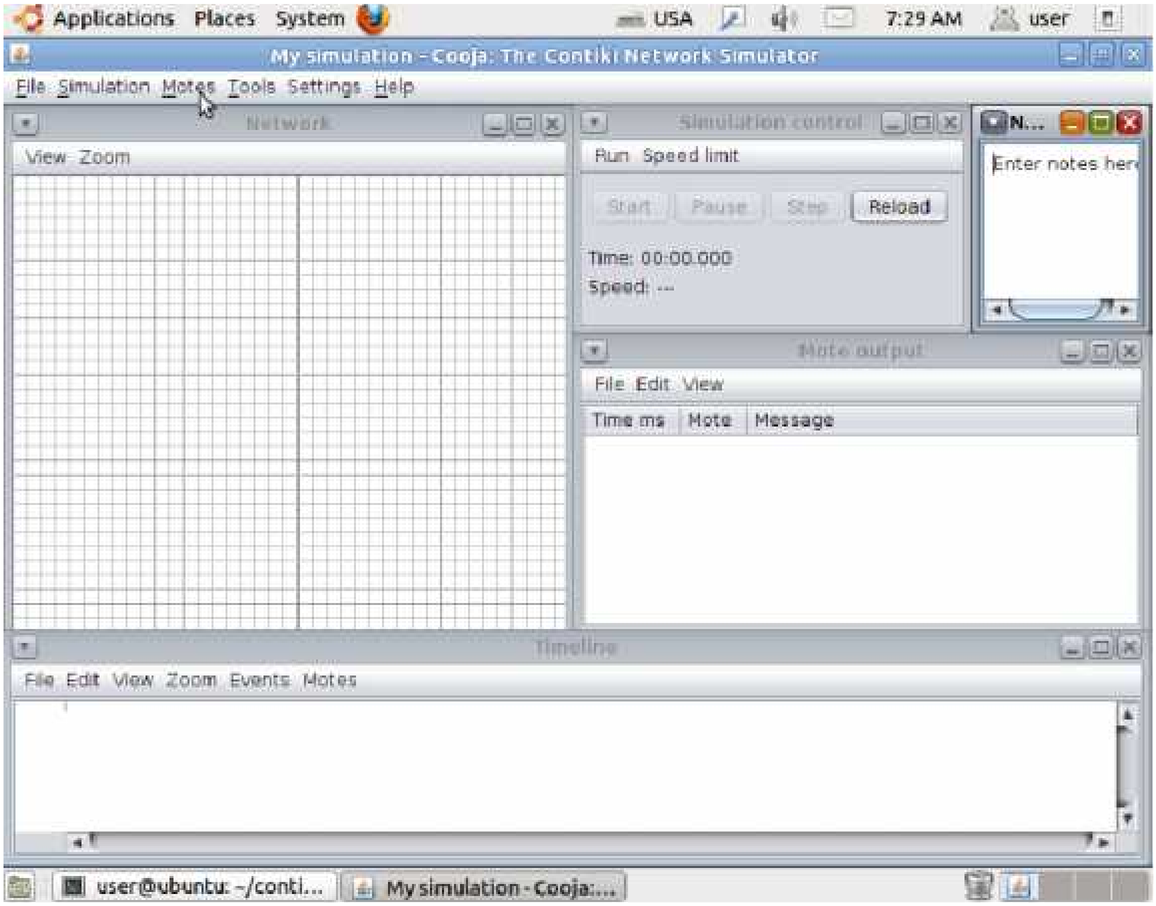

Figure 5.2:Cooja Create new simulation

5.1.2.3 Adding Motes and Running the Simulation

To add motes in the simulation the following main menu command is used:

Motes->Add motes->Create new motes type.

After that, the type of the mote EXP4530F5438 motes is chosen.

The “Create mote type: Compile Contiki” window opens showing a Description (Mote type) and the Contiki process/Firmware. With the click of the button, we have to choose the Contiki application. Then, in the tab Compile commands is shown in the following code:

make unicast_sender.EXP4530F5438 TARGET= EXP4530F5438

The EXP4530F5438 is a target platform or a Mote type name. After clicking on “Compile”, a Contiki program is compiled and the output is shown in the tab

"Compilation output" (Figure5.3). It displays all platform compiled parts and the errors in the application (if found). Then, "Create"is clicked to add motes in the simulation and the "Add Motes" window is displayed. The number of new motes, Positioning and Position interval are defined then the mote is added and the simulation is created. The motes can be seen in the Network window. In this experiment, two motes are used one for the sender and the others for the receiver as shown in figure 5.3

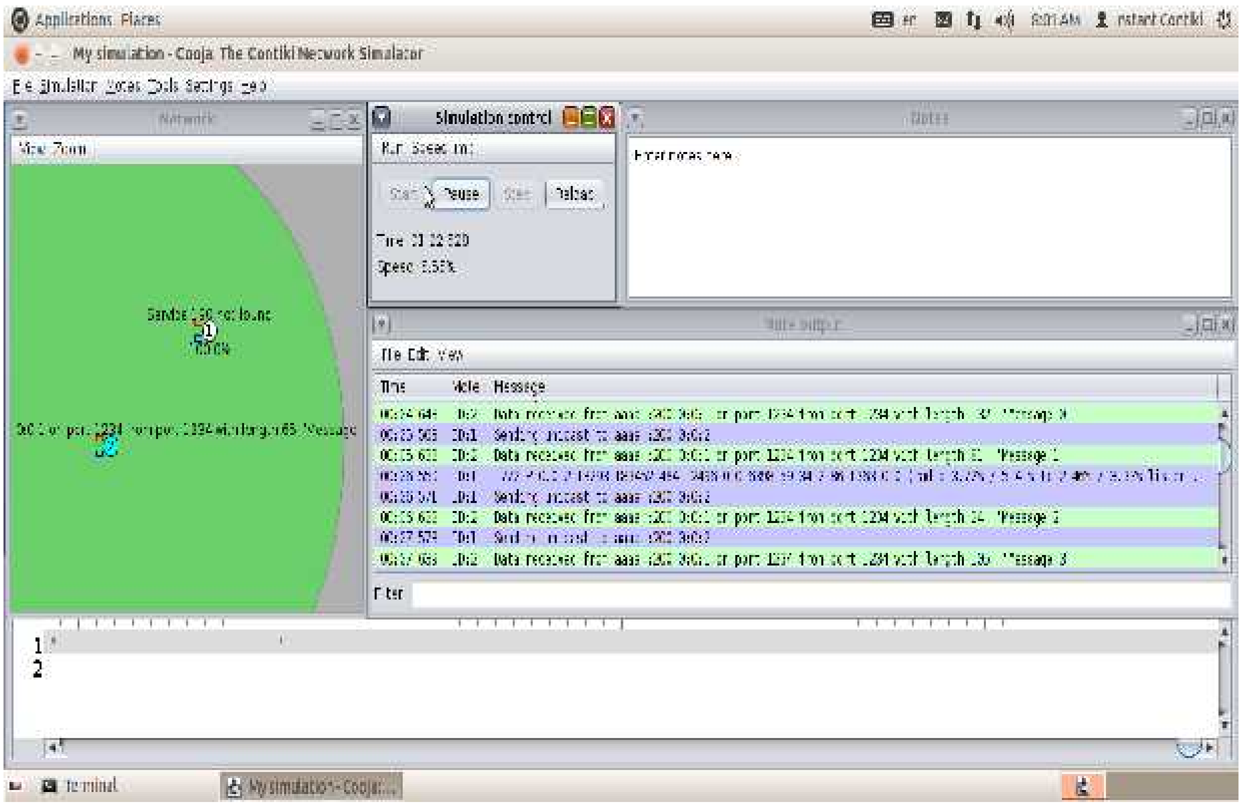

Figure 5.3:Unicast sender and receiver motes in Cooja

5.1.2.4 Network Options

Following is description of the various windows and their purpose, as well as the many other display options available before actually running the network test:

- The **Timeline** window at the bottom of Figure 5.4 shows events over a period of time such as layer 2 communication to motes to waken them up . This view can be filtered using the drop-down menu and the output can be saved to a file if necessary.

- The **Mote Output** window shows the output window that will display any printouts from the motes. This can be extremely useful in more complex networks which require more fine-grained analysis of results, as the actual source code of the motes can be altered at various levels, with messages produced which can be observed in this window. This is also where any print messages used to ascertain the flow of code will appear.

- The **Simulation Control** window is where the simulation is started, paused and stopped. The simulation can also be completely reloaded from this window although there are other options in this regard.

- The **Network** window displays the layout of the network motes and can be greatly modified in order to more easily display various factors of the network, as well as network traffic. The **View** drop-down menu shows the various options as shown in Figure5.4.

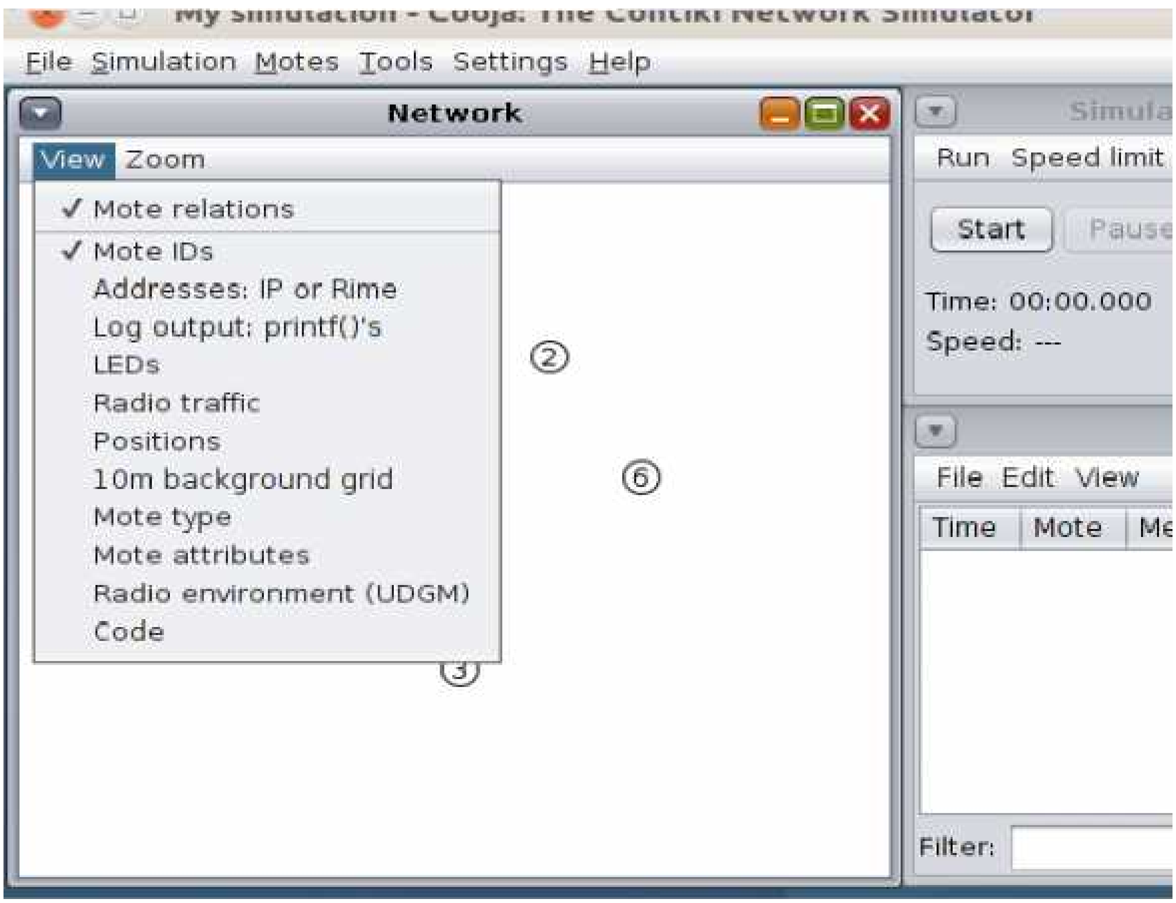

Figure 5.4:Cooja View Dropdown

should be noted that these options are best utilized selectively, as the window can become cluttered with too many options selected. As can be observed, **'Mote relations'** and '**Mote IDs'** are already ticked, showing the numbering for each mote. The rest of the options are detailed as follows:

- The **'Addresses: IP or Rime'** option displays the addresses, in this case, IPv6, of each mote. Given the length of an IPv6 address, it can be seen how this option should be used sparingly as in a fairly dense network the addresses simply will not be viewable.

- **'Log output: printf()'s'** displays printf messages from the mote code inside the actual view window. These messages also appear within the **Mote Output** window which could be argued as a better way of analyzing and filtering results.

- **'LEDs'** is used in order to observe the LED lights on the simulated motes.

- The **'Radio Traffic'** option is extremely useful in animating the network once the simulation is running. Using this option will display the exchange of messages between nodes allowing the observation of the selection of parents and how the Direction-Oriented Directed Acyclic Graph (DODAG) is built.

- **'Positions'** displays the relative location of each node.

- The **'10m background grid'** is extremely useful to give a sense of scale to the network layout. This displays a grid of 100m2 squares rather than a blank background.

- **'Mote type'** employs a color scheme to show the difference between motes of different types. In the case of this demonstration network, one types of motes are employed – one for senders and one for receiver.

- **'Mote attributes'** allows the use of code to alter the colorizing of motes as well as other options and is not relevant to this discussion.

- **'Radio environment'** displays the transmission range of any particular mote and is extremely useful when deciding upon the optimal position of motes within a simulated network.

