Multimedia Computer Networks Quality of Service Techniques Evaluation and Development

Aboagela A. Abolghasm Dogman

A thesis submitted in partial fulfilment of the requirements of Sheffield Hallam University

for the degree of

Doctor of Philosophy

February 2014

ABSTRACT

Quality of Service Evaluation and Improvement of Multimedia Computer Networks

The growth in the transmission of time-sensitive applications over computer networks means that Quality of Service (QoS) needs to be managed in an efficient manner. Network QoS management in this thesis refers to evaluation and improvement of QoS provided by integrated wired and wireless computer networks. Evaluation of QoS aims to analyse and quantify network performance with respect of meeting multimedia applications' transmission requirements. QoS improvement involves the ability to take actions to change network performance toward improved operation. Therefore, the main aims of this thesis are: (i) to develop techniques for evaluation QoS in multimedia computer networks, (ii) to develop techniques that uses the information from (i) to manage and improve network performance.

Multimedia traffic generates a large amount of data. Collecting this information poses a challenge as it needs to be sufficiently fast and accurate. A contribution of this thesis is that adaptive statistical sampling techniques to sample multimedia traffic were developed and their effectiveness was evaluated. Three different adjustment mechanisms were incorporated into statistical sampling techniques to adjust the traffic sampling rate: simple linear adjustment, quarter adjustment, and Fuzzy Inference System (FIS). The findings indicated that the developed methods outperformed the conventional non-adaptive sampling methods of systematic, stratified and random.

The data collected included important QoS parameters, i.e. delay, jitter, throughput, and packet loss that indicated network performance in delivering real-time applications. An issue is that QoS needs evaluation in an informative manner. Therefore, the second contribution of this thesis is that statistical and Artificial Intelligent (AI) techniques were developed to evaluate QoS for multimedia applications. The application's QoS parameters were initially analysed either by Fuzzy C-Means (FCM) clustering algorithm or by Kohonen neural network. The analysed QoS parameters were then used as inputs to a regression model or Multi-Layer Perceptron (MLP) neural network in order to quantify the overall QoS. The proposed QoS evaluation system differentiated the network's QoS into a number of levels (Poor to Good QoS) and based on this information, the overall network's QoS was successfully quantified. In order to facilitate QoS assessment, a portable hand-held device for assessing the QoS in multimedia networks was designed, regression model was implemented on the microcontroller board and its performance was successfully demonstrated.

Multimedia applications transmitted over computer networks require a large bandwidth that is a critical issue especially in wireless networks. The challenge is to enable end-to-end QoS by providing different treatments for different classes of traffic and efficient use of network resources. In this thesis, a new QoS enhancement scheme for wireless-wired networks is developed. This scheme consisted of an adaptive traffic allocation algorithm that is incorporated into the network's wireless side to improve the performance of IEEE 802.11e Enhanced Distributed Channel Access (EDCA) protocol, and a Weighted Round Robin (WRR) queuing scheduling mechanism that was incorporated into the wired side. The proposed scheme improved the QoS for Multimedia applications. The average QoS for voice, and video applications were increased from their original values by 72.5%, and 70.3% respectively.

DEDICATION

I dedicate this work to:

- *To the soul of my mother, I still owe to her. Without her encouragement when she was a live, I could not be able to reach this stage.*
- *My father, for his invaluable support throughout the years, who devoted his life to the achievement of this dream.*
- *My lovely wife whose unconditional love makes everything possible.*
- *My sons Yousf, Ahmed, and Abdul-Aziz who make every day new and precious.*
- *My brothers, sisters, and friends who shared with me my dream.*

ACKNOWLEDGMENTS

In the name of Allah, the Most Gracious, the Most Merciful.

All thanks to Almighty ALLAH for giving me the guidance and the strength to achieve my aim of obtaining PhD degree.

First, I would like to express my deep thanks to my director of studies, Dr. Reza Saatchi, for his guidance, support, encouragement, valuable suggestions and comments on various aspects throughout my PhD study. The initial help I have received from Dr. Reza in setting up my research focus and his subsequent professional support throughout my research process were very valuable. *I really appreciate your support!*

Special thanks to my supervisor, Dr. Samir Al-Khayatt for being very helpful and supportive. His valuable guidance, advice, and expertise throughout my research were really helpful. Dr. Samir Al-Khayatt taught me how a supervisor can be a sincere friend. *I will never forget your help and support!*

I also wish to thank researchers, who have been, or still are, PhD candidates within the C^3RI as well as many of the departmental staff, our friendly discussions helped me gain confidence in my research.

LIST OF PUBLICATIONS

- Dogman, A., Saatchi, R., and Al-Khayatt, S. (2014), Collaborative Scheme to Improve QoS in WLAN-Wired Networks, The Mediterranean Journal of Computers and Networks, Accepted for publication.
- Dogman, A., and Saatchi, R. (2014), Multimedia Traffic Quality of Service Management Using Statistical and Artificial Intelligence Techniques, Institute of Engineering and Technology Journal of Circuits, Devices, and Systems, Accepted for publication.
- Dogman, A., Saatchi, R., and Al-Khayatt, S. (2013), Network Quality of Service Assessment Implementation Using a Microcontroller Board, Malaysian Journal of Fundamental and Applied Sciences, Vol.9, No.2, pp 57 - 61.
- Dogman, A., Saatchi, R., and Al-Khayatt, S. (2012), Quality of Service Evaluation using a Combination of Fuzzy C-Means and Regression Model, International Journal of Electronics and Electrical Engineering, World Academy of Science Engineering and Technology (WASET12), Vol. 6, pp. 58 - 65.
- Dogman, A., Saatchi, R., and Al-Khayatt, S. (2012), Evaluation of Computer Network Quality of Service Using Neural Networks, In Proceedings of IEEE Symposium on Business, Engineering and Industrial Applications (ISBEIA12), 23rd - 26th September, Bandung, Indonesia, UK, IEEE Xplore, pp. 217 - 222.
- Dogman, A., Saatchi, R., and Al-Khayatt, S. (2012), Computer Network Quality of Service Monitoring Using KEIL ARM Microcontroller, In Proceedings of Regional Annual Fundamental Science Symposium (RAFSS12), 10th - 13th December, Persada Johor International Convention Centre, Johor Bahru, Malaysia.
- Dogman, A., Saatchi, R., and Al-Khayatt, S. (2012), Improving Quality of Service in IEEE 802.11e Enhanced Distributed Channel Access Protocol, In Proceedings of the 8th IEEE International Symposium on Communication Systems Networks and Digital Signal Processing (CSNDSP12), 18th - 20th July, Poznan University, Poznan, Poland, UK, IEEE Xplore, pp. 1 - 6.
- Dogman, A., Saatchi, R., Nwaizu, H., and Al-Khayatt, S. (2011), Adaptive Statistical Sampling of VoIP Traffic in WLAN and Wired Networks using Fuzzy Inference System, In Proceedings of the 7th IEEE International Wireless

Communications and Mobile Computing Conference (IWCMC11), 4th - 8th July, Bahcesehir University, Istanbul, Turkey, UK, IEEE Xplore, pp. 1731 - 1736.

- Dogman, A., Saatchi, R., and Al-Khayatt, S. (2010), An Adaptive Statistical Sampling Technique for Computer Network Traffic, In Proceedings of the 7th IEEE, IET International Symposium on Communication Systems Networks and Digital Signal Processing (CSNDSP10), 21st -23rd July, University of Northumbria, Newcastle, UK, IEEE Xplore, pp. 479 - 483.
- Dogman, A., Saatchi, R., and Al-Khayatt, S. (2010), Evaluation of Adaptive Statistical Sampling versus Random Sampling for Video Traffic, In Proceedings of the 11th International Arab Conference on Information Technology (ACIT10), 14th-16th December, University of Benghazi, Benghazi, Libya.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

GLOSSARY TERMS

AC	Access Category
ACR	Absolute Category Rating
ADC	Analog to Digital Converter
AI	Artificial Intelligent
AIFS	Arbitration Inter Frame Space
AIFSN	Arbitration Inter Frame Space Number
ANN	Artificial Neural Network
AODV	Ad hoc On-Demand Distance Vector
AP	Access Point
CAC	Call Admission Control
CAN	Controller Area Network
CAP	Controlled Access Phase
CBR	Constant Bit Rate
CFP	Contention Free Period
CF-Poll	Contention Free Poll
CW_{max}	Maximum Contention Window
CW_{min}	Minimum Contention Window
DAC	Digital to Analog Converter
DCF	Distributed Coordination Function
DiffServ	Differentiated Service
DIFS	Distributed Inter Frame Space
DoS	Denial of Service
DSDV	Destination Sequenced Distance Vector
EDCA	Enhanced Distributed Channel Access
FCM	Fuzzy C- means
FIFO	First In First Out
FIS	Fuzzy Inference System
FLC	Fuzzy Logic Controller
FQ	Fair Queuing
FTP	File Transfer Protocol
GloMoSIM	Global Mobile Information System Simulator
GOP	Group Of Picture
GUI	Graphical User Interface
HC	Hybrid Coordinator
HCCA	HCF Controlled Channel Access
HCF	Hybrid Coordination Function
HID	Human Interface Devise
IC	Integrated Circuit
IDE	Integrated Development Environment
IDS	Intrusion Detection System
IEEE	Internet of Electrical and Electronic Engineering
IntServ	Integrated Service
IP	Internet Protocol
IPTV	Internet Protocol Television
ISSI	Inter-Sampling Section Interval

ITU	International Telecommunication Union
KSOM	Kernel SOM
LCD	liquid crystal display
LED	light-emitting diode
LSP	Label Switched Path
MAC	Medium Access Control
MANET	Mobile Ad hoc Network
MCB	Microcontroller Board
MDK	Microcontroller Development Kit
MLP	Multi- Layer Perceptron
MOS	Mean Opinion Score
MSC	Massage Storage Class
MSDU	MAC Service Data Unit
MSE	Mean Square Error
NAM	Network Animator
NS-2	Network Simulator -2
OMNET++	Optical Micro Networks Plus Plus
OPNET	Optimized Network Evaluation Tool
OSI	Open System Interconnection
OTcl	Object Tool Command Language
OWD	One Way Delay
PAR	Packet Arrival Rate
PCF	Point Coordination Function
PCM	Plus Code Modulation
PESQ	Perceptual Evaluation of Speech Quality
PIFS	PCF Inter Frame Space
PLCP	Physical Layer Convergence Procedure
PLR	Packet Loss Ratio
PQ	Priority Queuing
PSNR	Peak Signal to Noise Ratio
PSQM	Perceptual Speech Quality Measure
QoS	Quality of Service
RAM	Random Access Memory
RED	Random Early Detection
RNN	Random Neural Network
ROM	Read Only Memory
RSE	Relative Standard Error
RST	Request to Send
SD	Secure Digital
SI	Service Interval
SID	Sample Interval Difference
SIFS	Short Inter Frame Space
SOM	Self Organising Map
TCL	Tool Command Language
TCP	Transmission Control Protocol
TD	Transmission opportunity Duration
TORA	Temporally Ordered Routing Algorithm
TSW	Time Sliding Window
TXOP	Transmission Opportunity
UDP	User Datagram Protocol

USB	Universal Serial Bus
VBR	Variable Bit Rate
VoIP	Voice over Internet Protocol
VVoIP	Voice and Video over Internet Protocol
WFQ	Weighted Fair Queuing
WLAN	Wireless Local Area Network
WRR	Weighted Round Robin

LIST OF VARIABLES

Symbol	Definition	Equation
$T_i\,(t)$	Throughput in bit per second (bps) during i^{th} time interval	2.1
$P_i(t)$	No. of bits of all successfully received packets during i^{th} interval	
D_i	Delay of the i^{th} packet arrives at its final destination	2.2
R_i	Sending timestamp of the i^{th} packet.	
S_i	Arrival timestamp of i^{th} packet.	
J_i	Jitter of i^{th} packet	2.4
PL_i	Packet Loss ratio in percentage (%) during i^{th} timeinterval	2.6
W_i	Associated weight for queue (i) in weithed round robin	2.8
R	Link capacity	
y	Dependent variable in regression model formula	2.9
$x_1, x_2, \ldots., x_n$	No. n of Independent variables in regression model formula	
$b_0, b_1, b_2, \ldots b_n$	No. n of regression coefficients determined from independent variables in regression model formula	
e	A column vector of n error terms in regression model formula	
μ_X	Degrees of membership functions for fuzzy sets X	2.14
μ_Y	Degrees of membership functions for fuzzy sets Y	
X	In FCM, is matrix of size $n \times N$ representing a given set of feature data	2.16
U	Membership matrix of size $n \times C$ generated by FCM	
V	Matrix of clusters' centres generated by FCM	
C	No. of clusters generated by FCM	
m	Controls the degree of fuzziness for the membership of the cluster	
μ_{ij}	Degree of membership between x_j to the centre v_i of cluster i	
D_{ij}^2	Euclidian distance between x_j to the centre v_i of cluster i	
s	Resulting value of summation function of MLP	2.21
x_i	Input i to MLP	
w_i	Associated neuron connection weight of input i	
y	The output of activation function $\varphi\,(s)$ of MLP	2.22
d	Desired output of training example	
e	Calculated error between d and y	
x_i	Input i to SOM	2.23
w_{ij}	Associated connection weight between input i and neuron j	
$w_{ij}\,(n)$	Current weight of winning neuron	2.24
$w_{ij}\,(n+1)$	Updated weight of winning neuron	
η	Learning rate	
mean1	Mean value of pre- sampling section for throughput of traffic being sampled	5.1

mean2	Mean value of post- sampling section for throughput of traffic being sampled	
median1	Median value of pre- sampling section for throughput of traffic being sampled	
median2	Median value of post- sampling section for throughput of traffic being sampled	5.1
std1	Standard deviation value of pre- sampling section for throughput of traffic being sampled	
std2	Standard deviation value of post- sampling section for throughput of traffic being sampled	
Updated ISSI	Updated length of Inter- Sampling Section Interval	5.2
$\mu 1$	Update the length of ISSI by increasing it linearly	
$\mu 2$	Update the length of ISSI by decreasing it linearly	5.3
μ	Update the length of ISSI in quarter adjustment mechanism	5.4 - 5.5
$\mu_{A^i}(x)$	Gaussian membership of value (x) of i^{th} fuzzy set A^i	
C_i	The mean of the i^{th} fuzzy set A^i	5.6
σ_i	Standard deviation of the i^{th} fuzzy set A^i	
Y	Deffuzzification output using centroid method	
y_i	Centroid of fuzzy region i	
μ_i	Output membership value	5.7
m	Number of fuzzy sets after implication process	
SID	Sample Interval Difference	5.8
Bias	Shows the difference between mean of sampled version of QoS parameter and the mean of its original population	
M_i	Mean of QoS parameters for the sampled version	
M	Mean of QoS parameters for original population	5.9
N	Number of simulation runs	
RES	Relative Standard Error	
SE	Standard Error of sampled version	5.10
n	Sample size	

Chapter 1 Introduction

Quality of Service (QoS) management is currently one of the principle technological fields of development in computer networks. Computer networks are increasingly integrated and carry a diverse set of traffic such as Voice over Internet Protocol (VoIP), video streaming, video conferencing, and traditional data. The interconnection of wired and wireless networks and the rapid growth of real-time and non real-time applications transmitted over these networks have made QoS management an area that requires further research and development. QoS management of these networks is important to both users as well as the network service providers. Users are interested in determining how well they receive applications. The network service providers and network managers need QoS information to determine how well their networks are preforming. Therefore, the main focus of the study is to develop mechanisms associated with QoS management processes for multimedia computer networks.

The purpose of this chapter is to: clarify the rationales behind this research which are presented in section 1.1, outline the aim and objectives in section 1.2, summarise the contribution of this research in section 1.3, and present the outline and organisation of this thesis in section 1.4.

1.1 Research Motivations

In this study, network QoS management refers to evaluation and improvement of QoS in wired and wireless computer networks. Evaluation of QoS aims to analyse and quantify a network performance with respect to meet the applications' transmission requirements. QoS improvement involves the ability to take actions to enhance network performance. However, there are complexities associated with realising QoS management and so the area is an important field of research. The following points summarise the QoS management issues considered in this study:

(i) Computer networks carry different traffic types containing video, audio and data. Analysis and interpretation of all traffic packets can be time consuming. Also, collecting all packets poses a challenge as the process needs to be sufficiently fast, should not load the network and interfere with the operation of the protocols responsible for managing the network. Therefore, ways need to be found to accurately sample packets and present them to the network management entities responsible for their interpretation and decision making. Most of current sampling techniques use a predetermined and fixed sampling rate irrespective of the extent of traffic fluctuation with time. The sampling therefore is not the optimal as the multimedia traffic is time varying. In a fixed rate sampling method, two situations could occur: (a) If the traffic fluctuation (i.e. frequency of variation) is high, there is a risk of losing important information (caused by under sampling). (b) If the traffic fluctuation is low, resources would be under utilised (by over sampling) (Giertl et al, 2006). In order to reduce the biasness presented in conventional fixed rate sampling and to enhance the process of gathering Quality of Service (QoS) information, traffic should be sampled adaptively. In other words, a small sample interval is required during the period of high activity, whereas a larger sample interval is required during the period of low traffic fluctuation.

(ii) Gathered network information is indicative of its performance in delivering real-time and non-real-time applications. The data collected include parameters related to QoS such as delay, jitter, throughput, and packet loss. These parameters need evaluating in an informative and effective manner. The evaluation of QoS is currently carried out either by analysis or measurement techniques. The analysis techniques are used to examine the characteristics of the traffic (Chen et al, 2009) and (Timo et al, 2002), whereas the measurement techniques are applied to determine how well the network treats the ongoing traffic (Palomar et al, 2008), (Teyeb et al, 2006), (Cranley and Davis, 2005), and (Mishra and Sharma, 2003). The current state-of-the-art of QoS analysis and measurement approaches have several limitations. For example, they are not combined to form a mechanism to evaluate QoS from an analysis and measurement point of view. Moreover, current QoS measurement techniques can generate an excessive traffic load that affect the operation of the network as in active measurement approach or they perform by measuring the actual network traffic that requires collecting and processing a large amount of recorded data packets in order to provide an indication of network performance as in passive approaches (Brekne et al, 2002). Subjective and objective QoS measurement approaches have also some limitations. The former approach cannot be automated since it requires a controlled environment, and it is time consuming to be repeated frequently due to its dependence on human subjects (Palomar et al, 2008). Objective approaches are computationally intensive because their operations are at the pixel level and they cannot take into the account all the affected network parameters (Mohammed et al, 2001). These limitations highlight the need for further development in evaluation of QoS. In order to evaluate QoS effectively, the proposed methods need to combine analysis and measurement techniques. The methods require evaluation of QoS in a manner similar to human subjects and quantify the QoS without the necessity for complex mathematical models, taking into the account the QoS requirements of every type of multimedia applications. In addition, the methods should not add a significant extra load to the network as is the case with active approaches, nor should it depend on collection of all traffic packets.

(iii) Delivery of multimedia applications is still with limitations, especially for wireless networks. A major bottleneck in transmission of real-time multimedia applications is insufficient channel bandwidth. Accordingly, QoS could be unpredictable. Therefore, deployment of network QoS enhancement techniques can be beneficial for multimedia transmission. However, most previous studies considered QoS support either in wireless local area networks (WLANs) or in wired networks. There are few studies to enable end-to-end QoS in wired and wireless networks (Skyrianoglou et al, 2002), (Park et al, 2003), and (Senkindu and Chan2008). The limitations of these studies were the inclusion of an intermediate layer between the MAC and IP layers in wireless stations, which in turn added more complexity in managing the wireless side of the network, or the low priority traffic was starved due to link congestions and QoS prioritisation. The challenge to enable end-to-end QoS is the inclusion of both parts of the network (i.e. wired and wireless) to provide different treatments for different classes of traffic and efficient use of network resources.

(iv) The existing QoS monitoring techniques are limited in scope. They obtain overall network QoS indirectly and are not stand-alone. For example, the QoS monitoring tool proposed by Graham et al (1998) was used to assess packet latency and loss as an indication of network performance. The Surveyor tool proposed by Zseby and Scheiner (2004) to assess end-to-end delay and packet loss required a further analysis using a server to measure the network performance. The disadvantage of these tools is that network managers have to do a variety of operations to assess the overall network QoS. From these limitations, it can be obvious that the process of monitoring QoS can be complicated, expensive, and time consuming.

Therefore, developing a portable hand-held device that accurately determines the overall network QoS for multimedia applications can be very valuable. In this study, a mechanism that assesses QoS by taking into the account the requirements of multimedia applications is implemented on a portable microprocessor board.

1.2 Research Aim and Objectives

The focus in this research is on network QoS management which entails evaluation and improvement of QoS. This raises the following research question: to what extent, the QoS evaluation and improvement can be interrelated, and do they require different mechanisms to be integrated.

The overall aim of this study is to develop techniques to evaluate QoS in multimedia networks and to use this information as part of a network management process to improve its performance. The study uses IEEE 802.11e standard for the wireless side of the network. The wireless is connected to a wired network. The objectives of the study are to:

i. Develop approaches that allow QoS parameters of multimedia applications to be collected efficiently and accurately.
ii. Develop techniques to analyse and interpret QoS parameters of multimedia networks.
iii. Develop techniques to allow the QoS information to be used as part of network management to improve its performance.
iv. Develop an electronic portable device that facilitates accurate QoS assessment of multimedia networks.
v. Critically evaluate the developed techniques to determine their effectiveness and accuracy.

This study will involve multimedia type networks that integrate both wired and wireless parts.

1.3 Research Contributions

Improvement in managing QoS of multimedia applications is essential, particularly when dealing with hybrid wired and wireless structures. The techniques proposed in this study contribute to extending the knowledge in QoS management. This section outlines the contributions of this research in line with its objectives. The research contributions of this study are included in relevant chapters. These contributions are outlined below.

(i) *Develop approaches that allow QoS parameters of multimedia applications to be collected efficiently and accurately.* In order to facilitate effective traffic data gathering, novel statistical adaptive sampling techniques are developed that utilised traffic's statistical features. These techniques adjust the sampling interval by considering the traffic's statistical features between two consecutive sampled sections using: quarter adjustment approach (Dogman, et al., 2010_a), simple linear adjustment approach (Dogman, et al., 2010_b) and fuzzy inference system (Dogman, et al., 2011). Chapter 5 of this thesis introduces the adaptive statistical sampling techniques which were based on three adjustment mechanisms: quarter adjustment mechanisms, simple linear adjustment mechanisms, and fuzzy inference system. Also, a comparison of the devised methods versus conventional non-adaptive sampling techniques (i.e. systematic sampling, stratified sampling, and random sampling) was carried out to validate the effectiveness of proposed sampling techniques.

(ii) *Develop techniques to analyse and assess QoS parameters of multimedia networks.* The contribution of this study is development of mechanisms that combines analysis and measurement techniques to evaluate QoS of multimedia applications in an effective manner. Two innovative QoS evaluation systems are proposed based on Artificial Intelligence (AI) and traditional techniques. The latter combines Fuzzy C-Means (FCM) and the regression model to analyse and assess the QoS of multimedia applications (Dogman, et al., 2012_a). The former system analyses and assesses the QoS of multimedia applications based on a combination of supervised and unsupervised neural networks (Dogman, et al., 2012_b). The proposed QoS evaluation systems are introduced in Chapter 6. The transmitted application's QoS parameters are initially analysed either by the FCM clustering algorithm or by the unsupervised learning Kohonen neural network (i.e. Self Organising Map (SOM)). The analysed QoS parameters are then used as inputs to a regression model or supervised learning Multi-Layer Perceptron (MLP) neural network in order to quantify the overall QoS. The proposed QoS evaluation systems provided relevant information about the network's QoS patterns and based on them, the overall network's QoS was successfully quantified. The process of evaluating QoS is explained in details in Chapter 6.

(iii) *Develop techniques to allow the QoS information to be used as part of network management to improve its performance.* A scheme that improves QoS in both wired and wireless domains of the network is proposed in (Dogman, et al., 2012_c). A description of the proposed QoS enhancement scheme is provided in Chapter 7. The scheme uses a combination of MAC layer QoS control in the wireless domains and network layer QoS control in the wired domains. A novel aspect of the scheme is that an adaptive access category (AC) traffic allocation is devised and incorporated into wireless access point (AP) in order to improve the QoS of IEEE 802.11e Enhanced Distributed Channel Access (EDCA) protocol, and Weighted Round Robin (WRR) queuing scheduling mechanism is implemented into congestion point (i.e. router) in wired networks to support fair distribution of bandwidth among different traffic classes. The adaptive traffic allocation algorithm determines the packet arrival rate (PAR) of the up- and down-link traffic for each access category (AC). It then dynamically allocates traffic of a lower priority AC to the next higher AC, when the higher AC is not receiving traffic at the time. Whereas, WRR shares the network resources, based on the traffic's QoS requirements. The results explained in Chapter 7 show the improvement in QoS provided by wired and wireless sides when the QoS enhancement scheme was implemented.

(iv) *Develop an electronic portable device that facilitates accurate QoS assessment of multimedia networks.* A hardware QoS monitoring system is designed (Dogman, et al., 2013). The system is used regression modelling, implemented on the MCB2300 KEIL ARM microcontroller board. More details about QoS monitoring system are provided in Chapter 8. The hardware QoS monitoring system analysed the QoS requirements (i.e. delay, jitter and packet loss ratio) of multimedia applications to determine their overall QoS. The evaluation of QoS monitoring system was carried out by comparing the results obtained with other QoS assessment methods to indicate the effectiveness of the developed system in monitoring multimedia QoS accurately.

1.4 Thesis Organisation

Figure 1-1 shows the schematic overview of the thesis. In addition to the introduction chapter, which outlines the rationales for this study, study's aim and objectives, and a summary of the contribution of the research to QoS management area are explained. There are eight further chapters in this thesis.

Chapter 2 covers the theoretical background essential for this research. This includes the definitions of QoS and its parameters, QoS requirements of multimedia applications, the service levels of QoS, and QoS components. Chapter 2 also provides a detailed description of IEEE 802.11e as an emerging WLANs standard to provide QoS, and packet scheduling mechanisms as the most commonly mechanisms implemented in wired network to support QoS. In addition to that, the theories of statistical and AI techniques used in this study are provided. Regression model as one of the most widely employed statistical analysis methods is described. The fundamental principles of three important paradigms in AI system: fuzzy logic, fuzzy clustering, and neural network with their operational steps are also discussed.

In Chapter 3, the previous studies relevant to the aforementioned issues of managing the QoS in multimedia networks in section 1.1 are critically analysed and discussed. The aim is to identify the potential limitations associated with these issues which in turn are further developed. Chapter 3 is organised to review the sampling techniques used to gather information from network traffic, and discuss the state-of-the-art of QoS analysis and assessment techniques used to evaluate network QoS. The relevant studies considering QoS support in the context of wireless-wired networks are also analysed, and the exiting monitoring tools used to assess the network performance are reviewed. Moreover, the applications of statistical and AI techniques used in this study into the field of network QoS management are reviewed in that chapter.

Chapter 4 provides an explanation of network evaluation approaches, network simulation tools, and the general experimental procedure used throughout this study. The description of Network Simulation 2 (NS-2) environment, transmission protocols, queuing mechanisms, and traffic type, and its characteristics are included. The measurement process, which includes a description of the QoS metrics and requirements, and the procedure for analysis simulation output are also provided.

The main results of this research are provided in Chapters 5-8. In Chapter 5, statistical adaptive sampling techniques to adjust sampling rate based on traffic's statistics are introduced. A detailed description of the proposed adaptive statistical sampling techniques which are based on three adjustment mechanisms: quarter adjustment mechanisms, simple linear adjustment mechanisms, and fuzzy inference system is provided. An implementation of conventional sampling techniques (i.e. systematic sampling, stratified sampling, and random sampling) is explained. The experimental results in Chapter 5 show the effectiveness of statistical sampling techniques by carrying out a comparison of devised statistical sampling technique versus the conventional sampling techniques using a simulated computer network.

Chapter 6 introduces the use of statistical and AI techniques in the area of QoS management. Two innovative QoS evaluation approaches are explained. The first approach combines Fuzzy C-Means (FCM) and regression model to analyse and assess QoS of multimedia applications in a simulated network, whereas the other approach analyses and assesses QoS in multimedia applications using a combination of supervised and unsupervised neural networks. The transmitted application's QoS parameters are initially analysed either by FCM clustering algorithm or by the unsupervised learning Kohonen neural network (i.e. Self-Organising Maps (SOM)). The analysed QoS parameters are then used as inputs to a regression model or supervised learning Multi-Layer Perceptron (MLP) neural network in order to quantify the overall QoS. Results in Chapter 6 show how the proposed QoS evaluation systems provide information about the network's QoS patterns and based on this information, how the overall network's QoS is quantified.

In Chapter 7, development of a new Quality of Service (QoS) enhancement scheme for WLAN-wired networks is introduced and its performance is evaluated. The proposed scheme consists of an adaptive Access Category (AC) traffic allocation algorithm that is incorporated into the network's wireless side to improve the performance of IEEE 802.11e Enhanced Distributed Channel Access (EDCA) protocol, and a Weighted Round Robin (WRR) queuing scheduling mechanism that is incorporated into the wired side of the network. A description of the proposed QoS enhancement scheme is provided in this Chapter. The results discussed in Chapter 7 show the efficiency of proposed enhancement scheme. The scheme provides an end-to-end QoS to be setup which in turn provides an improved delivery of a variety of applications in the context of wired-cum-wireless networks.

In Chapter 8, a network QoS monitoring system is designed and evaluated. The proposed monitoring system incorporates the QoS assessment approach developed by (Dogman et al, 2012_a) that is based on regression model. The microcontroller board MCB2300 KEIL ARM is used. Chapter 8 explains the MCB2300 KEIL ARM microcontroller board, and outlines how the QoS assessment technique using regression modelling is devised, and implemented on the MCB2300 KEIL ARM microcontroller board. The performance of QoS monitoring system is compared with other QoS assessment methods (e.g. QoS assessment using Fuzzy Inference System introduced by (Al-Sbou et al, 2006), and Neural Network QoS monitoring approach proposed by (Dogman et al, 2012_b). The results indicated that the developed system is capable of accurately assessing QoS.

Chapter 9 discusses the overall findings of this research, provides the conclusions, and highlights future research directions.

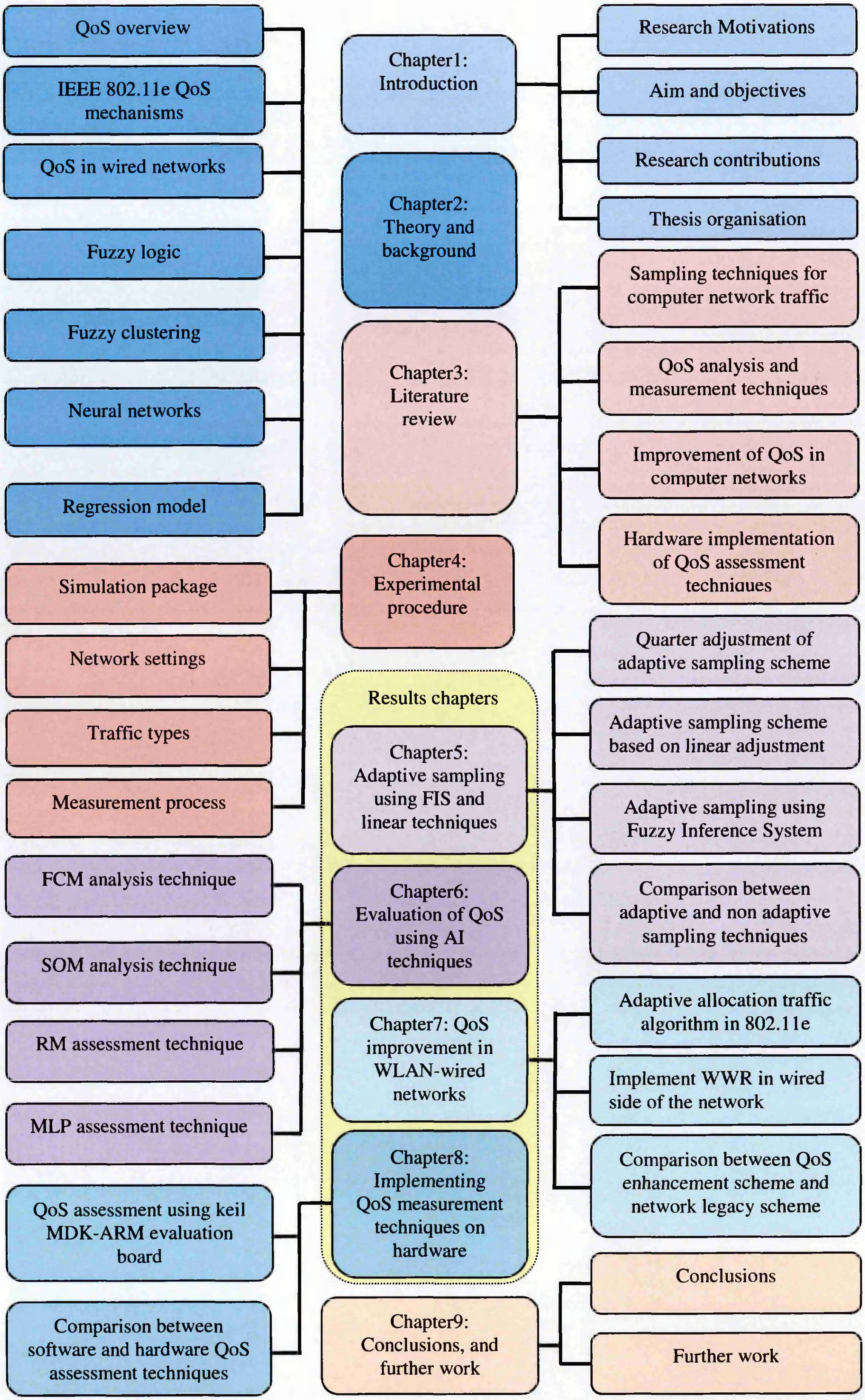


Figure 1-1. The schematic overview of the thesis.

Chapter 2 Relevant Theory and Background

2.1 Introduction

The aim of this chapter is to provide the background related to the main issues of this study. Section 2.2 of this chapter gives an overview of QoS in multimedia networks. This includes definitions of QoS and its parameters, QoS requirements of multimedia applications, the service levels required of QoS, and QoS components. Section 2.3 discusses the QoS in wireless and wired networks. This section includes a detailed description of IEEE 802.11e as an emerging WLAN standard to provide QoS, and packet scheduling mechanisms as the most commonly used mechanisms implemented in wired network to support QoS. Section 2.4 introduces the theory of statistical and AI techniques used in this study. This includes the basic concepts of regression analysis as one of the most widely employed statistical modelling methods and the fundamental principles of three important paradigms in AI system: fuzzy logic, fuzzy clustering, and neural network.

2.2 Quality of Service (QoS): An overview

Wired and wireless networks are becoming increasingly integrated. This is coupled with the rapid growth of real-time and non-real-time applications transmitted over these networks. These developments necessitated a greater emphasis on QoS of networks. As its aim is to provide guaranteed services for different applications, QoS is currently one of the principle research topics in computer network.

2.2.1 Definition of QoS:

In the world of telecommunications, each of the terms: "quality" and "service" has its own definition. The latter term means the ability of *network* as specified by the *service provider* to transmit information to the *end user,* whereas the term "quality" is used to assess the ability of the service whether it satisfies the stated and implied needs (Gozdecki et al, 2003). The interrelation between the two terms and the importance of their associations with three different parts: service provider, network manager, and the end user have produced the concept of QoS. Therefore, QoS could have a number of definitions. From the user point of view: QoS is "the collective effect of service performance, which determines the degree of satisfaction of the user of the service" (ITU, 1994). While from the technical aspects, QoS can be defined as the ability of network to provide a better service to selected traffic over a variety of technologies. QoS is thus "a collection of technologies, which allow network-aware applications to request and receive predictable service levels in terms of data throughput capacity (bandwidth), latency variations (jitter) or propagation latency (delay)" (Saliba et al, 2005). From service provider perspective, QoS is "a set of service requirements to be met by the network, while transporting a flow." (Crawley et al, 1998). In this study, the term QoS refers to the ability of network to assess and provide desired QoS requirements in terms of delay, jitter, throughput, and packet loss for transmitted multimedia applications.

2.2.2 QoS Parameters

In packet-switched networks, there are a number of QoS parameters that can be assessed and measured to determine QoS. These parameters can express how well the network treats packets during their journey from the source, throughout the network, and finally to their destinations. However, as different applications require different QoS parameters, a correct set of QoS parameters for a particular application being transmitted should be determined in order to effectively evaluate the QoS provided by the network for that particular application (Cheong and Lai, 1999). For instance, time sensitive applications such as video and Voice over IP (VoIP) are susceptible to transmission parameters such as packet delay, jitter, loss and throughput whereas time insensitive operations such as a file transfer are more sensitive to packet loss, but can tolerate transmission jitter (Kurose and Ross, 2005). In this study, multimedia applications will be considered. Therefore, the most important QoS parameters would be examined and quantified are throughput, delay, jitter, and packet loss ratio. The following subsections explain these parameters.

2.2.2.1 Throughput

Throughput is used to assess the capability of network to transmit data over a given period of time (Heckmann et al, 2002). It could be defined as the maximum transmission rate of packets that can be sustained between two endpoints. Wang et al (2000) defined the throughput as the amount of successfully received packets in a predefined time. Equation (2.1) is used to calculate the throughput:

$$T_i\ (t) = \frac{\sum P_i\,(t)}{t_i} \tag{2.1}$$

Where T_i is the measured throughput in bit per second (bps) during the i^{th} interval, $\sum P_i(t)$ is the total bits of all successfully received packets during the i^{th} interval, and t_i is the time duration of the i^{th} interval.

2.2.2.2 Delay

Delay is defined as the elapsed time for a packet to travel from its source to its destination. It can be also defined as the total amount of time that the packet takes to be sent from its source, through the network, until it reaches its destination (Heckmann et al, 2002). However, as the packet travels from its source to its destination, it suffers from various types of delay such as queuing delay, transmission delay, and processing delay. A detailed explanation of these types of delay can be found in (Kurose and Ross, 2005). According to Wang et al (2000), the delay can be calculated using equation (2.2):

$$D_{i=}\ R_i - S_i \tag{2.2}$$

Where D_i is the delay of the i^{th} packet arrived, R_i and S_i are the arrival and sending timestamps of the i^{th} packet. The average delay can be calculated using equation (2.3):

$$Average\ delay = \frac{1}{n}\sum_{i=1}^{n}\ D_i \tag{2.3}$$

Most of real-time applications are delay sensitive. For example, video conferencing and voice over IP (VoIP) have high sensitivity to delay because the transmitted packets need to be replayed back at the receiver in real time (Nortal Networks, 2003).

2.2.2.3 Jitter

Jitter is defined as the variation of delay between two consecutive packets for a given traffic flow (Nortal Networks, 2003). Jitter has a significant effect on real time applications (video and audio) because they require packets to be received with a fixed delay. For example, VoIP requires packets to be arrived at a constant rate, since the received packets will be played back at real time. Therefore, a small amount of jitter might be acceptable but when the jitter increases, its effect becomes obvious and might lead to a stuttering communication with pops and clicks (Heckmann et al, 2002). Wang et al (2000) pointed out that jitter can be calculated as the difference between the delay of two consecutive packets using equation (2.4):

$$J_i = |D_i - D_{i-1}| \quad while\ i>0 \tag{2.4}$$

Where J_i is the jitter of the i^{th} packet, D_i is the delay of packet i, and D_{i-1} is the delay of the previous packet. Average jitter can be calculated as in equation (2.5):

$$Average\ jitter = \frac{1}{n}\sum_{i=1}^{n} J_i \tag{2.5}$$

2.2.2.4 Packet Loss Ratio

Packets can be lost during their journey from source to destination. Packet loss occurs due to many effects such as inadequate physical transmission medium, collisions between packets, queuing overflow, and hardware failure (Nortal Networks, 2003). The effect of packet loss leads to "distortion, which results in a stuttering and snatch communication" (Heckmann et al, 2002). Packet loss can be defined as the percentage of transmitted packets that failed to reach their destinations as given by (Wang et al, 2000).

$$PL_i(t) = 100 \times \left(1 - \frac{\sum R_i(t)}{\sum S_i(t)}\right) \tag{2.6}$$

Where PL_i is the loss ratio in percentage (%) during the i^{th} interval, and $\sum R_i(t)$ and $\sum S_i(t)$ are the total number of received and transmitted packets with the i^{th} interval respectively.