- **‘Code’** merely displays code as it is being run and is not of relevance to this discussion.

• In the case of this demonstration the following options are used:

  - **Radio Traffic**
  - The **10m background grid**
  - **Mote type**
  - **Radio environment (UDGM)**

The motes have been reorganized into a tree formation and the Zoom option utilized in order to place them at a more realistic distance. In order to change the transmission range and interference range of the motes is right-clicked and the ‘Change transmission ranges’ is selected. Once the range has been changed the simulation must be reloaded.

5.1.2.5 Saving Simulation

Cooja simulation can be saved using the command:

File-> Save simulation as

The saved file has the extension “.csc”, and can later be loaded using that file.

In our research, we used the simple unicast sender and unicast receiver example after making some change in the code to transmit the video trace file We run the application on one type of wireless sensor network motes EXP430F5438 mote which is equipped with 8MHz Texas Instruments MSP430 low power microcontroller, 10 KB RAM and 48 KB flash [71].

5.1.2.6 Power Consumption

To estimate power consumption in Cooja for unicast sender examples a tool called power trace is used . this tool is added to the program Makefile using this line of code:

APPS += powertrace

In the code of the unicast sender.c file the following lines of code is added to print power consumptions in different states:

```
#include "power trace.h"
powertrace_start(CLOCK_SECOND * 2);
```

The value 2 is a Runtime. This means that the values for power consumptions will be printed every 2 seconds.

5.2 Experiment Setup for the Subjective Test

In our research two subjective test were conducted one for the context parameter such as gender and the other for the non-network parameter such as power.

5.2.1 The Experiment Setup for the First Subjective Test

In our experiment, a total of 34 people took part in the study on a volunteering basis. Participants have been chosen with respect to their gender (17 male and 17 female). Also, no participants were working in video quality assessment.

To verify the performance of the subjective test, 3 videos with video codec H.264 were employed in the experiment with a variety of content and quality as shown in the screenshots of videos (figure 5.5). The duration of each video was less than one minute and the videos parameters is shown in table (5.2). The laptop monitor that was used for display in our test was a 14-inch monitor. All participants were allowed to view the videos as many times as they wished before assessing videos, but they had to see them all at least once. Participants evaluated the experience of video service in an uncontrolled environment where they had the freedom of completing the evaluation at home or in their office. They were also asked if they like the video content.

Figure 5.5:From left, Akiyo, Bowing and Car phone videos

Table 5.2: Video parameter for H.264

Video sequence	akiyo,bowing,car phone
Video codec	H.264
Resolution	176x144
Encoder	Ffmpeg
Aspecr ratio	4:3
Bitrate(kbps)	9116,500kbps
Framerate (fps)	25,30
Size(yuv)	Qcif

5.2.2 The Experiment Setup for the Second Subjective Test

In our experiment, a total of 29 people took part in the study on a volunteering basis. no participants were working in video quality assessment.

To verify the performance of the subjective test, 3 videos with video codec H.264 were employed in the experiment with a variety of content as shown in the screenshots of videos figure 5.6. The duration of each video was less than ten second and the videos parameter is shown in table 3. The laptop monitor that was used for display in our test was a 14-inch monitor. All participants were allowed to view the videos as many times as they wished before assessing videos, but they had to see them all at least once. Participants evaluated the experience of video service in an uncontrolled environment where they had the freedom of completing the evaluation at home or in their office. They were also asked if they like the video content.

Figure 5.6: From left, bus, crew and suzie videos

Table 5.3: Video parameter for H.264

Video sequence	Bus,crew,suzie
Video codec	H.264
Resolution	176x144
Encoder	x264
Aspecr ratio	4:3
Bitrate(kbps)	30,40,50,60,70,80,90,100
Framerate (fps)	25,30
Size(yuv)	Qcif

CHAPTER VI

6. Result and Discussion

This chapter illustrates the results obtained by subjective QoE tests and proposed power model based on regression technique and the model evaluation .we demonstrate the impact bitrates, frame rate and video content on power consumption of IoT devices .

6.1 Impact of Bitrate and Frame rate on Power Consumption for IoT Devices

In this section the effects of the video bitrates and video frame rate and video content on power consumption of IoT devices are presented. In each test , one parameter was varied while keeping others fixed. Power (mW) is used as a metric to measure power consumption of IoT devices.

6.1.1 The Impact of Bitrates on Power Consumption

To evaluate the impact of bitrates on power consumption of IoT devices. We first use a video with fast movement in it is content (crew). The power consumption is evaluated for the central processing unit (CPU) which is the high power CPU time (CPU in active mode in this experiment is 12 second). Transmission (TX) is the total number of ticks in TX state (Transmit less than 5 second), and RX is the number of ticks in the RX state (Receive), and when the video is transmitted from sender to receiver after getting the power consumption from Cooja which is the fraction of time that a node remains in a particular power state we converted to mW by using equation (5). The frame rate and resolution were kept fixed at 25fps and 174x144 simultaneously and bitrates was varied from 30-100 kbps. According to the results that we obtained from the Cooja simulator shown in Figure 6.1, it was the videos with higher bitrates consume more power than the low bitrate video.

.

We repeat the same evaluation when fixing the frame rate at 30 fps while using the same values for the other parameters. In figure 6.2 we also observed that the videos with higher bitrates consume more power compared to videos with low bitrates.

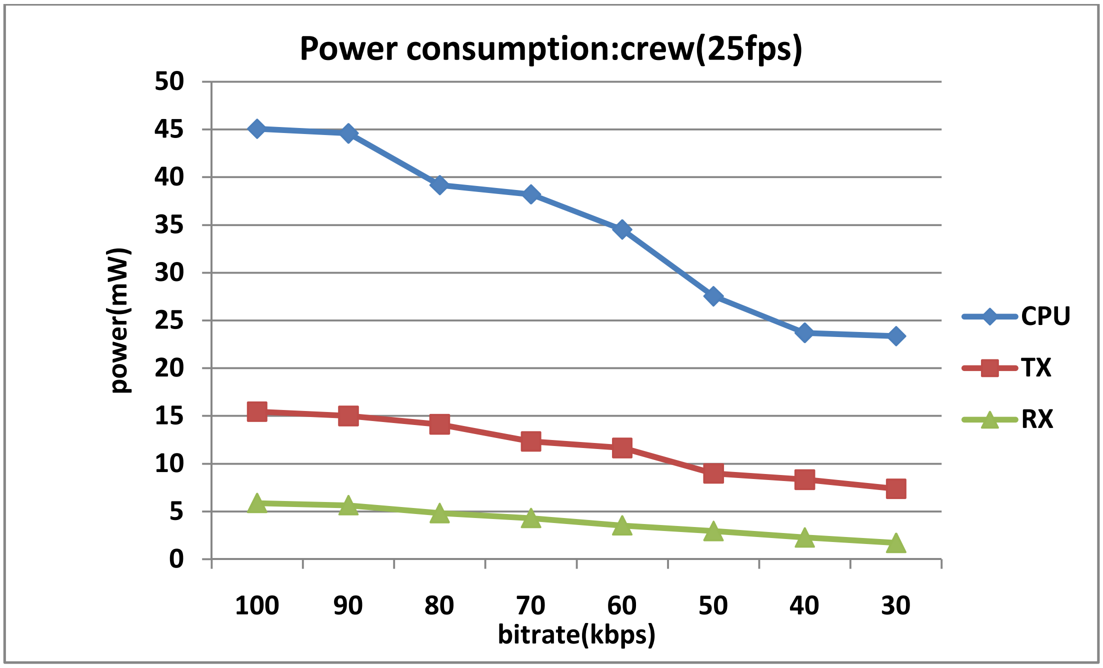

Figure 6.1:Power(mW)versus bitrate for crew sequence(25fps)

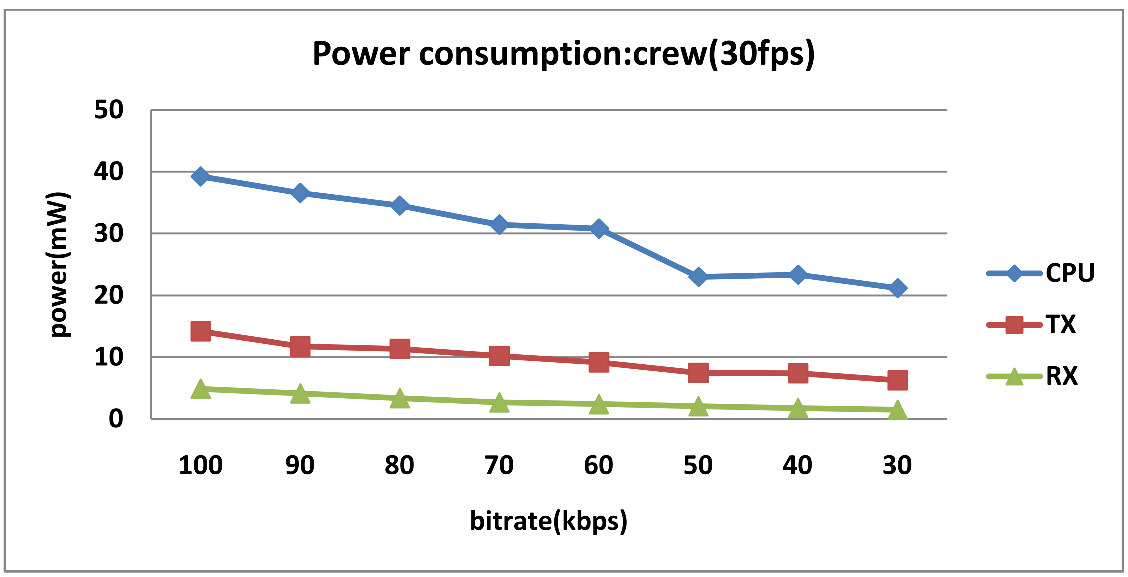

Figure 6.2: Power(mW)versus bitrate for crew sequence(30fps)

The same evaluation was carried out on a video with medium movement in it is content (bus).the resolution was fixed at 174x144 and the bitrate was varied from 30-100 kbps. In figures 6.3 and figure 6.4 we show the power consumption of IoT devices when the frame rate was 25fps and 30 fps , respectively . As can be seen in figure 6.3 and 6.4, and similar to previous results, videos with higher bitrates consume more power than video with lower bitrates.

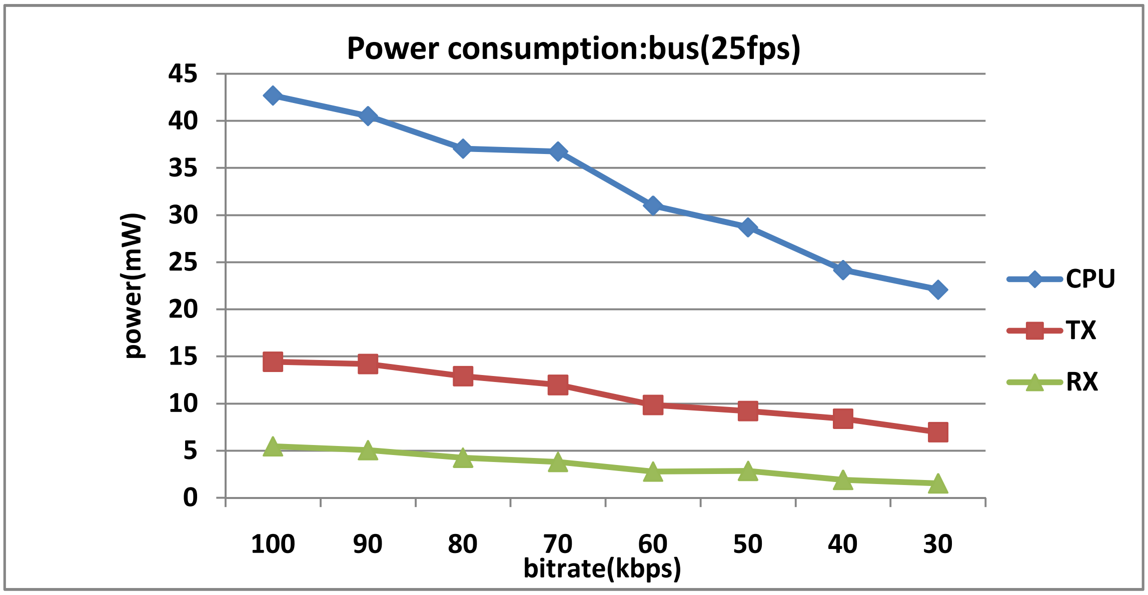

Figure **6.3**: Power (mW)versus bitrate for bus sequence(25fps)

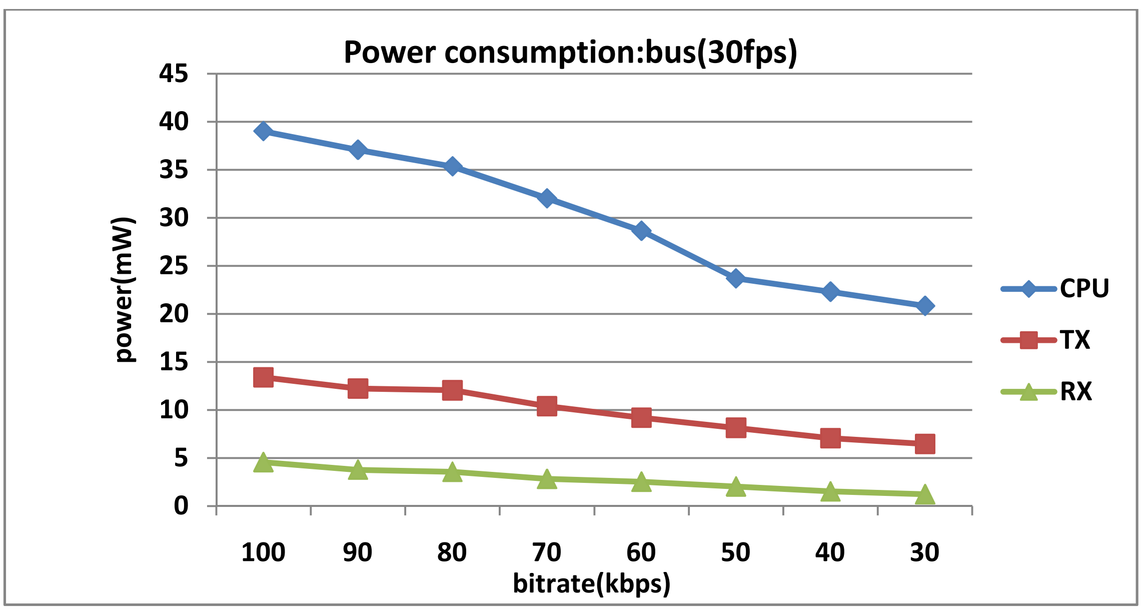

Figure 6.4: Power (mW)versus bitrate for bus sequence(30fps)

The same evaluation was carried out on a video with slow movement in it is content (suzie).the resolution was fixed at 174x144 and the bitrate was varied from 30-100 kbps. In figures 6.3 and figure 6.4 we show the power consumption of IoT devices when the frame rate was 25fps and 30 fps , respectively . As can be seen in figure 6.5 and figure 6.6 , and similar to previous results, videos with higher bitrates consume more power than video with lower bitrates.

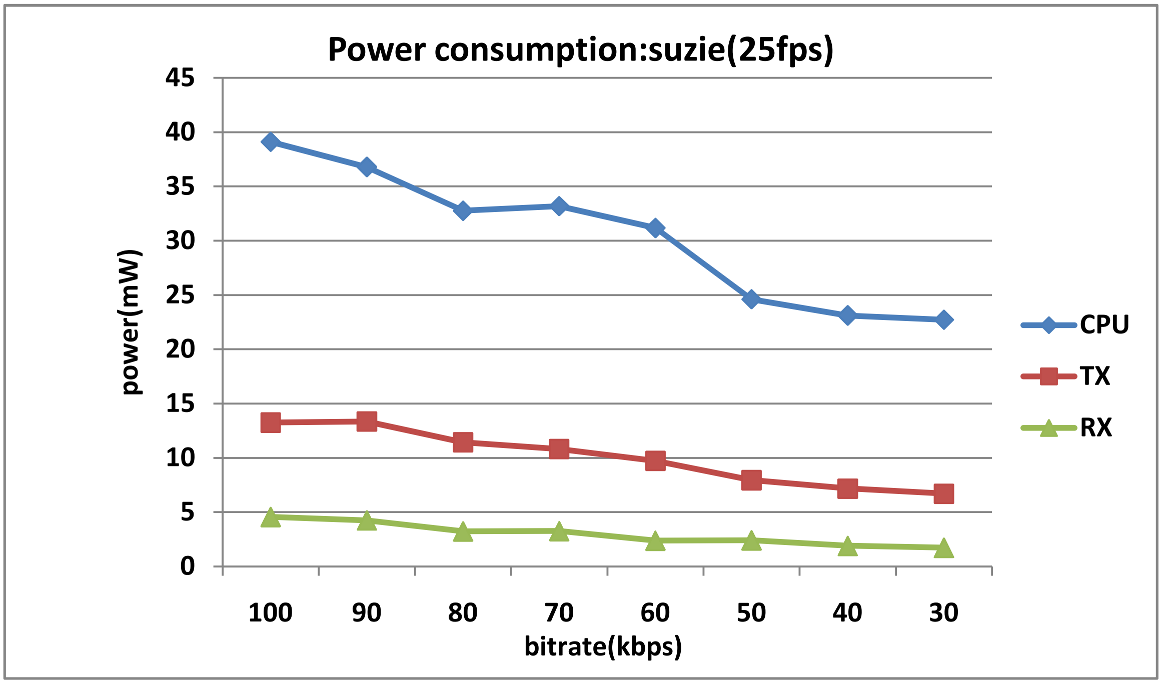

Figure 6.5:Power (mW)versus bitrates for suzie sequence(25fps)

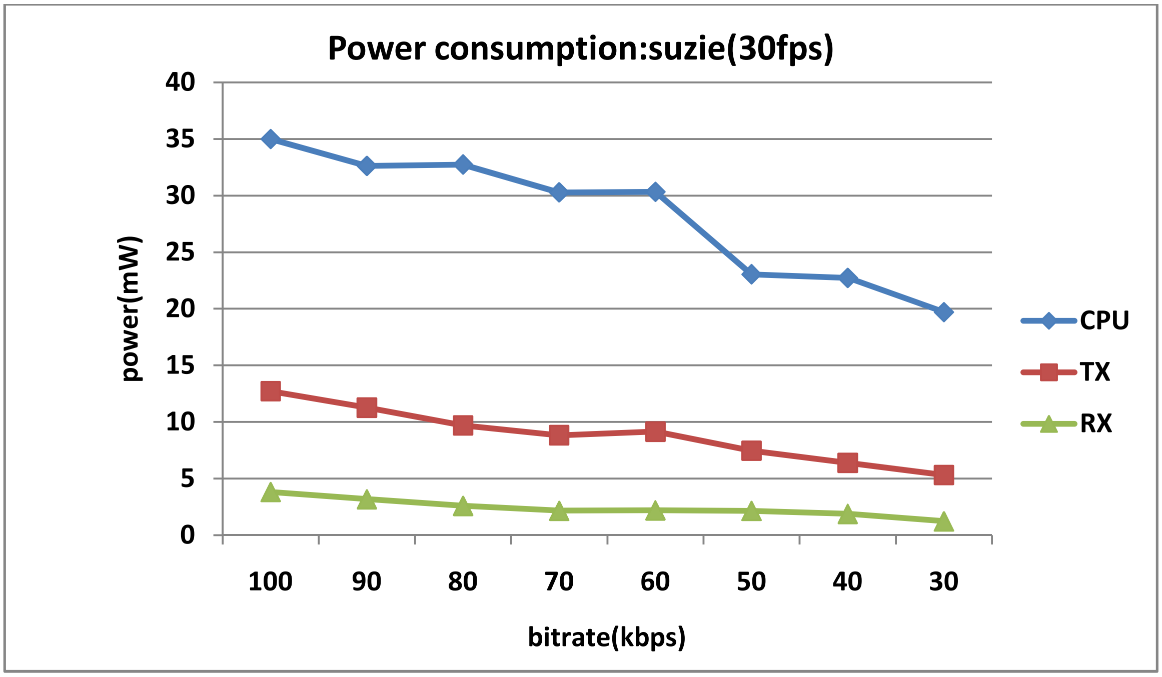

Figure 6.6:Power (mW)versus bitrate for suzie sequence(30fps)

6.1.2 The Impact of Frame Rate on Power Consumption

Figure 6.5 shows the power consumption in mW for the CPU,TX,RX when the fast movement (crew) video is transmitted from sender to the receiver, the resolution was kept fixed at 174x144 ,frame rate is 25 and 30 fps and bitrates was varied from 30-100 kbps. According to the results that we obtained from the Cooja simulator in Figure 6.7, it was observed that the video with higher bitrates with low frame rates consume more power than the video with low bitrates and high frame rates. This is because, more data is processed and sent in high bitrate videos than in slow and medium video sequences [72].

The same evaluation is carried out using the medium movement(bus) and slow movement (suzie) video the same results are achieved and shown in figure 6.8 and 6.9