2.2.3 QoS Requirements of Multimedia Applications

The QoS requirements of multimedia applications are significantly different from traditional applications. The latter applications such as email, web browsing, and file transfer can be elastic with some QoS parameters such as delay and jitter. However, the former applications (i.e. multimedia applications) such as video conferencing and VoIP have high sensitivity to QoS parameters and require a faster response from the network. A large delay or jitter can seriously degrade their quality (Kurose and Ross, 2005). The allocation of bandwidth usage for these applications can be also challenging to calculate. This is because of the number of different variables, such as codec usage, resolution, and transmission activity levels. In a computer network, some factors constrain realising an acceptable QoS. For instance, network congestion in wired networks and interference problems in wireless networks are critical issues to provide service guarantee for transmitted multimedia applications. Therefore, the acceptable range of QoS parameters for multimedia applications must be identified. Table (2-1) indicates the sensitivity of some common applications to QoS parameters (Zhai et al, 2005) and (ITU-T, 2001).

Table 2-1. The sensitivity of some common applications to their QoS parameters.

Class	Application	Typical bandwidth	Delay	Jitter	Packet loss ratio
Real-time	VoIP	16-128 kbps	< 150 ms preferred	< 1ms preferred	< 3 % preferred
	Video	16-384kbps	< 150 ms preferred	< 30ms preferred	< 1 % preferred
Non real-time	E-mail, file transfer, web browsing		Minutes	N/A	Zero

2.2.4 Service Levels of QoS

The service levels of QoS can be defined as the actual capability provided by a network to deliver service required by specific application. Computer network could provide three main levels of QoS agreements: best effort service (Floyd and Allman, 2008), Differentiated Service (DiffServ) (Black et al, 1998), and Integrated service (IntServ) (Braden et al, 1994). In the best effort service, the network QoS is unspecified. The packet delivery rate varies depending on the current network load. In this case, the network does not provide any guarantees or priority to the transmitted applications although the diversity of their QoS requirements. The DiffServ architecture provides a service differentiation mechanism to manage network resources based on QoS requirements of transmitted applications. In DiffServ, the transmitted applications are aggregated to a number of classes to receive a particular level of service. In IntServ architecture, end-to-end QoS guarantee per flow is provided. A flow requiring QoS guarantees must acquire sufficient network resources during its transmission to ensure its QoS requirements are met. This is achieved by a reservation of network resources for a specific flow (Shah, 2001) and (Kurose and Ross, 2005).

2.2.5 QoS Components

There are several components of QoS. These include: QoS mapping (Al-Kuwaiti et al, 2007), admission control (Georgoulas et al, 2004), traffic shaping (Lekcharoen, 2007), policing (Koutsakis, 2009), packet queue scheduling (Yu and Meng, 2009), and prioritisation (Senkindu and Chan, 2008).

The conversion of QoS representation between OSI model layers is referred to QoS mapping. In admission control, the network take a set of actions during traffic establishment phase to check if that particular traffic can be admitted or should be rejected. A new traffic is admitted to the network when its desired QoS can be satisfied, without causing any QoS violation to the already established services. The function of traffic shaping mechanism is to ensure that there is a smooth data rate in order to meet the specified QoS agreement. Traffic policing is used to monitor the transmitted traffic by discarding or remarking traffic that exceed limits in order to protect the network from malicious behaviour. Packet queue scheduling determines bandwidth allocation among transmitted packets and the manner to service various applications with different QoS requirements. In a prioritisation scheme, applications are classified based on their QoS requirements and resources are assigned according to classes of priority depending on network resource availability.

2.3 QoS in Wireless and Wired Networks

The QoS mechanisms in WLANs and wired networks are different. The QoS in WLANs is enabled at MAC layer, whereas the most wired networks enabled QoS at IP layer. This section discusses the QoS in WLANs and wired networks. A detailed descriptions of IEEE 802.11e as an emerging WLANs standard to provide QoS, and packet scheduling mechanisms as the most common mechanisms implemented in wired network to support QoS are provided.

2.3.1 QoS in Wireless Networks

WLANs have grown rapidly over the last decade, offering a range of practical and beneficial services for both home users and businesses. This growth relates to the fact that WLANs allow users to communicate without using physical medium. Also, the cost of installing WLANs is lower than wired networks because WLANs can be expanded by just adding access points (AP) rather than installing cables as in wired networks. Another influential factor in the growth of WLANs is the emergence the of IEEE 802.11 standard in 1997 and its subsequent amendments (1999 - 2005). Its cost effectiveness, ease of deployment, and mobility support made IEEE 802.11 WLANs to be used widely and became the dominating WLAN technology. IEEE 802.11 has reached an unprecedented maturity in providing ever-growing bit rates (Lin et al, 2009). However, with an emergence of multimedia communications over WLAN, the provision of QoS in WLANs that is capable of guaranteeing QoS requirements of multimedia applications becomes important. The demand for supporting QoS for various applications with different QoS requirements has led to the development of a WLAN standard so called IEEE 802.11e.

2.3.1.1 IEEE 802.11e Standard

The original standard of IEEE 802.11 was not designed to support QoS (IEEE Computer Society, 1999). In the legacy IEEE 802.11, applications with different QoS requirements were treated the same, and the service differentiation in terms of guaranteed bandwidth, bounded delay, and jitter for particular applications was disregarded. Due to the growth transmission of time-sensitive and time-insensitive applications over WLANs, the demand for supporting QoS for various applications has increased. Thus, the enhancement version of IEEE 802.11 standard called IEEE 802.11e was proposed to provide QoS support for applications with different QoS requirements (IEEE Computer Society, 2005). The IEEE 802.11e introduces Hybrid Coordination Function (HCF) which defines two medium access mechanisms: HCF Controlled Channel Access (HCCA) and Enhanced Distributed Channel Access (EDCA).

2.3.1.1.1 HCF Controlled Channel Access (HCCA)

HCCA is an enhanced version of Point Coordination Function (PCF) in the legacy IEEE 802.11. Similar to PCF, HCCA has centralised access scheme which uses Hybrid Coordinator (HC) implemented in the AP to access wireless medium, transmit at the Contention Free Period (CFP), and the QoS Contention Free (CF-Poll) frame is used to schedule the uplink traffic. However, HCCA has new operation parameters. These include: Service Interval (SI), Controlled Access Phase (CAP), and Transmission Opportunity (TXOP). SI is an interval between two successive TXOPs. CAP is a time period when the HC maintains control over the medium after gaining medium access by sensing the channel to be idle for the PCF Inter Frame Space (PIFS) period. TXOP is time period where stations can transmit a number of MAC Service Data Unites (MSDUs). Figure 2-1 shows the operation parameters of HCCA (Lee et al, 2011).

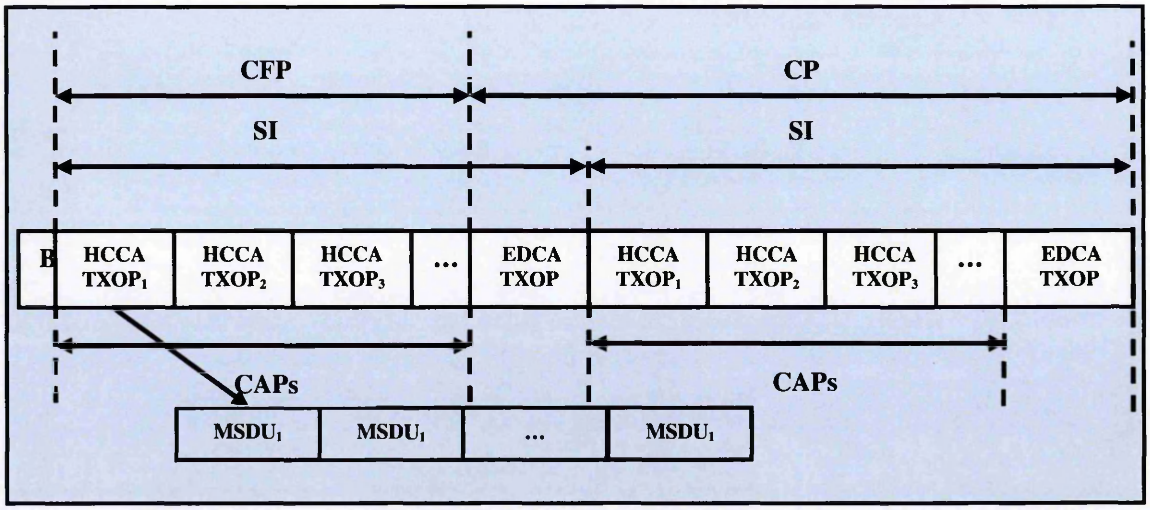


Figure 2-1. The construction of IEEE 802.11e HCCA.

In HCCA, each wireless station receives QoS CF-Poll frame from the HC before transmitting data. While the medium is idle for PIFS, the HC can start a CAP by sending the QoS CF-Poll frame to the station that is requesting to transmit data in order to inform that particular station of the time allocated for its transmission. The station should reply to this poll within a time interval equal to Short Inter Frame Space (SIFS). For downlink transmission, the HC waits for PIFS and then start its transmission. In HCCA, the HC has higher priority than other wireless stations. Thus, MSDUs for downlink traffic can be transferred faster than uplink traffic (Villalón et al, 2007). Figure 2-2 shows the operation of HCCA for uplink and downlink transmission (Lee et al, 2011).

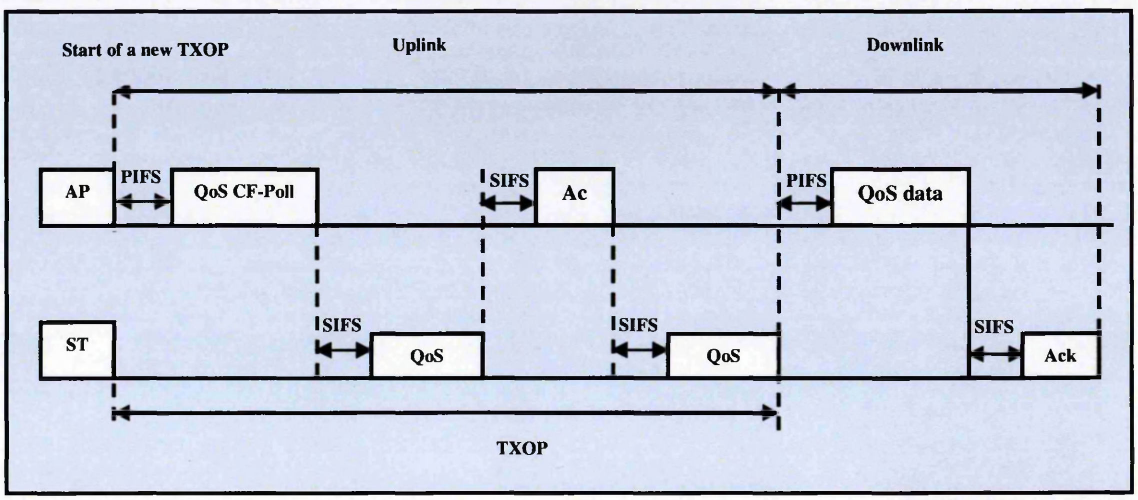


Figure 2-2. Uplink and downlink transmission between AP and wireless station.

A limitation of HCCA operation is its adaption to traffic rate. The allocation of time resources to the flows must consider the calculation of scheduling parameters such as

SI, and TXOP duration (TD) as these parameters significantly affect the performance of the scheduler. Another limitation of HCCA is the maximum TD which was set to 8160 μS. This value is an acceptable duration if the traffic rate is relatively small. However, a sudden increase in traffic rate can exceed the limit of TXOP which in turn affects the SI to be set to the optimal value. The computational complexity of HCCA and its overhead can deteriorate the performance of high priority traffic in heavily loaded networks (Rashid et al, 2007) and (He and Ma, 2011). Due to its limitations, the HCCA mechanism has not been implemented widely. Thus, the investigation of HCCA mechanism will be excluded from this study.

2.3.1.1.2 Enhanced Distributed Channel Access (EDCA)

The IEEE 802.11e EDCA is designed to enhance Distributed Coordination Function (DCF) in the legacy IEEE 802.11 standard. EDCA is based on a distributed access scheme and supports service differentiation among different traffic classes (IEEE Computer Society, 2005). The IEEE 802.11e EDCA mechanism classifies the traffic into four access categories (ACs) based on their QoS requirements as depicted in Figure 2-3 (Liang et al, 2006).

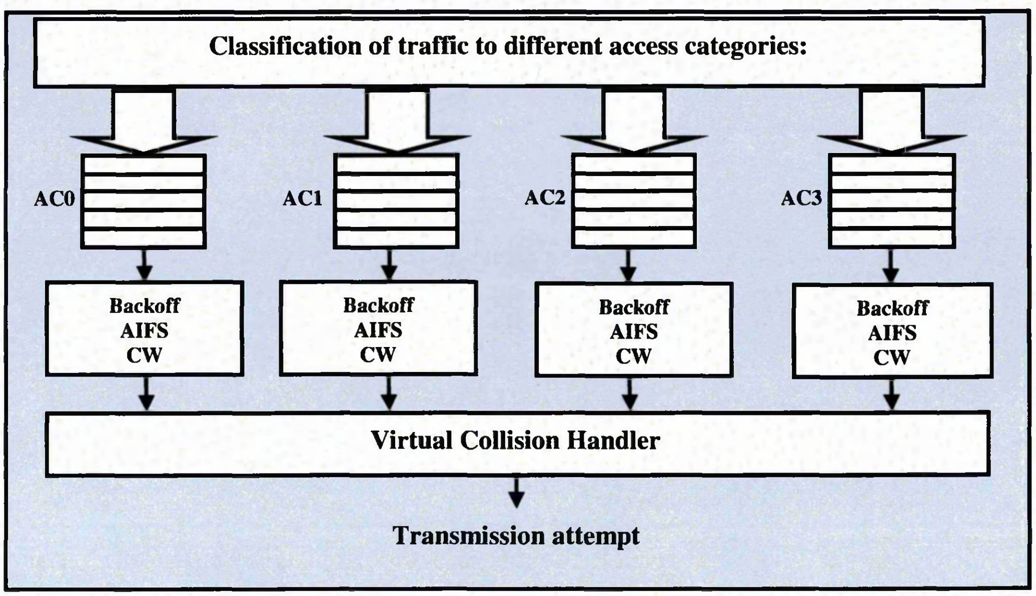


Figure 2-3. The IEEE 802.11e EDCA model.

Figure 2-3 shows that each AC forms an independent backoff entity with its own queue and corresponds to a different level of transmission priority. To simplify the concept, the four access categories (AC0 - AC3) assigned to different traffic priority. The traffic with the highest priority is assigned to AC0, whereas AC3 is assigned to the traffic with the lowest priority. The priority of an AC is determined by a set of parameters called

EDCA parameters. These parameters are: Arbitration Inter Frame Space (AIFS), Minimum Contention Window (CWmin), Maximum Contention Window (CWmax), and Transmission Opportunity (TXOP). The highest priority AC corresponds to the smallest AIFS, CWmin, CWmax and largest TXOP (Lin et al, 2009).

The AIFS is a replacement of the Distributed Inter Frame Space (DIFS) in IEEE 802.11 DCF access method. The value of AIFS can be determined based on the Arbitration Inter Frame Space Number (AIFSN) used to determine the length of AIFS, the time unit dictated by the physical layer characteristics SlotTime, and the Short Inter Frame Space period (SIFS) used to manage and control frames. AIFS for particulate AC can be calculated using equation 2.6 (Alahmadi and Madkour, 2008):

$$AIFS = AIFSN \times SlotTime + SIFS \tag{2.6}$$

The values of CWmin and CWmax determine the range of contention window for each AC. These values determine the range of random backoff slots which in turn control the waiting period for traffic before accessing the channel (Abeysekera et al, 2009). The backoff value of a particular AC can be determined using equation 2.7:

$$Backoff\,[AC] = random\,[0, min\,(2^K\,(CWmin\,[AC] + 1) - 1, CWmax\,[AC]] \tag{2.7}$$

where K is the number of collisions for the currently transmitted frame, CWmin, and CWmax are the minimum and maximum contention windows respectively. The high priority traffic with small CW values have small waiting period before accessing the medium whereas low priority traffic with a large CW values have a long waiting time to access the channel.

The transmission opportunity (TXOP) defines the transmission holding time for each AC. Each AC can transmit for a certain time interval whose length is determined by TXOP Limit.

During the operation of IEEE 802.11e EDCA as shown in Figure 2-4 (Lin et al, 2009), each AC contends to access the channel as an individual virtual station and start its backoff procedure after detecting the channel is idle for an AIFS period. AC with the smallest AIFS has the highest priority and needs to defer for its corresponding AIFS interval. When a particular station can initiate its transmission, it will be allowed to transmit multiple data frames from the same AC continuously during time interval defined by TXOP. The highest priority AC has a largest TXOP period, while the lowest priority AC has a smallest TXOP period. Similar to a real packets collision, when an internal collision occurs among the ACs within the same station, the higher priority AC has the rights to transmit whereas the lower priority AC suffers from a virtual collision.

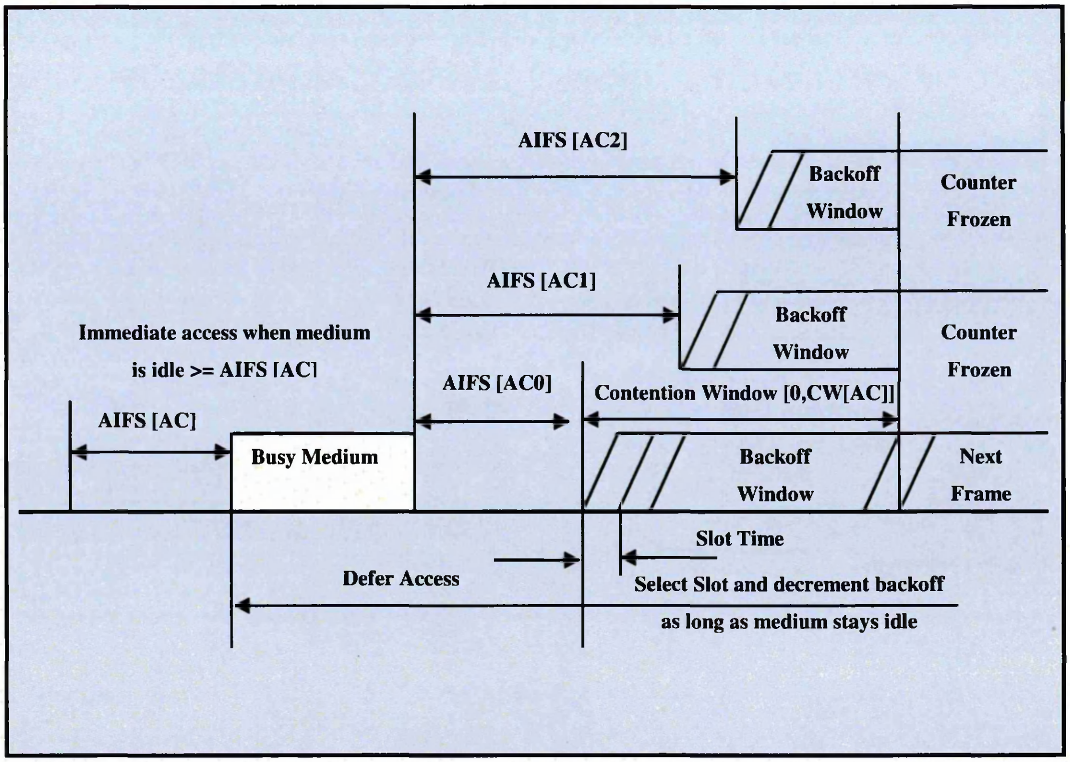


Figure 2-4. IEEE 802.11e EDCA operation.

IEEE 802.11e EDCA aims to ensure better service to high priority traffic and offer a reduced service for low priority traffic. The simplicity of EDCA operation as compared with HCCA means it is implemented widely. However, although EDCA mechanism improves the QoS of time sensitive applications, its performance is not optimal because EDCA parameters are not adjustable according to the network conditions. EDCA mechanism is reported to be unable to guarantee a good performance when the network traffic load was high. The main reason was the excessive number of packet collisions, which in turn was due to the fixed transmission parameters values assigned to the ACs (Villalón et al, 2007) and (Lin et al, 2009).

The limitations of EDCA mechanism make it an area of research to improve its performance. The study reported in this thesis, focused on the IEEE 802.11e EDCA. The allocation of traffic to the fixed AC will be investigated in order to enhance EDCA performance.

2.3.2 QoS in Wired Networks

Most current wired networks that are based on IEEE 802.3 have error rates when transmitting multimedia applications as compared with wireless networks. This is due to a higher bandwidth which ranges between 10Mbps - 100 Mbps. However, with the growth in transmission of real-time applications over wired networks, over-provisioning which refers to enhancing the network capability by simply providing the network with enough bandwidth, in order for all traffic to meet their QoS requirements might not be an optimal solution. This is because over-provisioning approach can be difficult and costly (Fraleigh et al, 2003). In addition, over provisioning bandwidth in the wired network may not prove effective in dealing with QoS requirements of multimedia applications (as bandwidth is a costly resource).

Multimedia applications must have priority over elastic applications because of their higher sensitivity to QoS parameters. For example, high delay variation of VoIP packets would affect its quality which in turn leads to a stuttering communication with pops and clicks.

An important issue that affects the QoS in wired networks is traffic prioritisation. When multiple packets are serviced through a bottleneck such as a router in the same manner, this would negatively affect the QoS, as it ignores the QoS requirements of transmitted applications. The transmitted packets could for instance experience several types of delay such as queuing delay, which occurs in the output buffer of a router. Also, when the buffer of the congested router is overflowed, the transmitted packets could be dropped regardless to their QoS requirements.

Therefore, an efficient utilisation of network resources must be considered during the transmission of different applications in order for the packets to be serviced according to their QoS requirements. This in turn improves network's QoS at the wired side of the network.

There are several mechanisms to support QoS in wired networks. These include: Call Admission Control (CAC), bandwidth reservation, congestion-management, congestion-avoidance, traffic policing, and shaping (Szigeti and Hatting, 2005). In this study, the network traffic prioritisation will be investigated as one of the most important issues that affect the QoS in wired networks. Therefore, packet scheduling mechanisms which are the most commonly congestion-management tools will be considered.

2.3.2.1 Packet Scheduling Mechanisms

Conceptually, the phrase "scheduling" refers to a set of rules that determine how a frame or packet is treated when congestion or bottleneck occurs at the convergence point. When bottleneck occurs, devices such as routers have buffers that allow packets to be stored temporarily in order to be scheduled subsequently. This process is known as "queuing".

The two terms "scheduling" and "queuing" are complementary and their processes are intertwined. Queuing process is engaged only when congestion occurs and deactivated after the congestion is cleared. Similar to queuing, scheduling takes place when packets experience congestion. However, the scheduler has to decide which packet should be transmitted next, even when there is no congestion. Packet queue scheduling determines bandwidth allocation among transmitted packets and the manner to service various applications with respect to different QoS requirements (Szigeti and Hatting, 2005).

There are a number of queuing scheduling mechanisms. These include: First-In-First-Out (FIFO), Priority Queuing (PQ), Fair Queuing (FQ), Weighted Fair Queuing (WFQ), and Weighted Round Robin (WRR) (Semeria, 2001). These mechanisms are explained in the following subsections.

2.3.2.1.1 First-In, First-Out (FIFO) queuing mechanism

FIFO queuing mechanism is the most basic queue scheduling algorithm. It is also known as First-Come-First-Served (FCFS). The incoming packets are accepted in order of arrivals. Figure 2-5 shows the process of FIFO scheduling mechanism.

FIFO serves the first packet in the queue regardless of any prioritisation or even fairness. This feature makes it to be the simplest scheduling mechanism in terms of implementation. However, FIFO is drop-tail based, when the buffer at the router implementing FIFO mechanism becomes full, the arrived packets are dropped. Therefore, FIFO can be insufficient mechanism in meeting QoS requirement for particular applications such as real-time applications (Semeria, 2001) and (Hasegawa et al, 2002).

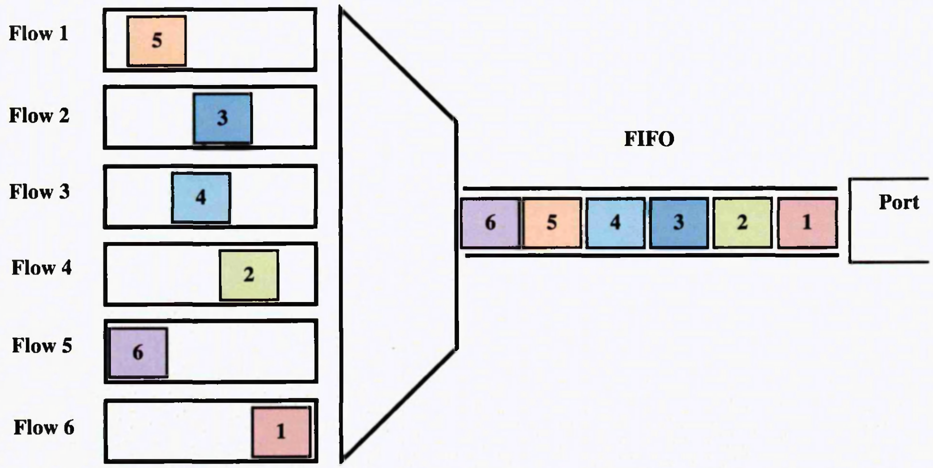


Figure 2-5. The process of FIFO scheduling mechanism.

2.3.2.1.2 Priority Queuing Mechanism (PQ)

PQ is designed to provide a simple method of supporting service differentiation among a variety of applications. The fundamental concept of PQ is to classify traffic in different classes according to their QoS requirements and then place them into different priority queues. Packets placed in the highest priority queue are served first, whereas the packets located in the lowest priority queue are severed only when higher priority queues are cleared. Lower classes could suffer from starving issue which in turn leads to a significant rate of packet dropping (Miaji and Hassan, 2010). In PQ mechanism, within each priority queues, packets are served by FIFO scheduling mechanism. Thus, any packet arrives in the lower priority queue will be dropped without any consideration if that particular queue is full. Figure 2-6 illustrates the process of PQ scheduling mechanism (Semeria, 2001).

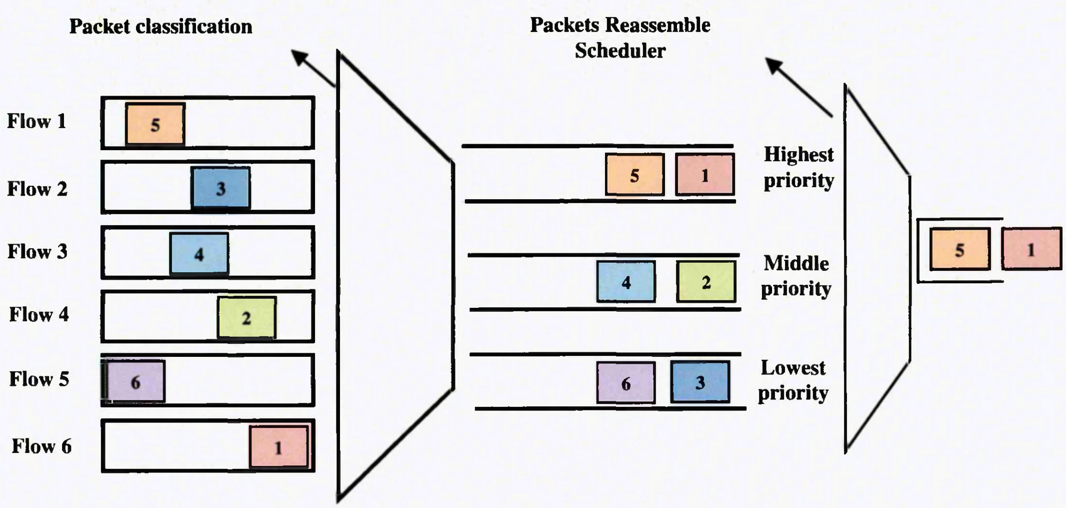


Figure 2-6. Priority queuing scheduling mechanism.

2.3.2.1.3 Fair Queuing Mechanism (FQ)

FQ scheduling mechanism was proposed by John Nagle in 1987 (Miaji and Hassan, 2010). It is designed to ensure that each flow has a fair distribution of the bandwidth regardless of the traffic transmission rate. In FQ, packets are first classified into flows and then each flow is assigned to a queue dedicated for that particular flow. During the scheduling process, flows are serviced one packet at a time in round robin order and empty queues are skipped. If a packet arrives at an empty queue after the scheduler is visited, the packet has to wait in that queue until the next visit of the scheduler. Figure 2-7 shows the principle of FQ (Semeria, 2001).

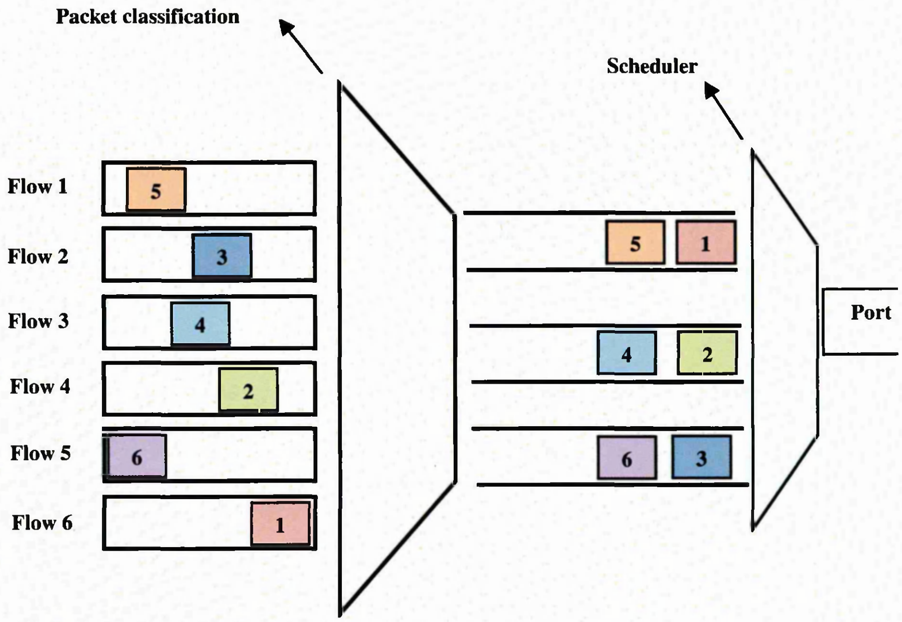


Figure 2-7. Fair queuing scheduling mechanism.

2.3.2.1.4 Weighted Fair Queuing Mechanism (WFQ)

WFQ was proposed by Zhang, Demers, Keshav and Schenke in 1989 to address the limitations of PQ, and FQ mechanisms. WFQ allocates bandwidth to different flows according to their assigned weights in order to satisfy the QoS requirements for different applications (Balogh and Medvecký, 2011). However, in WFQ, flows with large packet size are not allocated more bandwidth than flows with small packet size. Therefore, the distribution of bandwidth among variable length packets is carried out by a weighted bit-by-bit round robin scheduling. This approach supports fair distribution of bandwidth because it takes into the account the length of transmitted packets. Figure 2-8 shows a weighted bit-by-bit round robin scheduling serving three flows. 50% of the bandwidth is assigned to flow 1 whereas the remaining bandwidth are allocated to flows

2 and 3 (i.e. 25% to each flow). During scheduling, 2 bits are transmitted from flow 1, 1 bit from flow 2, and 1 bit from flow 3. This causes the packet with 600 byte to be transmitted before the packet with 350 byte, which in turn is transmitted before the packet with 450 byte (Semeria, 2001).

However, WFQ implements complex algorithm requiring a significant amount per-service class, and iterative check on each packet arrival and departure. The computational complexity affects the performance of WFQ to support a large number of flows on high speed network interfaces (Senkindu and Chan, 2008) and (Balogh and Medvecký, 2011).

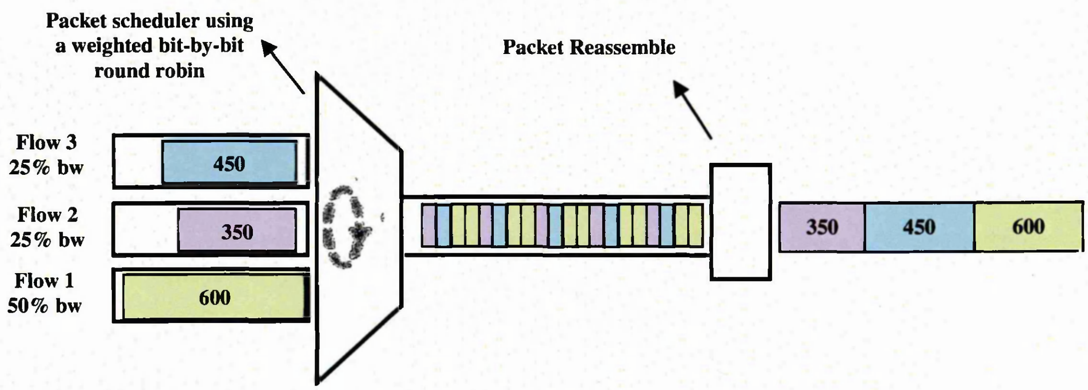


Figure 2-8. WFQ using a weighted bit-by-bit round robin scheduler.

2.3.2.1.5 Weighted Round Robin queuing mechanism (WRR)

WRR is also known as Class Based Queuing (CBQ). The operation of WRR is shown in Figure 2-9. The packets sent throughout the outgoing port are first classified into different classes and then assigned to a queue that is particularly dedicated to that class. The queues are in turn serviced using the weights associated with them. The weights indicate the number of packets to be sent for each class in a single service round. The number of packets transmitted for queue (*i*) is calculated using equation (2.8)

$$\textit{Number of serviced packets} = \frac{w_i}{\sum_{i=1}^{n} w_i} \times R \qquad (2.8)$$

Where W_i is the associated weight for queue (*i*), *n* determines the number of the queues, and *R* is the link capacity (Senkindu and Chan, 2008).

In WRR, capacity is allocated to different classes either by allowing high priority queues to send more packets in a single service round, or allowing each queue to send a single packet each time it is visited but high priority queues are visited multiple times during a single service round (Semeria, 2001).

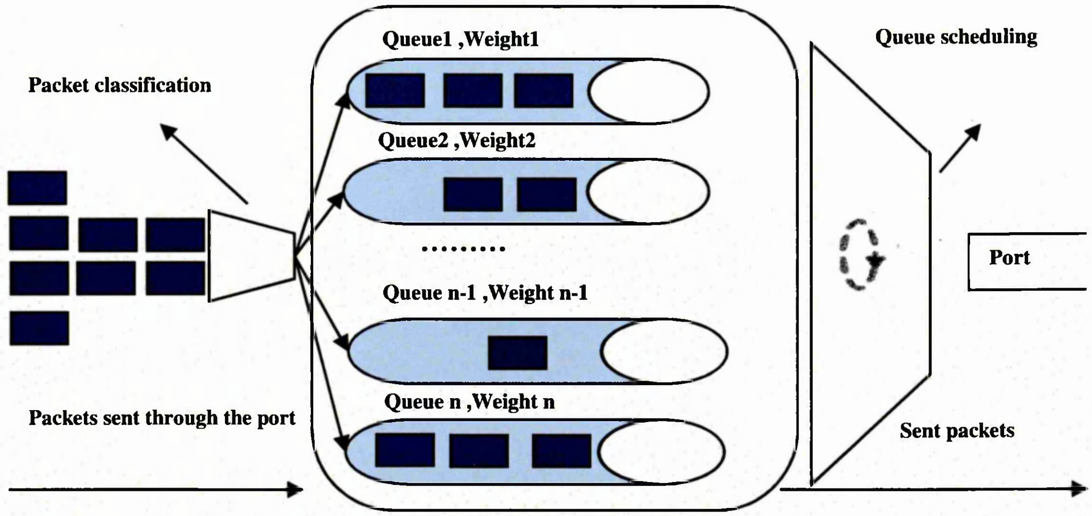


Figure 2-9. The operation of WRR.

From the aforementioned packet scheduling queuing mechanisms, it can be observed that the trade-off between these mechanisms is their complexity, control ability, and level of fairness. Therefore, an aspect of this study is an appropriate utilisation of a suitable queuing scheduling mechanism.

FIFO does not support QoS because it treats all traffic equally. It serves the first packet in the queue regardless of any prioritisation or fairness. PQ provides premium service to the high priority traffic at the expense of the low priority traffic. Low priority traffic are denied access to the buffer space, whenever high priority traffic is transmitted, which in turn causes the low priority traffic to experience excessive delay and high packet dropping. FQ is not designed to support traffic with different QoS requirements, as it allocates the same amount of bandwidth among multiple traffic. The computational complexity of WFQ algorithm affects its scalability to support lager traffic with different requirements at the edge of the network. WRR is designed to address the limitations of FIFO, PQ, and FQ by classifying traffic based on their QoS requirements, and ensuring that low priority traffic has access to buffer space and output port bandwidth. The implementation of WRR is more popular and its operation is less complex comparing with WFQ. Although WRR does not take packet size into account, an accurate bandwidth allocation could provide an optimal algorithm for usage in modern multimedia networks (Balogh and Medvecký, 2011). Therefore, in this study, WRR was considered to provide traffic prioritisation because of its practicality and low complexity.

2.4 Statistical and Artificial Intelligence Techniques

This section explains the principles of statistical and AI techniques used in this study. The basic concepts of regression analysis as one of the most widely employed statistical modelling tools is explained, and the fundamental principles of fuzzy logic, fuzzy clustering, and neural network as the most popular AI techniques are also discussed.

The aforementioned techniques were used in this study to fulfil the main objectives of this thesis. The use of these techniques is as follows:

i. Fuzzy Inference System (FIS) mechanism was employed to develop an approach that allowed QoS parameters of multimedia applications to be collected efficiently and accurately.
ii. Fuzzy C-Means (FCM) and Self Organizing Map (SOM) (i.e. Kohonen neural network) were used to develop techniques to analyse QoS parameters of multimedia networks.
iii. Regression model and Multi-Layer Perceptron (MLP) (i.e. a supervised neural network) were employed to develop techniques to assess QoS parameters of multimedia networks. The analysed QoS parameters by FCM and SOM were combined and then a single value that represented the overall network's QoS was produced.

2.4.1 Regression Model

Different regression models are used for prediction; they can be classified into linear and nonlinear model. Linear models include Auto Regressive (Box and Jenkins, 1976), Moving Average (Vandaele, 1983), and mixed of Auto Regressive and Moving Average (Box and Jenkins, 1976) and (Ljung, 1999). The nonlinear models include Bilinear Model, Threshold Auto Regressive Model, and Exponential Auto Regressive Model (Priestly, 1988). A detailed description about linear and nonlinear regression models is provided by (Box and Jenkins, 1976), (Vandaele, 1983), (Ljung, 1999, and (Priestly, 1988).

Due its simplicity and effectiveness, multi linear regression model is a commonly used method for prediction purposes (Chatterjee and Hadi, 2006) and (Sweet and Grace-Martin, 2010). Therefore, in this study, this type of regression is used.

2.4.1.1 Multi-Linear Regression Model

The multi-linear regression model is a widely employed statistical method due to its effectiveness for creating functional relationships among variables (Jain, 1991). Its aim is to analyse the relationship between several variables. One variable is considered to be the dependent or response variable and the others are considered to be independent or descriptive variables (Chatterjee and Hadi, 2006). In order for the regression model to be valid and accurate predictor, there are some assumptions that need to be followed (Jain, 1991) and (Chatterjee and Hadi, 2006). These are as follows: (i) dependent variable and independent variables need to be linearly related, (ii) the independent variables is non-stochastic and measured without error, and (iii) model errors are independent and normally distributed.

The formula of multi linear regression model as shown in equation (2.9) defines the relationship model between dependent variable (y) and independent variables ($x_1, x_2, \ldots., x_n$) as follows (Chatterjee and Price, 2006) and (Jain, 1991):

$$y = b_0 + b_1x_1 + b_2x_2 + \cdots + b_nx_n + e \tag{2.9}$$

In vector notation, the regression model can be written as in equation (2.10) or as in equation (2.11).

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_{11} & \cdots & x_{k1} \\ 1 & x_{12} & & x_{k2} \\ \vdots & & \ddots & \vdots \\ 1 & x_{1n} & \cdots & x_{kn} \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_n \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{bmatrix} \tag{2.10}$$

$$Y=XB + e \tag{2.11}$$

where Y is a $(n\times 1)$ vector of dependent variable, X is a matrix of independent variables with size of $(n\times (k+1))$, n is the number of observations, k is the number of independent variables, $B = \{b_0, b_1, b_2, \ldots b_n\}$ are the regression coefficients determined from recorded data, and e is a column vector of n error terms. The regression coefficients are calculated using equation (2.12).

$$B = (X^TX)^{-1}X^TY \tag{2.12}$$

where is X^T is an inverse of matrix X. The vector of residual e (i.e. error terms) is given by equation (2.13).

$$e = Y - XB \tag{2.13}$$

2.4.2 Fuzzy logic

Fuzzy logic was introduced by Lotfi Zadeh in 1965 (Zadeh, 1965) as a methodology for computing words rather than numbers. The concept of fuzzy logic is based on natural human communication language because it has similarities with human knowledge and reasoning (Klir and Yuan, 1995). The robustness of fuzzy logic due to the direct expression of input/output relationships without a physical derivation of the rules, and its flexibility to cope with imprecise and uncertain information and then draw definite conclusions makes it an excellent and powerful mechanism (Jantzen, 1998) (Khoukhi and Cherkaoui, 2008) and (Muyeen and Al-Durra, 2013). Unlike Binary logic (i.e. Classical logic) which has a sharp boundary between true and false states, fuzzy logic implements a gradient of possible states between true and false as shown in Figure (2-10) (Cirstea et al, 2002).

Fuzzy logic is applied to many applications in various domains such as control, decision making, optimisation, and evaluating systems (Klir and Yuan, 1995) (Naoum-Sawaya and Ghaddar, 2005) (Saraireh et al, 2008) and (Muyeen and Al-Durra, 2013).

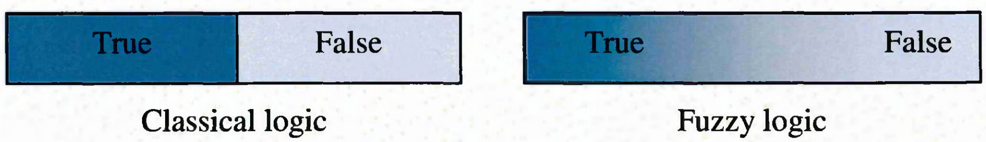


Figure 2-10. Binary logic versus fuzzy logic.

2.4.2.1 Fuzzy Inference System (FIS)

Fuzzy Inference System (FIS) is built upon the theory of fuzzy logic. FIS includes four main components: fuzzification, rules base, inference engine, and defuzzification as show in Figure (2-11) (Jantzen, 1998).

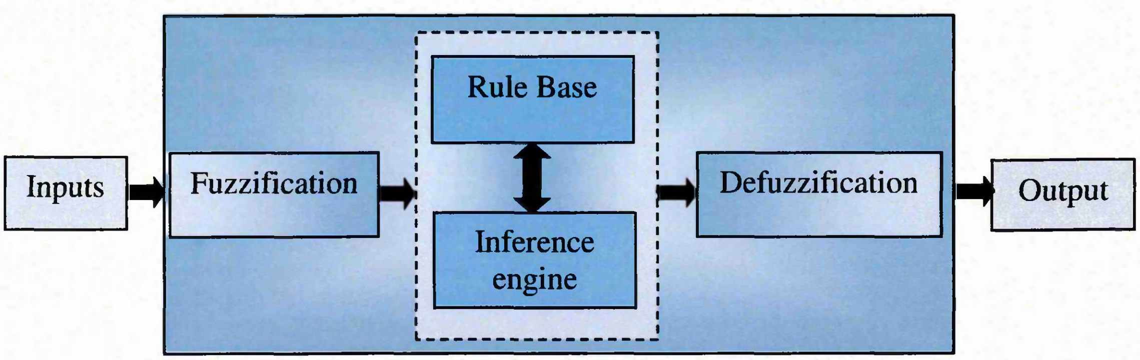


Figure 2-11. Block diagram of fuzzy inference system.

FIS is used to interpret (i.e fuzzify) the crisp values of inputs into linguistic terms, and based on a set of predefined rules, it calculates linguistic output value which in turn is converted (i.e. defuzzified) into real crisp output value (Naoum-Sawaya and Ghaddar, 2005) and (Saraireh et al, 2008). The following subsections outline each component of FIS.

2.4.2.1.1 Fuzzification

This is a process of converting numerical input values into linguistic terms and determining the degree of belonging to the appropriate fuzzy sets via membership functions. In fuzzy sets, Cirstea et al (2002) reported that an element (x_i) in the universe of discourse X is assigned a degree of membership $\mu(x_i)$ obtained by a membership function as shown in Figure (2-12). A membership function allows gradual transition from full-belonging to a fuzzy set to not-belonging at all with intermediate values presenting degrees of belonging (Al-Sbou et al, 2006). In fuzzification process, different types of membership functions can be employed. These include Triangular, Trapezoidal, Gaussian, Generalised, and bell Sigmoid (Mathworks, 2012$_{(a)}$).

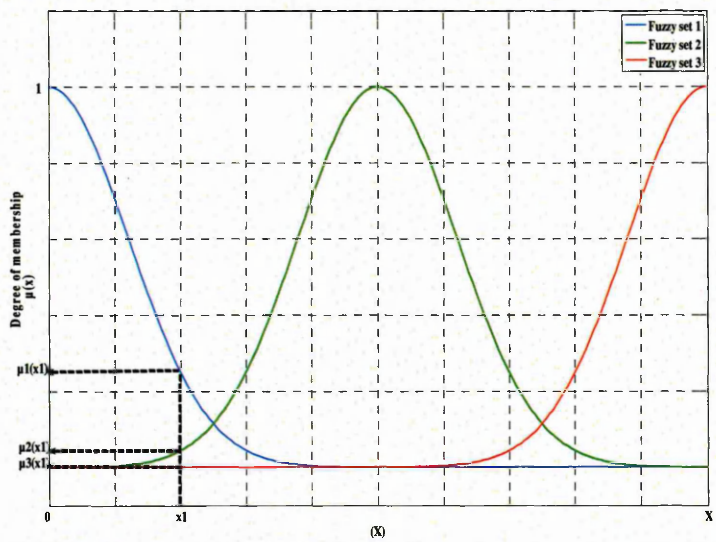


Figure 2-12. Degrees of membership in Gaussian membership function.

2.4.2.1.2 Rule Base

This component contains a set of *IF-THEN* rules represented in linguistic variables. The set of *IF-THEN* rules is the bases of decision making process of FIS. The number of rules depends on the number of inputs and outputs variables as well as the number of membership functions associated with them (Jantzen, 1998). The common form of *IF-THEN* rules as follows:

IF (Antecedent) AND (Antecedent) ......THEN (Consequent)

where the *Antecedent* relates the linguistic term to a fuzzy set, and the *Consequent* represents the conclusion from *IF* term. Each rule could have one or more connectives (i.e. fuzzy operators). The most common fuzzy operations applied on *IF-THEN* rules are Intersection, Union, and Complement which respectively defined by fuzzy operators *AND*, *OR*, and *NOT* (Klir and Yuan, 1995) (Ross, 1995) and (Mathworks, $2012_{(a)}$). For example, given that μ_X and μ_Y are the degrees of membership functions for fuzzy sets X and Y respectively, the application of fuzzy operators *AND*, *OR*, and *NOT* can be defined as given in equation (2.14) (Ross, 1995):

$$\begin{gathered} AND\!: \mu_{X\cap Y} = min\,(\mu_X\,, \mu_Y) \\ OR\!: \mu_{X\cup Y} = max\,(\mu_X\,, \mu_Y) \\ NOT\!: \mu_{\neg X} = 1 - \mu_X \end{gathered} \tag{2.14}$$

2.4.2.1.3 Inference Engine

Fuzzy inference engine uses fuzzified inputs along with the rules to perform inferencing (i.e. the process of implication and then aggregation) (Jantzen, 1998). The fuzzified inputs can be related to more than one rule to specify how adequately each rule describes the current situation by computing the degree of truth for *IF* condition. More than one rule might be triggered at the same time describing the same situation. Each of these rules produces *Consequent or Conclusion* to be taken in the *THEN* condition. This process is performed by implication method which is defined as the shaping of output membership functions. The input for the implication is a single number given by the *Antecedent* of the rule, and the output is a fuzzy set. The truncated output fuzzy sets from the implication process which describes the firing strength of the rules is then processed by an aggregation method. In the aggregation process, the truncated output fuzzy sets from the implication process are unified to produce one output fuzzy set (Ross, 1995).

2.4.2.1.4 Defuzzification

This is the process that converts the output linguistic value (i.e. the aggregate output fuzzy set) into a real numeric value. The input for the defuzzification process is the aggregate output fuzzy set and the output is a single number. However, the aggregate of a fuzzy set covers a range of output values which in turn must be defuzzified to produce a single output value from the set. There are several methods can be used in defuzzification process such as centroid, bisector, middle of maximum, largest of maximum, and smallest of maximum (Abdul Aziz and Parthiban, 2006).

Figure 2-13 shows the information flows through the process of fuzzy inference system: commencing from fuzzifying inputs, through the process of applying fuzzy operator, implication method, aggregation method, and terminating by defuzzification process (Mathwork,2012(a)).

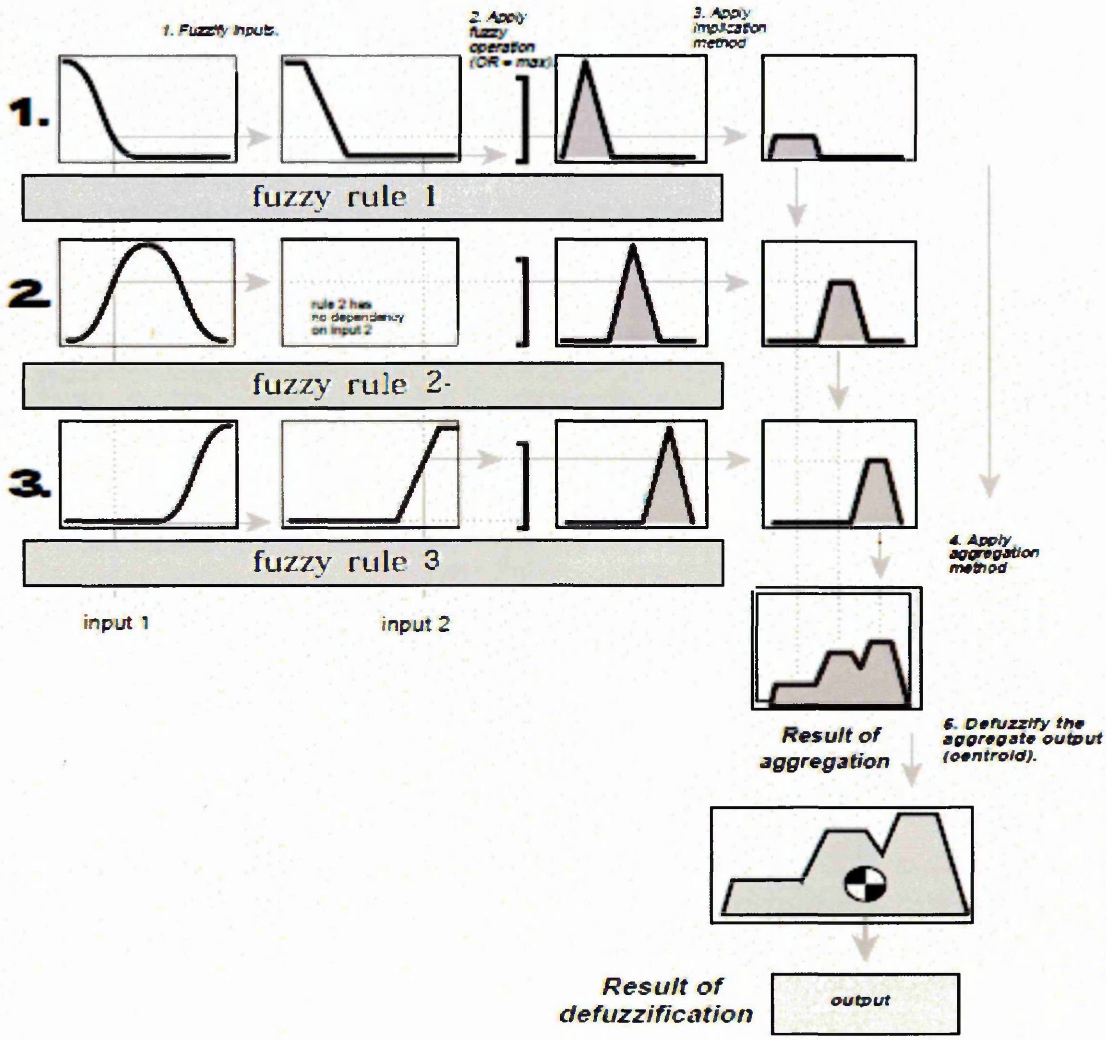


Figure 2-13. The process of fuzzy inference system.

Fuzzy inference system has two methods: Mamdani and Sugeno inference methods. The procedure of fuzzifying the inputs and applying the fuzzy operator during the fuzzy inference process are similar in both methods. However, the main difference between Mamdani and Sugeno is the manner the outputs are determined. Mamdani-type FIS is based on defuzzification process to generate crisp output from output fuzzy set, while Sugeno-type FIS uses weighted average to compute the crisp output. The advantage of

Mamdani FIS is that its outputs are expressed and interpreted. This feature is lost in the Sugeno FIS since the consequents of the rules are not fuzzy (Arshdeep and Amrit, 2012). The other difference is that Mamdani FIS has output membership functions whereas Sugeno FIS has no output membership functions. Due to the interpretable and intuitive nature of the rule base, Mamdani-type FIS is widely used particularly for decision support application (Haman and Geogranas, 2008). Therefore, in this study, Mamdani-type FIS will be used in this study.

2.4.3 Fuzzy clustering

Clustering techniques are used to partition data into correlated groups where different data objects should belong to different clusters and similar data objects are assigned to the same cluster. The aim of clustering is to reveal the underlying structures of data and explore its nature (Rokach and Maimon, 2005). Clustering techniques can be classified based on many criteria. For instance, clustering techniques can be divided into two main groups: hierarchical and partitioning techniques (Farley and Raftery, 1998). Another criterion divided clustering methods into: density-based methods, model-based clustering, grid based methods, and soft computing methods (Han and Kamber, 2006). In this study, the focus will be on Fuzzy C-means (FCM) as one of the most widely employed soft clustering methods. Unlike hard clustering methods where an object must belong to only one particular cluster, in FCM, it can belong to several clusters with different degrees of membership between 0 and 1 to indicate their partial membership (Parker et al, 2012). Therefore, in this study, FCM was used to analyse network QoS because the natural characteristics of network QoS patterns partly cover more than a single cluster.

2.4.3.1 Fuzzy C-Means Clustering (FCM)

FCM clustering algorithms was originally introduced by Dunn in 1973 and improved by Jim Bezdek in 1981 (Nascimento et al, 2000). This algorithm is one of the most widely used clustering algorithms. FCM is defined as a mechanism to discover certain features in a set of data and classify each element of data into a number of clusters with different degrees of memberships (Wang, 2009) and (Chaabane et al, 2008). FCM can be applied to partition a set of data with a form of matrix X of size $n \times N$ as shown in equation (2.15).

$$X = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1N} \\ x_{21} & x_{22} & & x_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nN} \end{bmatrix} \tag{2.15}$$

where each $x_j \in R^p, j = 1,2, \ldots, n$ is a given set of feature data representing by a number of features p. FCM operates on the matrix X and minimises the FCM objective function given in Equation (2.16) in order to partition matrix X into C clusters (Lei, et al , 2012).

$$J\,(X; U, V) = \sum_{i=1}^{C} \sum_{j=1}^{n} (\mu_{ij})^m \; D_{ij}^2 \tag{2.16}$$

The value m controls the degree of fuzziness for the membership of the cluster where $m \in [1, \infty]$. As the value of m decreased, the membership of the cluster becomes closer to the binary clustering. U the membership matrix includes $n \times C$ values and can be expressed as in formula (2.17):

$$U = \begin{bmatrix} \mu_{11} & \mu_{12} & \dots & \mu_{1C} \\ \mu_{21} & \mu_{22} & & \mu_{2C} \\ \vdots & \vdots & \ddots & \vdots \\ \mu_{n1} & \mu_{n2} & \cdots & \mu_{nC} \end{bmatrix} \tag{2.17}$$

where each value of the matrix $\mu_{ij}, j = 1, \ldots, n \; and \; i = 1, \ldots, C$ indicates the degree of membership between vector x_j and cluster C_i and must meet the following criteria:

- $\mu_{ij} \in [0,1], \forall_i = 1, \ldots, C, \forall_j = 1, \ldots, n$
- $\sum_{i=1}^{C} \mu_{ij} = 1, \; \forall_j = 1, \ldots, n$

During the clustering process, the objective function $J\,(U,\; V)$ is minimized with the following iterative steps (Parker et al, 2012) (Chen et al, 2009) and (Timo et al, 2002):

1. Membership matrix U is initialised with random values considering the aforementioned membership degree criteria.

2. The clusters' centres $V = \{v_1, v_2, \ldots, v_C\}$ are calculated according to Equation (2.18).

$$v_i = \frac{\sum_{j=1}^{n} (\mu_{ij})^m x_j}{\sum_{j=1}^{n} (\mu_{ij})^m} \quad , \forall_i = 1, \ldots, C \tag{2.18}$$

3. The distance D_{ij}^2 which is the Euclidian distance between x_j to the centre v_i of cluster i which is calculated using equation (2.19).

$$D_{ij}^2 = \left\| x_j - v_i \right\|^2 \quad (2.19)$$

4. The elements of matrix U are then updated using Equation (2.20).

$$\mu_{ij} = \frac{1}{\sum_{k=1}^{C} \left(\frac{\left\| x_j - v_i \right\|}{\left\| x_j - v_k \right\|} \right)^{\frac{2}{m-1}}} \quad (2.20)$$

Equations (2.18) - (2.20) are repeated until the termination criteria of FCM are met. The process of FCM can be terminated when the maximum number of iteration is reached or the objective function improvement between two consecutive iterations is less than the minimum amount of improvement (Timo et al, 2002).

2.4.4 Artificial Neural Network (ANN)

ANN is an adaptive parallel processing system capable of achieving results through the process of learning. It provides a mean to model the human brain in a simplified form. An ANN consists of a number of highly interconnected neurons that learn by interaction with each other as each interconnection has an associated weight (Abraham, 2005). In an ANN, the function of each neuron is to receive information, process it, and produce an output (Haykin, 1999). During the training process, the output is used to adjust the weight values to optimise the neural network performance. The mechanism to determine the amount of change in the weights is the neural network's learning algorithm (Cirstea et al, 2002) and (Zaknich, 2003). ANNs can be classified based on the manner of learning into supervised and unsupervised neural network. In supervised learning, they are provided with training examples from known classes together with their desired outcomes. In unsupervised learning, the neural network requires only training examples to be trained (Nogueira et al, 2006).

The advantage of ANN is the ability to derive meaning from imprecise values with highly parallel computing structure. This capability gives ANN the strength to model complex systems in efficient and effective manner and then achieve desired results. In this study, a combination of supervised and unsupervised neural networks was considered to evaluate the QoS parameters for multimedia applications. The transmitted application's QoS parameters were initially analysed by the unsupervised learning Kohonen neural network. The analysed QoS parameters were then used as inputs to a supervised learning Multi-Layer Perceptron (MLP) neural network in order to quantify the overall QoS. An explanation of MLP and Kohonen neural network (i.e. Self-Organising Maps SOM) are in the next sections.

2.4.4.1 Multi-Layer Perceptron (MLP)

In this study, a multi-layer perceptron (MLP) neural network was employed to assess the overall QoS due to its suitability and effectiveness. As one of the most popular supervised ANN, MLP needs to be provided with representative examples from each class, together with their corresponding class category. The architecture of MLP neural networks is shown in Figure 2-14 (a) (Abraham, 2005).

MLP composes of an input layer, one or two hidden layers and an output layer. At each layer, there are a number of neurons (i.e. processing elements). The function of each neuron is to receive information (i.e. the inputs with their associated weights), process them, and produce an output (Zaknich, 2003).

The operation of an MLP is shown in Figure 2-14 (b). During the training phase, a known pattern (x_i) is applied to the input layer of the MLP, and its target (i.e. desired value (d)) is applied to the output layer of the MLP. The elements of input (x_i) are multiplied with their associated connection weights (w_i) and the resulting value (s) is obtained using a summation function as in equation (2.21) (Cirstea et al, 2002):

$$s = \sum_{i=1}^{n} x_i . w_i \tag{2.21}$$

The output from the summation function (s) is processed by an activation function $\varphi(s)$ to provide the output (y). The activation function ensures that the neuron's output is limited to a predefined range (such as 0 to 1 or -1 to +1). There are several activation function types. These include threshold function, sigmoid, signum, and hyperbolic tangent (Cirstea et al, 2002) and (Mathworks, $2012_{(b)}$). Therefore, the value of y depends on (s) and the type of activation function. Subsequently, the calculated output (y) for each neuron at output layer is subtracted from the desired output (d) to produce an error (e) as expressed in equation (2.22):

$$e = d - y \tag{2.22}$$

The calculated error is then used by the learning algorithm to adjust the weights in order to reduce the error in the next iteration.

The process of learning and adapting are achieved by the learning algorithm. The function of the learning algorithm is to use the calculated error (e) and the input data (x_i) to adjust the values of the connections' weights (w_i) in such a way as to reduce the magnitude of the error in the following training iteration (Haykin, 1999). There are several methods to realise the required learning. These include gradient descent back-propagation, gradient descent with adaptive learning rate back-propagation, gradient descent with momentum back-propagation, and gradient descent with momentum and adaptive learning rate back-propagation (Abraham, 2005),(Cirstea et al, 2002).

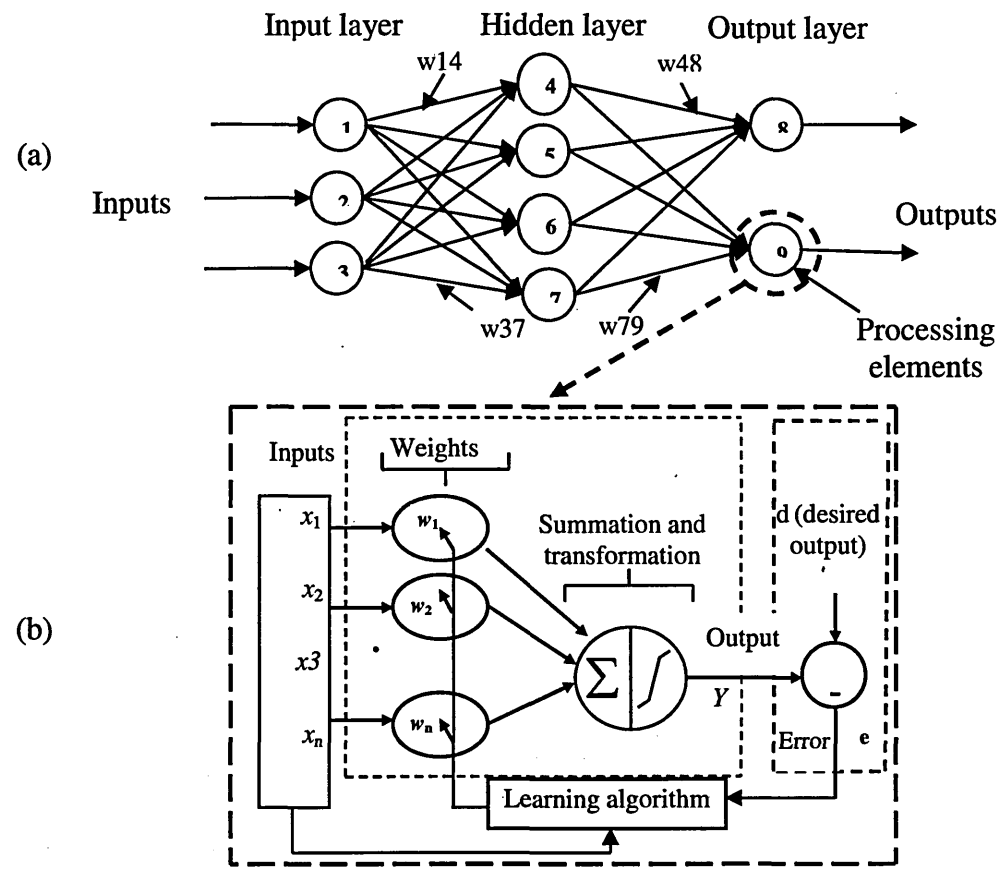


Figure 2-14. Multi-Layer Perceptron: (a) The architecture, (b) MLP operation.

The MLP training process is repeated until either the specified number of iterations is reached or when there is insignificant error between the network output (y) and the desired result (d) (Mathworks, $2012_{(b)}$). The termination criteria evaluate how effective the MLP is trained.

After the completion of training phase, the trained MLP is examined during the test phase. Unknown input patterns are fed to MLP input layer and outputs are produced by the output layer.

2.4.4.2 Kohonen Neural Network

In contrast to the multilayer perceptron, the Kohonen network (i.e Self Organising Maps SOM) is one of the most popular unsupervised neural networks require only training examples to learn (i.e. no desired output is required). Self-Organizing Map was introduced by Teuvo Kohonen as a data visualization technique (Kohonen, 1982). SOM visualizes data by reducing its dimensions to a map, and represents similar data objects into correlated clusters. An aspect of this study is that QoS parameters of multimedia applications were intelligently classified using SOM. In situations such as network QoS, where the natural characteristics of traffic cover multiple clusters, SOM could be an effective clustering technique to analyse network QoS patterns.

The Kohonen network has a single layer of neurons known as a Kohonen map as shown in Figure (2-15). The Kohonen map can be arranged in various topologies such as rectangular and hexagonal topology.

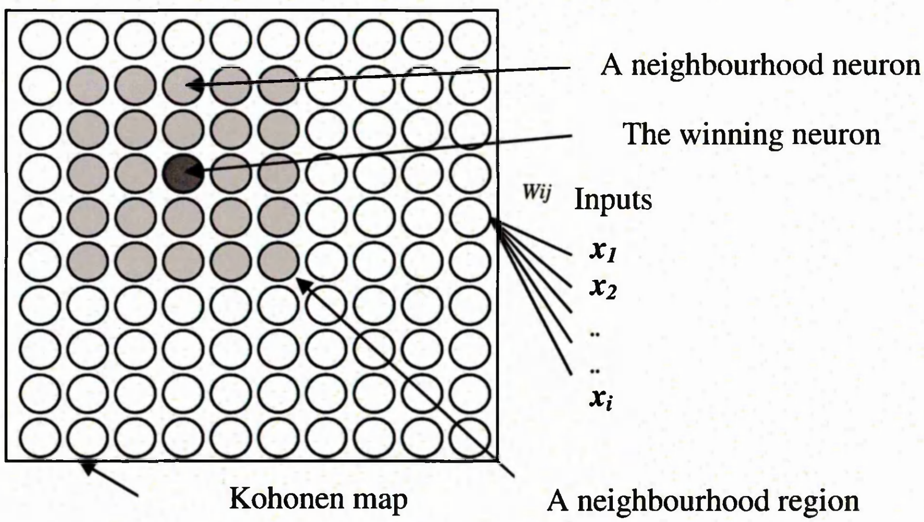


Figure 2-15. The structure of kohonen neural network.

As shown in Figure (2-15), the j^{th} neuron in Kohonen map is connected to i^{th} input feature of a certain pattern (x) to the neural network. Each connection has an associated weight(w_{ij}). The connections' weights are initially set to random values between 0 and 1. Then, the network learns by determining the Euclidean distance d_j between the features of normalised input pattern (x_i) and the neuron's weights. For each j^{th} neuron and N features for each example, the Euclidean distance is calculated using equation (2.23).

$$d_i = \sqrt{\sum_{i=0}^{N-1}(x_i - w_{ij})^2} \quad (2.23)$$

The neuron that obtained the smallest Euclidean distance with the input patterns is considered as the winning neuron. Its weights are adjusted using the learning algorithm as expressed in equation (2.24).

$$w_{ij}(n+1) = w_{ij}(n) + \eta\ (n)(x_i(n) - w_{ij}(n)) \quad (2.24)$$

where, $w_{ij}(n+1)$ is the updated weight, $\eta(n)$ is the learning rate $0 < \alpha < 1$, and the term $(x_i(n) - w_{ij}(n))$ represents the error. The learning rate is usually a value between 0 and 1 which in turn controls the adaptation speed. The learning algorithm ensures that the winning neuron's weights become iteratively closer to the input pattern category. This in turn allocates the winning neuron to become representative of that specific category.

The weights associated with a number of neurons around the winning neuron are also updated to a lesser extent. This enables specific regions of the Kohonen map to be associated with different pattern categories. The neurons around the winning neuron which their weights are updated are known as neighbourhood neurons, and the area of the Kohonen map covered by them is referred to as the neighbourhood region as shown in Figure (2-15) (Haykin, 1999).

2.5 Summary

This chapter provided a theoretical background related to the QoS of multimedia networks. This includes the definitions, QoS parameters, QoS requirements of multimedia applications, and QoS components. It also discussed the QoS in wireless and wired networks. The IEEE 802.11e as an emerging WLANs standard to provide QoS was explained, and packet scheduling mechanisms, the most commonly mechanisms implemented in wired network to support QoS were reviewed. This chapter also provided the relevant theoretical background to statistical and Artificial Intelligent (AI) techniques which were used in this study. The fundamental principles of regression analysis which is one of the most widely employed statistical modelling methods were discussed. The basic concepts of fuzzy logic, fuzzy C-means clustering algorithm, Multilayer Perceptron Neural Network, and Kohonen neural network were explained. The next chapter reviews the literature relevant to the process of managing QoS for multimedia computer networks

Chapter 3 Literature Review

3.1 Introduction

Managing Quality of Service (QoS) of multimedia applications is currently one of the principle research topics in the field of computer networks. Two important factors make the QoS management an issue of great importance. (i) Computer networks are increasingly integrated (i.e. networks consist of heterogeneous networks: wired and wireless). (ii) Computer networks carry diverse multimedia applications involving video, audio, and data which require a large bandwidth and perceived quality (Mohammed et al, 2001).

In this study, network QoS management refers to evaluation and improvement of QoS provided by computer networks. Evaluation of QoS aims to analyse and quantify network performance with respect to meeting the applications' transmission requirements. The QoS improvement involves the ability to take actions to enhance network QoS or change network performance toward a desired operation. However, there are many issues related to the process of managing multimedia computer networks. The main issues are: (i) Multimedia applications generate an extensive amount of data in the form of information packets. The collection and processing of all these packets in real time are not practically feasible. (ii) Gathered network information which represents network performance in delivering diverse applications include a multitude of parameters related to QoS. These parameters need evaluating in an effective manner. (iii) Multimedia applications require a large bandwidth, that in turn are considered to be as a critical issue because of the limitations of the physical communication channels as well as the interfering noise, particularly in the wireless side of the network. Accordingly, the QoS of the transmitted applications could be unpredictable. (iv) The existing monitoring tools are unable to get directly the overall network QoS. Network managers have to do an extra evaluation to assess the overall network QoS. This process could be complicated, expensive, and time consuming.

In this chapter, the previous studies relevant to the aforementioned issues of managing QoS of multimedia network are critically analysed and discussed. The aim of this chapter is to identify the potential gaps associated with these issues which in turn require a further development and investigation.

This chapter is organised as follows: section 3.2 reviews the sampling techniques used to gather information from network traffic. In section 3.3, the current QoS analysis and assessment techniques used to evaluate network QoS are discussed. The relevant studies considering QoS support in wireless and wired networks are evaluated in section 3.4. Section 3.5 critically analyses the exiting monitoring tools used to assess the network performance. Section 3.6 reviews the applications of statistical and artificial intelligent techniques (used within this study) into the field of computer network management. Finally, in section 3.7, a summary of this chapter is provided.

3.2 Sampling Approaches for Measuring QoS Parameters

The growth in real-time transmission of multimedia applications over computer networks means that the QoS parameters of these applications need to be recorded and measured in an efficient manner. The quantification of QoS parameters allow QoS provided by the network for the transmission of these applications to be assessed. However, most real-time applications generate an extensive amount of data in the form of information packets. The collection and processing of all these packets in real time are not practically feasible and computationally intensive. Therefore, in order to reduce the amount of collected data, sampling operation needs to be performed.

Sampling techniques are used to analyse the statistical characteristics of a population of packets and to produce subset traffic with smaller number of packets that represents the original traffic. Sampling techniques can be classified into two categories: adaptive sampling and non-adaptive sampling (Hernandez et al, 2001).

3.2.1 Non-adaptive Sampling Techniques

In non-adaptive sampling techniques, the packet selection method can be time interval based or packet number based. In the former method, the selection is based on predefined time intervals, whereas in the latter, the packet selection decision is based on a packet count (Zseby, 2004) and (Claffy et al, 1993).

Non-adaptive sampling techniques can be classified into systematic, random, and stratified. In a systematic sampling scheme, a packet is chosen at either a fixed time interval or a fixed number packet count. In random sampling, a packet is chosen from the parent population at a random time interval or in a random packet count number. In stratified sampling, a fixed interval of time is chosen and a packet is randomly selected from that interval (Claffy et al, 1993) and (Gan et al, 2009). Figure 3-1 (a) shows systematic, random, and stratified sampling techniques respectively (Claffy et al, 1993).

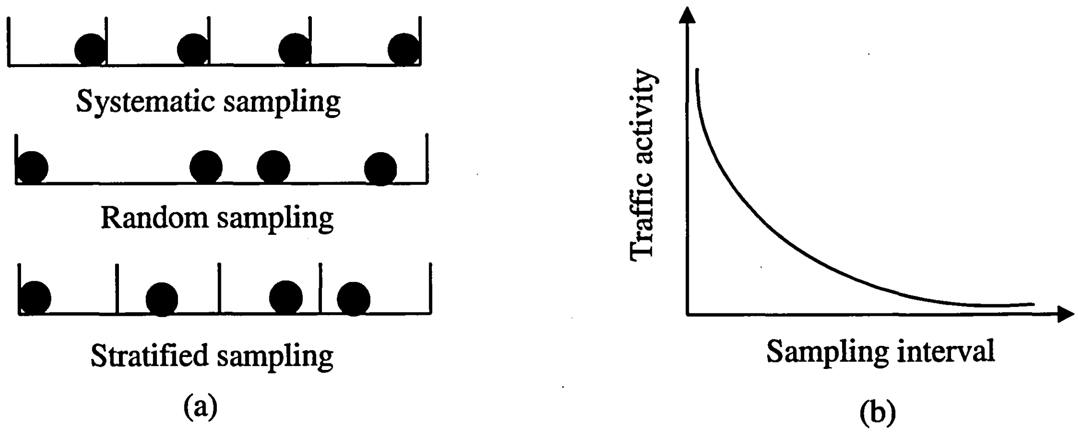


Figure 3-1. Classification of sampling techniques: (a) Non adaptive sampling, (b) The concept of adaptive sampling.

The use of predefined sampling rate in non-adaptive sampling schemes to determine how to sample packets may not be effective for sampling multimedia traffic. This is because multimedia traffic is time varying. In case of sampling multimedia traffic using non-adaptive sampling techniques, two situations might occur: (i) if the traffic rate is too high, there is a risk of losing information caused by under sampling. (ii) If the traffic rate is too low, resources would be underutilised by over sampling (Giertl et al, 2006).

Gathering traffic information from the network using conventional sampling techniques poses a challenge as the process should reflect on the nature of ongoing network traffic activity. Therefore, in order to reduce the biasness presented in conventional sampling methods and to increase the effectiveness of measuring QoS parameters, traffic should be sampled adaptively. This process is illustrated in Figure 3-1 (b).

3.2.2 Adaptive Sampling Techniques

Unlike non-adaptive sampling techniques, the sampling rate of adaptive sampling techniques (i.e. the packet selection method) is adjusted during the sampling process according to traffic characteristics. In other words, small sample interval are required during the period of high activity, whereas a larger sample interval is required during the period of low traffic activity (Giertl et al, 2006).

A number of studies have been conducted using adaptive sampling to gather network information. For example, in (Hernandez et al, 2001), two adaptive sampling schemes have been proposed based on linear prediction and fuzzy logic. The experimental results showed that both approaches achieved better results compared with systematic sampling.

An adaptive sampling based on fuzzy logic was reported in (Giertl et al, 2006). The contribution of the study was to assess the utilisation of bandwidth for real TCP/IP activity of network interface, connecting a local network to the Internet using adaptive, random and stratified sampling techniques. The study showed that adaptive sampling was more effective than the non-adaptive sampling techniques.

In (Ma et al, 2003, and 2004), an adaptive sampling technique was devised using the time-sliding window (TSW) algorithm to estimate traffic rate. A key element of the devised technique was to predict future behaviour based on observed behaviour. The experimental results showed that measuring QoS parameters of voice traffic using the devised adaptive sampling technique was more accurate than the conventional sampling techniques. Gan et al (2009) proposed another method to sample packets in an adaptive manner. Their adaptive sampling algorithm automatically adjusted the sampling granularity according to the packet arrival interval. Their results showed that the proposed adaptive sampling method is more accurate and economical than static sampling methods.

Adaptive sampling methods are not only used for traffic measurement, but they also can be used for a number of other applications (i.e. network security application). For instance, Zhang et al (2007) proposed a small packet threshold adaptive sampling algorithm to capture malicious packets, which speeded up the sampling process and achieved accurate results. Patcha and Park (2006) in their paper proposed an adaptive sampling algorithm based on weighted least squares prediction. The proposed sampling algorithm was tailored to enhance the capability of network based IDS at detecting denial of-Service (DoS) attacks. The algorithm was not only proposed to reduce data analysed by IDS but also, it maintained the intrinsic self-similar characteristic of network traffic. This feature can be used by IDS to detect DoS attacks by using the fact that a change in the self-similarity of network traffic could be an indicator of a DoS attack.