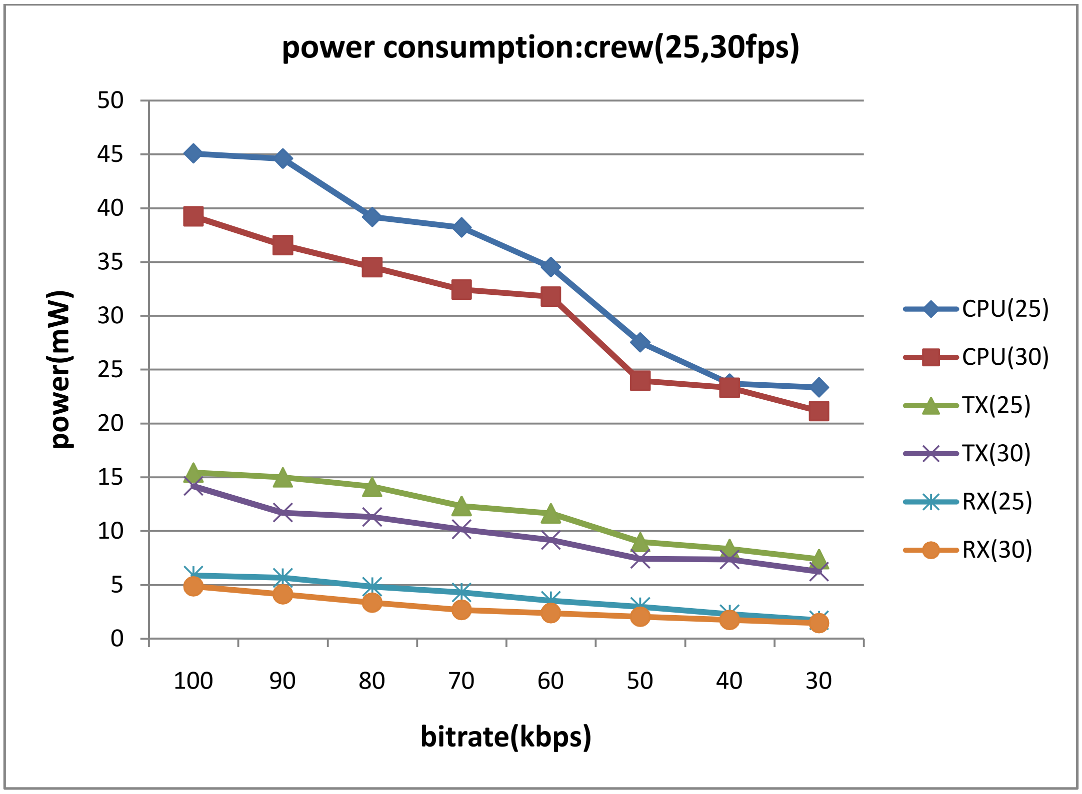

Figure 6.7:Power(mW)versus frame rate for crew sequence

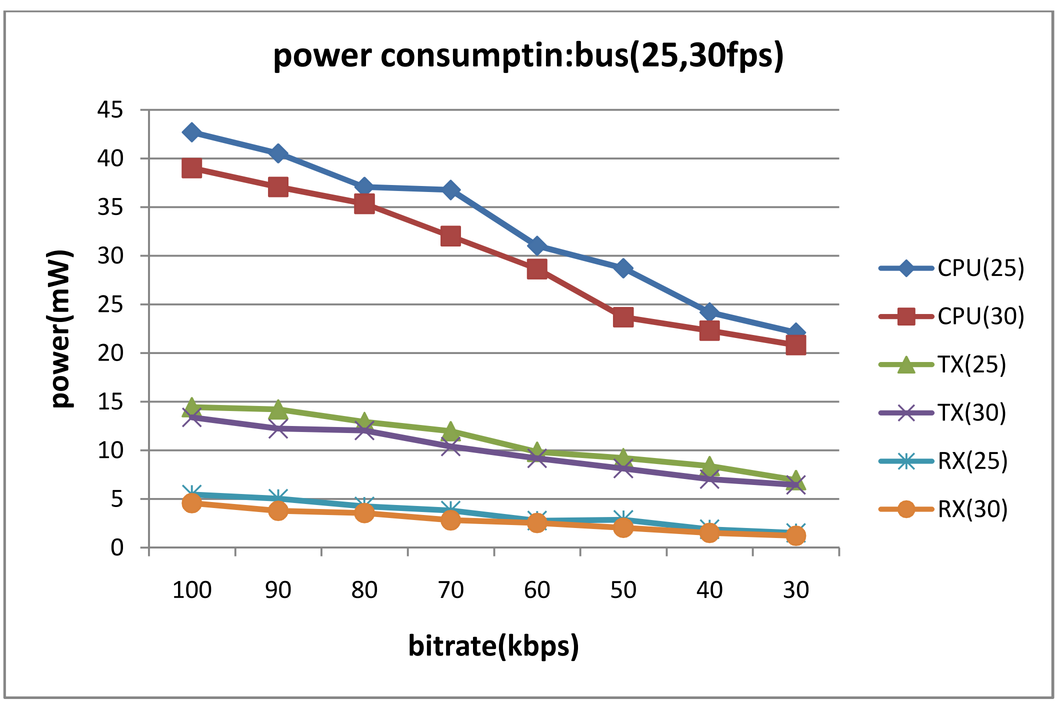

Figure 6.8:Power (mW)versus frame rate for bus sequence

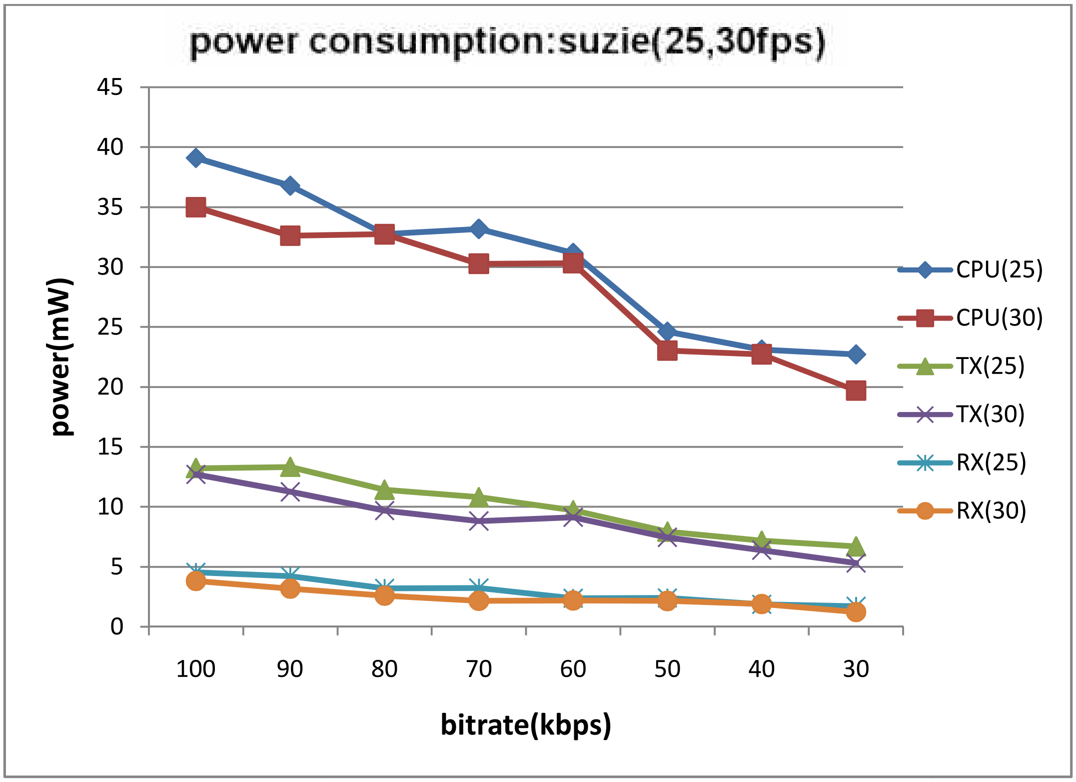

Figure 6.9: Power (mW)versus frame rate for suzie sequence

6.1.3 Impact of Video Content on Power Consumption for the Videos on IoT Devices

We evaluate the power consumption in mW for the CPU,TX,RX when the fast movement (crew) video is transmitted from sender to the receiver. The resolution was kept fixed at 174x144 ,frame rate is 25 and 30 fps and bitrates was varied from 30-100 kbps. Figure 6.10 and figure 6.11 show the power consumption of video transmission when the frame rate is 25 fps and 30 fps, respectively. With respect to the bitrate and frame rate parameters, we observed that videos with fast movement consume more power than the videos with medium and slow movement . This is also due to the fact that, fast movement video sequences are more complexity than slow and medium video sequences.

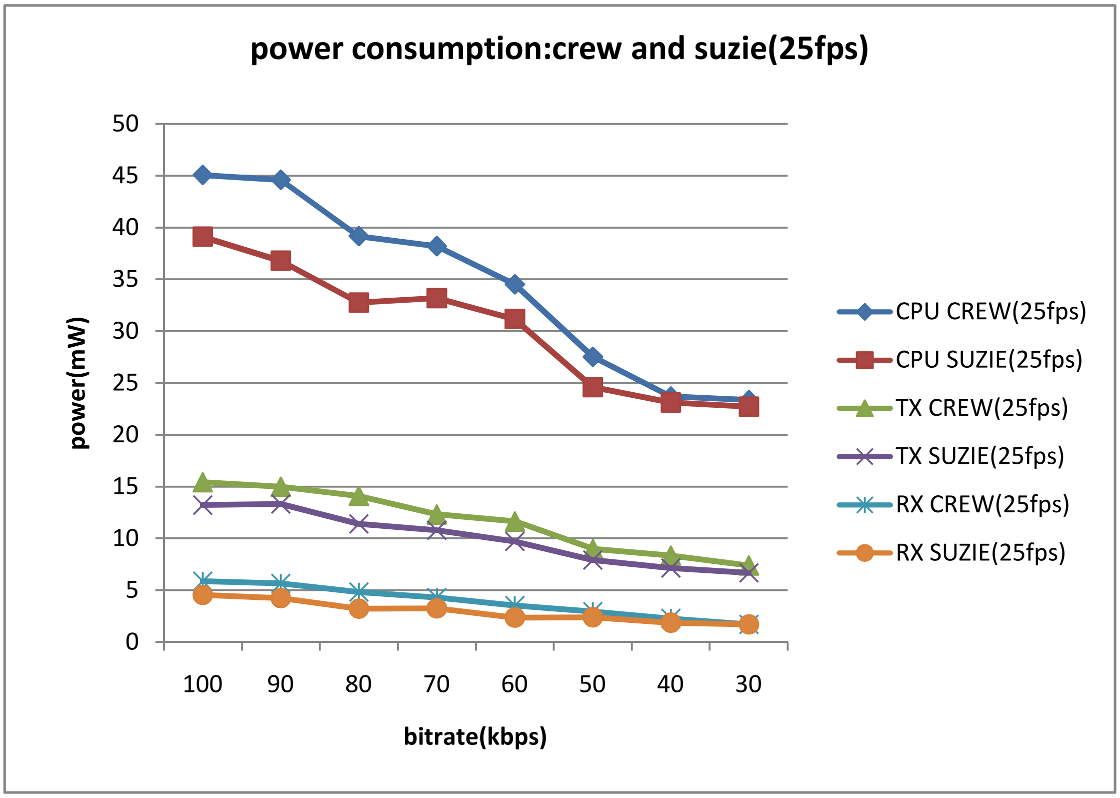

Figure 6.10: Power consumption for crew and suzie(25fps)

.

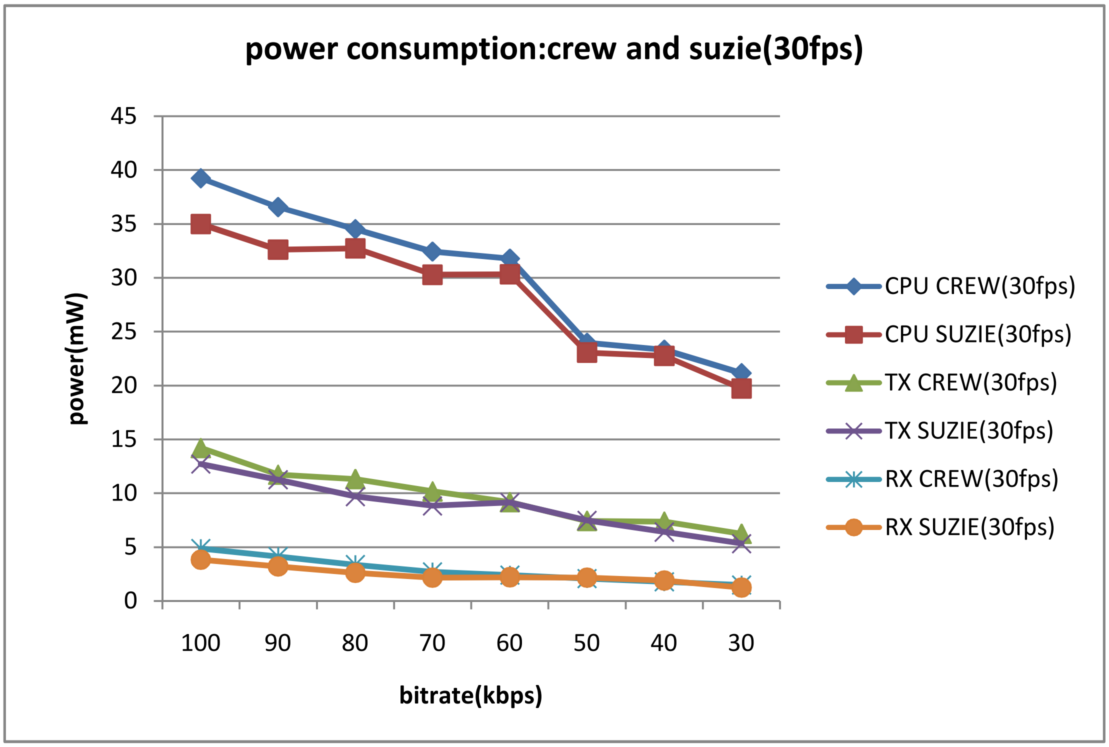

Figure 6.11:power consumption for crew and suzie (30fps)

6.2 Impact of Power Parameter on Video Quality for IoT Device

To evaluate the power aspects on the quality of experience, three videos with different content and different quality were employed from the reconstructed video sequences transmitted over the IoT devices. There were no packet losses in the entire process. As shown in Figure 6.12 it was observed that for the same bitrate, frame rate and resolution, slow and medium video sequences recorded better MOS values and hence, better quality than fast movement video sequences. This is because fast movement video sequences need high bit rates for better quality compared to slow and medium video sequences. For slow movement video sequence, low frame rate could suit them because there are not much temporal differences.

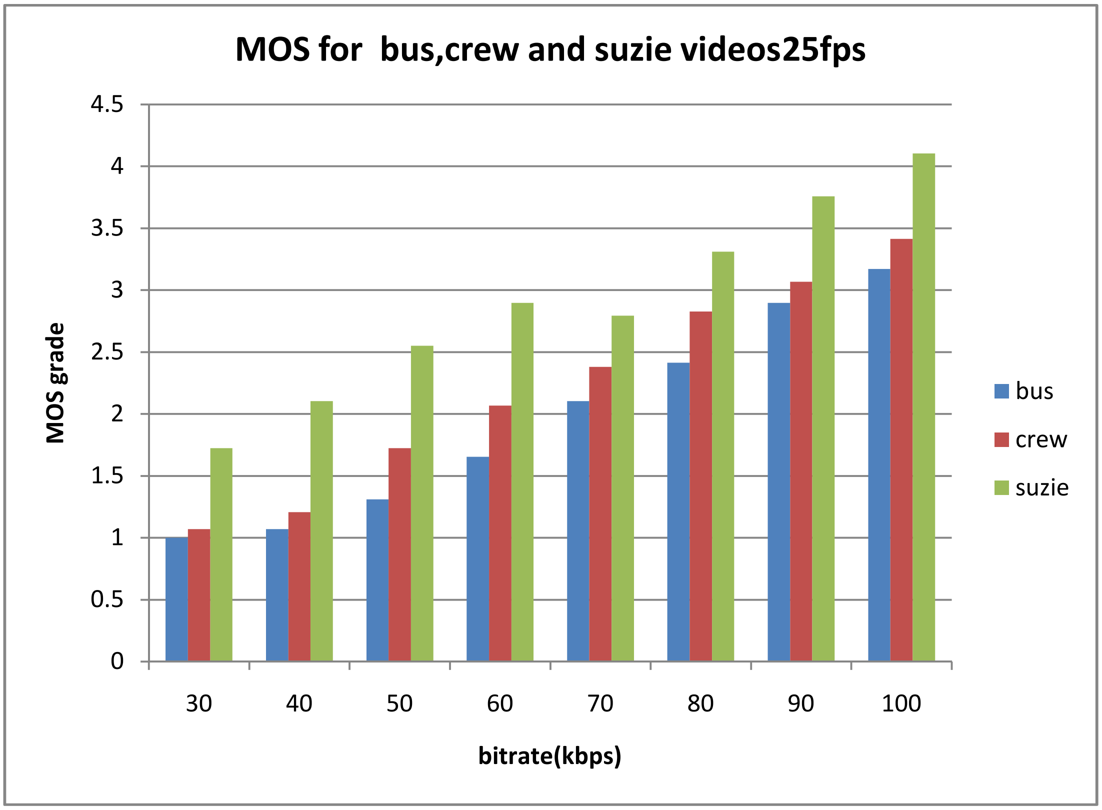

Figure 6.12:The Mean Opinion Score

The P-value results in Table 6.1 demonstrate that the bitrate parameter has significant (0.0029< 0.05) meaningful addition to the power consumption of the IoT device during video transmission.

Table 6.1:P-value for bitrates values

	Df	*SS*	*MS*	*F*	*P-value*
Regression	1	1924.67	1924.67	1.35	0.0029
Residual	7	9979.617	1425.66		
Total	8	11904.29			

The P-value in Table 6.2 illustrates that the frame rate parameter has significant (0.0016< 0.05) meaningful addition to the power consumption of the IoT devices. Since these p-values are less than the threshold (0.05), frame rate and bitrate parameters are statistically significant to the energy consumption of an IoT device during video transmission over the low power wireless network.

Table 6.2:P-value for bitrates values

	df	*SS*	*MS*	*F*	*P-value*
Regression	1	9295.485	9295.485	24.94	0.0016
Residual	7	2608.802	372.6859		
Total	8	11904.29			

6.3 Power Modeling Based on Regression Technique

The model for the power consumption is based on the encoding frame rate and bitrate while the encoding resolution was fixed. The smallest QCIF resolution and low bitrates were preferred due to IoT constraints in memory, storage and energy.

A non-linear regression has been used to model the power consumption with the dataset spilt into training (70% of the data) and testing (30% of the data). The total power consumption in mW, which is a combination of CPU, data transmission, data receiver and listening cycles, can be expressed as a function of frame rate and bitrate.

$$\text{Power} = f(br, fr) \qquad (10)$$

Where, br and fr denotes bitrate and frame rate respectively. From non-linear regression, the total power consumption model is then derived as,

$$\text{Power} = \alpha * fr * \ln(br) * \beta \quad (11)$$

Where α and β are coefficients (Table 6.3).