From the above discussion, most of the previous adaptive sampling schemes require sophisticated computations as in the case of (Hernandez et al, 2001), (Giertl et al, 2006), and (Giertl et al, 2008). Other schemes consider one parameter as the reference-coefficient of sampling granularity as the case in (Gan et al, 2009). None of the previous studies considered the traffic's statistics such as mean, median, and standard deviation during the process of sampling multimedia traffic in an adaptive manner.

The contribution of this study is to propose a statistical adaptive sampling method by considering the traffic's statistics. The method increases the interval between two consecutive sampled sections, when the overall statistic of the traffic does not change for those two sections, and the interval is decreased when the overall statistic of the traffic for the two sections significantly differs. The advantages of this sampling method are the ease of its implementation and its fast response to the variation of the input traffic.

3.3 Network Quality of Service Evaluation

The growth in transmission of critical real-time applications such as VoIP and video applications over computer networks means that their QoS needs evaluating in an effective manner. QoS evaluation of multimedia networks is very important for end users, network managers, and service providers. Users are interested in determining how well they are receiving services and whether the received services meet the agreed levels of service between them and the service providers (Gozdecki et al, 2003). Network managers need to evaluate QoS to determine how well their networks are operating in order to identify network failures and to optimise network performance. Service providers need to evaluate network QoS in order to comply with the level of QoS that the customer expects (Molina-Jimenez et al, 2004).

The current evaluation of QoS is achieved either by analysis or measurement techniques. The analysis techniques examine the characteristics of network traffic (Timo et al, 20022), (Wang et al, 2009), and (Ting et al, 2010), whereas the measurement techniques determine how well the network treats the ongoing traffic (Palomar et al, 2008), (Teyeb et al, 2006), (Mishra and Sharma, 2003), (Pias and

Wilbur, 2001), and (Malan and Jahanian, 1998). The following subsections discuss the current QoS analysis and measurement techniques used to evaluate network QoS.

3.3.1 Quality of Service Analysis Techniques

Analysis of network QoS plays a crucial role in practicing effective management for multimedia computer networks. The aim of QoS analysis is to obtain a comprehensive view about the state of the network and simultaneously discover important details from the transmitted traffic (Timo et al, 2002). The network traffic analysis has been investigated by a number of studies using several mechanisms. The most explorative techniques used to analyse the characteristics of network traffic are statistical analysis techniques and AI techniques (Liu et al, 2012) and (Ting et al, 2010).

Statistical parameters such as mean, standard deviation, and mode that are used to analyse network traffic may not be as effective as AI techniques. This is because outliers of the analysed traffic using statistical parameters could affect the final conclusion about traffic characteristics which in turn may give inaccurate results about the analysed traffic (Jain, 1991). Therefore, statistical parameters will not be used in this study to analyse QoS.

Due to the ability of AI techniques to derive meaning from imprecise values as demonstrated by (Cirstea et al, 2002) and (Haykin, 1999), and because network traffic is highly complex in natural as reported in (Wang et al, 2009), and (Ting et al, 2010), this study will include AI methods to analyse the characteristics network traffic.

Fuzzy C-means (FCM) and Self-Organising Maps (SOM) (i.e. Kohonen neural network) were previously used in a number of study to analyse network traffic..

FCM was used to cluster network traffic and produce application profiles which contained significant information about the current status of the network in order to manage network resources as reported in (Timo et al, 2002). A network administrator assistance system was proposed based on FCM. The proposed system utilised a FCM method to analyse users' network behaviours and traffic-load patterns based on the measured traffic data of an IP network. Analysed results were used to assist administrators to determine efficient controlling and configuring parameters of the network management (Chen et al, 2009). In wireless sensor networks, FCM algorithm was used to create clusters which reduced the spatial distance between sensors nodes

(Hoang et al, 2010). FCM was employed to detect routing attacks caused by abnormal flows in a wireless sensor network. The study demonstrated that FCM can be a valuable tool for intrusion detection (Wang et al, 2009).

SOM was also used by several studies to analyse network traffic. For example, Kernel-SOM (KSOM) was proposed to introduce network traffic classification approach (Ting et al, 2010). The experimental results showed that KSOM achieved high classification accuracy and successfully categorised network traffic characteristics. In (Kiziloren and Germen, 2007), network traffic types were analysed using SOM. The aim was to distinguish between normal traffic and anomaly traffic, such as port scanning and Denial of Service Attacks. The results demonstrated the usefulness of SOM to distinguish three traffic types: port scanning, heavy-download, and other traffic.

However, none of the previous studies discussed in this section have utilised FCM or SOM to analyse and classify QoS parameters (i.e. delay, jitter, and packet loss ratio). A novel aspect of this study is that QoS parameters of multimedia applications are intelligently classified using either FCM or SOM. In situations such as network QoS parameters where the natural characteristics of traffic cover more than a single cluster, FCM algorithm or SOM could be an effective QoS analysis technique.

3.3.2 Quality of Service Measurement Techniques

Measuring QoS of multimedia applications could be performed using different types of approaches. These can be objective or subjective. An objective approaches measures the QoS based on mathematical analysis that compares original and distorted multimedia signals (ČÁKY et al, 2006). Examples of some objective methods are Mean Square Error (MSE) and Peak Signal to Noise Ratio (PSNS) which measure the quality by a simple difference between frames (Mohammed et al, 2001). Other examples of objective methods are Perceptual Speech Quality Measure (PSQM) and Perceptual Evaluation of Speech Quality (PESQ) which used particularly to measure the voice quality (ČÁKY et al, 2006).

A number of studies quantified QoS based on objective approaches. For instance, Sun and Ifeachor (2002) examined the impact of packet loss and gender of talker on perceived speech quality using ITU PESQ measurement algorithm. In the study, they found that the packet loss and gender of talker have an impact on perceived speech quality. The quality for female talkers was worse than the male talkers for the same network conditions. Palomar et al (2008) assessed the quality of audio streaming over WLAN using EAQUAL which is a software tool based on ITU-R for objective assessment. Nevertheless, the experimental results showed that the EAQUAL was not a good approach to assess the audio quality in cases like the effect of packet loss.

Subjective approaches on the other hand measure the overall multimedia quality based on human subjects (ITU-T, 2008) and (Mohammed et al, 2001). Subjective approaches are carried out by having *n* human subjects viewing the distorted multimedia signals and then rate their quality based on predefined scale. The most common scalar used by subjective approaches is Mean Opinion Score (MOS) (ITU-T, 1998) (Patel et al, 2003).

Several studies have used subjective approaches to evaluate network performance. For instance, Bräuer et al (2008) assessed Voice and Video over IP (VVoIP) quality in IP networks. They used Mean Opinion Score (MOS) Absolute Category Rating (ACR) scalar to obtain the quality of VVoIP under certain conditions. Teyeb et al (2006) evaluated the performance of heterogeneous networks in subjective manner. In their study, the QoS of web browsing and video streaming services was subjectively evaluated through usability test. The aim was to find out the effect of network parameters on users' perception of quality of web browsing and video streaming.

However, objective and subjective approaches have some limitations. Subjective approaches cannot be automated, they require controlled environment, and they are time consuming to be repeated frequently due to their dependence on human subjects (Palomar et al, 2008). While objective approaches require high calculation power because their operation at the pixel level, and they cannot take into the account all the affected network parameters (Mohammed et al, 2001).

Another type of QoS assessment is based on passive or active measurements. Active measurements are carried out by generating probe packets and injecting them into the network or a portion of the network (Brekne et al, 2002). The concept behind this approach is that the performance experienced by probing packets gives an indication about the performance experienced by real traffic.

A number of studies assessed QoS based on active measurement approaches. For instance, Mishra and Sharma (2003) proposed active measurement approach to select an appropriate Label Switched Path (LSP) which satisfies the QoS of the new connection. The selection of LSP was based on the end-to-end delay of probing packets sent along each LSP. (Ma et al, 2003) and (Ma et al, 2004) used adaptive sampling and active measurement to evaluate the performance of voice traffic in Multiprotocol Label Switching MPLS-based IP networks.

In contrast with active measurements, passive measurements are carried out by measuring the actual network traffic in order to provide an indication about the network performance. This approach is non-intrusive as packet's information can be gathered without adding probing packets which might disturb the operation of the network (Brekne et al, 2002).

There are several studies based on passive approaches to quantify QoS. For instance, Cranley and Davis (2005) investigated the effect that the background traffic has on video streaming traffic. Their approach non-intrusively measured and recorded the bandwidth utilisation of video streaming traffic. Al-Sbou et al (2008) passively evaluated network performance of multimedia applications in mobile ad hoc networks (MANET). Their proposed system indicated that the measured QoS can be used as an indication of the network conditions and resource availability.

However, there are some drawbacks with both passive and active measurements. Performing active measurements that give representative results is not a trivial task, because excessive probing generates a significant load that might disturb the operation of the network, whereas infrequent probing might not reveal the performance characteristics of the network. The disadvantage of passive measurement however is the requirement of collecting and processing a large amount of recorded packets which is not practically feasible in real time (Brekne et al, 2002).

The limitations of the above QoS measurement approaches highlight the need for further investigation and development. The contribution of this study is to develop QoS assessment techniques that overcome some drawbacks in subjective and objective approaches. The techniques should evaluate QoS in a manner similar to human subjects and quantify the QoS without the necessity for complex mathematical models taking into the account the QoS requirements of each type of multimedia application. In addition, the QoS assessment techniques should not add extra load to the network as the case of active approaches, nor depend on the whole collected packets like passive approaches. The proposed assessment technique will be based on the analysed traffic generated from the proposed analysis techniques in order to overcome some drawbacks of both active and passive measurement methods. An aspect of this study is that a regression model is developed and Multi-Layer Perceptron (MLP) neural network was trained in order to combine the QoS parameters (i.e. delay, jitter, and packet loss ratio) for each QoS class (or cluster) identified by SOM or FCM to estimate the overall QoS. This is because a single QoS parameter could not reflect an application's transmission requirements. For instant, delay, jitter, and packet loss ratio could all have significant effects on VoIP quality (Al-Sbou, 2010).

Regression model is a widely employed statistical method in networking domain. Baldwin (1999) developed regression models from simulated data to predict network behaviour in terms of throughput, mean delay, missed deadline ratio, and collision ratio. An adaptive regression algorithm was proposed to monitor two arbitrary sensor nodes and dynamically learn the linear relation among their measurements. The algorithm then eliminated the redundant node, and estimated the deficient data without the need for base station assistance (Ollos and Vida, 2009). Regression model was used to predict the collision ratio, collision rate variation, and queue status ratio in participant wireless nodes in a mobile ad-hoc network and to subsequently adjust the Contention Window (CW), Distributed Inter-Frame Space (DIFS) and transmission rate in order to improve the network performance (Saraireh, 2006).

Neural networks were also used in the area of network QoS. For instance, in (Nogueira et al, 2006), a modelling approach based on neural network was proposed to predict the media access delay in a Wireless Local Area Network (WLAN). MLP was used to predict packet loss in a real-time video transmission. The results showed that MLP could predict packet loss rate with accuracy of 96% (Lavington et al, 1999). Random Neural Networks (RNNs) were devised to automatically quantify the quality of video flows. The achieved results correlated well with human perception methods (Mohamed, 2002). In (Radhakrishnan and Larijani, 2011), three non-intrusive RNNs models (simple feed-forward model, Multilayer feed forward model, and recurrent architecture) were employed to measure and monitor voice quality transmission. The results indicated that the feed forward architecture produced the best results as compared with the other two architectures.

3.4 Improving Quality of Service in Computer Networks

QoS improvement in this study involves the ability to take actions to improve QoS or change network performance toward desired situation. Improving QoS has become more pronounced in particular with the presence of multimedia applications and integration of wired and wireless networks.

In many instances, users in WLAN exchange and access multimedia applications with users in wired networks. Figure 3-2 shows how voice calls can take place between a user connected to wireless network and user connected to wired network. Considering that multimedia applications which are sensitive to transmission parameters, it is vital to improve and sustain QoS in the integrated networks in order to enable the effective delivery of multimedia services.

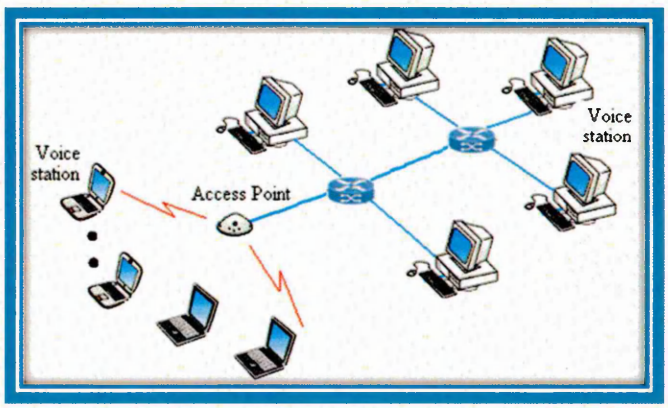


Figure 3-2. Integrated WLAN-wired network.

The integration of wired and wireless networks poses challenges to overcome QoS issues in both network sides: wired and wireless. This is because the QoS issues are different between wired and wireless networks. The QoS support in WLAN is enabled at MAC layer whereas the most wired networks enable QoS at the IP layer (Senkindu and Chan, 2008). Therefore, providing and improving network QoS for traffic being exchanged between two different environments is considered to be a challenging task.

In terms of wireless networks, the demand for supporting QoS for various applications has led to the development of a WLAN standard so called IEEE 802.11e. This standard provides two schemes to access the wireless channel: Enhanced Distributed Channel Access (EDCA) and Hybrid Coordination Function (HCF) Controlled Channel Access

(HCCA) (IEEE Computer Society, 2005). An explanation of these schemes can be found in Chapter 2. The simplicity of EDCA operation as compared with HCCA allows it to be implemented widely. Therefore, this study focuses on the IEEE 802.11e EDCA.

In contrast with the earlier legacy IEEE 802.11 DCF access method, IEEE 802.11e EDCA differentiates traffic according to their QoS requirements as reported in (Liang et al, 2006), (Rauf et al, 2009), (Alahmadi et al, 2008), and (Peng et al, 2010). For instance, in (Liang et al, 2006), stations operating the IEEE 802.11e EDCA were compared with IEEE 802.11b for transmitting a number of applications. Results showed that IEEE 802.11e EDCA mechanism provided a better QoS as compared with IEEE 802.11b DCF. An evaluation reported in (Rauf et al, 2009) showed that IEEE 802.11e EDCA introduced an effective service differentiation mechanism and provided QoS support under light network load. Peng et al (2010) presented an analytical model to study the performance of EDCA service differentiation schemes. The general effectiveness of EDCA service differentiation was proved through analytical and simulation results.

However, a number of studies have demonstrated the limitations of IEEE 802.11e EDCA for efficiently handling a variety of traffic types in congested networks. The main reason was the static nature of resource allocation inherent in IEEE 802.11e. For instance, the inadequate QoS support for multimedia traffic in high load conditions using IEEE 802.11e EDCA wireless infrastructure was demonstrated in (Politis et al, 2011). The proposed mechanism so called X-EDCA was designed to cope with high load traffic situations in IEEE 802.11e. The scheme improved the QoS for multimedia traffic in infrastructure IEEE 802.11e networks. The transmission performance of IEEE 802.11e EDCA was evaluated and reported in (Villalón, et al, 2007). The study showed that EDCA was unable to guarantee a good performance when the network traffic load was high. The main reason was the excessive number of packet collisions, which in turn was due to the fixed transmission parameters values assigned to the ACs. Lin et al (2009) proposed an adaptive cross layer mapping algorithm was devised to improve the quality of MPEG-4 video transmitted over IEEE 802.11e EDCA. The proposed algorithm outperformed the 802.11e protocol by dynamically mapping MPEG-4 video packets to appropriate access categories, according to the network's traffic load and the significance of video frames.

Most previous studies either supported QoS provided by IEEE 802.11 EDCA for high priority traffic, which may starve other transmitted traffic as in (Politis et al, 2011), or required modification of all wireless stations, which in turn complicates the WLAN operation as in (Lin et al, 2009). The contribution of this study is the development of an adaptive allocation algorithm at the wireless Access Point (AP) to further improve the QoS of IEEE 802.11e EDCA. The algorithm determines the Packet Arrival Rate (PAR) of the uplink and downlink traffic for each Access Category (AC). The algorithm then dynamically allocates the traffic from a lower priority AC to the next higher AC, when there is no traffic for the higher AC. The algorithm enables lower priority traffic to have access to a higher priority AC ensuring more efficient use of network resources.

Most current wired networks which are based on IEEE 802.3 have low bit error rates in transmitting multimedia applications as compared with wireless networks. However, one of the most challenging tasks that affects the QoS in wired is traffic congestion. Congestion occurs when multiple stations transmit traffic simultaneously using the same router. Congestion has a negative impact on the network performance as increases the probability of packet loss ratio and collision.

Several mechanisms and approaches were used to overcome the QoS issues and enhance network performance (Farrel, 2008). One of these approaches is over-provisioning which refers to enhancing the network capability by simply providing the network with enough bandwidth in order for all traffic to meet their QoS requirements (Fraleigh et al, 2003). Other approaches such as traffic class, resource reservation, and queuing mechanism utilise available network resources according to the application's QoS (Farrel, 2008).

Over provisioning might be preferred in core networks, such as Internet backbones (Fraleigh et al, 2003) (Menth et al, 2006). However, in the context of integrated wired and wireless, it is preferred to guarantee QoS using service differentiation mechanisms rather than over provisioning because the latter can be difficult and costly. For instance, telecommunication providers prefer Admission Control (AC) to guarantee QoS in packet-switched communication rather than Capacity Over provisioning (CO) which can be complicated and expensive task (Menth et al, 2006). D'Antonio (2003) pointed out that committing resources is not sufficient since QoS degradation is often unavoidable due to any fault in the behaviour of network elements or lack of using sufficient resources.

There are several mechanisms to support QoS in wired networks as demonstrated in Section 2.3.2 in Chapter 2. In this study, the network traffic prioritisation will be investigated as one of the most important issues that affect the QoS in wired networks. Therefore, packet scheduling mechanisms which are the most commonly employed congestion-management tools will be considered.

Several studies have used packet scheduling mechanisms to improve network QoS. For instance, Frantti and Jutila (2009) proposed Adaptive Weighted Fair Queuing (AWFQ) to differentiate service for traffic according to QoS requirements. The study showed that AWFQ achieved improved results than traditional WFQ. Epiphaniou et al (2010) discussed the performance of three different mechanisms: FIFO, RED and DiffServ, and their effects on real-time voice traffic. The experimentation results proved that under burst traffic conditions up to a congestion level, DiffServ seems to perform better on all three categories examined, by mainly providing a better queue management and throughput, reduced packet drop rates based on data transmitted, and One Way Delay (OWD) within the acceptable levels for an acceptable voice call. Miaji and Hassan (2010) investigated the performance of three scheduling mechanisms: First Come-First-Serve, Priority Queuing, and Weighted Fair Queuing. According the their simulation results, WFQ provided a better enhancement for multimedia applications and hence a higher QoS. Balogh and Medvecký (2011) carried out a comparison between Weighted Fair Queuing (WFQ), Worst-case Fair Weighted Fair Queuing+ (WF2Q+) and Weighted Round Robin (WRR) from the point of their usage in modern converged telecommunication networks. Simulation results showed that WFQ and WF2Q+ provide fair distribution of bandwidth for variable length packets due to the calculation of packet size at cost of high computing and memory usage requirements which limited their usages in high speed backbone network. On the other hand, WRR was a quick algorithm with low computing requirements which allows its usage in high speed backbone networks though WRR does not take into the account the length of packet size.

From the above discussion, the trade-off between different queuing mechanisms is their complexity, control ability and level of fairness. An aspect of this study is an appropriate utilisation of a suitable queuing scheduling mechanism. FIFO does not support QoS because it treats traffic equally. PQ provides premium service to the high priority traffic at the expense of the lower priority traffic, causing the latter traffic to experience excessive delay. FQ is not designed to support traffic with different QoS requirements, as it allocates the same amount of bandwidth among multiple traffic. The computational complexity of WFQ algorithm affects its scalability to support larger traffic with different requirements at the edge of the network (Semeria, 2001), (Epiphaniou et al, 2010), and (Balogh and Medvecký, 2011).

WRR addresses the limitations of FIFO, PQ, and FQ by classifying traffic based on their QoS requirements, and ensuring that low priority traffic can access to buffer space and output port bandwidth. The implementation of WRR is more popular and its operation is less complex as compared with WFQ. Therefore, in this study, WRR was considered to provide traffic prioritisation because of its practicality and low complexity.

3.5 Quality of Service Monitoring Tools

With the provision of transmitting multimedia applications over computer networks, it is crucial to monitor network performance to ensure that the QoS of these applications is being sustained. Therefore, based on the need of monitoring QoS, many tools have been proposed to monitor network performance. For example, the QoS monitoring tool proposed by Graham et al (1998) was used to assess packet delay, jitter, and packet loss as an indication of network performance. The Surveyor tool proposed by Zseby and Scheiner (2004) was used to assess end-to-end delay and packet loss to measure network performance. The Surveyor tool included three main components: the measurement machine, the database, and the analysis server. The network QoS matrices were collected periodically by the measurement machine which in turn buffered them to the database. Finally, the network QoS matrices were analysed by the analysis server. In (Malhotra et al, 2011), a tool for QoS monitoring of multimedia sessions so called StreamTrack was proposed. The tool was deployed in an IP infrastructure with heterogeneous network access technologies. StreamTrack was used to calculate the QoS parameters (i.e. bandwidth, delay, and jitter) of multimedia sessions such as Voice/Video Call, and Internet Protocol Television (IPTV) sessions. Another QoS monitoring tool that has been proposed and implemented for multiservice networks was QMon (Carvalho et al, 2009). This tool was used to measure the QoS of distinct service classes between network measurement points. The measurement results can be accessed directly from the monitoring database or from a web interface available on the QServer. QMon, as a multiplatform and generic tool, is a versatile and cost-effective QoS monitoring solution to be deployed in multiservice network environments, being useful to assist traffic engineering tasks, service management and auditing.

However, the existing QoS monitoring tools have some limitations. For instance, they are not determine directly the overall network QoS as in (Graham et al, 1998) and (Malhotra et al, 2011). Network managers have to do a variety of operations to assess the overall network QoS. Also, these tools were not developed to work as stand-alone device as in (Zseby and Scheiner, 2004) and (Carvalho et al, 2009). From these limitations, it can be concluded that the process of monitoring QoS can be complicated, expensive, and time consuming. Therefore, developing a portable hand-held device that accurately determines the overall network QoS for multimedia applications can be very valuable. In this study, a mechanism that assesses QoS which in turn taking into the account the QoS requirements of multimedia applications is implemented on a portable microprocessor board to build QoS monitoring tool. The proposed tool could work on its own to assess the QoS of multimedia applications based on their QoS requirements.

3.6 Application of Statistical and Artificial Intelligent Techniques to Computer Network

Statistical and artificial intelligent techniques have been widely used in the area of computer network for various tasks such as analysis, optimisation, and evaluation. A number of studies have applied regression model as one of the most popular statistical techniques in networking domain. For example, Akinaga et al (2005) proposed a method based on regression analysis for forecasting network traffic using the user's properties and information about mobile network environment. The technique was used to predict traffic fluctuations for a mobile network area. An enhancement of the Peak Signal-to-Noise Ratio PSNR method to evaluate video streaming quality was introduced by (Chan et al, 2010). The modified PSNR was based on linear regression technique which in turn was used to derive two specific objective video quality metrics, PSNR-based Objective MOS (POMOS) and Rates-based Objective MOS (ROMOS). Linear regression prediction model was proposed to evaluate network security situation (Xia and Wang, 2010).

Artificial intelligence techniques on the other hand were used widely in the area of computer networks. For instance, Khoukhi and Cherkaoui (2008) proposed a fuzzy logic system called FuzzyMARS for call admission control and service differentiation for wireless ad hoc networks. Their results showed that FuzzyMARS achieved service differentiation delivery and reduced delay transmission of multimedia applications. Frantti and Jutila (2009) embedded fuzzy expert system for Adaptive Weighted Fair Queuing (AWFQ) to differentiate service for traffic according to QoS requirements. The simulation results showed that AWFQ reacted faster to differentiate traffic classes than traditional WFQ. Fuzzy logic was used to assess QoS and adjust the minimum contention window (CW_{min}) in mobile ad hoc networks (MANET) (Seriareh et al, 2008) and (Al-Sbou, 2010). The devised approach effectively assessed and improved the QoS of multimedia applications in MANET.

Several studies have used FCM clustering in network traffic domain. For instance, a network administrator assistance system was proposed based on FCM (Chen et al, 2009). The proposed system utilised a FCM method to analyse users' network behaviours and traffic-load patterns based on the measured traffic data of an IP network. Analysis results can be used to assist administrators to determine efficient controlling and configuring parameters of the network management. In wireless sensor networks, a FCM algorithm was used in order to create clusters which reduced the spatial distance between sensors nodes (Hoang et al, 2010). A FCM clustering algorithm was also developed to detect routing attacks caused by abnormal flows in a wireless sensor network. The study demonstrated that FCM can be a valuable tool for intrusion detection (Wang et al, 2009).

Neural network has been also used in the area of computer networks. In (Nogueira et al, 2006), a modelling approach based on a neural network was proposed to predict the media access delay in WLAN. The prediction of the model was accurate even when the number of active nodes was changed significantly. Multilayer perceptron MLP was used to predict packet loss of real-time video transmission. The results showed that MLP was capable of predicting the number of lost packets with 96% accuracy (Lavington et al, 1999). Self-organising map was used to cluster network traffic types and produce application profiles, which contained significant information about the current status of the network, in order to manage network resources (Timo et al, 2002) and (Kiziloren, and Germen, 2007). A network traffic classification approach based on Kernel-SOM (KSOM) was proposed in (Ting et al, 2010). The experimental results showed that the Kohonen SOM achieved high classification accuracy and successfully categorised network traffic characteristics.

The contribution of this research is to develop approaches using the aforementioned statistical and AI techniques to intelligently manage QoS for integrated wired and wireless network carrying multimedia applications. Fuzzy Inference System will be used to sample multimedia traffic in adaptive manner, Fuzzy C-Means clustering techniques and Self Organizing Map will be devised to analyse the QoS parameters in order to provide valuable information about the network's QoS patterns, and regression model and Multi-Layer Perceptron will be employed to combine the analysed QoS parameters and then produce a single value that represented the overall network's QoS.

3.7 Summary

The main objective in this chapter was to provide an extensive literature review of previous studies in the area of managing QoS of multimedia computer networks. The main QoS management issues of multimedia networks that need further development and investigations were discussed. Firstly, the state of the art of sampling techniques for multimedia traffic was reviewed. Then, the current QoS analysis and assessment techniques used to evaluate network QoS were reviewed. The relevant studies that considered the QoS support in wireless and wired networks were discussed. The exiting monitoring tools used to assess the network performance were critically analysed. The applications of statistical and artificial intelligent techniques used within this study into the field of computer network management were reviewed.

The next chapter focuses on a description of the experimental approaches that were carried out to evaluate and validate the proposed techniques involved in the process of managing QoS.

Chapter 4 Experimental Methodology

4.1 Introduction

An explanation of network evaluation methodologies, tools, and general experimental procedure used throughout this study is provided in this chapter. A more detailed discussion of the procedures which are specific to individual studies are discussed later in the relevant chapters. Section 4.2 of this chapter covers an overview of network evaluation approaches, network simulation tools, description of simulation environment and protocols, and traffic type and characteristics. The measurement processes which include a description of QoS metrics and requirements, and analysis procedure of simulation output are included in section 4.3. The main issues are summarised in section 4.4.

4.2 Network Evaluation Approaches

The aim of this section is to reveal the appropriate network evaluation approach that suited the objectives of this study. Three main approaches were used to evaluate network performance: analytical modelling, measuring physical networks, and simulation (Jain, 1991). It was not practically feasible to use the analytical modelling techniques in this study since computer networks, in particular for wireless networks, have a dynamic behaviour. For instance, new traffic comes online, while others terminate, the route that packets take can vary, and the bandwidth availability can vary considerably over time. Evaluating network performance based on measuring physical networks was also excluded because it is time consuming as well as the most expensive approach among other approaches. Also, implementation of the devised methods in realistic size and complexity networks could not be implemented in the duration of this study. Therefore, an appropriate approach that fitted to this research was based on simulation. Simulation could provide a rich environment for experimentation at low cost. In contrast with analytical modelling techniques, simulation techniques may achieve more accurate results since they are often closer to the reality. Comparing with measuring real networks, simulations could have more control over the network conditions and allow changes to the network settings in more effective manner (Bajaj et al, 1999).

4.2.1 Network Simulation

Network simulation approach can be used to serve a variety of network engineering needs. It allows engineers to simulate networks with realistic topologies in an effective and inexpensive manner (Siraj et al, 2012). There are many simulation tools used in network engineering research community (Jain, 2008), (Weingärtner et al, 2009), (Sarkar and Halim, 2011) and (Siraj et al, 2012). The most popular simulation tools are Optimized Network Evaluation Tool OPNET (OPNET, 2012), Global Mobile Information System Simulator (GloMoSim) (GloMoSim, 2012), Optical Micro-Networks Plus Plus (OMNET++) (OMNET++, 2012), and NS-2 from Virtual Internetwork Testbed project (VINT) (NS, 2012) and (Bajaj et al, 1999). The full version of OPNET has a complete set of features with a well-developed Graphical User Interface (GUI), but it is not open source, restricting the scope for customising its operation. Although GloMoSim and OMNET++ are open source simulation tools, they only support wireless networks (Jain, 2008) and (Sarkar and Halim, 2011). NS-2 on the other hand is an open source and freeware simulation tool in nature. Hence, network engineers tend to use NS-2 in order to test new protocols or modifying the existing ones in a controlled, reliable and reproducible environment (Lucio et al, 2003). Moreover, NS-2 can carry out trace-driven simulation using a record of events from a real system (Caro, 2003). Therefore, this study will be based on NS-2.

NS-2 is a discrete event simulator based on two object oriented languages: C++ with an Object Tool Command Language (OTcl) as shown in Figure 4-1. NS-2 has a rich set of protocols such as TCP and UPD and traffic source behaviour such as FTP, Telnet, Web, CBR and VBR (Chung and Claypool, 2004). NS-2 is also capable of transmitting video streaming such as MPEG-4 and H.264 using Evalvid framework tool-set (Ke et al, 2008). Full details about the general architecture of the network components in NS-2 can be found in the documentation supported by NS-2 group (Fall and Varadhan, 2011). In the simulation process of NS-2 as shown in Figure 4-1, users generate Tool Command Language (TCL) script files to specify network topology, traffic applications, and all the required settings. The TCL files are then handled by the C++ libraries and OTcl interpreter. After the termination of simulation process, NS-2 produces simulation results in two output files: NAM file and trace file. The NAM file is used to support graphical tool called Network Animator (NAM) to visualize simulation traces. The trace file contains information about packets (i.e. transmitted packets, received packets, dropped packet, packet types, packet ID, etc) (NS, 2012). NS-2 Users can extract relevant information from the output trace file using script languages such as Awk, Perl, or Grep and plot them using other script languages such as Xgraph, Gunplot, or Matlab (Kumar, 2008).

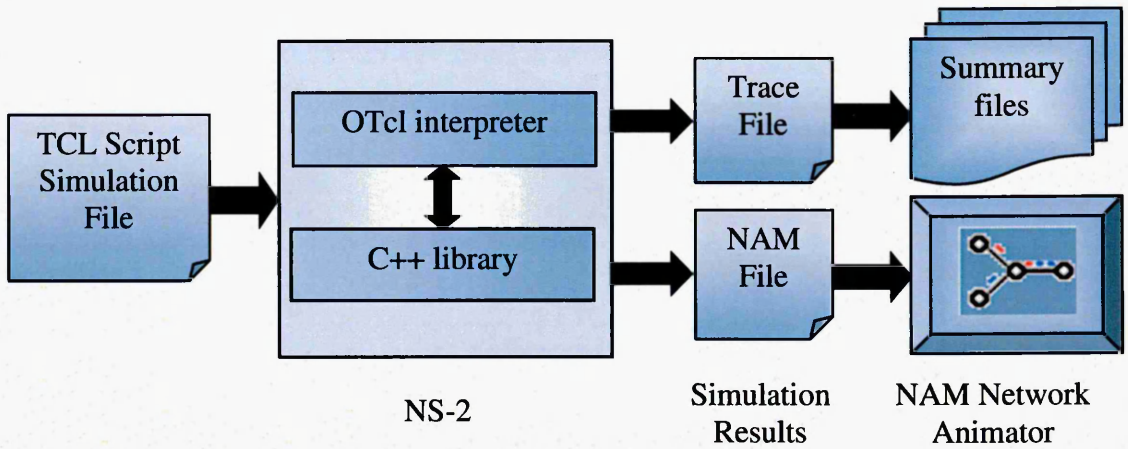


Figure 4-1. The simulation process of NS-2.

4.2.1.1 An overview of Evalvid Framework

Evalvid framework tool set can be integrated within NS-2 simulator to facilitate the transmission of real video applications over simulated networks. The formats of the video applications supported by Evalvid framework which can be later transmitted using NS-2 are YUV QCIF (176 × 144) or YUV CIF (352 × 288) (Zhou and Sik-Jang, 2008). Figure 4-2 shows the component of Evalvid framework and NS-2 simulator used to transmit real video applications. The process of transmitting a video application using Evalvid framework and NS-2 simulator is as follows (Ke et al, 2008) and (Abdel-Hady and Ward 2007):

- **i.** Video Encoder: the encoder is used to convert the source video file from YUV format to a compressed H.264 or MPEG4 format at the sender side.
- **ii.** Video Trace Generator: this component reads the compressed H.264 or MPEG4 file generated by video encoder and then fragments each video frame into smaller segments of 1000 bytes for transmission, if the size of the video frame is larger than the preset maximum packet size (Maximum packet length is 1028 bytes, including 20 bytes for IP header and 8bytes for UDP header). Video trace generator produces a video trace file that contains information about every frame in the real video file.
- **iii.** During the simulation process of NS-2, there are three agents implemented between Evalvid framework and NS-2. These are MyTrafficTrace, MyUDP, and

MyUDPSink agent. MyTrafficTrace agent reads the frame type and the frame size from the video trace file, and sends these frame segments to the transport layer according to the preset time settings specified in the simulation script file. At the transport layer, MyUDP which is an extension of UDP agent allows users to specify the output file name of the sender trace file and then records the timestamp of each transmitted packet, the packet ID, and the packet size. At the transport layer of the receiver side, MyUDPSink agent receives the fragmented video frame packets sent by MyUDP and then records the timestamp for each received packet, packet ID, and packet size in the receiver trace file specified by user.

iv. Video Reconstruction: after simulation is terminated, the three trace files (i.e. video trace file, sender trace file, and receiver trace file) are reconstructed by the video reconstruction component to produce the received video file in a compressed H.264 or MPEG4 format.

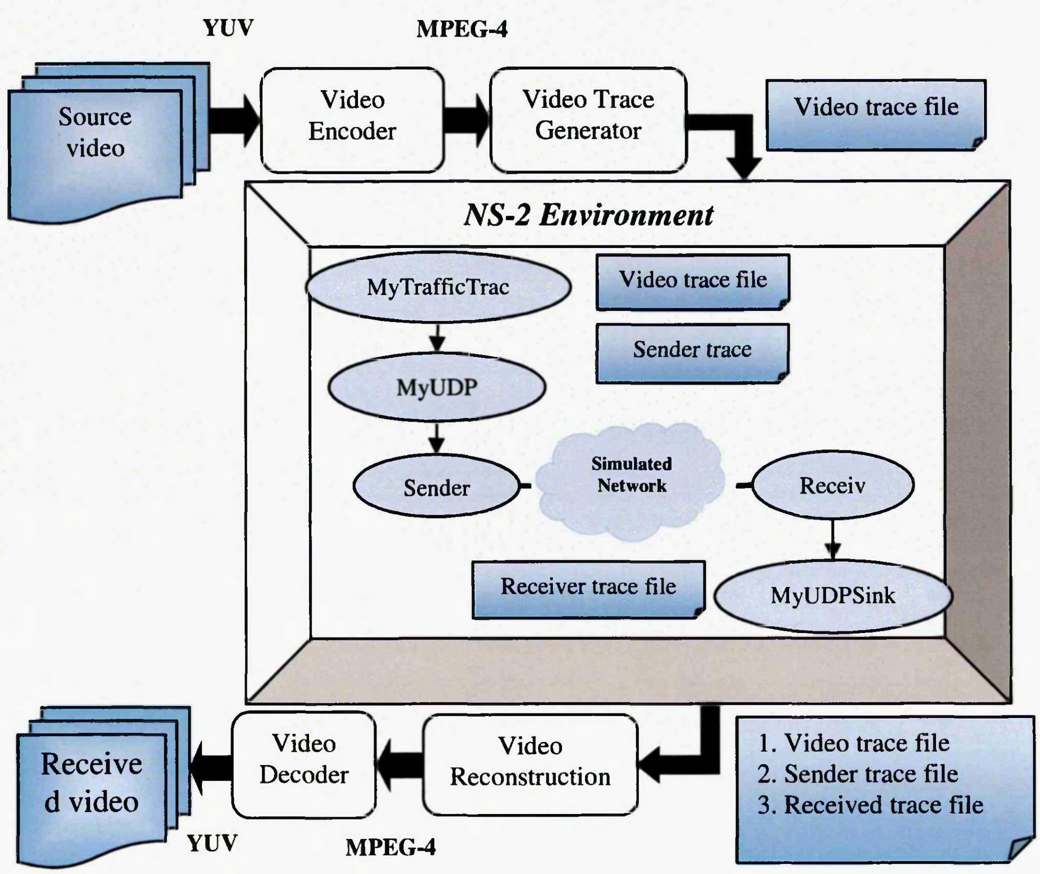


Figure 4-2. The process of transmitting video using Evalvid framework and NS-2.

4.2.2 Network Topologies

Wireless-cum-wired network topologies with different sizes (i.e. small, medium, and large network) were simulated using the Network Simulator-2 (NS2) as shown in Figure 4-3. As the number of stations plays an important role in network performance, the number of stations in the simulated network was varied from 8 to 64 stations according to selected scenario. The connections between stations in wireless-cum-wired network topology were unidirectional. The number of these connections was varied from 4 to 32 according to the number of transmitted stations. In most simulation scenarios, half of the connections transmitted traffic from wireless to wired network, whereas the other half transmitted traffic from wired to wireless. Each station transmitted one type of traffic to its corresponding destination. At the wired side of network, all links had 5 Mbps bandwidth and 2 ms propagation delay. The WLAN side of the network was based on IEEE 802.11e, and it used the Enhanced Distributed Channel Access (EDCA) scheme. The main parameters that modelled the wireless channel were the default settings for IEEE 802.11e as shown in Table 4-1.

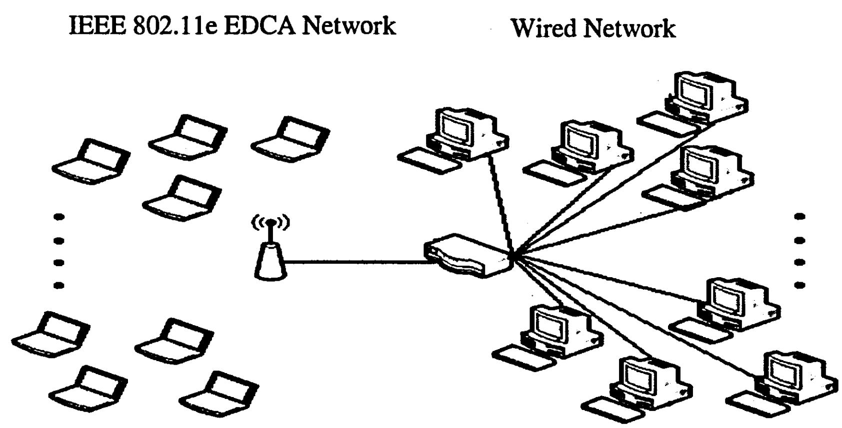


Figure 4-3. The simulated network topology.