The proposed power model in (11) was evaluated by using of 30% of the dataset for validating the model. The accuracy of the model was determined by using the Root Mean Squared Error (RMSE) and correlation coefficient R^2 (Table 6.4).

Table 6.3: Model coefficients

Coefficient	fast movement	medium movement	slow movement
	Crew	Bus	Suzie
α	0.168	0.148	0.125
β	15.027	16.392	17.490

6.4 Model Evaluation

The propose QoE model is based on the encoding frame rate and bitrate while the encoding resolution was fixed was evaluated by the use of 30% of the dataset for validation. RMSE and correlation coefficient were used to find the accuracy of the proposed model in (11). The correlation coefficients and RMSE values are tabulated in Table 6.4

Table 6.4: RMSE and correlation coefficient R2

T	fast movement	medium movement	slow movement
	Crew	Bus	Suzie
R^2	0.974	0.986	0.967
RMSE	2.318	1.491	1.946

Figure 6.10-6.11 depicts the correlation predicted and actual power values of video sequences used in this thesis .It can therefore be argued that the proposed power consumption modeling IoT with video quality communication can be used by and network and service providers as a simple but yet an accurate objective model. The model can be used to control and optimize video quality in IoT devices.

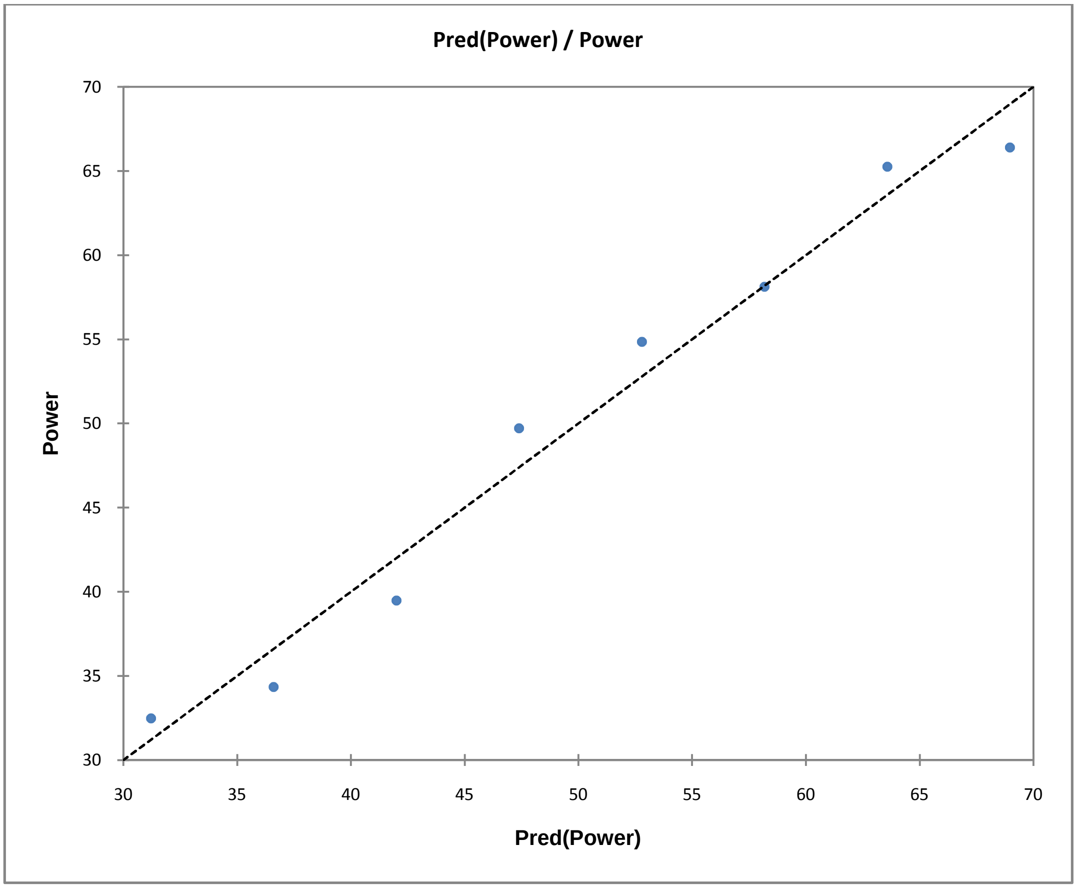

Figure 6.13:Model validation for fast movement sequences

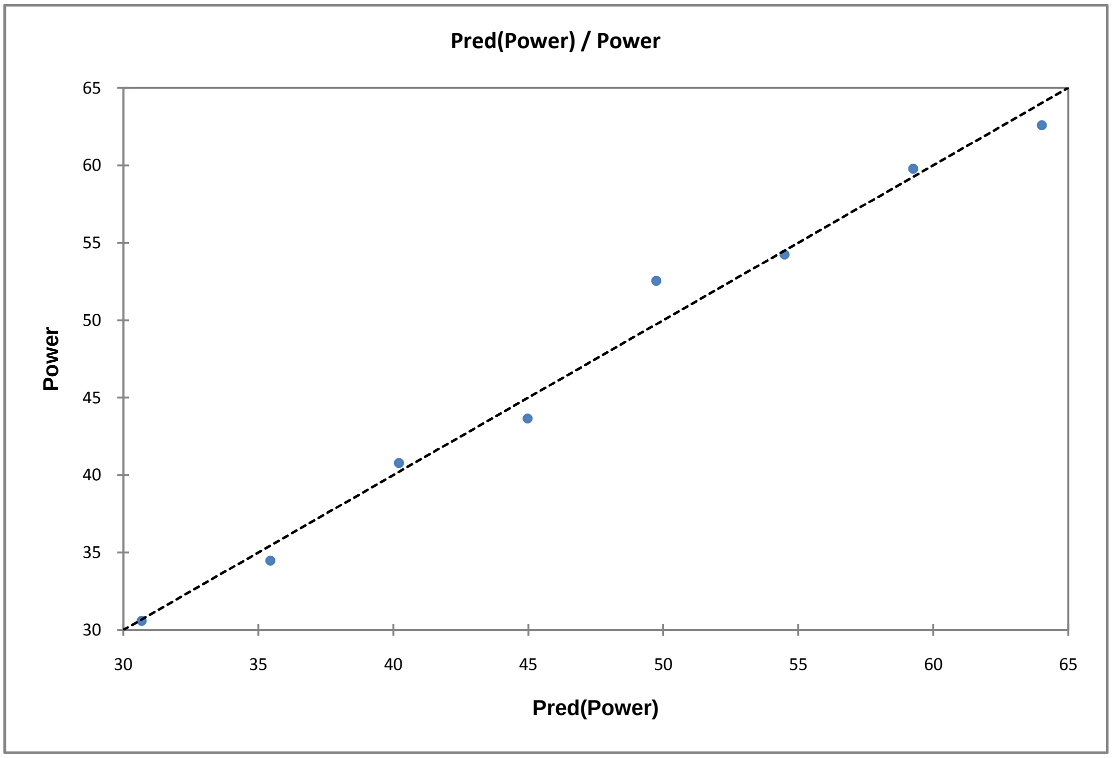

Figure 6.14:Model validation for medium movement sequences

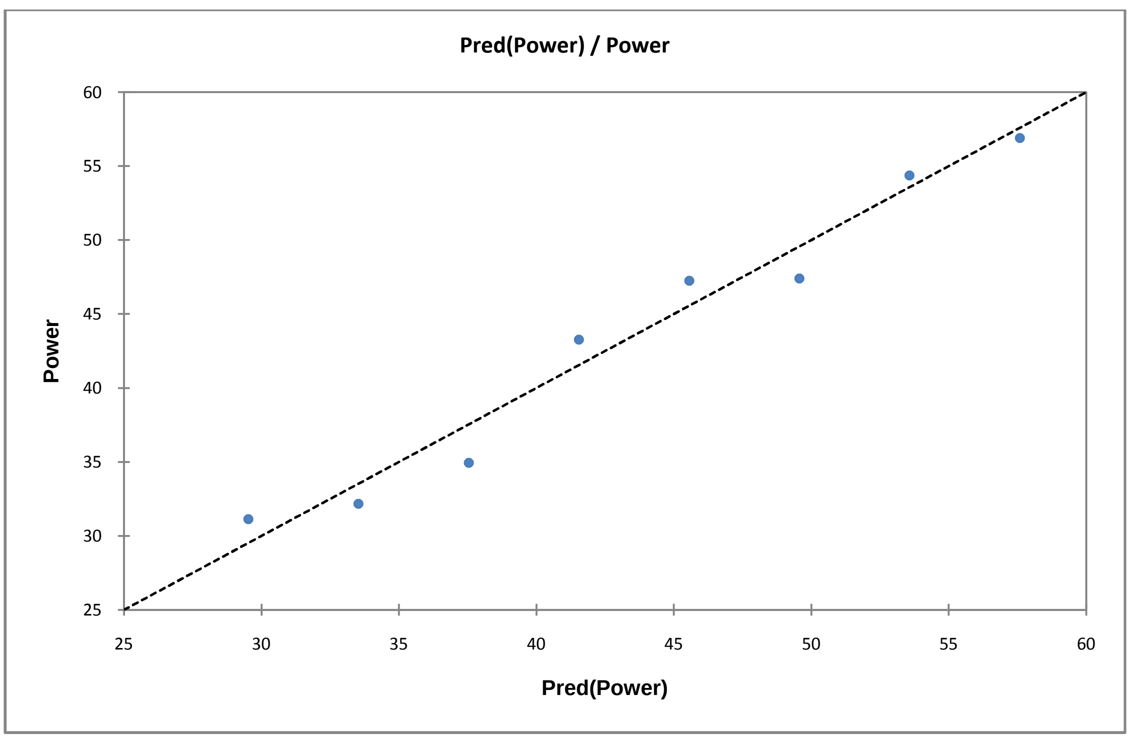

Figure 6.15:Model validation for slow movement sequences

6.5 Impact the Gender on QoE for the Video Services

To evaluate the gender aspects on the quality of experience, three videos with different content and different quality were employed. We used gender (male, female) parameter to assess the quality of those videos. In addition, MOS was initialized for the subjective test.

According to the results that we obtained from the subjective test, considering the gender parameter as in Figure 6.13, we have found that people's desire in quality of video showed a statistically significant difference with the variation in gender ($p < 0.05$, from t-test method).

With respect to the gender parameter, we observed that the male viewers of videos may request a higher quality compared to female viewers as shown in Figure6.14. We also observed that male viewers do not concentrate on the video content in contrast to female viewers who concentrate on the video content; if they don't like the video content they will give a low score.

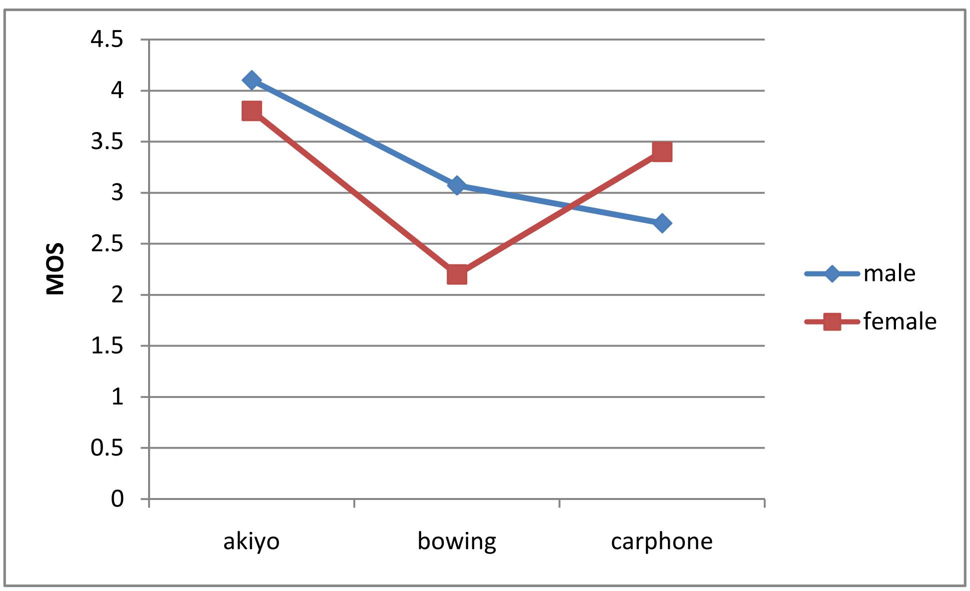

Figure 6.16:The Mean Opinion Score (gender)

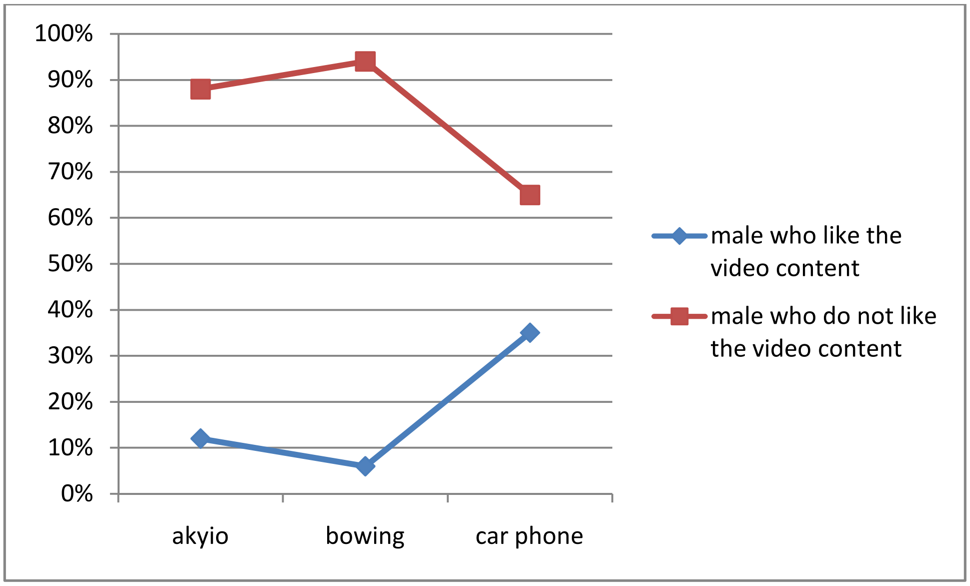

Figure 6.17:Male and female who like the content

CHAPTER VII

7. Conclusions and Future Work

7.1 Conclusions

This dissertation has presented the impact of bitrate and frame rate on the power consumption of IoT devices. It has also proposed the power consumption model, which included bitrates and frame rates as non-network parameters. Experimental results showed that the videos with higher bitrates and low frame rates consumed more power than the videos with low bitrates and high frame rates. It was also found that videos with high movement consume more power than videos with medium and slow movement. The power consumption model can be used to predict power usage of IoT devices. For IoT devices, memory and power are one of the major constraints. Therefore, in order to deliver acceptable video quality, video encoding parameters should be carefully chosen in order to save devices' energy during video transmission without jeopardizing the QoE.