The network topology covered an area of 500m × 500m and the stations were positioned randomly within the specified area. This position was fixed during the simulation time. Simulations were repeated 10 times. Each time a different initial seed value was used to randomly position the stations and manage which node transmitted first, as all nodes were requested to transmit at a given time. The randomness introduced using different seeds avoided the bias of random number generation. The results of the 10 simulations were then averaged. Simulation time was between 300 - 500 seconds. These settings were considered appropriate to examine the long term behaviour of the IEEE 802.11e protocol.

4.2.3 Physical Layer (PHY) Parameters

At the wireless side of the network, the main physical layer parameter considered was channel bit rate which includes basic rate for control frame transmission and data rate for data transmission. Basic rate was set at 1 Mbps, while data rate was set to be 2 Mbps or 11 Mbps for some selected simulation scenarios. The physical layer was modelled to work as Lucent WaveLAN at a frequency of 914 MHz and DSSS radio interface card (Fall and Varadhan, 2011). A summary of PHY layer parameters of this model is listed in Table 4-1 (NS, 2012).

4.2.4 Medium Access Layer (MAC) Parameters

In this research study, the MAC layer was based on IEEE 802.11e at the wireless side of the network. This standard provides two schemes to access the wireless channel. These are: Enhanced Distributed Channel Access (EDCA) and Hybrid Controlled Channel Access (HCCA) (IEEE, 2005). This study focused on the IEEE 802.11e EDCA due to its simplicity as compared with HCCA. The main parameters that modelled EDCA were the default settings for IEEE 802.11e. A summary of MAC parameters of this model is listed in Table 4-1 (NS, 2012).

Table 4-1. Simulation settings of MAC and PHY parameters in IEEE 802.11e.

Parameter	Value
Capture Threshold	10
Carrier Sense Threshold	1.559e-11
Receiving Threshold	3.652e-11
Power Transmission	0.28183815
Frequency Band	914e+6
Data Rate	2.0 - 11.0 Mbps
Basic Rate	1.0 Mbps
Modulation Technique	DSSS
PHY Header	24 bytes
MAC Header	28 bytes
SlotTime	20μsecs
SIFS	10μsecs
Preamble Length	144 bits
PLCP Header Length	48 bits
RST Threshold	3000
ShortRetryLimit	7
LongRetryLimit	4

In this study, the transmitted traffic over IEEE 802.11e EDCA were VoIP, video, best effort traffic, and background traffic. These traffics were mapped into different Access Categories (ACs) to represent different levels of priority as shown in Table 4-2 (IEEE, 2005).

Table 4-2. IEEE 802.11e access categories parameters.

Type of traffic / Parameters	VoIP	Video	Best effort traffic	Background traffic
AIFS	2	2	3	7
CW_{min}	7	15	31	31
CW_{max}	15	31	1023	1023
TXOP	3.008	6.016	0	0

Classifying the traffic into different ACs was based on their QoS requirements. Due to the high sensitivity of VoIP to QoS parameters, it was assigned to AC with the smallest values of AIFS, CWmin, CWmax and largest value of TXOP. In contrast, background traffic was assigned to AC with the largest values of AIFS, CWmin, CWmax and smallest value of TXOP because of its tolerance to some QoS parameters such as delay. Accordingly, VoIP had the highest priority, whereas the background traffic had the lowest priority.

4.2.5 Routing Protocols

Several routing protocols are available in NS-2 environment. These include, Destination-Sequenced Distance Vector (DSDV), Ad Hoc On-Demand Distance Vector Routing (AODV), Temporally Ordered Routing Algorithm (TORA), and Dynamic Source Routing (DSR) (Gupta and Saket, 2011) and (Said and Saatchi, 2009). DSDV protocol is based on the Bellman-Ford routing algorithm. It uses the proactive table-driven routing strategy. Whereas AODV, DSR, and TORA are reactive on-demand routing protocols which initiate route discovery mechanism to establish a route between the source and destination nodes (Shah et al, 2008). In this study, DSDV was chosen as it maintains the routing information for all the nodes in the network and adds a new route or update the existing routes periodically. This ensures the routes to any destination are ready to use when needed (Elashheb, 2012).

4.2.6 Queuing Mechanisms

In the network, when multiple packets are serviced through a congestion point such as a router, queuing mechanisms are required to determine the bandwidth allocation among transmitted packets and the manner in which to service various applications with different QoS requirements. In this study, two queuing scheduling mechanisms were employed. These were First-In-First-Out (FIFO), and Weighted Round Robin (WRR). Although no preference was given to the transmitted traffic regarding to its QoS requirement in FIFO queuing scheme as its basis was first packet come first packet served, FIFO was employed in most simulation scenarios due to its management simplicity and implementation popularity. The default queue size of FIFO used in this study was 50 packets. In some experimental scenarios in Chapter 7, WRR queue scheduling mechanism was implemented between the router and the Access Point (AP) at the wired side of the network to improve its QoS. WRR addresses the limitations of FIFO by classifying traffic based on their QoS requirements, and ensuring that different traffic priorities can access the buffer space and output port bandwidth (Semeria, 2001). The operation of WRR is explained under Section 2.3.2 in Chapter 2. In this study, time-sensitive applications had a higher priority than time-insensitive applications. This was because the former had larger weights than the latter as shown in Table 4-3.

Table 4-3. WRR Parameters.

Preset No. of queues in WRR	Queue 1	Queue 2	Queue 3	Queue 4
Application type	VoIP	Video streaming	Best effort traffic	Background traffic
WRR weights	3	3	2	2
Queue length	25	25	25	25

4.2.7 Traffic Type and Traffic Characteristics

In this study, different types of traffic were transmitted over the simulated networks. These were: Voice over IP (VoIP), videoconferencing, video streaming, best effort traffic, and background traffic. Constant Bit Rate (CBR) traffic was adapted to model VoIP, videoconferencing, and best effort traffic. The VoIP packet size was 160 bytes and its inter-packet interval was 20 ms, corresponding to G.711 voice encoding scheme with 64 kbps Pulse Code Modulation (PCM) voice flows. The packet size of the video traffic was 512 bytes and the inter-packet interval was 10 ms. This generated a packet transmission rate of 384 kbps (Markopoulou et al, 2003) and (Saraireh et al, 2007).

The video streaming sources were YUV QCIF (176 × 144) Foreman (400 frame) and YUV QCIF (176 × 144) Highway (2000 frame) (YUV QCIF, 2012). Prior to the transmission, each video frame was fragmented into packets, which in turn had a maximum length of 1024 bytes. Video streaming applications were encoded using MPEG-4 encoding scheme which defines three types of video frames: I (Intra-coded) frame, P (Predictive-coded) frame, and B (Bidirectionally predictive-coded) frame as shown in Table 4-4.

Table 4-4. The number of video frames and packets of the video streaming sources.

Video	format	Number of frames			Total	Number of packets			Total
		I	P	B		I	P	B	
Foreman	QCIF	45	89	266	400	237	149	273	659
Highway	QCIF	223	445	1332	2000	461	451	1333	2245

The I frame is encoded and decoded independent of previous or successive frames. The encoding of P frame requires information from preceding I or P frame in the video sequence. The predictions from previous and successive I or P frames are also required to encode the B frame. According to MPEG-4 scheme, the I frame is the most important frame among other types of frame. Comparing P and B frames, MPEG-4 scheme specifies that the former frame is more important than the latter (Lin et al, 2009). During the decoding process, the video frames can be decompressed into Group Of Pictures (GOP), which its pattern is defined by two parameters G (M, N), where N is the I-to-I frame distance and M is the I-to-P frame distance as shown in Figure 4-4 (Zhou and Sik-Jang, 2008).

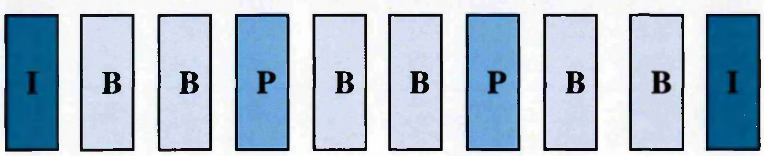


Figure 4-4. GOP sequence in MPEG-4

The other types of traffic transmitted over the simulated networks were the best effort traffic which modelled using CBR with different packet sizes and generation rates that corresponded to non-videoconferencing or VoIP usage. The packet size of 200 bytes with 12.5 ms inter-packet interval was used to generate 128 kbps data rate, while large packet size of 1000 bytes with 12.5 ms packet interval was to generate 500 kbps data rate. File Transfer Protocol (FTP) application was also used as background traffic. FTP was transmitted over TCP, whereas other traffics were transmitted using UDP transport protocol.

4.3 Analysis of Simulation Output

Due to their importance in the process of QoS evaluation, the transmission requirements of applications are considered during the analysis the simulation results. Throughout this research, delay, jitter, and packet loss ratio are considered to be the main QoS parameters. These parameters and the manner they were calculated are explained in the following sections.

4.3.1 Transmission Requirements of Applications

According to the application type, the QoS requirements of time-sensitive applications are significantly different from time-insensitive applications. The latter applications such as email, web browsing, and file transfer can be tolerant with QoS parameters such as delay and jitter. However, the former applications (i.e. multimedia applications) such as VoIP and video applications are highly sensitive to QoS parameters, thus requiring a faster response from the network. A large delay, jitter, or packet loss ratio can seriously degrade their quality (Kurose and Ross, 2005). There are some factors that pose challenges to prevent network providing sustained QoS for transmitted applications. These include network congestion in wired networks and interference problems in wireless networks. Therefore, the QoS requirements for traditional and multimedia applications must be considered to provide QoS for these applications. Table 4-5 summaries the QoS requirements for time sensitive and insensitive applications as recommended by ITU group (ITU-T, 2001) and (Zhai et al, 2005).

Table 4-5. QoS requirements for voice, video, and data as recommended by ITU group (Zhai et al, 2005).

Class	Application	Delay	Jitter	Packet loss rate
Time-sensitive	VoIP	< 150 ms preferred	< 1ms preferred	< 3 % preferred
	Video	< 150 ms preferred	< 1ms preferred	< 1 % preferred
Time-insensitive	E-mail, file transfer, web browsing	Minutes	N/A	Zero

4.3.2 Calculation of QoS Parameters

This section explains the calculation method of QoS parameters (i.e. delay, jitter, and packet loss ratio). After the TCL script file is simulated by NS-2, a detailed trace file is generated which contains extensive information about transmitted traffic. This information include: packet status (i.e. departed, arrived, and dropped), its timestamp, packet ID, packet type, packet size, flow ID, sequence number, node ID, source and destination addresses.

In this study, the QoS parameters were extracted from the data trace file using AWK script language (Aho et al, 1988). Delay calculation process was associated with three main fields: packet sent time, packet received time, and its unique ID. To calculate delay, the sent time a packet was subtracted from the received time for the same packet. Equation 4.1 illustrates how delay is calculated for particular packet:

$$D_i = R_i - S_i \tag{4.1}$$

Where D_i is the delay of the i^{th} packet arrived, R_i and S_i are the arrival and sending timestamps of the i^{th} packet.

Jitter was computed by calculating the absolute value of the difference between two consecutive packets delays as shown in equation 4.2:

$$J_i = abs\ (D_i - D_{i-1}) \tag{4.2}$$

Where J_i is the absolute jitter of the i^{th} packet, D_i is the delay of packet i, and D_{i-1} is the delay of the previous packet.

The percentage of packet loss ratio during certain time interval was calculated based on the total number of received packets with respect to the total number of transmitted packets during that time interval as in equation 4.3:

$$PL_i(t) = 100 \times \left(1 - \frac{\sum R_i(t)}{\sum S_i(t)}\right) \tag{4.3}$$

Where PL_i is the loss ratio in percentage (%) during the i^{th} interval, and $\sum R_i\ (t)$ and $\sum S_i\ (t)$ are the total number of received and transmitted packets with the i^{th} interval respectively.

When the values of QoS parameters were calculated, they were fed to MATLAB for further analysis (MATLAB, 2012). This analysis included averaging the values of QoS parameter for every n consecutive packets. The averaged values were then normalised and limited in order to ensure that all values had the same contribution in QoS evaluation process.

4.4 Summary

This chapter described the experimental procedure used to evaluate and validate the techniques proposed throughout this study. The use of NS-2 to simulate computer network scenarios (wireless-cum-wired network) was discussed. The network settings including routing protocols, queuing mechanisms, PHY, and MAC layer parameters were also explained. The characteristics and the QoS requirements of applications transmitted over the simulated network were clarified. The chapter concluded by describing the calculation of QoS parameters using AWK script language and then MATLAB for further qualitative analysis. Chapters 5, 6, 7, and 8 rely on the outlined experimental approach discussed in this chapter in order to test, validate, and evaluate the proposed techniques to manage QoS of multimedia computer networks.

Chapter 5 Development and Evaluation of Adaptive Statistical Sampling Techniques for Multimedia Traffic

5.1 Introduction

The rapid growth of real-time applications transmitted over multimedia networks, makes measurement of their traffic increasingly important. These measurements allow Quality of Service (QoS) for the transmission of the applications to be assessed. However, most real-time applications such as VoIP and video-conferencing generate an extensive amount of traffic data. Analysing these data in real-time is computationally intensive. Therefore, in order to reduce the amount of collected data and their processing, appropriate sampling techniques are required.

In fixed rate sampling techniques, the number of data packets processed remains unchanged even when traffic characteristics change. However, in adaptive sampling, the number of packets sampled varies in accordance with traffic fluctuations. This ensures appropriate amounts of data are processed.

In this chapter, statistical adaptive sampling techniques to adjust sampling rate based on traffic's statistics were developed based on a linear adjustment approach, quarter adjustment approach, and fuzzy inference system. A comparison of the devised methods versus conventional sampling techniques (i.e. systematic sampling, stratified sampling, and random sampling) was also carried out using a simulated computer network.

The organisation of this chapter is as follows: Section 5.2 reviews the state-of-the-art of adaptive sampling approaches which are used to reduce network QoS parameters. Section 5.3 includes a description of the proposed adaptive statistical sampling approaches, implementation of conventional sampling techniques, calculation of QoS parameters from sampled versions, methods of sampling analysis, and demonstrates the simulation set up. The experimental results are discussed in section 5.4. The summary of this chapter is provided in section 5.5.

5.2 Related Work

There were a number of studies conducted using adaptive sampling approaches to gather network information. These studies were reviewed in chapter 3. A number of these studies used fuzzy logic to sample traffic in an adaptive manner, such as (Hernandez, et al., 2001), (Giertl, et al., 2006), and (Giertl, et al., 2008). A further improvement of adaptive sampling, based on fuzzy logic control (FLC), was proposed by (Xin, et al., 2009). Modified FLC method can realize dynamic adaptive sampling making it more suitable for high speed networks.

Other adaptive sampling approaches were used for a variety of applications. These applications were: a control of resources allocation for network performance reported by (Graci`a et al, 2008), an estimate of traffic rate proposed by (Ma, et al., 2004), and capturing Denial of Service (DoS) attack packet as in (Zhang, et al., 2007).

In our work, novel statistical adaptive sampling methods were developed based on traffic's statistics (Dogman, et al., 2010_a), (Dogman, et al., 2010_b), and (Dogman, et al., 2011). The methods adjusted the sampling interval by considering the traffic's statistics between two consecutive sampled sections.

In this chapter, simple linear adjustment mechanism, quarter adjustment mechanism, and fuzzy inference system were devised considering traffic's statistics to adaptively adjust the sampling rate. A comparison of the devised methods versus conventional sampling techniques (i.e. systematic, stratified, and random sampling) was also carried out using a simulated computer network.

5.3 Adaptive Statistical Sampling Approaches

The applications of intelligent and non-intelligent methods for sampling multimedia traffic in an adaptive manner are described in this section. Three adaptive sampling approaches are proposed. The traffic length of the sampling interval for the three devised sampling techniques was controlled using three different mechanisms: simple linear adjustment mechanism, quarter adjustment mechanism, and fuzzy inference system. The approach that used fuzzy inference system was devised based on the theory of fuzzy logic, discussed in Chapter 2 (Section 2.4.2).

5.3.1 Description of Statistical Sampling Algorithm

The algorithm samples packets by considering the statistics of QoS parameters of transmitted traffic. Analysing QoS parameters of the traffic is important in order to ensure sampling process becomes tuned to the traffic characteristics. In this study, the statistics of throughput were considered during the sampling process. Multimedia applications such as videoconferencing and VoIP have high sensitivity to QoS parameters, such as throughput (Alkahtani, et al., 2003). Moreover, the statistics of throughput can be easily computed during the sampling process which gives an advantage to the algorithm to respond quickly to traffic changes. In this study, throughput was calculated using equation (2.1) (Wang, et al., 2000).

The devised statistical adaptive sampling algorithm had a number of operating parameters as shown in Figure 5-1. These parameters were:

- Pre- and post-sampling sections of the traffic: these traffic sections contained the packets to be sampled. The number of packets in these sections was not changed during the sampling operation.
- Inter-Sampling Section Interval (ISSI) of the traffic: the position of this traffic section was between the pre- and post-sampling sections. The length of this section was adaptively determined during the sampling process. A simple linear adjustment mechanism, quarter adjustment mechanism, and fuzzy inference system (FIS) were developed to adjust the length of ISSI. Section 5.3.2 demonstrates how the ISSI was adjusted using these mechanisms.
- Threshold value ρ : this value was used to assess the variation between the statistics of the pre- and post- sampling sections. The threshold value influenced when the ISSI was increased or decreased.

Pre-Sampling Section	Inter-Sampling Section Interval (ISSI)	Post-Sampling Section

Figure 5-1. Operating parameters of adaptive statistical sampling algorithm.

A flow chart of the statistical sampling algorithm's operation is provided in Figure 5-2. The user is required to initialise the length of the pre- and post- sampling sections, length of Inter-Sampling Section Interval (ISSI), and threshold value ρ . The lengths pre- and post-sampling sections are the same and are not changed during the algorithm's operation. An increase in the length of pre- and post- sampling sections enlarges the sample size. As the goal of the algorithm is to produce smallest sample size with the highest accuracy, the user needs to select a suitable value for the length of the pre- and post-sampling sections, through experimenting with a number of different values. The initial length of ISSI was used as the current sample interval for the first iteration of the sampling process. However, the initial length of ISSI was not critical during the first iteration of sampling process because it will be adjusted in the following iterations.

The algorithm determined the statistics (i.e. mean, median and standard deviation) of the QoS parameter (i.e. throughput) for the pre- and post- sampling sections and then quantified their overall statistic using equation (5.1).

$$\textit{Overall Statistic} = (abs\ (mean1-mean2)/mean1) + (abs\ (median1-median2)/median1) + (abs\ (std1-std2)/std1) \quad (5.1)$$

Where *abs* represents the absolute value, *mean1* and *mean2* are the mean values of pre- and post- sampling sections respectively for the throughput of traffic being sampled, *median1* and *median2* are the median values of pre- and post-sampling sections respectively for the traffic throughput, and *std1* and *std2* are the standard deviation values of pre- and post-sampling sections respectively for the traffic throughput. The overall statistic assessed the discrepancy of statistics between pre- and post-sampling sections. The algorithm updates the new length of ISSI based on the comparison between the quantified overall statistic and threshold value ρ using simple linear adjustment mechanism, quarter adjustment mechanism, and fuzzy inference system. An explanation of these mechanisms is in section 5.3.2.

The above operation represents one update of the ISSI length. During the next step, the updated ISSI is the current sample interval, the current post-sampling section becomes the next pre-sampling section, and the location of next post-sampling section is determined based on the updated value of ISSI. This process is repeated until the traffic is fully sampled.

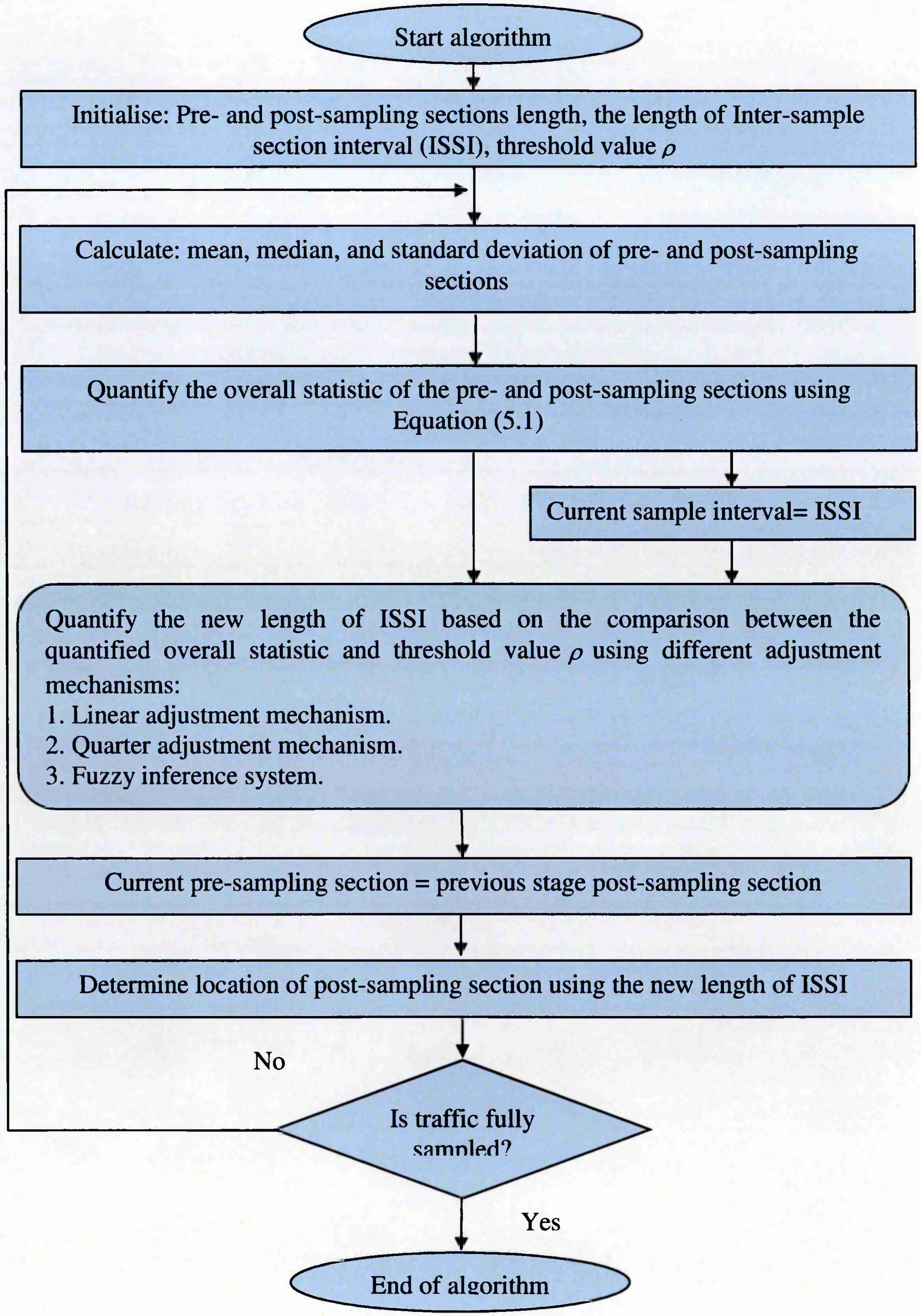


Figure 5-2. The flow chart of the adaptive statistical sampling algorithm.

5.3.2 Adjustment of Inter-Sampling Section Interval (ISSI)

Three adjustment mechanisms were proposed and incorporated into statistical sampling algorithm to adaptively adjust the length of Inter-Sampling Section Interval (ISSI).

These mechanisms are simple linear adjustment mechanism, quarter adjustment mechanism, and Fuzzy Inference System (FIS). These mechanisms are explained in the following subsections.

5.3.2.1 Linear Adjustment Mechanism of ISSI

This scheme examines the overall statistic along with predefined threshold value ρ and then linearly increases or decreases the length of ISSI. If the overall statistic value is less than the threshold, then the length of ISSI is updated using equation (5.2).

$$\textit{Updated ISSI} = \textit{current sample interval} + \mu_1 \tag{5.2}$$

If the overall statistics value is more than or equal to the threshold value, then the ISSI is updated using equation (5.3).

$$\textit{Updated ISSI} = \textit{current sample interval} - \mu_2 \tag{5.3}$$

The terms μ_1 and μ_2 in equations (5.2) and (5.3) control the update magnitude. The initial value for both μ_1 and μ_2 was 1. During the sampling operation, a further increase to the value of μ_1 and μ_2 is applied which in turn increases or decreases the length of ISSI linearly. However, in case of the overall statistic value is equal or more than threshold value ρ, the value of μ_2 was less than the length of current sample interval, at least by one.

5.3.2.2 Quarter Adjustment Mechanism of ISSI

In this approach, the calculated overall statistic was compared with the user specified threshold value ρ. If the overall statistic value is less than the threshold, then the Inter-Sampling Section Interval (ISSI) was updated using equation (5.4)

$$\textit{Updated ISSI} = \textit{current sample interval} + \textit{round}\ (\textit{ISSI}/\mu) \tag{5.4}$$

If the overall statistic value was more than or equal to the threshold value, then the ISSI was updated by equation (5.5)

$$\textit{Updated ISSI} = \textit{current sample interval} - \textit{round}\ (\textit{ISSI}/\mu) \tag{5.5}$$

The round function ensured that fractional values were rounded to the nearest integer value. The term μ in equations (5.4) and (5.5) updates the length of ISSI. The value of μ was determined though experimenting with different values. Small value of μ may significantly change the length of ISSI, whereas as larger values of μ make finer changes when ISSI length was updated. As the goal of the sampling algorithm is to produce smallest sample size with the highest accuracy, the user needs to select a suitable value of μ, through experimenting with a number of different values. In this study, the value of μ was determined though experimenting with different values and a value of 4 was chosen for this parameter. Therefore, the amount of ISSI change was a quarter of its previous length.

5.3.2.3 ISSI Adjustment using Fuzzy Inference System

In this approach, the Mamdani type of Fuzzy Inference System (FIS) was used to adjust the length of ISSI. Two inputs were fed into the FIS: the length of current sample interval and the overall statistic, which was used to measure the discrepancy of statistics between pre- and post-sampling sections. The overall statistic was calculated using equation (5.1).

Each fuzzy input variable was represented by five fuzzy sets to create input membership functions. The amount of overlap and the range of each variable were determined by experimenting with a number of suitable values and selecting the ones that gave best outcomes. The locations, the degree of overlap between the generated membership functions, and their corresponding fuzzy linguistics variables are shown in Figure 5-3 (a).

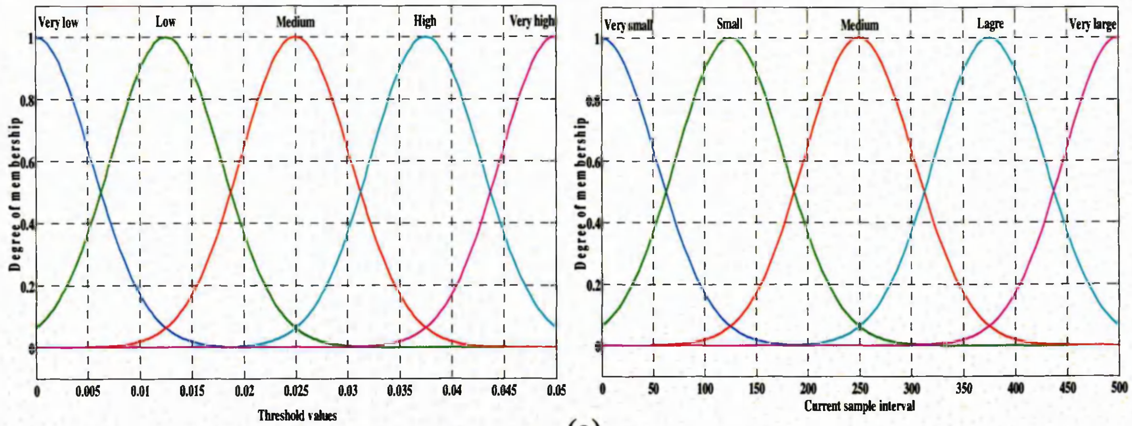
The fuzzy input variables were used to produce a single fuzzy output called Sample Interval Difference (SID) which was then used to determine whether ISSI should be increased or decreased or remain unchanged. The fuzzy output variable was also partitioned into five membership functions as shown in Figure 5-3 (b).

The fuzzy inputs and the output were fuzzified using the Gaussian membership function. This membership function is smooth and has concise notation. The mathematical formula of Gaussian membership function is expressed in equation (5.6):

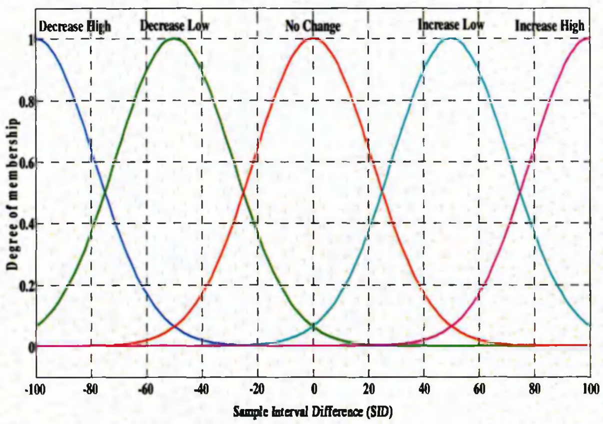
$$\mu_{A^i}(x) = exp\left(-\frac{(c_i - x)^2}{2\sigma_i^2}\right) \quad (5.6)$$

where c_i and σ_i are the mean and standard deviation of the i^{th} fuzzy set A^i, respectively (Saraireh et al., 2007).

Tables (5.1) - (5.3) show respectively the values of membership function parameters for fuzzy inputs (i.e. the length of current sample interval and the overall statistic) and fuzzy output (i.e. Sample Interval Difference SID).



(a)



(b)

Figure 5-3. Graphs of FIS for adjusting ISSI: (a) Fuzzy inputs, (b) Fuzzy output.

Table 5-1. Mean and standard deviation of overall statistic input fuzzy membership functions.

Membership functions	Overall statistics	
	Mean	St. dev.
Very low	0	0.005
Low	0.012	0.005
Medium	0.025	0.005
High	0.037	0.005
Very high	0.05	0.005

Table 5-2. Mean and standard deviation of current sample interval input fuzzy membership functions.

Membership functions	Current sample interval	
	Mean	St. dev.
Very small	0	53.09
Small	125	53.09
Medium	250	53.09
Large	375	53.09
Very large	500	53.09

Table 5-3. Mean and standard deviation of sample interval difference output fuzzy membership functions.

Membership functions	Sample interval difference	
	Mean	St. dev.
Decrease low (DL)	-100	21.24
Decrease High (DH)	-50	21.24
No change (NC)	0	21.24
Increase low (IL)	50	21.24
Increase high (IH)	100	21.24

The relationship between the inputs and the output was defined by a set of fuzzy rules. The number of fuzzy rules was set according to the number of inputs and their associated fuzzy sets. Examples of the rules are provides in Table 5-4.

Table 5-4. The fuzzy rules used by FIS to adjust ISSI.

Current sample interval	Overall statistic	Sample interval difference (SID value)
Very small	Very low	Increase high (IH)
Small	Very low	Increase high (IH)
Medium	Very low	Increase low (IL)
Large	Very low	Increase low (IL)
Very large	Very low	No change (NC)
Very small	Low	Increase high (IH)
Small	Low	Increase low (IL)
Medium	Low	Increase low (IL)
Large	Low	No change (NC)
Very large	Low	Decrease low (DL)
Very small	Medium	Increase low (IL)
Small	Medium	Increase low (IL)
Medium	Medium	No change (NC)
Large	Medium	Decrease low (DL)
Very large	Medium	Decrease low (DL)
Very small	High	No change (NC)
Small	High	Decrease low (DL)
Medium	High	Decrease low (DL)
Large	High	Decrease High (DH)
Very large	High	Decrease High (DH)
Very small	Very high	No change (NC)
Small	Very high	Decrease low (DL)
Medium	Very high	Decrease low (DL)
Large	Very high	Decrease High (DH)
Very large	Very high	Decrease High (DH)

Fuzzy reasoning (i.e. the process of implication and then aggregation) is based on (minimum-maximum) inference method. Each rule is applied to the corresponding membership function and the minimum is mapped into associated output membership function. The output fuzzy set from the implication process for each rule is combined together via aggregation process to produce one fuzzy set. In this study, the FIS output was generated from aggregated fuzzy set (i.e. defuzzification) using the centriod scheme. The centriod method returns the centre of area under the curve of the aggregated output values using equation (Al-Sbou et al, 2006) (5.7).

$$Y = \sum_{i=1}^{m} y_i \times \mu_i / \sum_{i=1}^{m} \mu_i \tag{5.7}$$

where m is the number of fuzzy sets obtained after implication, y_i is the centriod of fuzzy region i, and μ_i is the output membership value.

The adaptive sampling algorithm used the generated fuzzy output (i.e. Sample Interval Difference SID) along with the current sample interval to update the length of ISSI. Equation (5.8) is used to calculate the new length of ISSI.

*Updated ISSI = round ((SID/current sample interval)*100+current sample interval)* (5.8)

5.3.3 Implementations of Conventional Sampling Techniques

In this study, conventional sampling techniques (i.e. systematic, stratified, and random sampling) were implemented using the count based approach where the packet selection decision was based on the packet count. This approach was chosen due to its simplicity (Zseby, 2002). The implementations of conventional sampling techniques are as follows:

i. Systematic sampling: in this approach, for every n packets, the n^{th} packet is selected. In implementation of systematic sampling a counter is set initially to n and it is decreasing it by 1 on receiving each packet. When the counter is zero, the packet is selected. This operation represents one packet selection. During the next step, the counter is reset and the process is repeated. For several experimental runs, the starting point of packet selection is chosen randomly to get different sets of samples for the same size.

ii. Random sampling: in this approach, for a sample size of n taken from a population of N, n random numbers need to be generated for a range 1 to N, and then the packet selection is performed according to the position of the packets in the flow. For each experiment, a new n random numbers should be generated in order to get different sets of samples of the same size.

iii. Stratified sampling: is similar to the implementation of random sampling. For every strata which has a size of N packets, random numbers n are generated in the range [1, N], and the packets are selected according to their position. For every run, a new n random numbers are generated for the same sample size.

5.3.4 Calculation of Sampling QoS Parameters and Sampling Analysis

This section explains how the QoS parameters can be calculated from sampled versions obtained using adaptive statistical sampling techniques and non-adaptive sampling techniques. It also demonstrates the methods used to compare the sampled versions versus original populations.

i. Throughput calculation: throughput was calculated by multiplying the number of received packets (*N*) by the packet size (*PS*) and then dividing it by the difference between receiving time of two successive sampled packets.
ii. Delay calculation: end-to-end delay was calculated by the difference of arrival and sending times of packets which were selected during the sampling process.
iii. Jitter calculation: the difference between the delays of two consecutive sampled packets for specific flow was used in order to calculate the jitter.
iv. Packet loss calculation: the percentage of loss ratio during the i^{th} time interval was calculated based on the total number of received and transmitted packets during that time interval. The number of received packets during the i^{th} time interval was obtained from the sampled version, whereas the number of sent packets was computed from the difference between the sequence numbers of two successive sampled packets.

After calculating the QoS parameters (i.e. throughput, delay, jitter, and packet loss ratio) of the sampled versions, obtained using adaptive statistical sampling and non-adaptive sampling approaches, a comparison between the QoS parameters of sampled versions and the QoS parameters of original population was carried out. The aim of the comparison was to determine which sampling approach could be used to produce sampled version that effectively represented the whole population. The comparison was carried out by calculating the mean, and standard deviation of the original population and its sampled versions using adaptive and non-adaptive sampling techniques.

In order to compare the sampled version versus original populations, the mean and standard deviation of the sampled version may not be sufficient to assess the accuracy of sampled version in terms of representing the original population as they are affected by the outliers (Brase, 2010). Therefore, additional criteria were used to assess the discrepancy between the original population and its sampled version. These were:

i. Bias: the bias shows how far the mean of the sampled version lies from the mean of the original traffic (Zseby, 2004). Bias is the averaged difference of all samples of the same size. Equation (5.9) was used to calculate the bias:

$$Bias = \frac{1}{N}\sum_{i=1}^{N} M_i - M \tag{5.9}$$

where N is the number of simulation runs, Mi and M are the means of the QoS parameters for the sampled version and its original population respectively.

ii. Relative Standard Error (RSE): RSE measures the reliability of sampled version. RSE is expressed as a percentage and can be defined as the standard error of the sample (*SE*) divided by the sample size (*n*) as show in equation (5.10)

$$RSE = \frac{SE}{n} \times 100 \tag{5.10}$$

iii. Curve fitting (i.e. data trend): another criterion to examine the behaviour of sampled version in terms of representing the original population is evaluating the trend of sampled data versus its original counterpart by applying the curve fitting. Curve fitting is a useful tool for representing a data set in a linear, quadratic or polynomial fashion. Curve fitting could be based on two functions, polynomial curve fitting function, and polynomial evaluation function which can quickly and easily fit a polynomial to a set of data points. The general formula for a polynomial is given in equation (5.11):

$$f(x) = a_0x^N + a_1x^{N-1} + a_2x^{N-2} + \ldots\ldots + a_{N-1}x + a_N \tag{5.11}$$

The degree of a polynomial is equal to the largest value of the exponents *(N)*, *(x)* is a set of data, and *(a)* is a set of polynomial coefficients. Polynomial curve fitting function computes a least squares polynomial for a given set of data *(x)* and generates the coefficients of the polynomial which can be used to simulate a curve to fit the data according to the specified degree *(N)*. Whereas, polynomial evaluation function evaluates a polynomial for a given set of *(x)* values and then generates a curve to fit the data based on the coefficients found using curve fitting function (Lindfield and Penny, 2012).

5.3.5 Network Topology and Traffic Characteristics

To validate the performance of adaptive statistical sampling schemes, wireless-cum-wired network topology with a size of 16 unidirectional connections as shown in Figure 4-3 were simulated using NS2. Half of the connections transmitted traffic from wireless to wired network whereas the rest transmitted traffic from wired to wireless. The WLAN was based on IEEE 802.11e, and it used the Enhanced Distributed Channel Access (EDCA) scheme. The main parameters that modelled the wireless channel were the default settings for IEEE 802.11e. These parameters are shown in Table 4-1.

The traffic transmitted over the simulated network were: VoIP, video streaming, best effort traffic, and background traffic. Constant Bit Rate (CBR) traffic was adapted to model VoIP. VoIP was modelled as G.711 voice encoding scheme. The packet size of VoIP was 160 bytes and the transmission rate was 64 kbps. The video streaming source was YUV QCIF (176 × 144) Foreman (400 frame) (YUV QCIF, 2012). Prior to its transmission, each video frame was fragmented into packets, which in turn had a maximum length of 1024 bytes. The best-effort traffic had a fixed packet size of 1000 bytes and 125 kbps transmission rate. File Transfer Protocol (FTP) was used for the background traffic. FTP was transmitted over TCP, whereas other traffics were transmitted using the UDP transport protocol. The transmitted traffic over IEEE 802.11e EDCA (i.e. VoIP, video, best effort traffic, and background traffic) were mapped into different access categories to represent different levels of priorities as shown in Table 4-2. VoIP had the highest priority, whereas the background traffic had the lowest priority. Each simulation lasted 500 seconds. Simulations were repeated 10 times for each experiment. Each time, a different initial seed was used in order to randomly manage the node's transmission period as all the nodes were requested to transmit and stop at a given time. The randomness introduced for different seeds avoided the bias of random number generation.