In this dissertation we have also presented the impact of gender on the Quality of Experience for video services. Gender is part of user context. We mentioned several challenges related to QoE modeling. We have also discussed existing methods and explained their advantages and weaknesses. In addition, we have also shown some initial results for a subjective QoE assessment based on gender. They have suggested that gender influences the expected QoE ratings where male viewers notice the quality level of the video, while female viewers possibly pay more attention to the content of the video.

7.2 Limitations of the Current Research

This thesis has successfully met all the requirements and objectives set out in the Introduction chapter, there are some limitations that can be addressed in the future work.

- Considerations of high bitrates options

The send bit rates used in this research ranged from 30 Kbps to 100 Kbps . Future work could consider high bit rates from 300 Kbps for 240p, 360p to 480p. Higher resolutions could be considered in the future with corresponding bit rates from 300 Kbps to 1000 Kbps. The frame rate of 60 fps and 15 fps could also be considered in the future corresponding. This will make the proposed model more generic for a wide range of values of frame rates, bit rates and resolutions.

- Improved validation of the work:

Although the new Scale Factor model has been validated by the data that was not used to derive the model, external dataset and large scale dataset is still needed to further validate the model.

- Content types consideration:

In this thesis, three different video sequences were considered. These video sequences covered slow movement video (suzie) , medium movement video (bus) to fast movement video (crew). The spatial and temporal features of movie sequences may have an impact on the content type and therefore, on the new proposed model

7.3 Future Work

The following point can be considered in future research to improve existing work:

1. Implement the proposed model in IoT systems in order to control and optimize video quality transmission.
2. Develop a model to evaluate video quality by considering the non-network parameter (power).
3. Testing a wide range of video send bit rates, frame rates and resolutions.
4. Proposing Objective methods of measuring and predicting video quality on IoT device.
5. Develop a model to evaluate video quality by considering additional Context parameters related to users.

REFERENCES

[1] D. Guinard, V. Trifa, F. Mattern, and E. Wilde, “From the Internet of Things to the Web of Things : Resource Oriented Architecture and Best Practices,” *Archit. Internet Things*, pp. 97–129, 2011.

[2] A. Rao, A. Legout, Y. Lim, D. Towsley, C. Barakat, and W. Dabbous, “Network characteristics of video streaming traffic,” *Proc. Seventh Conf. Emerg. Netw. Exp. Technol. - Conex. ’11*, pp. 1–12, 2011.

[3] N. Thomos, S. Argyropoulos, N. V. Boulgouris, and M. G. Strintzis, “Robust transmission of H.264/AVC streams using adaptive group slicing and unequal error protection,” *EURASIP J. Appl. Signal Processing*, vol. 2006, pp. 1–13, 2006.

[4] L. Plissonneau and E. Biersack, “A longitudinal view of HTTP video streaming performance,” *Proc. 3rd Multimed. Syst. Conf. - MMSys ’12*, p. 203, 2012.

[5] V. K. Adhikari, S. Jain, Y. Chen, and Z. L. Zhang, “Vivisecting YouTube: An active measurement study,” *Proc. - IEEE INFOCOM*, pp. 2521–2525, 2012.

[6] J. Guo, E. W. M. Wong, S. Chan, P. Taylor, M. Zukerman, and K.-S. Tang, “Performance Analysis of Resource Selection Schemes for a Large Scale Video-on-Demand System,” *Multimedia, IEEE Trans.*, vol. 10, no. 1, pp. 153–159, 2008.

[7] J. Summers, T. Brecht, D. Eager, and B. Wong, “To chunk or not to chunk: Implications for HTTP streaming video server performance,” *… Digit. Audio Video*, pp. 15–20, 2012.

[8] P. Martinez-Julia, E. T. Garcia, J. O. Murillo, and A. F. Skarmeta, “Evaluating video streaming in network architectures for the internet of things,” *Proc. - 7th Int. Conf. Innov. Mob. Internet Serv. Ubiquitous Comput. IMIS 2013*, pp. 411–415, 2013.

[9] T. S. Noor and N. I. Osman, “The Influnce of Gender on QoE Subjective Assessment for Video Services,” vol. 7, no. 4, pp. 198–203, 2018.

[10] T. S. Noor, N. I. Osman, and I.-H. M. Mkwawa, “Analysis and Modelling of Power Consumption in IOT with Video Quality Communication,” *Int. J. Multimed. Its Appl.*, vol. 10, no. 1/2/3, pp. 15–27, 2018.

[11] ITU-T P.10/G.100 amd2, “Vocabulary for performance and quality of service,” *Ser. P Teleph. Transm. Qual. Teleph. Install. Local Line Networks*, vol. 100, no. 2006, pp. 1–10, 2008.

[12] ITU-T Recommendation P.910, “Subjective video quality assessment methods for multimedia applications,” *Int. Telecommun. Union, Geneva*, pp. 1–42, 2009.

[13] ITU, “Methodology for the subjective assessment of the quality of television pictures,” *Itu-R Bt.500-11*, vol. 211, no. BT.500-11, pp. 1–48, 2002.

[14] M. Ghanbari, *Standard Codecs*. 2011.

[15] "ITU-T SG 9, Q 12/9, Hybrid perceptual/bitstream models." .

[16] Z. Wang, L. Lu, and A. C. Bovik, "Video Quality Assessment Based on Structural Distortion Measurement," vol. 19, no. 1, pp. 1–9, 2004.

[17] A. Takahashi, D. Hands, and V. Barriac, "Standardization Activities in the ITU for a QoE Assessment of IPTV," *IEEE Communications Magazine*, 2008. .

[18] B.Girod, "EE368b Image and Video Compression." .

[19] M. R. Color, S. Transforms, D. Marpe, H. Kirchhoffer, V. George, and T. Wiegand, "_ t X ~ FixedRGB," pp. 3157–3160, 2006.

[20] G. Wei, L. Wu, S. Wang, and C. Qi, "Fast mode selection for H . 264 video coding standard based on motion region classification," pp. 453–466, 2012.

[21] "OVERVIEW OF THE SCALABLE H . 264 / MPEG4-AVC EXTENSION Heiko Schwarz , Detlev Marpe , and Thomas Wiegand Fraunhofer Institute for Telecommunications – Heinrich Hertz Institute , Image Processing Department."

[22] I. Transactions, O. N. Circuits, S. For, and V. Technology, "Introduction to the Special Issue on the H . 264 / AVC Video Coding Standard," vol. 13, no. 7, pp. 557–559, 2003.

[23] P. Brooks and B. Hestnes, "User measures of quality of experience: Why being objective and quantitative is important," *IEEE Netw.*, vol. 24, no. 2, pp. 8–13, 2010.

[24] K. Mitra, C. Åhlund, and A. Zaslavsky, "A decision-theoretic approach for quality-of-experience measurement and prediction," *Proc. - IEEE Int. Conf. Multimed. Expo*, pp. 1–4, 2011.

[25] W. Wu, A. Arefin, and R. Rivas, "Quality of experience in distributed interactive multimedia environments: toward a theoretical framework," *… Multimed.*, pp. 481–490, 2009.

[26] S. Möller, K.-P. Engelbrecht, C. Kuhnel, I. Wechsung, and B. Weiss, "A taxonomy of quality of service and Quality of Experience of multimodal human-machine interaction," *2009 Int. Work. Qual. Multimed. Exp.*, pp. 7–12, 2009.

[27] J. Song, F. Yang, and S. Wan, "QOE evaluation of video services considering users' behavior," *2014 IEEE Int. Conf. Multimed. Expo Work.*, pp. 1–6, 2014.

[28] W. Song, D. Tjondronegoro, and M. Docherty, "Understanding User Experience of Mobile Video: Framework, Measurement, and Optimization," *Mob. Multimed. - User Technol. Perspect.*, 2012.

[29] A. Perkis, S. Munkeby, and O. Hillestad, "A model for measuring Quality of Experience," *Proc. 7th Nord. Signal Process. Symp. - NORSIG 2006*, pp. 198–201, 2006.

[30] K. Kilkki, "Quality of Experience in Communications Ecosystem," *J. Univers. Comput. Sci.*, vol. 14, no. 5, pp. 615–624, 2008.

[31] R. Laghari and N. Crespi, "QoE aware Service Delivery in Distributed Environment," pp. 837–842, 2011.

[32] A. Floris and L. Atzori, "Quality of Experience in the Multimedia Internet of Things: Definition and practical use-cases," *2015 IEEE Int. Conf. Commun. Work. ICCW 2015*, pp. 1747–1752, 2015.

[33] B. Martinez, M. Montón, I. Vilajosana, and J. D. Prades, "The Power of Models: Modeling Power Consumption for IoT Devices," *IEEE Sens. J.*, vol. 15, no. 10, pp. 5777–5789, 2015.

[34] I. H. Mkwawa and L. Sun, "Power-driven VoIP quality adaptation over WLAN in mobile devices," *2012 IEEE Globecom Work. GC Wkshps 2012*, pp. 1276–1281, 2012.

[35] M. Csernai and A. Gulyás, "Wireless adapter sleep scheduling based on video QoE: How to improve battery life when watching streaming video?," *Proc. - Int. Conf. Comput. Commun. Networks, ICCCN*, 2011.

[36] B. H. Chunlei Xie , Xin Zhang , Yijie Li, "QoE-driven energy efficiency promotion for mobile video service," pp. 1030–1035, 2015.

[37] Julio Arauz and Tony Fynn-Cudjoe, "Data Quality in the Internet of Things," pp. 34–42, 2013.

[38] C. Mulligan and D. Boyle, *From Machine-to-Machine to the Internet of Things Introduction to a New Age of Intelligence From Machine-to-Machine to the Internet of Things Introduction to a New Age of Intelligence Stefan Avesand*. .

[39] "Internet of Things (IoT)." [Online]. Available: http://www.cisco.com/c/en/us/solutions/internet-of. [Accessed: 25-Jul-2017].

[40] R. Duan, X. Chen, and T. Xing, "A QoS architecture for IOT," *Proc. - 2011 IEEE Int. Conf. Internet Things Cyber, Phys. Soc. Comput. iThings/CPSCom 2011*, pp. 717–720, 2011.

[41] and M. D. Adam Dunkels, Oliver Schmidt, Niclas Finne, Joakim Eriksson, Fredrik Osterlind, Nicolas Tsiftes, "The Contiki os: The operating system for the internet of things," 2011.

[42] B. Sobhan, P. L. Padmaja, T. Ramanjaneyulu, I. L. Narayana, and K. Srikanth, "Web Site : www.ijettcs.org Email : editor@ijettcs.org Role of COOJA Simulator in IoT For such requirements the traditional network simulators," vol. 6, no. 2, pp. 139–143, 2017.

[43] G. D. Levis P, "TinyOS Programming." 2009.

[44] P. Pu, J. Eliasson, and J. Delsing, "An Authentication and Access Control Framework for CoAP-based Internet of Things," no. D.

[45] O. M. Alliance, *Lightweight Machine to Machine Technical Specification*. 2017.

[46] Eclipse Foundation, "Leshan LWM2M Server." [Online]. Available: http://www.eclipse.org/leshan.

[47] Tetcos, “Netsim emulator.”

[48] A. Varga and R. Hornig, “AN OVERVIEW OF THE OMNeT ++ SIMULATION ENVIRONMENT,” 2008.

[49] T. R. Henderson and G. F. Riley, “Network Simulations with the ns-3 Simulator,” p. 2006, 2006.

[50] R. Technologies, “Opnet simulator,” 2017.

[51] J. Ahrenholz, C. Danilov, T. R. Henderson, and J. H. Kim, “Core: A real-time network emulator.”

[52] B. A. Klein, “RPL : IPv6 Routing Protocol for Low Power and Lossy Networks,” no. July, pp. 59–66, 2011.

[53] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis, “Collection Tree Protocol,” 2009.

[54] Z. Alliance, “Zigbee specification.” [Online]. Available: http://www.zigbee.org. [Accessed: 25-Apr-2017].

[55] and C. S. . Kushalnagar, G. Montenegro, “IPv6 over Low-Power Wireless Personal Area Networks,” pp. 1–12, 2007.

[56] Z. Shelby and C. Bormann, *6LoWPAN : The Wireless Embedded Internet*. .

[57] S. A. Alvi, B. Afzal, G. A. Shah, L. Atzori, and W. Mahmood, “Internet of multimedia things: Vision and challenges,” *Ad Hoc Networks*, vol. 33, no. April, pp. 87–111, 2015.

[58] R. Yao, W. Wang, M. Farrokh-Baroughi, H. Wang, and Y. Qian, “Quality-driven energy-neutralized power and relay selection for smart grid wireless multimedia sensor based IoTs,” *IEEE Sens. J.*, vol. 13, no. 10, pp. 3637–3644, 2013.

[59] S. . P. Le Callet, P.; Möller, “Qualinet White Paper on Definitions of Quality of Experience To cite this version : HAL Id : hal-00977812 Qualinet White Paper on Definitions of Quality of Experience Output from the fifth Qualinet meeting , Novi Sad ,” 2014.

[60] A. Karaadi, L. Sun, and I.-H. Mkwawa, “Multimedia Communications in Internet of Things QoT or QoE?,” *2017 IEEE Int. Conf. Internet Things IEEE Green Comput. Commun. IEEE Cyber, Phys. Soc. Comput. IEEE Smart Data*, pp. 23–29, 2017.

[61] et al. Wu Q, Ding G, Xu Y, “Cognitive Internet of Things: A New Paradigm Beyond Connection,” *IEEE Internet Things J.*, vol. 1, no. 2, pp. 129–143, 2014.

[62] V. Gazis, “A Survey of Standards for Machine-to-Machine and the Internet of Things,” *IEEE Commun. Surv. Tutorials*, vol. 19, no. 1, pp. 482–511, 2017.

[63] L. Tan, “Future internet: The Internet of Things,” *2010 3rd Int. Conf. Adv. Comput. Theory Eng.*, pp. V5-376-V5-380, 2010.

[64] “Contiki OS: Using Powertrace to estimate power consumption.” [Online].

Available: http://thingschat.blogspot.mk/2015/04/contikios- using-powertrace-and.html. [Accessed: 03-Sep-2017].

[65] . A. Ponemon and J. P. Wendell, “Judgmental versus random sampling in auditing: An experimental investigation,” vol. 14, p. 17, 1995.

[66] ITU-T Rec. J.148, “Requirements for an objective perceptual multimedia quality model ITU-T,” vol. 144, 2003.

[67] L. Janowski, “SUBJECT BIAS : INTRODUCING A THEORETICAL USER MODEL Department of Telecommunications AGH University of Science and Technology , Margaret Pinson U . S . Department of Commerce , National Telecommunications and Information Administration (NTIA),” pp. 2–7, 2014.

[68] I. T. Rec, *Methods of subjective determination of transmission quality*. 1996.

[69] M. J. Crawley, *Statistical computing: An introduction to data analysis using*. 2002.

[70] and G. G. V. D. C. Montgomery, E. A. Peck, *Introduction to linear regression analysis. John Wiley & Sons*. 2015.

[71] E. Board, “MSP-EXP430F5438 Experimenter Board User â€™ s Guide,” no. June, 2011.

[72] A. Khan, L. Sun, and E. Ifeachor, “Impact of Video Content on Video Quality for Video over Wireless Networks,” *2009 Fifth Int. Conf. Auton. Auton. Syst.*, pp. 277–282, 2009.

APPENDICES

APPENDIX A: Unicat sender code

```
/*
 * Copyright (c) 2011, Swedish Institute of Computer Science.
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 * 3. Neither the name of the Institute nor the names of its contributors
 *    may be used to endorse or promote products derived from this software
 *    without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE INSTITUTE AND CONTRIBUTORS
``AS IS'' AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED
TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
PARTICULAR PURPOSE
 * ARE DISCLAIMED.  IN NO EVENT SHALL THE INSTITUTE OR
CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING
IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
POSSIBILITY OF
 * SUCH DAMAGE.
 *
 * This file is part of the Contiki operating system.
 *
 */

#include "contiki.h"
```

```
#include "lib/random.h"
#include "sys/ctimer.h"
#include "sys/etimer.h"
#include "net/uip.h"
#include "net/uip-ds6.h"
#include "net/uip-debug.h"

#include "sys/node-id.h"
#include "powertrace.h"
#include "simple-udp.h"
#include "servreg-hack.h"
#include <stdio.h>
#include <string.h>
#define UDP_PORT 1234
#define SERVICE_ID 190
#define SEND_INTERVAL		(60 * CLOCK_SECOND)
#define SEND_TIME		(random_rand() % (SEND_INTERVAL))
static struct simple_udp_connection unicast_connection;
/*---------------------------------------------------------------------------*/
PROCESS(unicast_sender_process, "Unicast sender example process");
AUTOSTART_PROCESSES(&unicast_sender_process);
/*---------------------------------------------------------------------------*/
static void receiver(struct simple_udp_connection *c,
         const uip_ipaddr_t *sender_addr,
         uint16_t sender_port,
         const uip_ipaddr_t *receiver_addr,
         uint16_t receiver_port,
         const uint8_t *data,
         uint16_t datalen)
{
  printf("Data received on port %d from port %d with length %d\n",
         receiver_port, sender_port, datalen);
}
/*---------------------------------------------------------------------------*/
static void
set_global_address(void)
{
  uip_ipaddr_t ipaddr;
  int i;
  uint8_t state;
  uip_ip6addr(&ipaddr, 0xaaaa, 0, 0, 0, 0, 0, 0, 0);
  uip_ds6_set_addr_iid(&ipaddr, &uip_lladdr);
  uip_ds6_addr_add(&ipaddr, 0, ADDR_AUTOCONF);
  printf("IPv6 addresses: ");
  for(i = 0; i < UIP_DS6_ADDR_NB; i++) {
    state = uip_ds6_if.addr_list[i].state;
```

```
    if(uip_ds6_if.addr_list[i].isused &&
      (state == ADDR_TENTATIVE || state == ADDR_PREFERRED)) {
     uip_debug_ipaddr_print(&uip_ds6_if.addr_list[i].ipaddr);
     printf("\n");
    }
  }
}
/*---------------------------------------------------------------------------*/
PROCESS_THREAD(unicast_sender_process, ev, data)
{
  //static struct etimer periodic_timer;
  static struct etimer send_timer;
  uip_ipaddr_t *addr;
  PROCESS_BEGIN();
 powertrace_start(CLOCK_SECOND * 2);
  servreg_hack_init();
  set_global_address();
  simple_udp_register(&unicast_connection, UDP_PORT,
                      NULL, UDP_PORT, receiver);
  //etimer_set(&periodic_timer, SEND_INTERVAL);
  while(1) {
     static unsigned int message_number=0;
//-------------------bustime-------------------------------------
//floatb[19]={1000,0.001,0.042,0.08,0.12,0.16,0.198,0.238,0.278,0.318,0.358,0.396,0.43
7,
0.477,0.515,0.555,0.595,0.635,0.674};/*bus60-25*/
//floatb[19]={1000,0.001,0.034,0.067,0.1,0.134,0.167,0.2,0.232,0.266,0.298,0.331,0.364,
0.398,0.431,0.464,0.497,0.529,0.563};/*bus60-30*/
//float
b[19]={1000,0.001,0.041,0.08,0.12,0.16,0.199,0.24,0.277,0.317,0.358,0.397,0.438,
0.477,0.516,0.555,0.595,0.636,0.674};/*bus70-25*/
//float b[19]={1000,0,0.034,0.066,0.101,0.135,0.166,0.198,0.232,0.264,0.297,0.33,0.363,
0.397,0.43,0.462,0.495,0.529,0.561};/*bus70-30*/
//float b[19]={1000,0.003,0.042,0.08,0.124,0.159,0.199,0.238,0.278,0.317,0.358,0.396,
0.437,0.476,0.515,0.555,0.595,0.635,0.674};/*bus80-25*/
//float
b[19]={1000,0,0.034,0.066,0.101,0.136,0.166,0.198,0.232,0.264,0.298,0.331,0.363,
0.397,0.43,0.462,0.495,0.529,0.562};/*bus80-30*/
//float b[19]={1000,0.001,0.04,0.08,0.121,0.16,0.2,0.239,0.279,0.318,0.357,0.397,0.437,
0.477,0.515,0.557,0.596,0.634,0.674};/*bus90-25*/
//float b[19]={1000,0.001,0.034,0.067,0.104,0.141,0.169,0.198,0.232,0.265,0.298,0.331,
0.363,0.397,0.431,0.463,0.495,0.529,0.561};/*bus90-30*/
//float b[19]={1000,0.001,0.041,0.08,0.12,0.16,0.199,0.238,0.278,0.317,0.358,0.398,
0.437,0.477,0.515,0.555,0.595,0.635,0.675};/*bus100-25*/
//float b[19]={1000,0,0.034,0.067,0.099,0.14,0.167,0.199,0.233,0.265,0.298,0.331,0.364,
0.397,0.431,0.463,0.496,0.529,0.562};/*bus100-30*/
```

```
//---------------crewtime--------------------------------------------------
//float b[19]={1000,0,0.04,0.081,0.119,0.16,0.198,0.238,0.278,0.318,0.358,0.397,0.436,
0.477,0.515,0.556,0.594,0.634,0.675};/*crew60-25*/
//float b[19]={1000,0,0.033,0.066,0.099,0.135,0.166,0.199,0.232,0.266,0.298,0.331,
0.364,0.398,0.429,0.462,0.495,0.529,0.562};/*crew60-30*/
//float
b[19]={1000,0,0.04,0.081,0.12,0.16,0.2,0.238,0.278,0.318,0.358,0.397,0.436,0.477,
0.515,0.562,0.595,0.635,0.674};/*crew70-25*/
//float
b[19]={1000,0,0.036,0.067,0.1,0.134,0.166,0.2,0.233,0.265,0.298,0.331,0.364,0.398,
0.431,0.464,0.495,0.529,0.563};/*crew70-30*/
//float
b[19]={1000,0,0.04,0.081,0.122,0.16,0.199,0.24,0.279,0.318,0.357,0.398,0.437,0.476,
0.516,0.556,0.594,0.636,0.675};/*crew80-25*/
//float b[19]={1000,0.001,0.034,0.068,0.099,0.133,0.166,0.2,0.232,0.266,0.298,0.332,
0.365,0.397,0.43,0.462,0.496,0.529,0.562};/*crew80-30*/
//float b[19]={1000,0.005,0.04,0.08,0.129,0.159,0.199,0.239,0.278,0.318,0.357,
0.398,0.437,0.476,0.516,0.556,0.595,0.635,0.674};/*crew90-25*/
//float b[19]={1000,0.001,0.033,0.066,0.1,0.133,0.165,0.199,0.232,0.265,0.297,0.33,
0.363,0.397,0.43,0.463,0.495,0.53,0.562};/*crew90-30*/
//float b[19]={1000,0,0.041,0.079,0.121,0.158,0.198,0.238,0.277,0.317,0.357,0.397,
0.436,0.477,0.516,0.556,0.594,0.635,0.679};/*crew100-25*/
//float b[19]={1000,0,0.034,0.068,0.102,0.133,0.166,0.199,0.232,0.266,0.299,0.332,
0.365,0.396,0.449,0.463,0.496,0.53,0.564};/*crew100-30*/
//----------------------suzietime-------------------------------------
//float b[19]={1000,0.001,0.04,0.08,0.122,0.159,0.199,0.239,0.279,0.318,0.357,0.397,
0.436,0.477,0.515,0.556,0.595,0.634,0.799};/*suzie60-25*/
//float b[19]={1000,0,0.035,0.068,0.106,0.133,0.167,0.199,0.232,0.265,0.299,0.331,
0.363,0.397,0.429,0.463,0.496,0.529,0.562};/*suzie60-30*/
//float b[19]={1000,0,0.04,0.081,0.119,0.162,0.201,0.238,0.278,0.318,0.358,0.401,
0.436,0.476,0.516,0.556,0.596,0.635,0.673};/*suzie70-25*/
//float b[19]={1000,0,0.033,0.067,0.099,0.132,0.165,0.199,0.232,0.266,0.298,0.331,
0.364,0.399,0.429,0.464,0.497,0.53,0.563};/*suzie70-30*/
//float b[19]={1000,0,0.041,0.081,0.12,0.159,0.199,0.243,0.279,0.317,0.357,0.397,
0.437,0.477,0.516,0.556,0.595,0.635,0.674};/*suzie80-25*/
//float b[19]={1000,0,0.036,0.068,0.104,0.138,0.17,0.198,0.232,0.266,0.299,0.334,
0.364,0.397,0.43,0.464,0.496,0.53,0.562};/*suzie80-30*/
//float b[19]={1000,0,0.041,0.081,0.122,0.159,0.2,0.239,0.278,0.318,0.358,0.398,0.436,
0.477,0.516,0.555,0.595,0.634,0.674};/*suzie90-25*/
//float b[19]={1000,0.001,0.034,0.067,0.1,0.133,0.166,0.199,0.232,0.271,0.297,0.331,
0.364,0.396,0.434,0.463,0.495,0.528,0.562};/*suzie90-30*/
//float b[19]={1000,0,0.04,0.081,0.121,0.161,0.198,0.239,0.278,0.317,0.357,0.398,0.437,
0.476,0.515,0.556,0.595,0.634,0.674};/*suzie100-25*/
//float b[19]={1000,0.001,0.034,0.068,0.105,0.133,0.165,0.198,0.232,0.265,0.297,0.33,
0.363,0.398,0.43,0.539,0.539,0.539,0.565};/*suzie100-30*/
//----------------------footballtime-----------------------------------
```

```
//float b[19]={1000,0.001,0.04,0.08,0.123,0.16,0.2,0.238,0.278,0.319,0.358,0.398,0.436,
0.476,0.517,0.555,0.596,0.634,0.675};/*footbal60-25*/
float b[19]={1000,0,0.034,0.067,0.101,0.146,0.17,0.2,0.235,0.265,0.299,0.331,0.364,
0.398,0.43,0.464,0.496,0.53,0.562};/*footbal60-30*/
//float b[19]={1000,0,0.041,0.08,0.12,0.161,0.198,0.238,0.277,0.317,0.358,0.397,0.436,
0.476,0.515,0.556,0.595,0.634,0.674};/*footbal70-25*/
//float b[19]={1000,0,0.034,0.067,0.105,0.134,0.166,0.199,0.231,0.265,0.298,0.331,
0.364,0.397,0.429,0.464,0.495,0.529,0.562};/*footbal70-30*/
//float b[19]={1000,0,0.042,0.081,0.12,0.16,0.2,0.238,0.278,0.318,0.358,0.397,0.436,
0.478,0.517,0.555,0.595,0.635,0.675};/*footbal80-25*/
//float b[19]={1000,0,0.038,0.066,0.11,0.133,0.165,0.199,0.232,0.264,0.297,0.33,0.363,
0.397,0.431,0.463,0.496,0.529,0.562};/*footbal80-30*/
//float b[19]={1000,0.001,0.041,0.08,0.121,0.16,0.199,0.238,0.278,0.318,0.357,0.398,
0.437,0.477,0.515,0.556,0.596,0.635,0.675};/*footbal90-25*/
//float
b[19]={1000,0.001,0.037,0.068,0.101,0.135,0.166,0.2,0.232,0.266,0.301,0.331,0.365,
0.398,0.43,0.462,0.496,0.529,0.562};/*footbal90-30*/
//float b[19]={1000,0.001,0.04,0.081,0.12,0.159,0.199,0.239,0.28,0.318,0.357,0.397,
0.437,0.476,0.515,0.555,0.596,0.635,0.674};/*footbal100-25*/
//float b[19]={1000,0,0.034,0.067,0.103,0.135,0.166,0.199,0.232,0.266,0.298,0.332,
0.364,0.397,0.43,0.463,0.496,0.53,0.562};/*footbal100-30*/
float x= b[message_number];
//float x=message_number;
etimer_set(&send_timer,((x/1000)+1)* CLOCK_SECOND);
   PROCESS_WAIT_EVENT_UNTIL(etimer_expired(&send_timer));
   addr = servreg_hack_lookup(SERVICE_ID);
/*else{
etimer_set(&send_timer, 1* CLOCK_SECOND);
   PROCESS_WAIT_EVENT_UNTIL(etimer_expired(&send_timer));
   addr = servreg_hack_lookup(SERVICE_ID);
}*/
   if(addr != NULL && message_number <18) {
/*----------------------------bus-------------------------------*/
//uint64_t  a[18]={549,159,500,138,503,151,488,136,484,133,524,123,497,156,514,
117,496,102};/*bus60-25*/
//uint64_t  a[18]={409,129,400,111,350,142,367,117,346,107,406,107,332,124,403,
95,356,80};/*bus60-30*/
//uint64_t  a[18]={674,157,650,145,635,153,644,139,632,148,625,150,655,162,617,
122,611,102};/*bus70-25*/
//uint64_t  a[18]={463,136,473,138,464,142,477,133,454,136,485,130,478,152,474,
116,464,101};/*bus70-30*/
//uint64_t  a[18]={804,188,754,176,754,164,736,154,757,194,760,160,800,169,795,
133,699,108};/*bus80-25*/
//uint64_t  a[18]={597,158,544,160,548,158,559,148,555,149,595,133,588,152,555,
110,552,95};/*bus80-30*/
//uint64_t  a[18]={914,199,903,187,836,182,867,173,931,208,922,168,878,199,901,
```

```
126,810,107};/*bus90-25*/
//uint64_t  a[18]={674,159,639,159,677,154,664,149,674,190,670,148,685,157,674,
124,635,109};/*bus90-30*/
//uint64_t
a[18]={1006,239,1013,234,984,236,1013,187,1048,224,1057,182,1025,203,1042,
129,907,113};/*bus100-25*/
//uint64_t  a[18]={758,187,755,194,743,167,746,166,800,198,789,163,793,178,781,
120,718,119};/*bus100-30*/
//--------------------crew--------------------------------------------
//uint64_t  a[18]={315,208,369,51,373,73,409,65,441,69,461,83,490,94,550,100,
521,75};/*crew60-25*/
//uint64_t  a[18]={257,120,318,52,306,67,327,61,342,62,351,69,417,87,463,
61,425,50};/*crew60-30*/
//uint64_t  a[18]={371,246,446,63,462,112,486,82,511,85,531,89,629,117,651,
117,641,89};/*crew70-25*/
//uint64_t  a[18]={297,164,345,59,355,66,370,71,409,76,432,79,485,88,508,
99,490,45};/*crew70-30*/
//uint64_t  a[18]={423,331,526,66,517,125,551,100,603,98,629,84,728,133,750,
105,737,97};/*crew80-25*/
//uint64_t  a[18]={331,241,395,59,414,104,442,82,457,91,490,76,535,110,610,
100,588,69};/*crew80-30*/
//uint64_t  a[18]={475,359,604,74,647,110,660,103,681,126,733,110,832,142,875,119,
884,106};/*crew90-25*/
//uint64_t  a[18]={366,288,449,65,471,104,511,91,517,100,602,92,635,113,673,103,
673,79};/*crew90-30*/
//uint64_t  a[18]={567,433,702,80,697,140,748,117,785,129,848,112,882,144,1007,147,
997,125};/*crew100-25*/
//uint64_t  a[18]={422,335,526,65,550,120,556,103,620,101,635,90,732,130,755,113,
736,105};/*crew100-30*/
//--------------------suzie--------------------------------------------
//uint64_t  a[18]={399,43,427,40,436,47,449,42,470,57,458,57,464,49,450,68,
428,28};/*suzie60-25*/
//uint64_t  a[18]={333,46,345,42,346,45,351,48,360,42,358,48,358,46,361,54,
339,26};/*suzie60-30*/
//uint64_t  a[18]={453,50,494,44,494,46,523,49,545,61,525,56,541,56,558,62,
511,35};/*suzie70-25*/
//uint64_t  a[18]={371,46,395,40,418,47,430,42,451,53,432,49,439,46,433,56,
395,26};/*suzie70-30*/
//uint64_t  a[18]={493,55,538,46,604,48,623,49,628,63,616,50,636,56,665,75,
570,35};/*suzie80-25*/
//uint64_t  a[18]={420,50,444,43,453,49,471,46,493,56,498,54,498,51,498,64,
471,29};/*suzie80-30*/
//uint64_t  a[18]={567,66,636,40,643,51,691,52,700,69,725,64,744,60,733,
72,682,45};/*suzie90-25*/
//uint64_t  a[18]={452,50,477,48,530,42,541,54,550,54,539,57,600,61,570,63,
535,38};/*suzie90-30*/
```

```
//uint64_t  a[18]={649,60,716,42,731,48,797,58,797,71,801,67,869,66,861,75,
789,49};/*suzie100-25*/
//uint64_t  a[18]={493,53,539,44,601,48,623,49,632,62,623,48,634,58,674,66,
598,32};/*suzie100-30*/
//------------------------football------------------
//uint64_t  a[18]={622,181,470,321,162,461,312,311,309,342,259,395,289,328,224,
392,258,246};/*footbal60-25*/
uint64_t  a[18]={465,152,375,243,155,367,251,251,245,279,178,314,248,272,198,
315,212,218};/*footbal60-30*/
//uint64_t  a[18]={753,223,605,398,238,508,355,345,355,432,270,439,354,404,308,451,
308,281};/*footbal70-25*/
//uint64_t a[18]={565,175,448,279,158,413,304,298,292,335,243,369,287,320,239,
371,250,246};/*footbal70-30*/
//uint64_t a[18]={884,242,650,458,271,608,397,428,435,477,330,483,428,464,339,
483,331,341};/*footbal80-25*/
//uint64_t a[18]={640,196,514,335,209,491,347,350,347,418,279,424,337,372,292,425,
295,280};/*footbal80-30*/
//uint64_t a[18]={1001,303,747,486,297,679,478,476,457,527,373,537,453,517,372,
547,361,399};/*footbal90-25*/
//uint64_t a[18]={734,236,574,421,252,525,392,423,418,475,312,462,384,430,295,
466,321,327};/*footbal90-30*/
//uint64_t a[18]={1100,324,857,537,327,734,496,527,542,614,426,585,491,559,475,
625,379,440};/*footbal100-25*/
//uint64_t a[18]={834,256,674,453,268,608,440,440,429,496,357,500,431,484,374,
513,334,373};/*footbal100-30*/
      printf("Sending unicast to ");
      uip_debug_ipaddr_print(addr);
      printf("\n");
      sprintf(buf, "Message %d", message_number);
simple_udp_sendto(&unicast_connection, buf, a[message_number], addr);
      message_number++;
    } else {
      printf("Service %d not found\n", SERVICE_ID);
    }
  }

  PROCESS_END();
}
/*---------------------------------------------------------------------------*/
```