5.4 Results and Discussion

In this section, the main QoS parameters of VoIP (i.e. throughput, delay, jitter, and packet loss ratio) with their sampled versions using adaptive and non-adaptive sampling techniques are presented in the following subsections.

5.4.1 Throughput

Figures 5-4 (a) - (g) show respectively the actual throughput with its sampled versions using adaptive statistical sampling based on fuzzy inference system (FIS), adaptive statistical sampling based on linear adjustment mechanism, adaptive statistical sampling based on quarter adjustment mechanism, systematic sampling, stratified sampling, and random sampling. The results obtained using adaptive statistical sampling approaches were based on the initial parameters settings shown in Table 5-5. These initial parameters settings were chosen experimentally, i.e. different values were tested to monitor the response of the adaptive statistical sampling approaches and the most suitable settings were chosen.

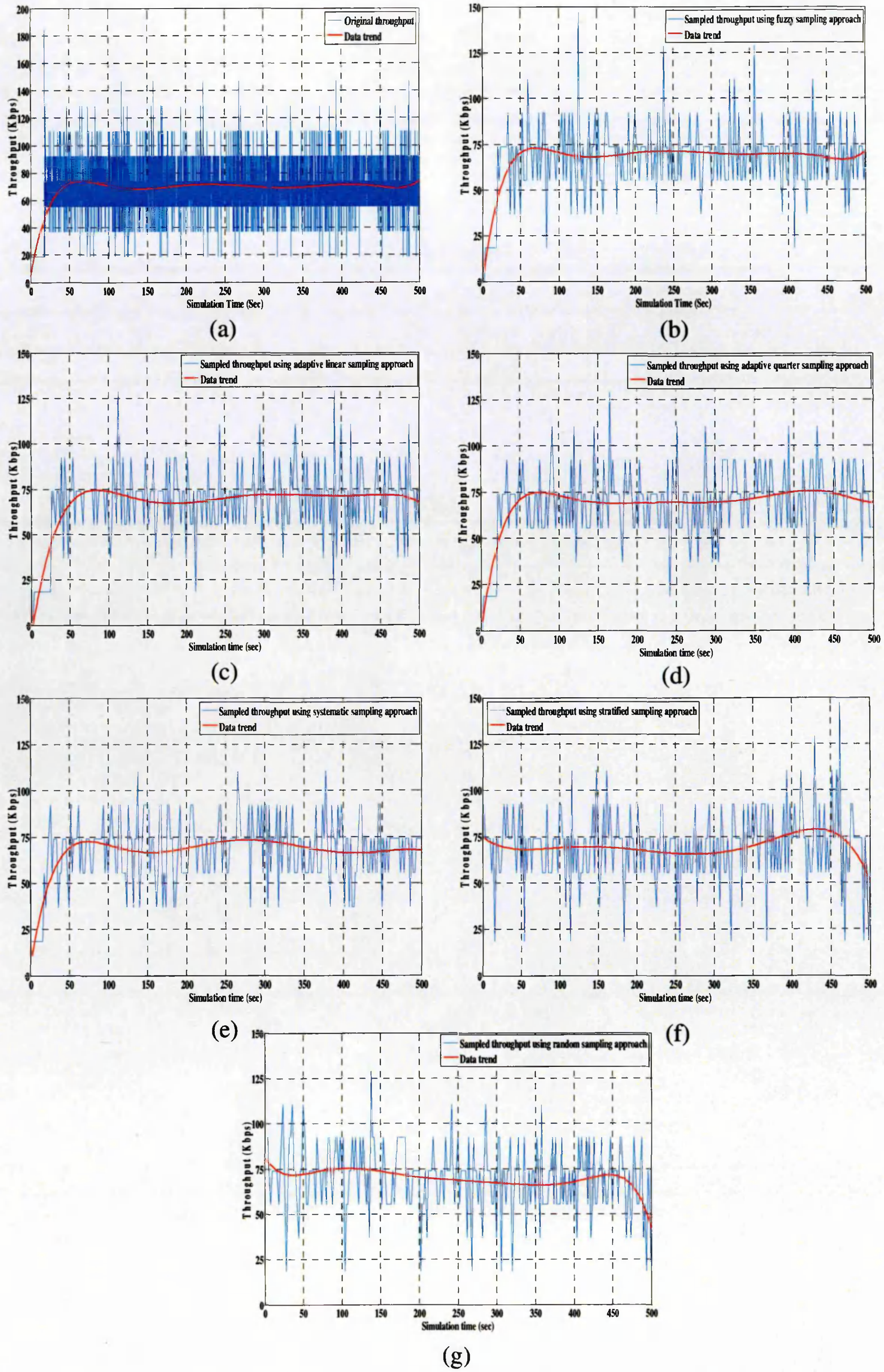


Figure 5-4. Comparison of throughput with its sampled versions using: (a) Actual, (b) Adaptive sampling based on FIS, (c) Adaptive sampling linear adjustment, (d) Adaptive sampling quarter adjustment, (e) Systematic, (f) Stratified, (g) Random sampling.

Table 5-5. Operating parameters of adaptive statistical sampling approaches.

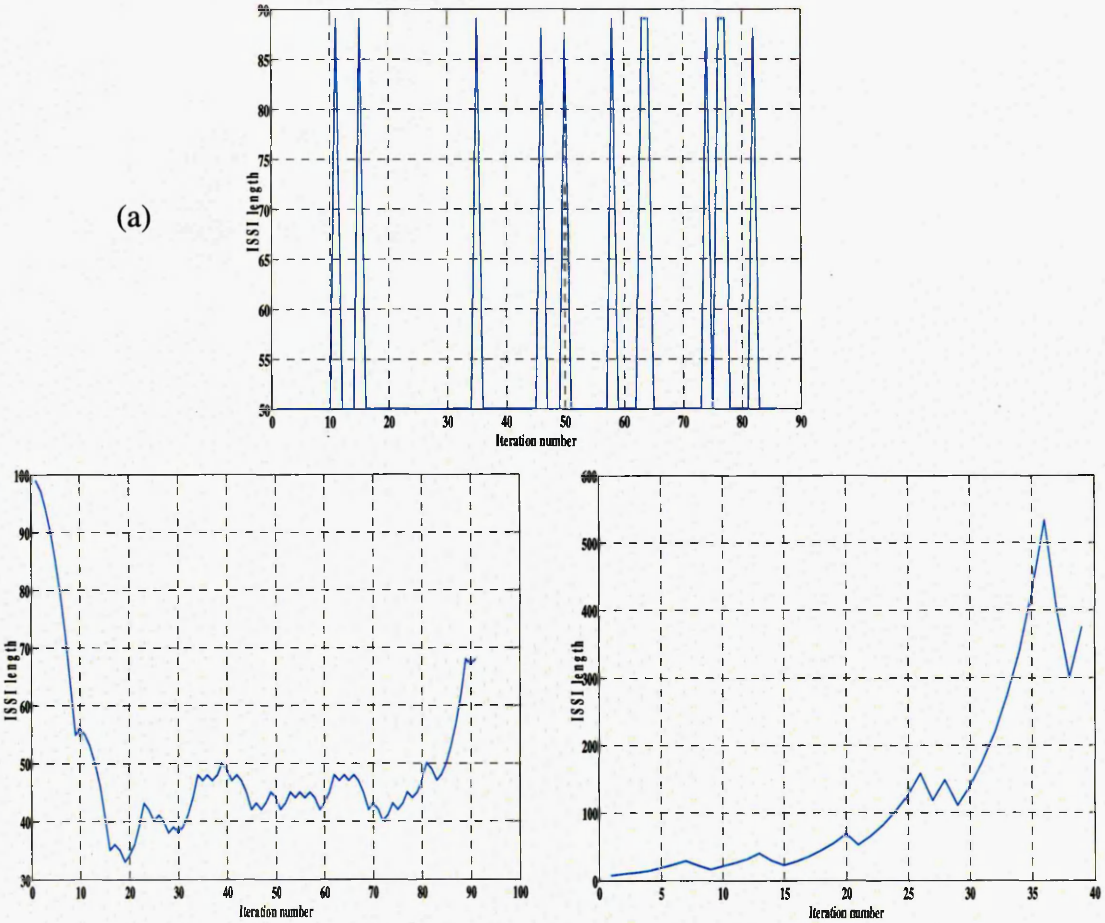
Adaptive statistical sampling approaches \ Operating parameters	Initial value of ISSI	Pre- and post-sampling sections	Threshold values
Adaptive sampling based on FIS	100	2	0.1
Adaptive sampling based on linear adjustment mechanism	100	5	1
Adaptive sampling based on quarter adjustment mechanism	10	10	0.6

In Figures 5-4 (b) - (g), the sample size for each sampled version was 240 packets (i.e. sample fraction was 4.8 % of the actual traffic). The data trends shown in Figures 5-4 (a)-(g) were used to describe the behaviour of the observed sampled versions of throughput and to illustrate the degree of discrepancy for each sampling method from the original population (i.e. actual throughput).

It can be noticed from the trend of data in Figures 5-4 (a) - (g), how different sampling approaches represent the actual throughput. As each sampling approach follows certain procedure, the degree of discrepancy from the actual throughput is different for each sampling method. However, it is observed that the sampled versions of throughput obtained using adaptive statistical sampling approaches is closer to the actual throughput as compared with the non-adaptive sampling approaches (i.e. systematic, stratified, and random sampling). This is because the adaptive statistical sampling approaches based on FIS, linear adjustment mechanism, and quarter adjustment mechanism varied the length of Inter Sampling Section Interval (ISSI) according to the statistical variations of throughput.

Figures 5-5 (a) - (c) show respectively the response of adaptive statistical sampling approaches based on FIS, linear adjustment mechanism, and quarter adjustment mechanism. It can be seen from Figures 5.5 (a) - (c) that during the sampling process of adaptive statistical sampling approaches based on linear adjustment mechanism, quarter adjustment mechanism, and FIS, whenever the calculated overall statistic of throughput was less than pre-defined threshold values indicated in Table 5-5, the ISSI was increased according to equation (5.2), (5.4), and (5.8) respectively. Otherwise, the ISSI was decreased using equations (5.3), (5.5), and (5.8).

(a)



(b) (c)

Figure 5-5. The length of ISSI for adaptive statistical sampling using: (a) Fuzzy inference system, (b) Linear adjustment mechanism, (c) Quarter adjustment mechanism.

Tables 5-6 (a) - (f) summarise respectively the throughput measurement for different sample fractions using adaptive statistical sampling based on (FIS), linear adjustment mechanism, quarter adjustment mechanism, and non-adaptive sampling methods (i.e. systematic, stratified, and random). It is established from Tables 5-6 (a) - (f) that the variations of sampling versions using all sampling methods from actual mean and actual standard deviation are increased as the sample size is decreased. The calculated absolute error is also increased for all sampling methods as the sample fraction is decreased and vice versa.

The summary statistics provided in Tables 5.6 (a) - (f) also indicate that sampled versions of throughput for different sampling fractions obtained using the three adaptive sampling approaches are closer to the original throughput as compared with the versions obtained using non adaptive sampling approaches. This indicates that the proposed adaptive sampling techniques outperform conventional sampling techniques.

Table 5-6. Throughput measurement results using different sampling methods: (a) Adaptive sampling based on FIS, (b) Adaptive sampling based on linear adjustment, (c) Adaptive sampling based on quarter adjustment, (d) Systematic, (e) Stratified, (f) Random sampling.

(a)

Units: (kbps)	Actual values	Sample fraction (%)			
		15.7	8.18	4.8	3.34
Mean throughput	68.35	68.37	68.45	67.57	69.33
Standard deviation	19.73	19.58	19.46	19.15	20.5
Absolute error		0.02	0.1	0.78	0.98

(b)

Units: (kbps)	Actual values	Sample fraction (%)			
		15.7	8.18	4.8	3.34
Mean throughput	68.35	68.1	67.9	67.5	69.3
Standard deviation	19.73	19.31	20.32	19.06	22.8
Absolute error		0.25	0.45	0.85	0.95

(c)

Units: (kbps)	Actual values	Sample fraction (%)			
		15.7	8.18	4.8	3.34
Mean throughput	68.35	67.74	67.59	69.37	67.15
Standard deviation	19.73	21.4	21.87	22.41	23.27
Absolute error		0.61	0.76	1.02	1.2

(d)

Units: (kbps)	Actual values	Sample fraction (%)			
		15.7	8.18	4.8	3.34
Mean throughput	68.35	67.58	69.25	67.17	69.69
Standard deviation	19.73	22.32	22.71	22.85	23.8
Absolute error		0.77	0.9	1.18	1.34

(e)

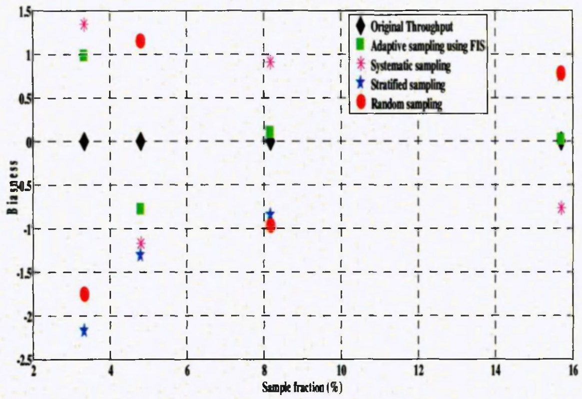
Units: (kbps)	Actual values	Sample fraction (%)			
		15.7	8.18	4.8	3.34
Mean throughput	68.35	69.1	67.5	67.04	66.17
Standard deviation	19.73	24.19	22.33	23.8	24.6
Absolute error		0.75	0.85	1.31	2.18

(f)

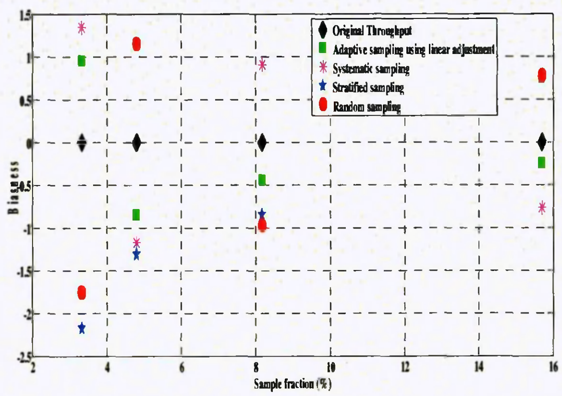
Units: (kbps)	Actual values	Sample fraction (%)			
		15.7	8.18	4.8	3.34
Mean throughput	68.35	69.12	67.38	69.5	66.59
Standard deviation	19.73	23.09	23.16	23.42	24.82
Absolute error		0.77	0.97	1.15	1.76

The mean and standard deviation were not only the criteria used to access the accuracy of throughput sampled version for representing the actual population. Bias and RSE calculated using equations 5.9 and 5.10 respectively were also used to assess the discrepancy between the original throughput and its sampled versions.

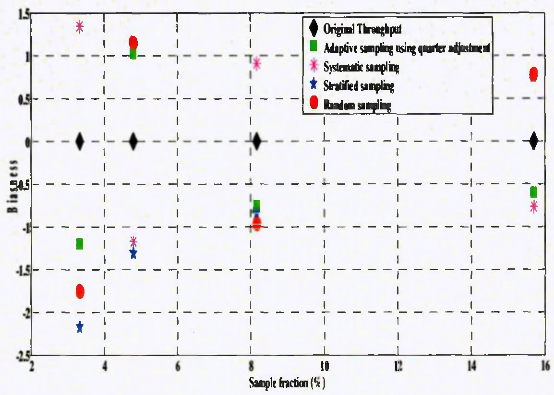
Figures 5-6 (a) - (c) illustrate respectively the comparison of the bias of sampled throughput versions from the mean of actual throughput for different sample fractions using conventional sampling approaches versus adaptive sampling based on FIS, linear adjustment mechanism, and quarter adjustment mechanism. The results illustrate that the bias was decreased and became closer to zero for all sampling approaches when the sample size was increased. However, it can be observed from the figures that the adaptive statistical sampling approaches have a lower bias as compared with systematic, stratified, and random sampling techniques. For example, the bias of sampled throughput for 8.18% sample fraction (i.e. sample size 409 packets) using adaptive statistical sampling based on FIS was 0.1, whereas the bias values when using systematic, stratified, and random sampling were 0.9, -0.85, and -0.97 respectively.



(a)



(b)



(c)

Figure 5-6. Comparison of bias of sampled throughput obtained from non-adaptive sampling with bias obtained using: (a) Adaptive sampling based on FIS, (b) Adaptive sampling based on linear adjustment, (c) Adaptive sampling based on quarter adjustment.

RSE calculated from throughput sampled versions obtained from conventional sampling approaches (i.e. systematic, stratified, and random) compared with the RSE obtained using adaptive statistical approaches based on FIS, linear adjustment, and quarter adjustment as shown in Figures 5-7 (a) - (c) respectively. It is established that the results obtained from the three adaptive sampling approaches are more accurate as compared with systematic, stratified, and random sampling. For example, the overall reductions of RSE were respectively 13.46%, 16.67% and 16.67% when using adaptive sampling based on FIS as compared with systematic, stratified, and random sampling.

(a)

(b)

(c)

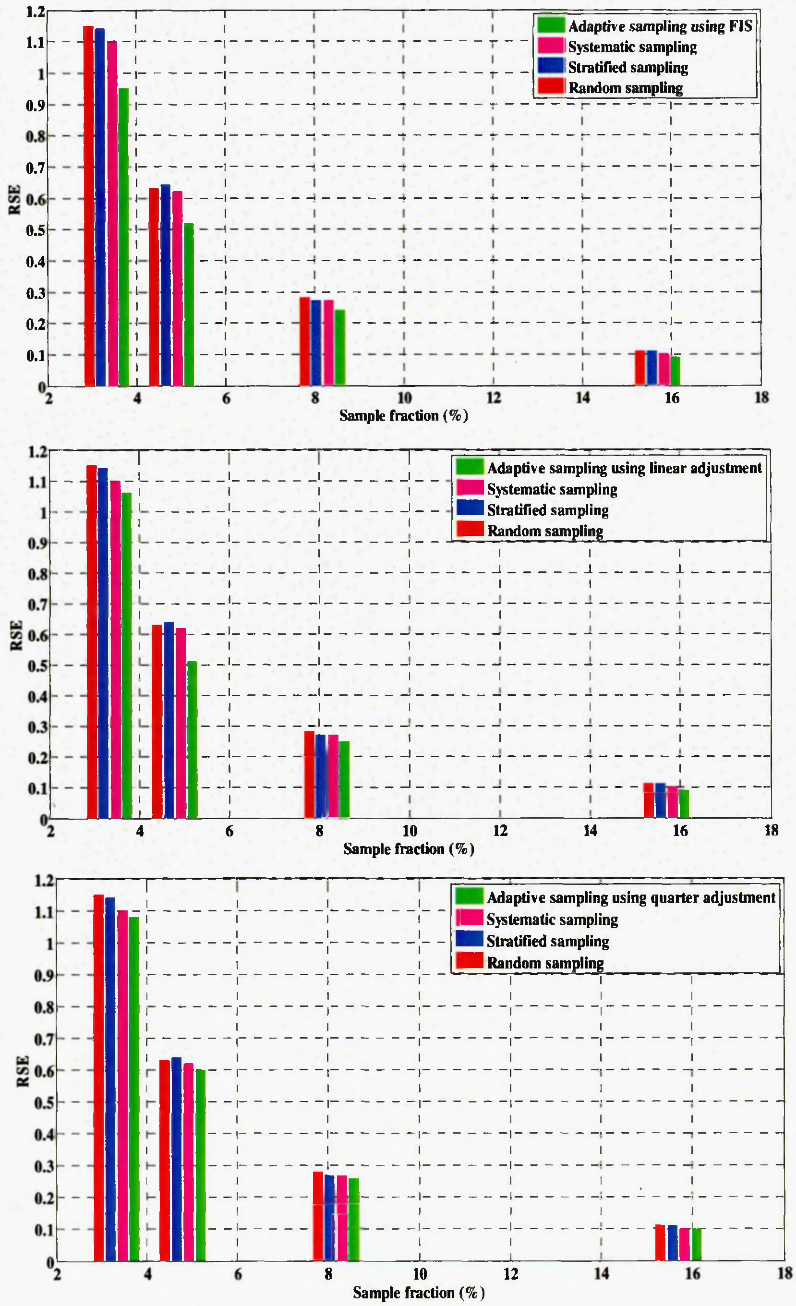


Figure 5-7. RSE of sampled throughput using conventional sampling versus: (a) Adaptive sampling based on FIS, (b) Adaptive sampling based on linear adjustment, (c) Adaptive sampling based on quarter adjustment.

5.4.2 Delay

Delay was another QoS parameter measured from sampled versions obtained using adaptive statistical sampling based on FIS, linear adjustment mechanism, and quarter adjustment mechanism, along with non-adaptive sampling techniques (i.e. systematic, stratified, and random sampling) as depicted in Figure 5-8 (b) - (g).

Figures 5-8 (a) - (g) show the comparison between the actual delay and its sampled versions using the above mentioned sampling techniques respectively. The mean and standard deviation of data trend for original delay were 33.5 msec and 7 msec respectively, whereas the sampled versions of delay obtained from the adaptive sampling based on FIS, linear adjustment mechanism, and quarter adjustment mechanism had the mean of 32.67 msec, 33.99 msec, and 32.33 msec and standard deviation of 7.7 msec, 5.7 msec, and 7.5 msec respectively. However, the mean and standard deviation of data trend for sampled delay using systematic, stratified, and random sampling were (35.82 msec, 34.95 msec, and 35.85 msec) and (10.73 msec, 5.42 msec, and 5.3 msec) respectively. This indicates that sampled versions of delay using adaptive statistical sampling approaches represented the original delay more accurately and effectively.

Tables 5-7 (a) - (f) statically demonstrate how different sample fractions obtained using adaptive and non-adaptive sampling approaches represent the actual delay. For all sampling methods, as the sample size was increased, the variation of sampled mean, standard deviation from the actual mean and standard deviation decreased accordingly. This is because a large sample size includes more packets that in turn increase the probability of obtaining more details from the actual delay. However, the delay sampled versions obtained from the proposed adaptive sampling techniques represent the actual delay more closely than the conventional sampling techniques. For example, the absolute error of 8.18% sample fraction of systematic, stratified, and random sampling was increased by 32.2%, 31.03%, and 33.33%, as compared with the absolute error obtained using the adaptive sampling using linear adjustment mechanism.

The improvement of adaptive sampling approaches over conventional sampling techniques is due to the selection of packets in the former approaches depending on the statistical variation of the traffic whereas the packet selection in the latter techniques depended either on a fixed or random sample rate.

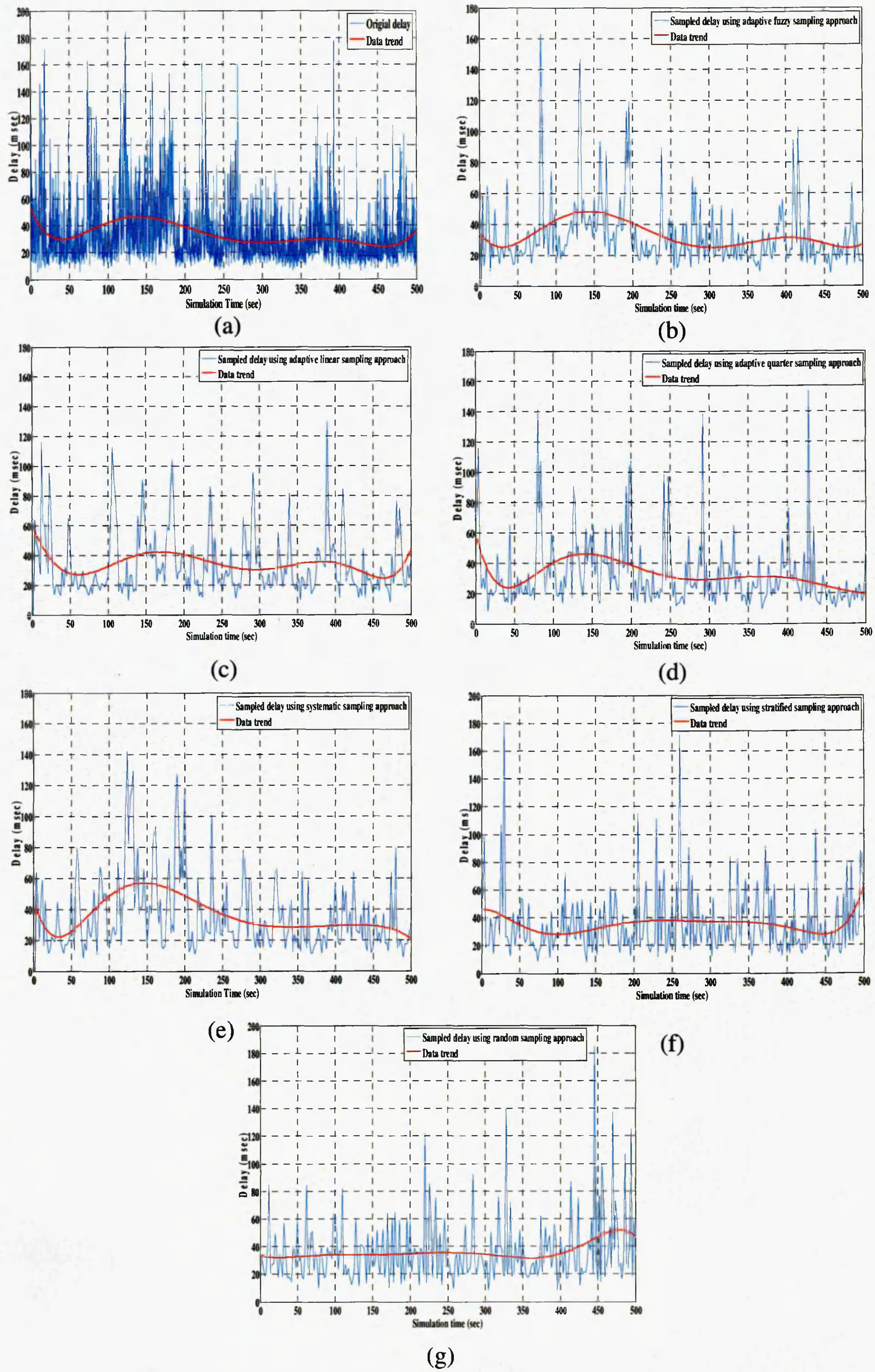


Figure 5-8. Comparison of delay with its sampled versions using: (a) Actual, (b) Adaptive sampling based on FIS, (c) Adaptive sampling linear adjustment, (d) Adaptive sampling quarter adjustment, (e) Systematic, (f) Stratified, (g) Random sampling.

Table 5-7. Delay measurement results using different sampling methods: (a) Adaptive sampling based on FIS, (b) Adaptive sampling based on linear adjustment, (c) Adaptive sampling based on quarter adjustment, (d) Systematic, (e) Stratified, (f) Random sampling.

(a)

Units: (msec)	Actual values	Sample fraction (%)			
		15.7	8.18	4.8	3.34
Mean delay	33.5	33.61	32.64	32.6	34.56
Standard deviation	22.63	22.09	22.01	21.44	20.47
Absolute error		0.11	0.86	0.9	1.06

(b)

Units: (msec)	Actual values	Sample fraction (%)			
		15.7	8.18	4.8	3.34
Mean delay	33.5	33.98	34.3	34.48	32
Standard deviation	22.63	22.7	22.2	21.93	21.75
Absolute error		0.48	0.8	0.98	1.5

(c)

Units: (msec)	Actual values	Sample fraction (%)			
		15.7	8.18	4.8	3.34
Mean delay	33.5	32.85	34.45	34.94	35.26
Standard deviation	22.63	22.88	23.08	23.81	24.05
Absolute error		0.65	0.95	1.44	1.76

(d)

Units: (msec)	Actual values	Sample fraction (%)			
		15.7	8.18	4.8	3.34
Mean delay	33.5	32.64	34.68	35.24	31.57
Standard deviation	22.63	24.19	24.81	24.54	24.82
Absolute error		0.86	1.18	1.74	1.93

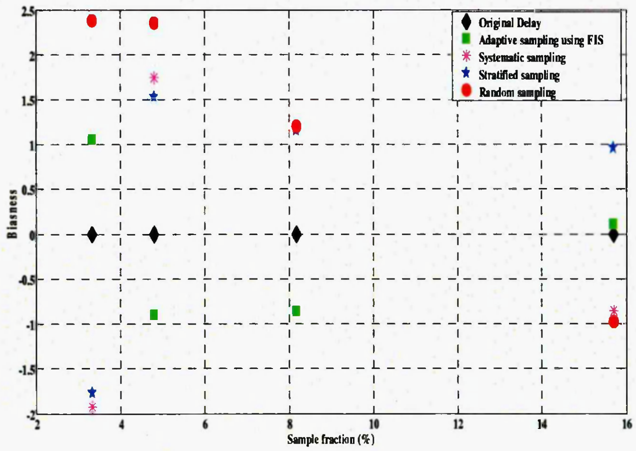
(e)

Units: (msec)	Actual values	Sample fraction (%)			
		15.7	8.18	4.8	3.34
Mean delay	33.5	34.46	34.66	35.03	31.73
Standard deviation	22.63	24.19	25.64	24.32	24.17
Absolute error		0.96	1.16	1.53	1.77

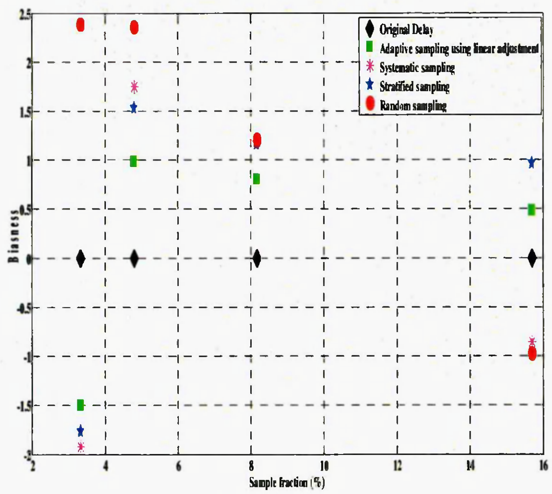
(f)

Units: (msec)	Actual values	Sample fraction (%)			
		15.7	8.18	4.8	3.34
Mean delay	33.5	32.52	34.7	35.85	35.88
Standard deviation	22.63	24.19	24.94	25.68	24.61
Absolute error		0.98	1.2	2.35	2.38

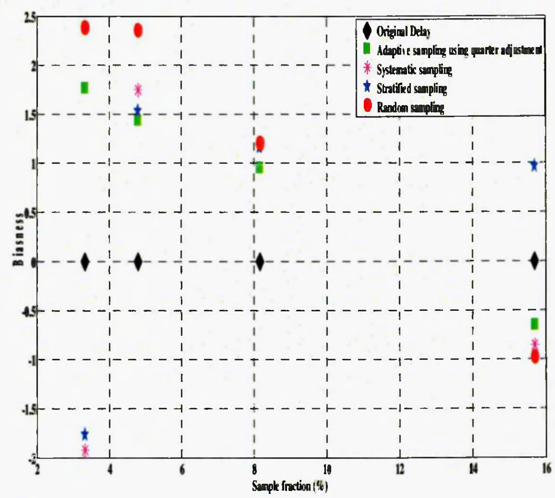
Bias and RSE were also used to assess the accuracy of sampled delay. The bias for adaptive sampling based on FIS, linear adjustment mechanism, quarter adjustment mechanism were closer to zero as compared with the non-adaptive sampling approaches as shown in Figures 5-9 (a) - (c) respectively. For example, the bias of 3.34% sample fraction was reduced by 55.46%, 36.97%, and 26.05% in case of adaptive sampling based on FIS, linear adjustment mechanism, quarter adjustment mechanism respectively comparing with random sampling approach.



(a)



(b)



(c)

Figure 5-9. Comparison of bias of sampled delay obtained from non-adaptive sampling with bias obtained using: (a) Adaptive sampling based on FIS, (b) Adaptive sampling based on linear adjustment, (c) Adaptive sampling based on quarter adjustment.

Relative Standard Error RSE as shown in Figures 5-10 (a) - (c) proves that adaptive statistical approaches based on FIS, linear adjustment, and quarter adjustment respectively outperformed conventional sampling approaches.

Although the value of RSE decreased and become closer to zero for all sampling methods when the sample size was increased, the RSE values obtained from the three adaptive sampling approaches were less than RSE calculated from sampled delay using systematic, stratified, and random sampling. The overall RSE was increased by 3.7%, 1.85% and 3.7% when using systematic, stratified, and random sampling respectively compared with adaptive statistical sampling based on quarter adjustment.

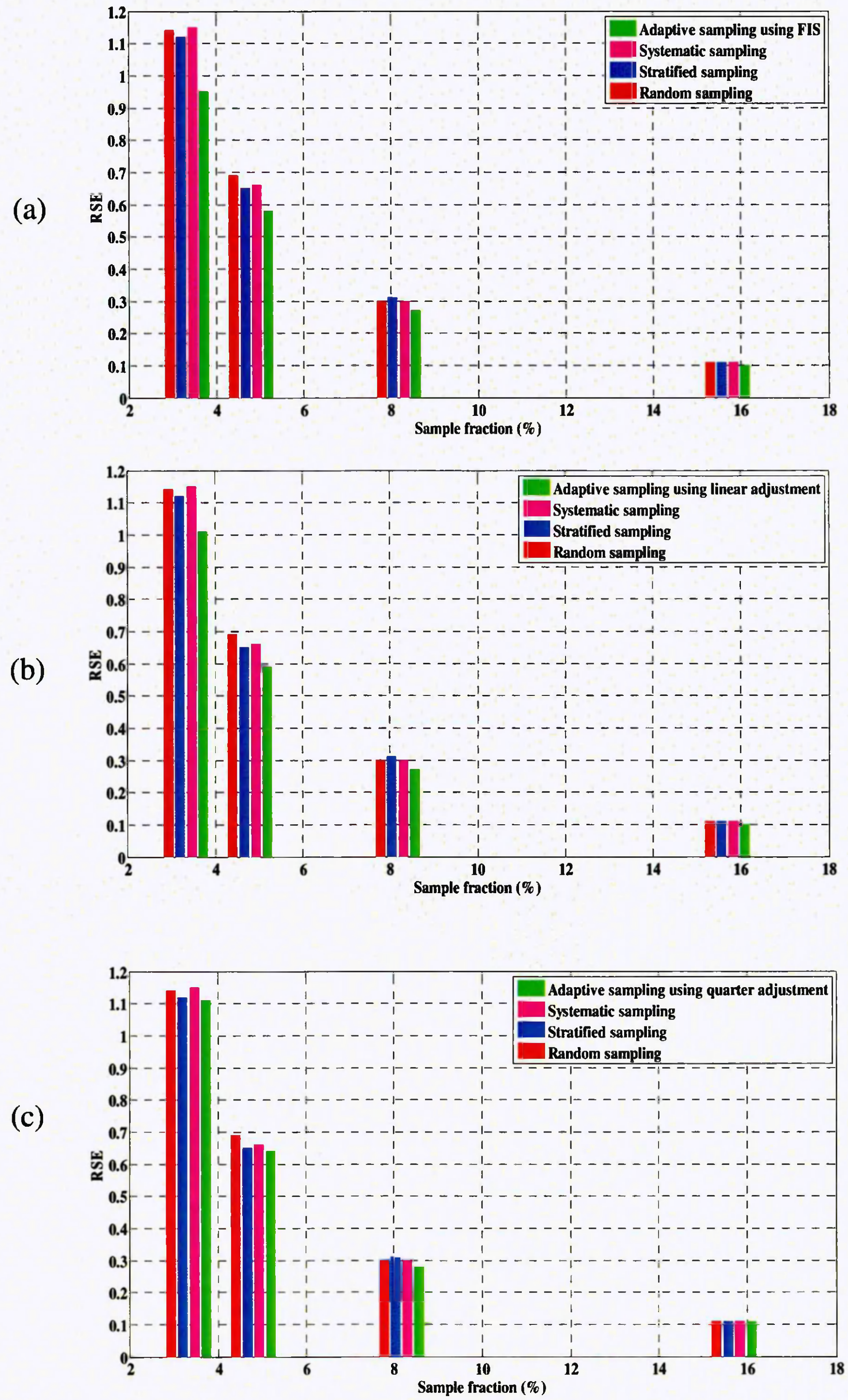


Figure 5-10. RSE of sampled delay using conventional sampling versus: (a) Adaptive sampling based on FIS, (b) Adaptive sampling based on linear adjustment, (c) Adaptive sampling based on quarter adjustment.

5.4.3 Jitter

Jitter was another important QoS parameter to be measured from sampled versions obtained using the adaptive and non-adaptive sampling approaches. The results shown in Figures 5-11 (a) - (g) highlighted the potential of sampling the actual jitter using different sampling approaches.

It can be perceived from Figures 5-11 (b) - (d) how adaptive statistical sampling based on FIS, linear adjustment, and quarter adjustment represent the actual jitter shown in Figure 5-11 (a). The statistic of data trend obtained from jitter sampled versions using adaptive statistical sampling based on FIS, linear adjustment, and quarter adjustment had respectively; 12.99 msec, 12.63 msec, and 12.34 msec for their means and 0.76 msec, 1.18 msec, and 1.57 msec for their standard deviations. These values were very close to the means and standard deviations of the trend of actual jitter which were 12.92 msec, and 0.81 msec. However, the mean and standard deviation of data trend obtained using the non-adaptive sampling techniques as shown in Figures 5-11 (e) - (g) were respectively 13.19 msec, 2.14 msec for systematic, 13.27 msec, 3.17 msec for stratified, and 13.7 msec, 1.81 msec for random sampling. This indicates that the three proposed adaptive statistical sampling approaches provided results closer to the actual jitter as compared with the non-adaptive sampling techniques. The statistic results of jitter for different sample fractions using adaptive statistical sampling based on FIS, linear adjustment, quarter adjustment approaches and conventional sampling techniques are summarised in Tables 5-8 (a) - (f) respectively.

From the Tables 5-8 (a) - (f), the percentage of difference between the values of mean and standard deviation for 3.34% sample fraction obtained from adaptive sampling approaches and the actual mean and standard deviation were respectively 4.01%, 2.33% for fuzzy adjustment approach, 7.45%, 5.5% for linear adjustment approach, and 3.37%, 10.53% for quarter adjustment approach. Whereas the percentage of difference between the values of the actual mean and standard deviation, and the values of mean and standard deviation obtained from conventional sampling techniques were respectively 8.24%, 16.56% for systematic sampling, 9.31%, 20.42%, for stratified sampling, and 11.57%, 29.01% for random sampling. This analysis verified that the proposed adaptive statistical sampling with fuzzy adjustment, linear adjustment, and quarter adjustment mechanisms generated sampled versions which are very close to the original jitter as compared with non-adaptive sampling approaches.