APPENDIX B : Unicat Receiver Code

```
/*
 * Copyright (c) 2011, Swedish Institute of Computer Science.
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 * 3. Neither the name of the Institute nor the names of its contributors
 *    may be used to endorse or promote products derived from this software
 *    without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE INSTITUTE AND CONTRIBUTORS
``AS IS'' AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED
TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
PARTICULAR PURPOSE
 * ARE DISCLAIMED.  IN NO EVENT SHALL THE INSTITUTE OR
CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING
IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
POSSIBILITY OF
 * SUCH DAMAGE.
 *
 * This file is part of the Contiki operating system.
 *
 */

#include "contiki.h"
```

```
#include "lib/random.h"
#include "sys/ctimer.h"
#include "sys/etimer.h"
#include "net/uip.h"
#include "net/uip-ds6.h"
#include "net/uip-debug.h"

#include "simple-udp.h"
#include "servreg-hack.h"

#include "net/rpl/rpl.h"

#include <stdio.h>
#include <string.h>

#define UDP_PORT 1234
#define SERVICE_ID 190

#define SEND_INTERVAL		(10 * CLOCK_SECOND)
#define SEND_TIME		(random_rand() % (SEND_INTERVAL))

static struct simple_udp_connection unicast_connection;

/*---------------------------------------------------------------------------*/
PROCESS(unicast_receiver_process, "Unicast receiver example process");
AUTOSTART_PROCESSES(&unicast_receiver_process);
/*---------------------------------------------------------------------------*/
static void
receiver(struct simple_udp_connection *c,
         const uip_ipaddr_t *sender_addr,
         uint16_t sender_port,
         const uip_ipaddr_t *receiver_addr,
         uint16_t receiver_port,
         const uint8_t *data,
         uint16_t datalen)
{
  printf("Data received from ");
  uip_debug_ipaddr_print(sender_addr);
  printf(" on port %d from port %d with length %d: '%s'\n",
         receiver_port, sender_port, datalen, data);
}
/*---------------------------------------------------------------------------*/
static uip_ipaddr_t *
set_global_address(void)
{
  static uip_ipaddr_t ipaddr;
```

```
  int i;
  uint8_t state;

  uip_ip6addr(&ipaddr, 0xaaaa, 0, 0, 0, 0, 0, 0, 0);
  uip_ds6_set_addr_iid(&ipaddr, &uip_lladdr);
  uip_ds6_addr_add(&ipaddr, 0, ADDR_AUTOCONF);

  printf("IPv6 addresses: ");
  for(i = 0; i < UIP_DS6_ADDR_NB; i++) {
    state = uip_ds6_if.addr_list[i].state;
    if(uip_ds6_if.addr_list[i].isused &&
       (state == ADDR_TENTATIVE || state == ADDR_PREFERRED)) {
      uip_debug_ipaddr_print(&uip_ds6_if.addr_list[i].ipaddr);
      printf("\n");
    }
  }

  return &ipaddr;
}
/*---------------------------------------------------------------------------*/
static void
create_rpl_dag(uip_ipaddr_t *ipaddr)
{
  struct uip_ds6_addr *root_if;

  root_if = uip_ds6_addr_lookup(ipaddr);
  if(root_if != NULL) {
    rpl_dag_t *dag;
    uip_ipaddr_t prefix;

    rpl_set_root(RPL_DEFAULT_INSTANCE, ipaddr);
    dag = rpl_get_any_dag();
    uip_ip6addr(&prefix, 0xaaaa, 0, 0, 0, 0, 0, 0, 0);
    rpl_set_prefix(dag, &prefix, 64);
    PRINTF("created a new RPL dag\n");
  } else {
    PRINTF("failed to create a new RPL DAG\n");
  }
}
/*---------------------------------------------------------------------------*/
PROCESS_THREAD(unicast_receiver_process, ev, data)
{
  uip_ipaddr_t *ipaddr;

  PROCESS_BEGIN();
```

```
  servreg_hack_init();

  ipaddr = set_global_address();

  create_rpl_dag(ipaddr);

  servreg_hack_register(SERVICE_ID, ipaddr);

  simple_udp_register(&unicast_connection, UDP_PORT,
                      NULL, UDP_PORT, receiver);

  while(1) {
    PROCESS_WAIT_EVENT();
  }
  PROCESS_END();
}
/*---------------------------------------------------------------------------*/
```