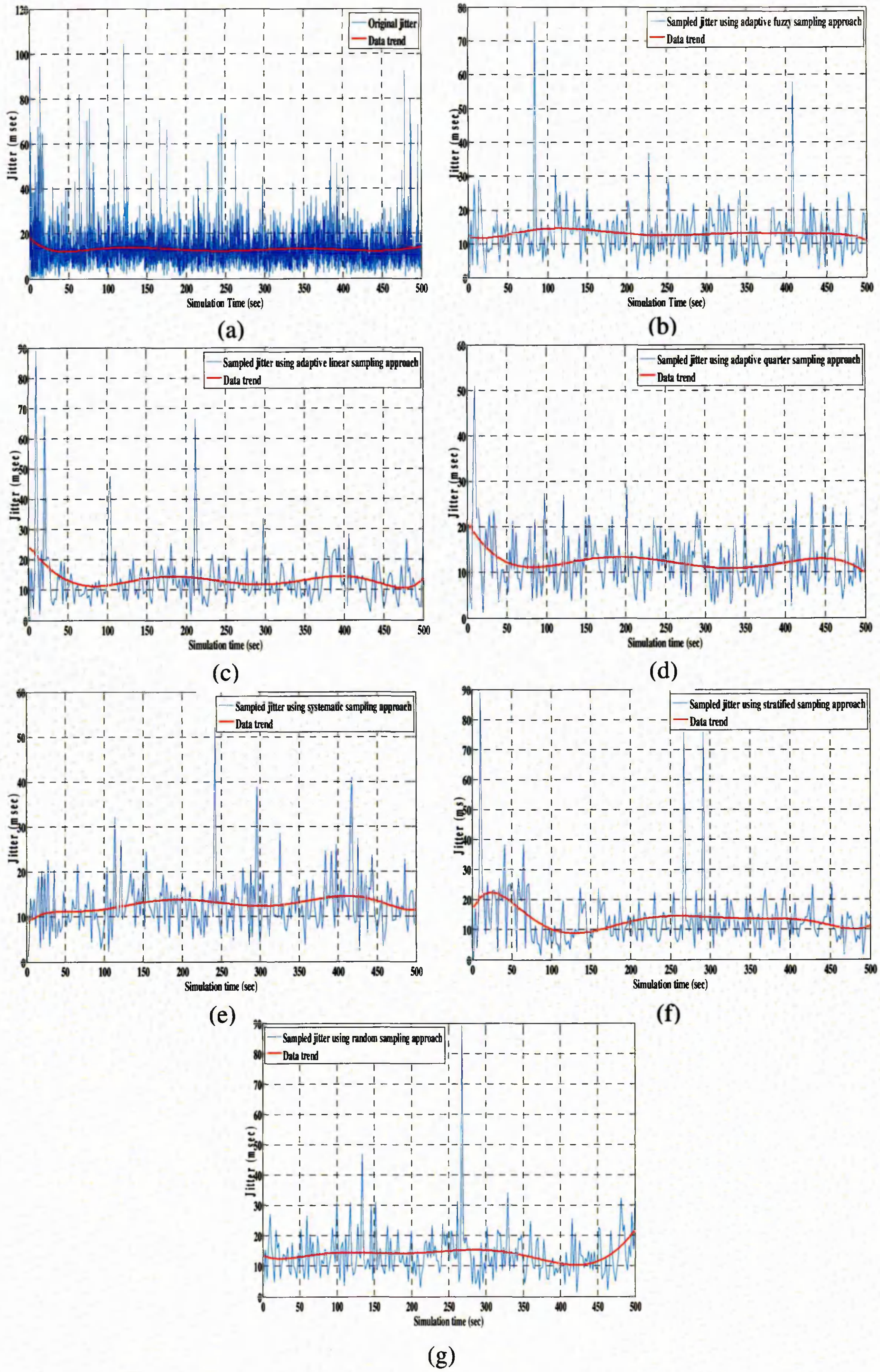


Figure 5-11. Comparison of jitter with its sampled versions using: (a) Actual, (b) Adaptive sampling based on FIS, (c) Adaptive sampling linear adjustment, (d) Adaptive sampling quarter adjustment, (e) Systematic, (f) Stratified, (g) Random sampling.

Table 5-8. Jitter measurement results using different sampling methods: (a) Adaptive sampling based on FIS, (b) Adaptive sampling based on linear adjustment, (c) Adaptive sampling based on quarter adjustment, (d) Systematic, (e) Stratified, (f) Random sampling.

(a)

Units: (msc)	Actual values	Sample fraction (%)			
		15.7	8.18	4.8	3.34
Mean jitter	12.92	12.99	13.02	13.09	13.46
Standard deviation	7.56	7.57	7.47	7.67	7.74
Absolute error		0.07	0.1	0.17	0.54

(b)

Units: (msc)	Actual values	Sample fraction (%)			
		15.7	8.18	4.8	3.34
Mean jitter	12.92	13.1	13.18	13.42	13.96
Standard deviation	7.56	7.39	7.77	7.11	8
Absolute error		0.18	0.26	0.5	1.04

(c)

Units: (msc)	Actual values	Sample fraction (%)			
		15.7	8.18	4.8	3.34
Mean jitter	12.92	13.01	13.1	12.64	13.37
Standard deviation	7.56	7.9	8.01	8.23	8.45
Absolute error		0.09	0.18	0.28	0.45

(d)

Units: (msc)	Actual values	Sample fraction (%)			
		15.7	8.18	4.8	3.34
Mean jitter	12.92	12.72	12.63	13.71	14.08
Standard deviation	7.56	9.89	9.92	8.9	9.06
Absolute error		0.2	0.29	0.79	1.16

(e)

Units: (msc)	Actual values	Sample fraction (%)			
		15.7	8.18	4.8	3.34
Mean jitter	12.92	12.7	13.23	12.22	11.82
Standard deviation	7.56	9.68	9.1	9.67	9.5
Absolute error		0.22	0.31	0.7	1.1

(f)

Units: (msc)	Actual values	Sample fraction (%)			
		15.7	8.18	4.8	3.34
Mean jitter	12.92	13.16	13.27	13.7	11.58
Standard deviation	7.56	10.01	8.72	10.04	10.65
Absolute error		0.24	0.35	0.78	1.34

The bias values shown in Figures 5-12 (a) - (c) demonstrate that the bias obtained from adaptive and non-adaptive sampling approaches decreased accordingly as the sample size increased.

The bias values obtained from adaptive sampling based on fuzzy approach, linear adjustment approach, and quarter adjustment approach were closer to zero as compared with non-adaptive sampling approaches. For instance, the biases of sample size of 409 packets (i.e. 8.18% sample fraction) obtained using adaptive sampling were 0.1 for fuzzy approach, 0.26 for linear adjustment approach, and 0.18 for quarter adjustment approach whereas the biases obtained from the same sample size using conventional sampling techniques were 0.29 for systematic sampling, 0.31 stratified sampling, and 0.35 for random sampling. This indicates that actual jitter could be represented effectively using adaptive sampling approaches rather than conventional sampling approaches.

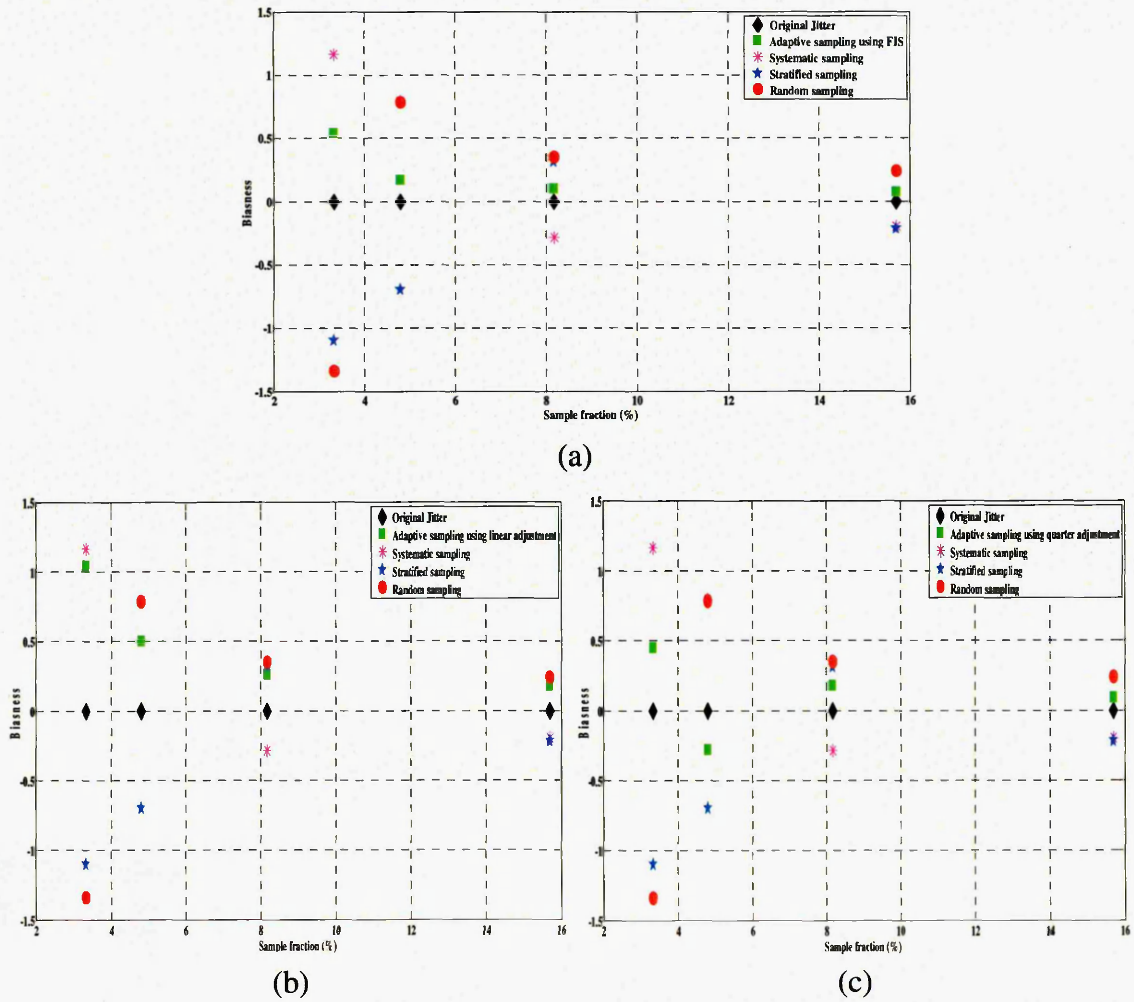


Figure 5-12. Comparison of bias of sampled jitter obtained from non-adaptive sampling with bias obtained using: (a) Adaptive sampling based on FIS, (b) Adaptive sampling based on linear adjustment, (c) Adaptive sampling based on quarter adjustment.

Relative Standard Error RSE as shown in Figures 5-13 (a) - (c) also illustrated how the sampling errors for all sample fractions obtained from jitter sampled versions using adaptive statistical sampling approaches were closer to zero as compared with non-adaptive sampling approaches.

(a)

(b)

(c)

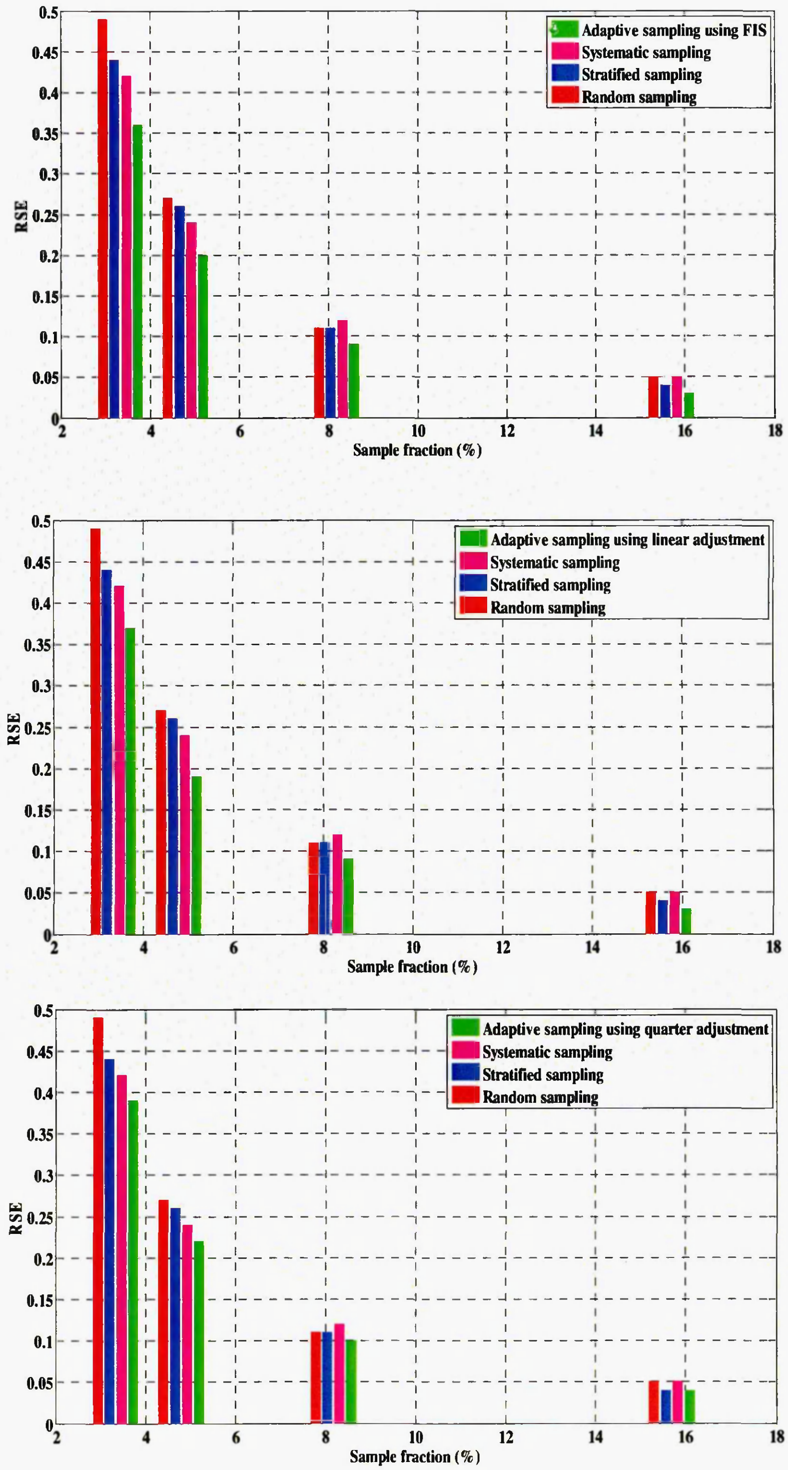


Figure 5-13. RSE of sampled jitter using conventional sampling versus: (a) Adaptive sampling based on FIS, (b) Adaptive sampling based on linear adjustment, (c) Adaptive sampling based on quarter adjustment.

As an example of the accuracy of adaptive statistical sampling, the RSE value for 4.8% sample fraction as shown Figure 5-13 (a) was reduced by 16.67%, 23.08%, and 25.93% respectively when fuzzy adjustment approach was applied as compared with systematic, stratified, and random sampling.

5.4.4 Packet Loss Ratio

Figures 5-14 (a) - (g) illustrate the actual packet loss ratio in addition to the sampled versions of loss ratio using adaptive and non-adaptive sampling approaches with a sampling fraction of 3.34%. These figures show how adaptive and non-adaptive sampling approaches tracked and sampled the actual traffic losses.

It can be observed from Figures 5-14 (a) - (g) that the accurate estimation of packet losses were obtained using adaptive sampling techniques based on fuzzy adjustment approach, linear adjustment approach, and quarter adjustment approach. The mean and standard deviation of sampled packet losses obtained from adaptive sampling techniques were 0.27, and 0.82 for fuzzy adjustment approach, 0.26, and 0.77 for linear adjustment approach, and 0.38, 0.91 for quarter adjustment approach. These results were very close and comparable with the mean and standard deviation of actual packet loss ratio which were respectively 0.31, and 0.81. The data trends shown in Figures 5.14 (a) - (g) also demonstrate that sampled versions of packet loss ratio obtained using the adaptive statistical sampling techniques with the three different approaches closely represented the actual packet loss ratio as compared with non adaptive sampling approaches.

Tables 5-9 (a) - (f) demonstrate how the actual packet loss ratio was represented by different sample fractions obtained using adaptive and non adaptive sampling approaches. For all sampling approaches, as the sample size was decreased, the variation of sampled mean, standard deviation from the actual mean and actual standard deviation increased accordingly. However, the packet loss ratio calculated from sampled versions obtained using the proposed adaptive sampling techniques represented the actual loss ratio better than the conventional sampling techniques. For example, the absolute error of sample fraction of 4.8% was increased by 15%, 25%, and 60% in case of systematic, stratified, and random sampling comparing with the absolute error obtained using adaptive sampling based on quarter adjustment which was 0.2.

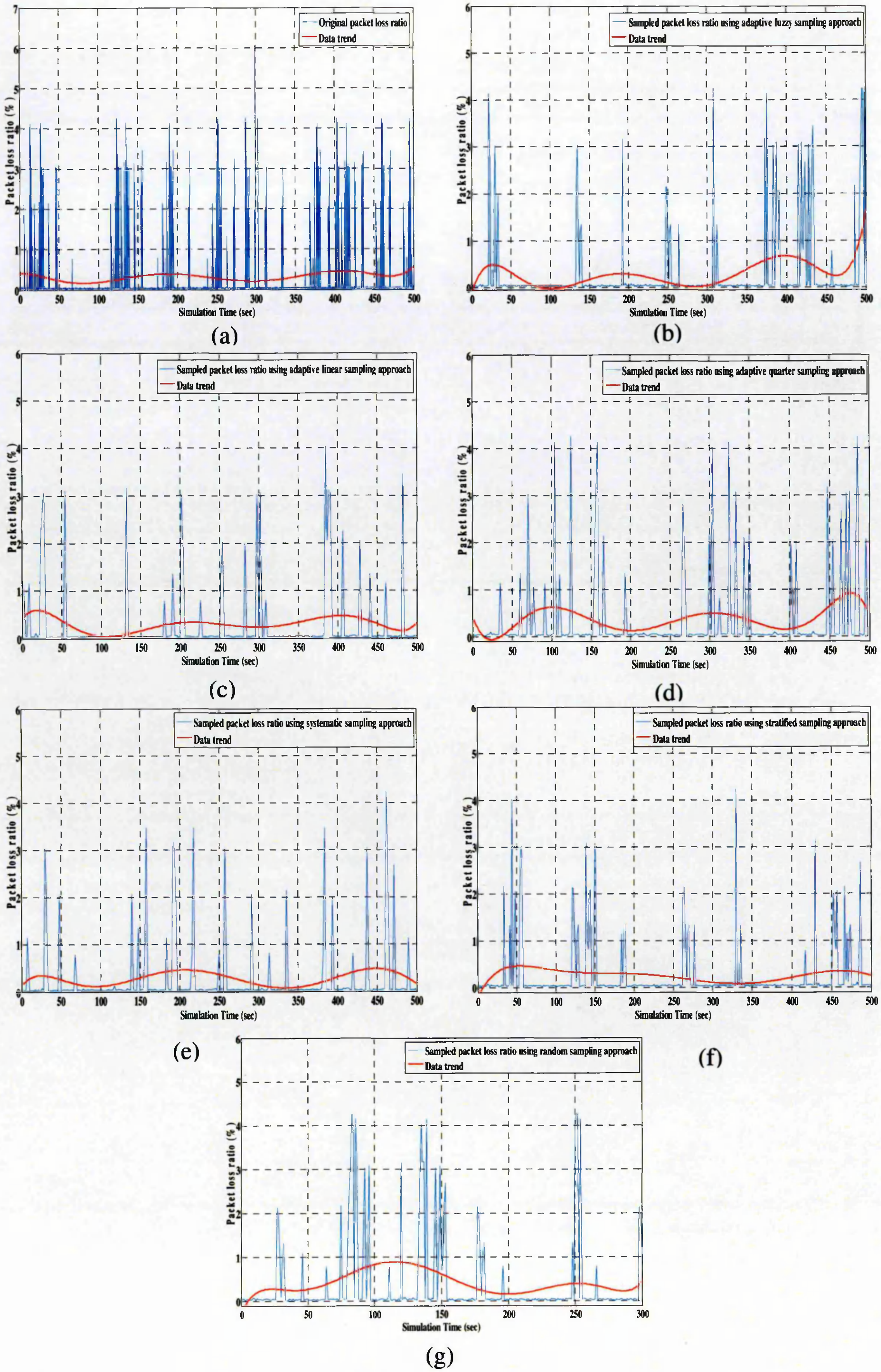


Figure 5-14. Packet loss ratio with its sampled versions using: (a) Actual, (b) Adaptive sampling based on FIS, (c) Adaptive sampling linear adjustment, (d) Adaptive sampling quarter adjustment, (e) Systematic, (f) Stratified, (g) Random sampling.

Table 5-9. Packet loss ratio measurement results using different sampling methods: (a) Adaptive sampling based on FIS, (b) Adaptive sampling based on linear adjustment, (c) Adaptive sampling based on quarter adjustment, (d) Systematic, (e) Stratified, (f) Random sampling.

(a)

Units: (%)	Actual values	Sample fraction (%)			
		15.7	8.18	4.8	3.34
Mean packet loss ratio	0.31	0.31	0.31	0.29	0.27
Standard deviation	0.81	0.8	0.82	0.8	0.82
Absolute error		0.005	0.01	0.07	0.11

(b)

Units: (%)	Actual values	Sample fraction (%)			
		15.7	8.18	4.8	3.34
Mean packet loss ratio	0.31	0.3	0.32	0.29	0.26
Standard deviation	0.81	0.8	0.83	0.76	0.77
Absolute error		0.02	0.03	0.07	0.16

(c)

Units: (%)	Actual values	Sample fraction (%)			
		15.7	8.18	4.8	3.34
Mean packet loss ratio	0.31	0.3	0.29	0.37	0.38
Standard deviation	0.81	0.81	0.76	0.9	0.91
Absolute error		0.015	0.05	0.2	0.21

(d)

Units: (%)	Actual values	Sample fraction (%)			
		15.7	8.18	4.8	3.34
Mean packet loss ratio	0.31	0.3	0.33	0.27	0.38
Standard deviation	0.81	0.8	0.83	0.75	0.95
Absolute error		0.018	0.07	0.23	0.23

(e)

Units: (%)	Actual values	Sample fraction (%)			
		15.7	8.18	4.8	3.34
Mean packet loss ratio	0.31	0.33	0.37	0.23	0.39
Standard deviation	0.81	0.83	0.82	0.63	0.93
Absolute error		0.08	0.18	0.25	0.26

(f)

Units: (%)	Actual values	Sample fraction (%)			
		15.7	8.18	4.8	3.34
Mean packet loss ratio	0.31	0.37	0.38	0.41	0.47
Standard deviation	0.81	0.88	0.9	0.96	0.99
Absolute error		0.21	0.23	0.32	0.54

The accuracy of sampled packet loss ratio was also measured using bias as shown in Figures 5-15. As shown in the Figures, the bias of sampled packet loss ratio using adaptive sampling based on FIS, linear adjustment mechanism, quarter adjustment mechanism were closer to zero for all sampling fractions as compared with the non-adaptive sampling approaches. For example, the bias of 4.8% sample fraction was reduced by 75%, 75%, and 25% in case of adaptive sampling based on FIS, linear adjustment mechanism, quarter adjustment mechanism respectively comparing with stratified sampling approach.

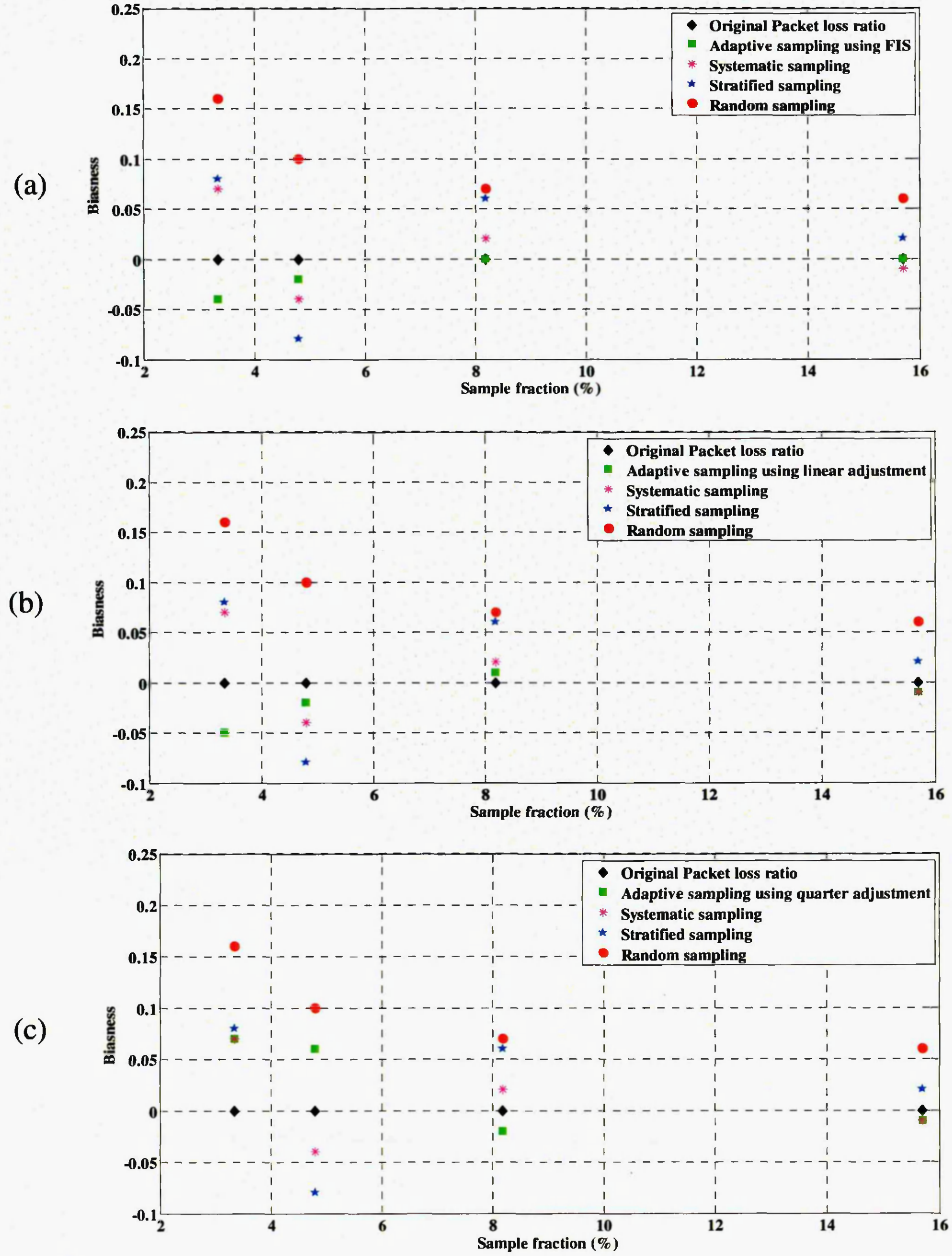


Figure 5-15. Comparison of bias of sampled packet losses obtained from non-adaptive sampling with bias obtained from adaptive sampling based on: (a) FIS, (b) linear adjustment approach, (c) quarter adjustment approach.

The comparisons between RSE of sampled packet losses obtained using conventional sampling approaches, and the RSE obtained using adaptive statistical approaches based on FIS, linear and quarter adjustments are shown in Figure 5-16 (a) - (c) respectively. The results obtained from the three adaptive sampling approaches were more accurate as compared with systematic, stratified, and random sampling. The overall reductions of RSE were respectively 10.5%, 15% and 22.7% when using adaptive sampling based on FIS compared with systematic, stratified, and random sampling.

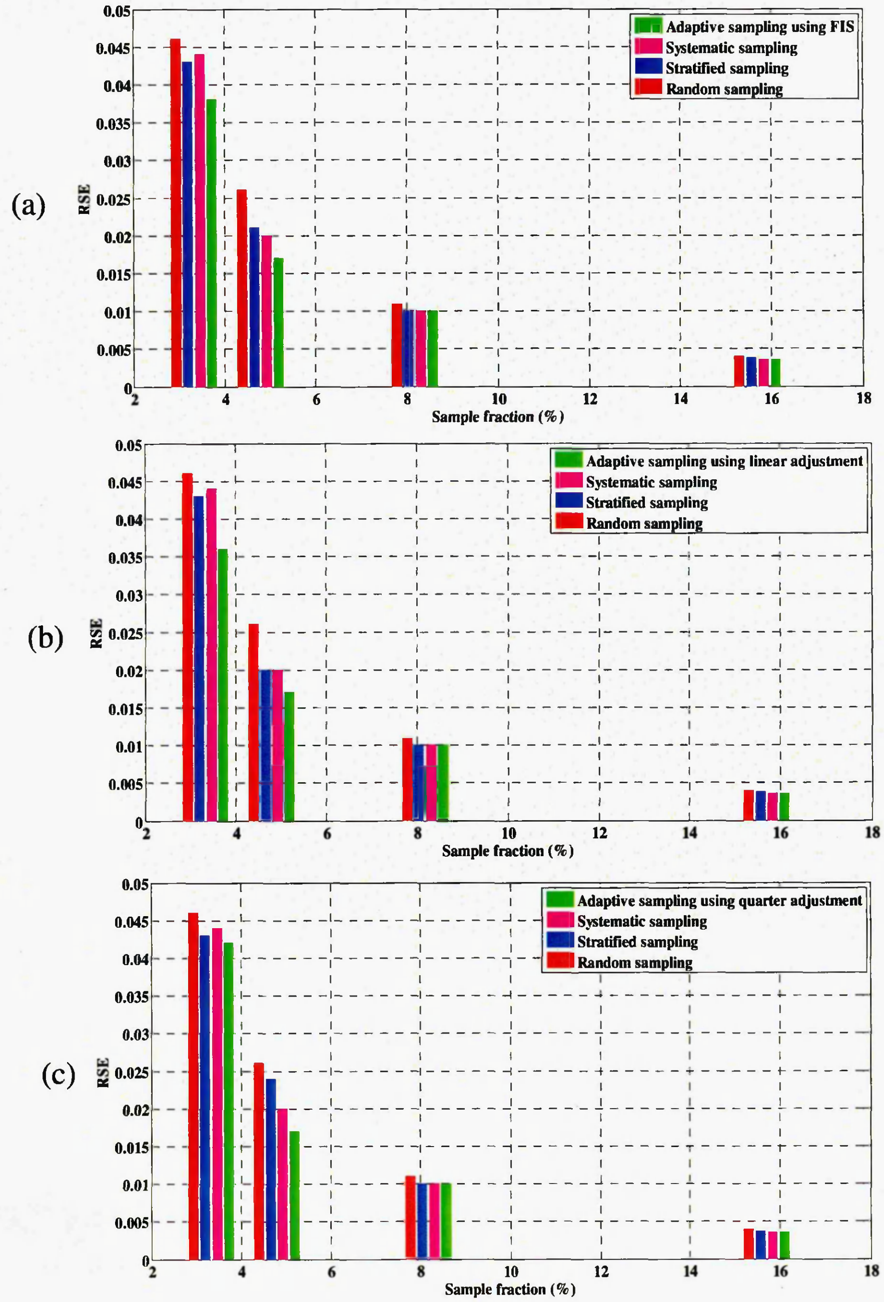


Figure 5-16. RSE of sampled packet loss ratio using conventional sampling versus: (a) Adaptive sampling based on FIS, (b) Adaptive sampling based on linear adjustment, (c) Adaptive sampling based on quarter adjustment.

5.5 Summary

In this chapter, three adaptive statistical sampling techniques to adjust the sampling rate of multimedia traffic were developed and evaluated. A novel aspect of the proposed techniques was an adjustment of sampling rate based on traffic's statistics. The sampling rate of the three devised sampling techniques was controlled using three different mechanisms: a simple linear adjustment mechanism, a quarter adjustment mechanism, and a fuzzy inference system. The proposed techniques decreased the sampling rate when the statistics of the traffic did not significantly change and increased the sampling rate when the statistics of the traffic significantly changed.

A comparison of the three proposed adaptive statistical sampling techniques with conventional sampling techniques (i.e. systematic, stratified, and random sampling) was also carried out in this chapter. The findings indicated that the developed adaptive sampling methods were more effective than conventional sampling methods. The sampled versions produced using adaptive statistical sampling techniques were more representative to the original population than the sampled versions produced using conventional sampling techniques. The significant difference between adaptive statistical sampling techniques and conventional sampling techniques was the manner of sampling traffic. The sample interval was adjusted during the sampling process according to traffic's statistics in case of adaptive statistical sampling approaches based on linear adjustment mechanism, quarter adjustment mechanism, and FIS. In other words, the sampling rate was decreased when the statistics of the traffic did not significantly change and increased when the statistics of the traffic significantly changed.

Conversely, the sampling rate of conventional sampling techniques was either constant as in systematic sampling or changed randomly as in stratified and random sampling. The fixed and random sampling rates may result in a significant discrepancy between the actual data and its sampled version.

Advantages of the proposed adaptive statistical sampling techniques were the ease of implementation and the quick response to traffic changes.

Chapter 6 Techniques to Evaluate Network Quality of Service Using Statistical and Artificial Intelligence

6.1 Introduction

Evaluation of QoS is an important task in managing computer networks. This is currently carried out by analysis or measurement techniques. Analysis techniques are used to examine the characteristics of the traffic, whereas measurement techniques are applied to determine how well the network treats an ongoing traffic. The contribution of this study is to propose mechanisms that combine analysis and measurement techniques to evaluate QoS in multimedia applications in an effective manner. In this chapter, two innovative QoS evaluation approaches are proposed. The first approach combined Fuzzy C-Means (FCM) and regression model to analyse and assess QoS of multimedia applications in a simulated network, whereas the other analysed and assessed QoS in multimedia applications using a combination of supervised and unsupervised neural networks. The transmitted application's QoS parameters were initially analysed either by FCM clustering algorithm or by the unsupervised learning Kohonen neural network (i.e. Self-Organising Maps (SOM)). The analysed QoS parameters were then used as inputs to a regression model or supervised learning Multi-Layer Perceptron (MLP) neural network in order to quantify the overall QoS. The proposed QoS evaluation system provided information about the network's QoS patterns and based on this information, the overall network's QoS was successfully quantified.

Section 6.2 of this chapter presents a discussion of the related studies. The proposed FCM clustering algorithm, regression model, Kohonen neural network, and Multi-Layer Perceptron (MLP) neural network in addition to the simulation of experiments and traffic models are discussed in section 6.3. The findings of these studies are presented in section 6.4. Finally, the summary of this chapter is provided in section 6.5.

6.2 Related Works

In section 3.3 (see Chapter 3), taxonomy of quality of service evaluation with the state of the art of recent related studies were discussed in details. Previous studies generally aim to evaluate network quality of service either by analysing the characterises of network traffic as reported in (Chen et al, 2009), (Timo et al, 2002), (Hoang et al, 2010), (Wang et al, 2009), (Ting et al, 2010), and (Kiziloren and Germen, 2007) or by measuring the network performance to determine how well the network treats the ongoing traffic as reported in (Palomar et al, 2008), (Bräuer et al, 2008), (Mishra and Sharma, 2003), (Al-Sbou et al, 2008), (Brekne et al, 2002), and (Mohammed et al, 2001). A novel aspect of this study is to propose an evaluation system that combines analysis and measurement techniques to effectively evaluate network QoS. The first contribution of this study is to analyse and classify network QoS parameters (i.e. delay, jitter, and packet loss ratio) using either Fuzzy C-means (FCM) or Self Organizing Maps (SOM). Due to the ability of FCM and SOM to derive meaning from imprecise values as reported in (Cirstea et al, 2002) and because of the natural characteristics of network QoS parameters, where a single cluster could not be clearly identified, FCM algorithm and SOM are suitable for QoS analysis.

Another contribution of this study is to develop QoS assessment techniques. The proposed techniques should evaluate QoS in a manner similar to human subjects and quantify the QoS without the necessity for complex mathematical models as in objective approaches taking into the account the QoS requirements of each type of multimedia application. Also, the proposed QoS assessment techniques should not add extra load to the network as the case of active approaches, nor depend on the whole collected packets, like passive approaches. The proposed assessment techniques are based on the analysed traffic generated from the proposed analysis techniques in order to overcome some drawbacks of both active and passive measurement methods. A regression model was developed and multi-layer perceptron (MLP) was trained to combine the QoS parameters (i.e. delay, jitter, and packet loss ratio) for each QoS class identified by SOM or FCM to estimate the overall QoS.

6.3 Description of the Approaches

A schematic diagram of the proposed network QoS evaluation systems is shown in Figure 6-1. The evaluation system combined analysis and measurement techniques in order to effectively evaluate network QoS. The evaluation system was a combination of FCM algorithm and regression model or a combination of supervised neural network (i.e. MLP architecture) and unsupervised neural networks (i.e. Kohonen network). As shown in Figure 6-1, the QoS parameters (i.e. delay, jitter, and packet loss ratio) were obtained from the simulated network. The extracted QoS parameters were then used as inputs to the FCM clustering algorithm or to the Kohonen network to be analysed and organised into correlated groups. The regression model and MLP relied on the identified groups of QoS parameters generated by FCM or Kohonen network to assess the overall QoS. The following subsections explain how the aforementioned mechanisms were developed to perform the QoS evaluation process.

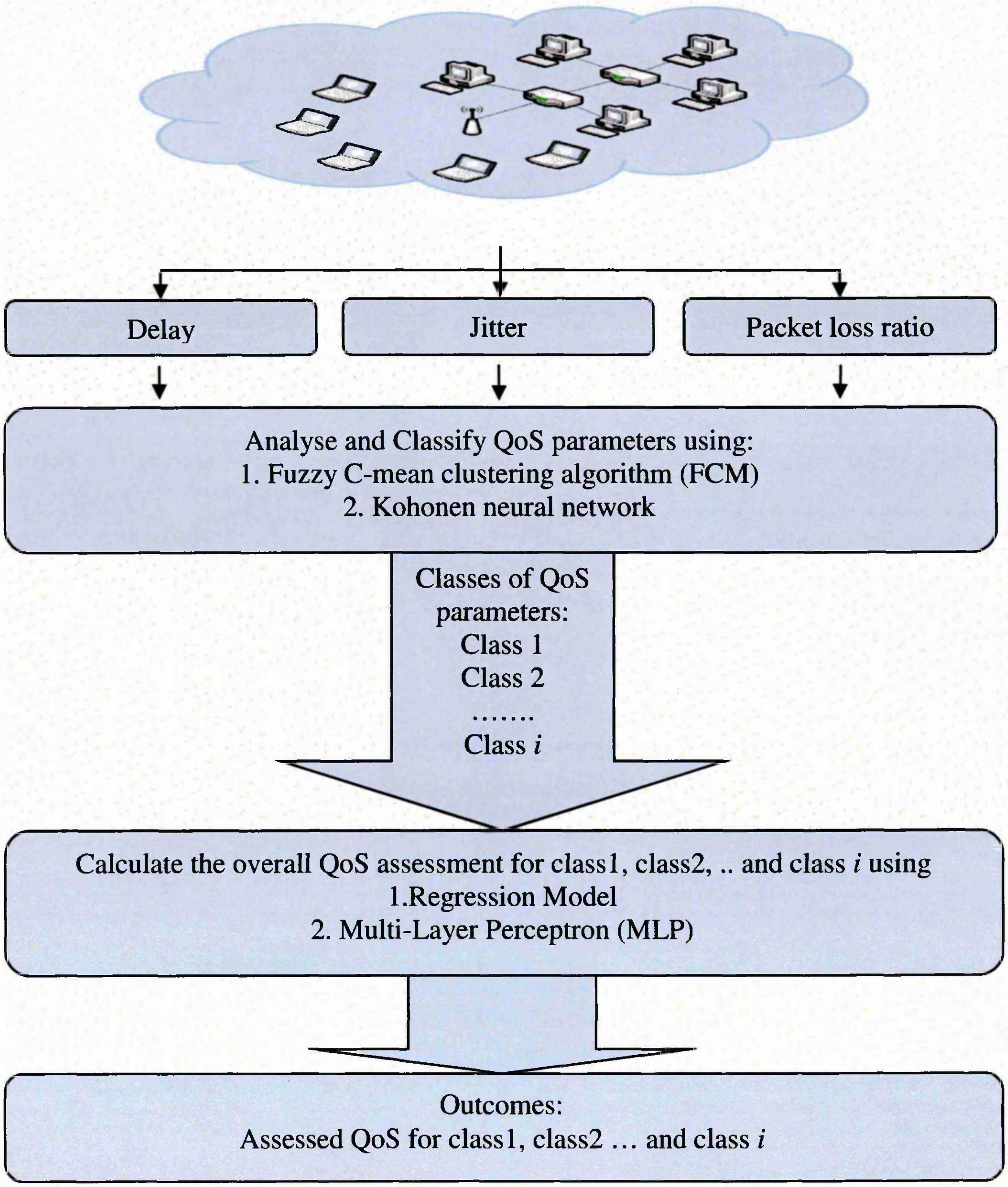


Figure 6-1. QoS evaluation system.

6.3.1 Analysis of QoS using Fuzzy C-means Clustering Algorithm

FCM was developed to partition the QoS parameters (i.e. delay, jitter, and packet loss ratio) of critical real-time applications (i.e. VoIP and video applications) into clusters. This classification provided an informative view about multimedia traffic behaviour and consequently discovered valuable information from ongoing traffic.

The values of delay, jitter, and packet loss ratio were measured respectively using equations (2.2), (2.4), and (2.6) which can be found in section 2.2.2, chapter 2. The measured QoS parameters values were then represented by a matrix (*QoS_P*) to be used as inputs to FCM algorithm as shown in equation (6.1)

$$QoS\_P = \begin{bmatrix} D_1 & J_1 & PLR_1 \\ D_2 & J_2 & PLR_2 \\ \vdots & \vdots & \vdots \\ D_n & J_n & PLR_n \end{bmatrix} \tag{6.1}$$

where *QoS_P* is the matrix of QoS parameters, $D_j, J_j,\ and\ PLR_j$, $j = 1,2,\dots,n$ are the measured delay, jitter, and packet loss ratio respectively. In this study, FCM was employed at predefined time interval. FCM operated on the matrix (*QoS_P*) and minimised the FCM objective function given in equation (6.2) in order to partition (*QoS_P*) matrix into (*C*) clusters, generate membership matrix (*U*), and produce clusters centres (*V*).

$$J\,(QoS\_P; U,V) = \sum_{i=1}^{C}\sum_{j=1}^{n}(\mu_{ij})^m\ D_{ij}^2 \tag{6.2}$$

The exponent value for partition (i.e. *m*) controls the degree of fuzziness for the membership of the clusters (Chen et al, 2009). This is commonly set to 2 as other values either introduce insufficient or too much fuzziness.

During the clustering process as explained in section 2.4.3 (chapter 2), the elements of U were updated using Equation (2.17), the clusters centres $V = \{v_1, v_2, \dots, v_C\}$ were calculated according to Equation (2.18), and the Euclidian distance D_{ij}^2 between $D_j, J_j,\ and\ PLR_j$ to the centre v_i of cluster i were calculated using equation (2.19). The clustering process was terminated when the maximum number of iteration was performed or the objective function improvement between two consecutive iterations was less than the minimum set amount of improvement.

In this study, the maximum number of iterations was 200, and the minimum set amount of improvement was 10^{-5}. These parameters were chosen experimentally, i.e. different values were experimented to monitor the FCM clustering response and the best values were selected.

The generated membership matrix U contained values indicating the degree of membership between vectors D_j, J_j, PLR_j and cluster C_i. The generated membership matrix U expressed as

$$U = \begin{bmatrix} \mu_{d1} & \mu_{j1} & \mu_{PLR1} \\ \mu_{d2} & \mu_{j2} & \mu_{PLR2} \\ \vdots & \vdots & \vdots \\ \mu_{dn} & \mu_{jn} & \mu_{PLRn} \end{bmatrix} \tag{6.3}$$

The number of clusters C was chosen based on the Xie-Beni index cluster validity method (Xie and Beni, 1991). The function of Xie-Beni index method is defined by

$$S = \frac{\sum_{i=1}^{C}\sum_{j=1}^{n} \mu_{ij}{}^{2}\left\|v_i - x_j\right\|^2}{n\, min_{i,j}\left\|v_i - v_j\right\|^2} \tag{6.4}$$

In this study, the optimal number of clusters for VoIP and video traffic was three, associated with small Xie-Beni index S=0.0002 for VoIP, and S=0.0004 for video traffic. The FCM algorithm produced three clusters classifying the QoS parameters into three categories: Low, Medium, and High. These clusters were represented by clusters centres matrix V as expressed as

$$Clusters\ centres\ (V) = \begin{bmatrix} D_{c1} & J_{c1} & PLR_{c1} \\ D_{c2} & J_{c2} & PLR_{c2} \\ D_{c3} & J_{c3} & PLR_{c3} \end{bmatrix} \tag{6.5}$$

where D_{ci}, J_{ci}, PLR_{ci} , $i = 1,2,3$ are the cluster centres of delay, jitter, and packet loss ratio respectively.

6.3.2 Analysis of QoS using Kohonen Neural Network

Kohonen neural network (i.e. Self Organising Map SOM) with size of 100 neurons (i.e. a 10 by 10 neuron grid) as shown in Figure 6-2 was trained to partition the QoS parameters of multimedia applications (i.e. VoIP and video) into their correlated groups. The aim of partition using SOM was to provide information about multimedia traffics behaviour and their inherent groups of QoS.

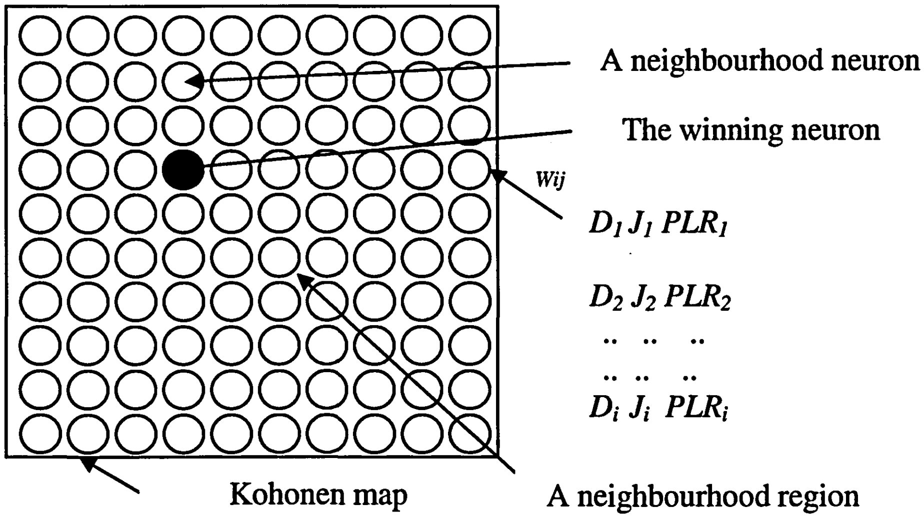


Figure 6-2. Kohonen neural network.

Prior to the training process, the values of QoS parameters for VoIP or video application were pre-processed and arranged to form examples to train the Kohonen neural network. The values of QoS parameters (delay, jitter, and packet loss ratio) were measured using equations (2.2), (2.4), and (2.6). The training examples considered the classification of QoS parameters of VoIP and video applications listed in Table 2-1 (see chapter 2). The QoS parameters of VoIP (i.e. delay, jitter, and packet loss ratio) were classified into: Low (i.e. delay $\leq$ 150 ms, jitter $\leq$ 1 ms, and packet loss ratio $\leq$ 2%), Medium (i.e. 150 < delay $\leq$ 400 ms, 1 < jitter $\leq$ 3 ms, and 2% < packet loss ratio $\leq$ 4%), and High (i.e. delay > 400 ms, jitter > 3 ms, and packet loss ratio > 4%), whereas the QoS parameters of video application were classified into: Low (i.e. delay $\leq$ 150 ms, jitter $\leq$ 10 ms, and packet loss ratio $\leq$ 1%), Medium (i.e. 150 < delay $\leq$ 400 ms, 10 < jitter $\leq$ 20 ms, and 1% < packet loss ratio $\leq$ 2%), and High (i.e. delay > 400 ms, jitter > 20 ms, and packet loss ratio > 2%). After considering the classification of QoS parameters and during the arrangement phase, the QoS parameters (delay, jitter, and packet loss ratio) were labelled to provide visual differentiation between the generated QoS classes. Label (L) indicated Low range for delay, jitter, and packet loss ratio. Label (M) indicated Medium range for delay, jitter, and packet loss ratio, whereas label (H) indicated High range for delay, jitter, or packet loss ratio. The values of QoS parameters and their labels were represented in a matrix notation as shown equation (6.6) in order to train the Kohonen network.

$$QoS\ parametres = \begin{bmatrix} D_1 & J_1 & PLR_1 & L_1 \\ D_2 & J_2 & PLR_2 & L_2 \\ \vdots & \vdots & \vdots & \\ D_n & J_n & PLR_n & L_n \end{bmatrix} \tag{6.6}$$

where $D_i, J_i, PLR_i\ L_i,\ i = 1,2,\dots,n$ were respectively the measured delay, jitter, packet loss ratio, and their labels (i.e. L,M, or H).

During the training process, the i^{th} input feature of delay, jitter, or packet loss ratio was connected to j^{th} neuron in Kohonen map. Each connection was associated with weight w_{ij}. The network learned by determining the Euclidean distance d_j between input patterns with N elements and the connection weights w_{ij} for j^{th} neuron as explained in Equation (2.23) (see chapter 2). The neuron where its associated weights provided the smallest Euclidean distance to the input pattern was considered as the winning neuron. The weights associated with the winning neuron were then updated according to the Kohonen learning rule represented by Equation (2.24). This ensured that the winning neuron's weights iteratively moved closer to the specific input pattern category. However, the weights associated with a number of neurons around the winning neuron were also updated to a lesser extent. This allowed improved training.

In this study, the maximum number of training iterations was 1000. According to (Vesanto et al, 1999), the number of training steps should be at least 10 times of the number of neurons. The initial learning rate of learning Kohonen algorithm η was set to (0.5). This value was decreased inversely proportional to the number of iterations (Haykin, 1999). This allowed the training to be course to start with and then it became gradually finer. The analysis process of Kohonen network produced three distinct groups that correlated with low, medium, and high QoS.

During the test phase, the trained Kohonen neural network was fed with other values of QoS parameters in order to classify the QoS parameters into their correlated groups. The groups produced by the Kohonen network provided information about the relationships between different QoS parameters of transmitted VoIP or video traffics and subsequently discovered relevant information about network operation.

6.3.3 QoS Assessment using Regression Model

Due to the relevance between the quality of multimedia applications and their QoS parameters (delay, jitter, and packet loss ratio) as shown in Table 2-1 (see Chapter 2), regression model was devised in this study to combine the QoS parameters of VoIP, and video applications in order to quantify the overall QoS.

The regression model was developed by using the theory of regression analysis discussed in Section 2.4.1, Chapter 2. The values of the independent variables (x_1, x_2, x_3) in the regression model were represented by delay, jitter, and packet loss ratio respectively, whereas the values of dependent variable (y) were represented by the overall QoS. The regression expression was determined by considering the QoS requirements listed in Table 2-1 in order to provide the outputs that reflect the overall QoS. The QoS parameters shown in Table 2-1 were categorised as: Low, Medium, and High. The overall QoS on the other hand was classified as Good, Average, and Poor quality corresponding to the categories of QoS parameters. In this study, the overall QoS spanned between (0-100%). QoS=0% represented the worst case of network performance, whereas the best network performance was for QoS=100%.

The regression formula for VoIP application was calculated by arranging the values of independent variables (i.e. delay, jitter, and packet loss ratio) and the values of dependent variable (i.e. overall QoS) into matrices as follows: Low QoS parameters (i.e. delay $\leq$ 150 ms, jitter $\leq$ 1 ms, and packet loss ratio $\leq$ 2%) corresponded to good overall QoS which ranged between (67-100%), medium QoS parameters (i.e. 150 $<$ delay $\leq$ 400 ms, 1 $<$ jitter $\leq$ 3 ms, and 2% $<$ packet loss ratio $\leq$ 4%) corresponded to average QoS (i.e.33% $<$ QoS $\leq$ 67%), whereas high QoS parameters (i.e. delay $>$ 400 ms, jitter $>$ 3 ms, and packet loss ratio $>$ 4%) corresponded to poor QoS (i.e. QoS $\leq$ 33%).

The regression formula for video application was calculated using the same mapping used in VoIP application. However, good overall QoS which ranged between (67-100%) corresponded to low value of delay $\leq$ 150 ms, jitter $\leq$ 10 ms, and packet loss ratio $\leq$ 1%), average QoS (i.e.33% $<$ QoS $\leq$ 67%) corresponded to medium QoS parameters (i.e. 150 $<$ delay $\leq$ 400 ms, 10 $<$ jitter $\leq$ 20 ms, and 1% $<$ packet loss ratio $\leq$ 2%), whereas high QoS parameters (i.e. delay $>$ 400 ms, jitter $>$ 20 ms, and packet loss ratio $>$ 2%) corresponded to poor QoS (i.e. QoS $\leq$ 33%).

After the mapping process of QoS parameters to the overall QoS, the matrices of QoS parameters and the overall QoS were then organised to form the regression formula as expressed by

$$\begin{bmatrix} QoS_1 \\ QoS_2 \\ \vdots \\ QoS_n \end{bmatrix} = \begin{bmatrix} 1 & D_1 & J_1 & PLR_1 \\ 1 & D_2 & J_2 & PLR_2 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & D_n & J_n & PLR_n \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{bmatrix} \tag{6.7}$$

where $D_i, J_i, PLR_i, QoS_i, i = 1,2,\dots,n$ are delay, jitter, packet loss ratio, and overall QoS of VoIP or video applications. The regression coefficients b_0, b_1, b_2, b_3 were determined from the recorded data using equation (2.12). The vector of residual (i.e. error terms) was then calculated using equation (2.13). In this study, the calculated errors produced from regression formula for VoIP and video traffics were normally distributed. This indicated that the mean of error terms $e_i, i = 1,2,\dots,n$ was zero. This implied that the estimated regression model determined was:

$$QoS_i = b_0 + b_1 * D_i + b_2 * J_i + b_3 * PLR_i \tag{6.8}$$

where $QoS_i, D_i, J_i, PLR_i, i = 1,2,\dots,n$ are the overall QoS, delay, jitter, packet loss ratio for i^{th}packet respectively.

6.3.4 QoS Assessment using Multi-Layer Perceptron

A Multi-Layer Perceptron (MLP) neural network was chosen to assess the overall QoS due to its suitability and effectiveness (Abraham, 2005). The proposed MLP neural network model in this study composed of an input layer with three neurons, a hidden layer with three neurons, and an output layer with one neuron as shown in Figure 6-3.

The inputs D_i, J_i, and PLR_i fed into the input layer of MLP were delay, jitter, and packet loss ratio respectively, whereas the values of desired output (QoS_i) were represented by the overall QoS.

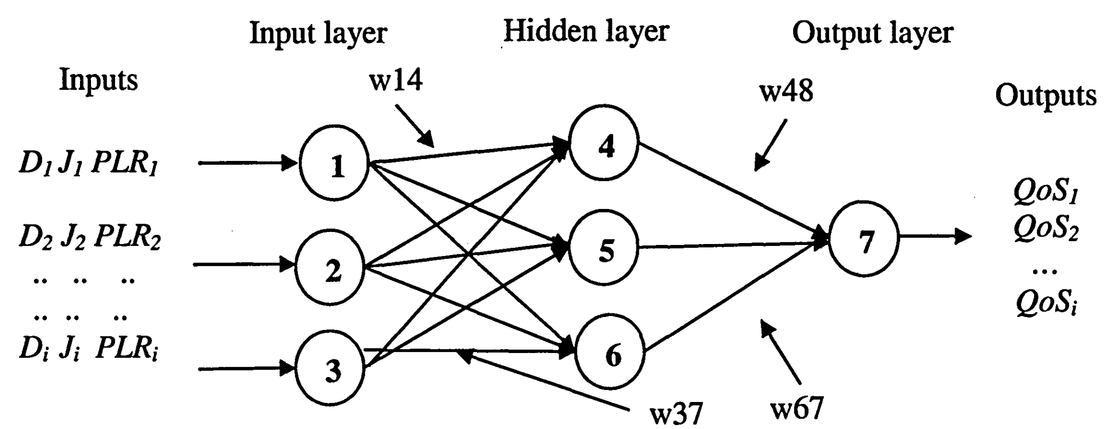


Figure 6-3. Structure of an MLP.

The inputs (i.e. delay, jitter, and packet loss ratio) and their desired output(i.e. QoS) were arranged in a matrix by considering the QoS requirements listed in Table 2-1 in order to provide outputs that reflected the overall QoS. The QoS parameters of VoIP and video application shown in Table 2-1 were categorised into three classes: Low, Medium, and High. The overall QoS on the other hand was classified as Good, Average, and Poor quality corresponding to the categories of QoS parameters. The overall QoS spanned between (0-100%).

Low QoS parameters of VoIP application (i.e. delay < 150 ms, jitter < 1 ms, and packet loss ratio $< 2\%$) corresponded to good overall QoS which ranged between (67-100%), medium QoS parameters (i.e. $150 \le$ delay ≤ 400 ms, $1 \le$ jitter ≤ 3 ms, and $2\% \le$ packet loss ratio $\le 4\%$) corresponded to average QoS (i.e.$33\% <$ QoS $\le 67\%$), and high QoS parameters (i.e. delay > 400 ms, jitter > 3 ms, and packet loss ratio $> 4\%$) corresponded to poor QoS (i.e. QoS $\le 33\%$). Whereas in video application, good overall QoS which ranged between (67%-100%) corresponded to low value of delay $\le$ 150 ms, jitter ≤ 10 ms, and packet loss ratio $\le 1\%$), average QoS (i.e.$33\% <$ QoS $\le$ 67%) corresponded to medium QoS parameters (i.e. $150 <$ delay ≤ 400 ms, $10 <$ jitter ≤ 20 ms, and $1\% <$ packet loss ratio $\le 2\%$), and high QoS parameters (i.e. delay > 400 ms, jitter > 20 ms, and packet loss ratio $> 2\%$) corresponded to poor QoS (i.e. QoS $\le$ 33%).

The matrix expressed in equation (6.9) which included training examples was fed into MLP for its training. The matrix represented the QoS parameters and the overall QoS where $D_i, J_i, PLR_i, QoS_i, i = 1,2,\dots,n$ were delay, jitter, packet loss ratio, and the overall QoS respectively:

$$\text{Training examples} = \begin{bmatrix} D_1 & J_1 & PLR_1 & QoS_1 \\ D_2 & J_2 & PLR_2 & QoS_2 \\ \vdots & \vdots & \vdots & \vdots \\ D_n & J_n & PLR_n & QoS_n \end{bmatrix} \tag{6.9}$$

During the training phase of MLP, the inputs (D_i, J_i, PLR_i) were multiplied with their associated weights (W_i) and the resulting values are summed by the summation function using equation (2.21) (see Chapter 2). The output from summation function (s) was then processed by the activation function to produce the output (y). In this study, the hyperbolic tangent activation function was used as it gives continuous output between -1 and +1 and thus it gave a larger range than sigmoid activation function which provides a range between 0 and 1 (Karlik and Olgac, 2010). Equation (6.10) formulates the hyperbolic tangent function:

$$\varphi(s) = \frac{1-e^{-s}}{1+e^{-s}} \tag{6.10}$$

The calculated output $\varphi(s)$ (i.e. y) was subtracted from the desired output (QoS_i) as in equation (2.22) (See Chapter 2) to produce an error (e) which in turn was used by the learning algorithm in order to reduce the magnitude of the error in the next iteration.

In this study, the commonly used a gradient descent with momentum learning algorithm was employed. The function of this algorithm was to use the calculated error (e) and the input data (D_i, J_i, PLR_i) to the processing neuron to adjust the values of the connections' weights (W_i) which in turn reduced the magnitude of the error in the following training iteration. Gradient descent with momentum learning algorithm updated the weights using equation (6.11) (Eberhart and Dobbins, 1990):

$$W_{new} = W_{old} + \eta\ (e)x + \alpha\ [\Delta\, W_{old}] \tag{6.11}$$

where W_{new} is the new updated weight, W_{old} is the previous weight, the term η is the learning rate parameter, (e) is the calculated error, x is the input to the processing element, the momentum term (α) ensures that the learning algorithm does not get stuck in a local minimum and finds the desired global minimum, and the term $\Delta\, W_{old}$ represents the amount of change in the weights from the previous iteration.

The learning rate parameter η is a positive constant limited to the range $0 < \eta \leq 1$ whereas the momentum factor α can take values between 0 and 1. In this study, the learning rate parameter η and the momentum factor α were set to their default values which were 0.01 and 0.9 respectively. These values provided the most reliable results.

The training process of MLP was terminated when the maximum number of iterations reached (in this study was 1000) or when there was insignificant error (i.e. 0.001) between network output (y) and desired result (QoS_i). During the test phase, the trained MLP was fed with other values of QoS parameters (D_i, J_i, PLR_i) in order to assess the overall QoS. The actual output values used to train MLP and the MLP output values during test phase were then correlated to ensure that MLP had been trained effectively.

6.3.5 Measuring Predication Accuracy

The prediction accuracy of proposed QoS assessment methods was measured using a correlation coefficient. The correlation coefficient (R) is widely used to evaluate the prediction accuracy (Chatterjee and Hadi, 2006). The magnitude of R is between 0 and 1. The magnitude closest to 1 indicates a perfect correlation, whereas a correlation less than 0.5 would be described as weak correlation. The correlation between the actual values (y_A) and predicted values (y_P) is calculated using equation (6.12).

$$R_{y_A\ ,\ y_P} = \frac{\Sigma(y_{Ai} - \overline{y_A})(y_{Pi} - \overline{y_P})}{\sqrt{\Sigma(y_{Ai} - \overline{y_A})^2\, \Sigma(y_{Pi} - \overline{y_P})^2}} \tag{6.12}$$

where y_A is the actual value, $\overline{y_A}$ is the mean of the actual values, y_P is the predicted value, and $\overline{y_P}$ is the mean of the predicted values.

6.3.6 Network Simulation and Traffic Models

A wireless-cum-wired network topology illustrated in Figure 4-3 was simulated using Network Simulator- 2 (NS-2). The network topology consisted of 10 wireless nodes, 10 wired nodes, and 1 base station to form 10 unidirectional connections. At the wired side of the network, all the links had 5 Mbps bandwidth and 2 ms propagation delay. The queue management mechanism was Drop-Tail and the queue size was 50 packets. The WLAN side of the network was based on IEEE 802.11e, and it used the Enhanced Distributed Channel Access (EDCA) scheme. The main parameters that modelled the wireless channel were the default settings for IEEE 802.11e as shown in Table 4-1.

The types of traffic transmitted over the simulated network were: VoIP, video streaming, best effort and background. VoIP was modelled as G.711 voice encoding scheme by adapting Constant Bit Rate (CBR) traffic. The packet size for VoIP was 160 bytes and the transmission rate was 64 kbps. The video streaming source was YUV QCIF (176 × 144) Highway (2000 frame) (YUV QCIF, 2012). Prior to its transmission, each video frame was fragmented into packets that in turn had a maximum length of 1024 bytes. The best-effort traffic had a fixed packet size of 1000 bytes and 125 kbps transmission rate. File Transfer Protocol (FTP) application was used for the background traffic. FTP was transmitted over TCP, whereas other traffics were transmitted using UDP transport protocol. The transmitted traffic over IEEE 802.11e EDCA (i.e. VoIP, video, best effort traffic, and background traffic) were mapped into different access categories to represent different priority levels, as shown in Table 4-2. VoIP had highest priority, whereas the priority of the background traffic was lowest.

The simulation time was 500 seconds. During the first third of the simulation, two VoIP applications, and two best effort traffics were transmitted. During the period (170s-

340s), the number of transmitted applications increased to three VoIP, two video, two best effort traffics, and background traffic making the network load heavy and approximately 100% of channel capacity was used. However, after the transmission of video traffic was completed, the network load became light during the last third of simulation time.

6.4 Results and Discussion

This section is divided into four subsections: two sections provide QoS analysis results for Fuzzy C-Means and Kohonen Neural Network. The remaining two sections provide QoS assessment results using Regression Model and Multi-Layer Perceptron.

6.4.1 QoS Analysis using FCM Clustering Algorithm

In this study, FCM clustering algorithm was applied at predefined regular time intervals to analyse VoIP and video traffic. FCM analysis results for VoIP traffic during the simulation time interval 350 - 400 seconds are shown in Figures 6-4. The values of QoS parameters (i.e. delay, jitter, and packet loss ratio) of VoIP were grouped into three clusters, representing Low, Medium, and High values. Each cluster was represented by its own centre. Figure 6-4 shows that the packet loss ratio of VoIP is zero. This is because VoIP was mapped to the access category that had the highest priority. However, due to the heavy network load, a number of VoIP packets experienced fluctuated delay that in turn resulted in high jitter values.

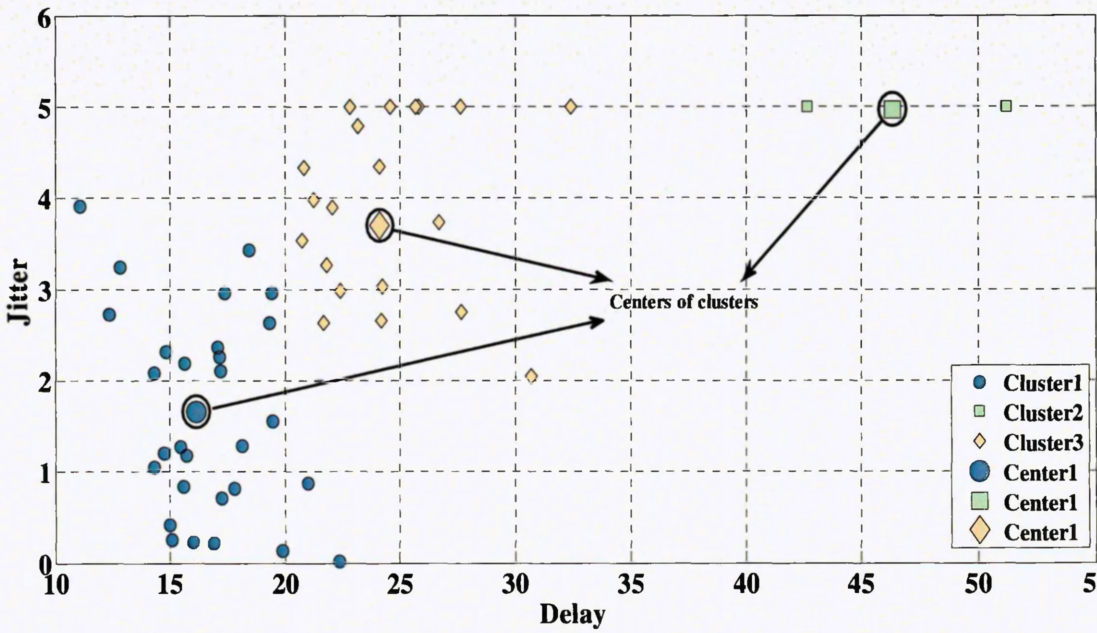


Figure 6-4. Clustering QoS parameters of VoIP application at predefined time interval.

FCM analysis result for video traffic during the simulation time interval 200 - 250s is presented is presented in Figure 6-5. Due to the mapping of video application to the Access Category (AC) that had a lower priority as compared with the (AC) assigned to VoIP, and the heavy load on the network, the QoS parameters (i.e. delay, jitter, and packet loss ratio) varied extensively. These variations resulted in three clusters representing Low, Medium, and High QoS parameters. During this interval, the video quality was affected by the high values of QoS parameters in cluster 3. The centre of cluster 3 characterised high delay, high jitter, and medium packet loss ratio which were respectively 540.6 ms, 24.4 ms, and 1.2%.

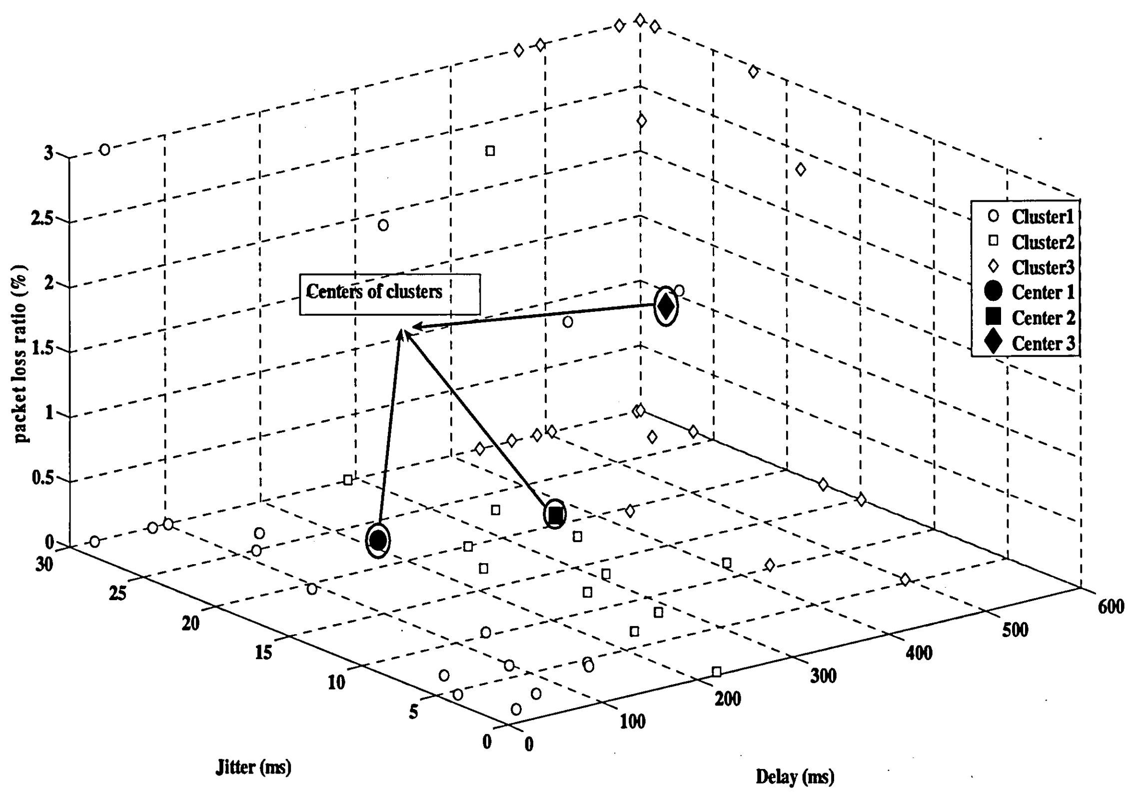
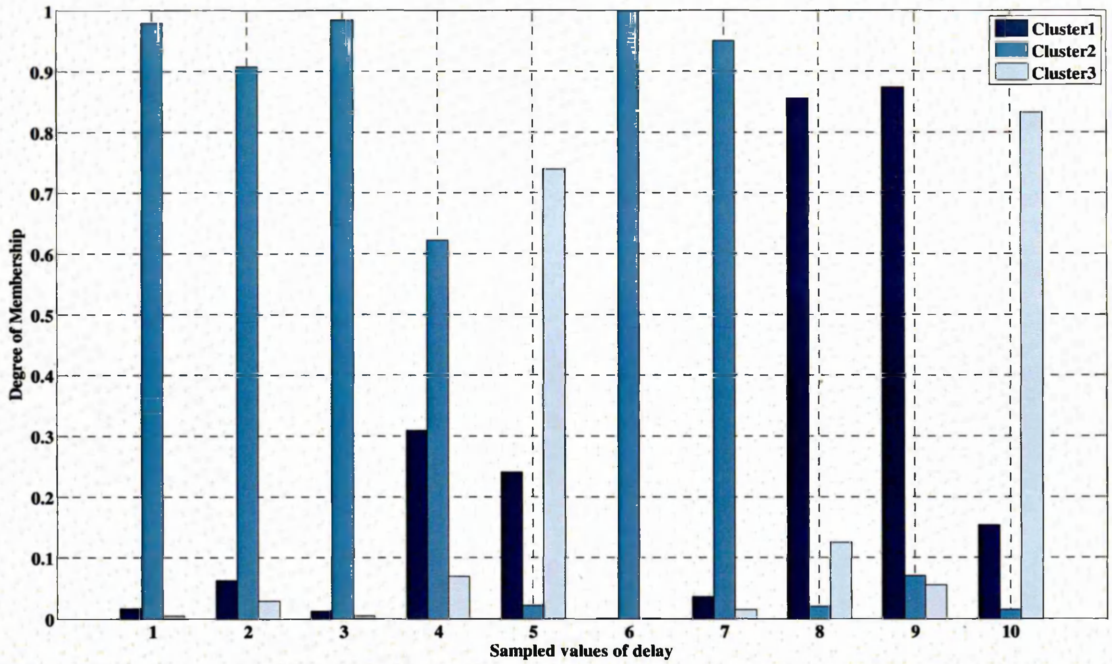
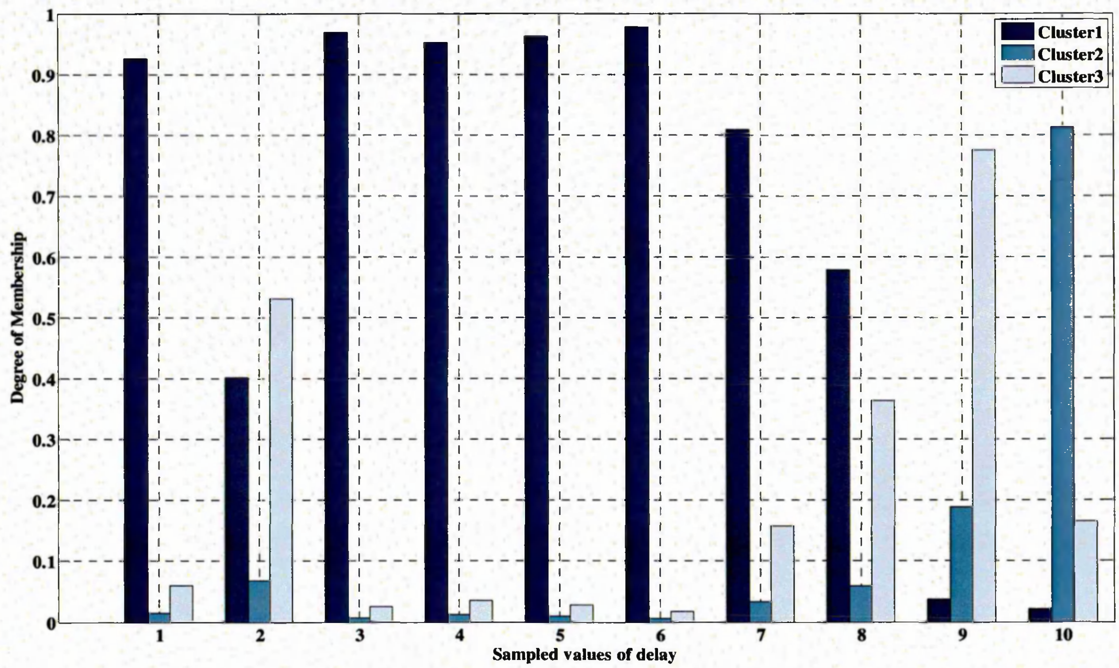


Figure 6-5. Clustering QoS parameters of video application at predefined time interval.

The fuzzy partition matrix (produced by FCM algorithm) indicated the degree of membership of each QoS parameter to each cluster. Figures 6-6 (a) - (b) show respectively the degree of membership for a sample of delay for VoIP, video application, and their clusters' centres. As indicated in the Figures, each value had different degree of membership to the three clusters between 0 and 1 with the total being 1. In this study, the fuzzy partition feature of FCM made it a valuable clustering tool because the characteristics of QoS parameters did not allow crisp (binary) partition.



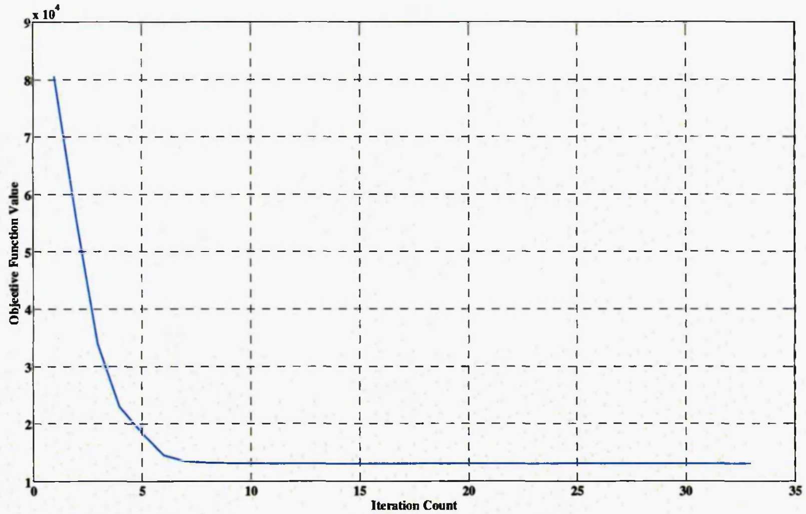
(a)



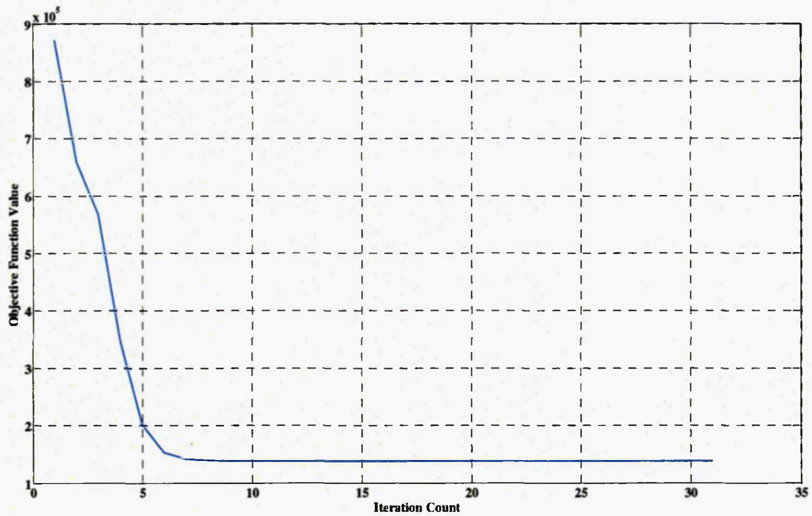
(b)

Figure 6-6. The degree of membership between a sample of delay and the cluster's center for: (a) VoIP, and (b) Video application.

During the clustering process of VoIP and video application, the objective function indicated the progress of FCM clustering algorithm over the number of iterations performed. This is shown in Figures 6-7 (a) - (b). The clustering process of FCM stopped either when it reached the maximum number of iterations or when the objective function improvement between two consecutive iterations was less than the predefined minimum amount of improvement. From Figures 6-7 (a) - (b), it can be noticed that the clustering process of VoIP and video application terminated when the objective function improvement between two consecutive iterations was less than 1e-5 (predefined minimum improvement) although the maximum number of iterations was set as 200.



(a)



(b)

Figure 6-7. The progress of objective functions during FCM analysis of: (a) VoIP, (b) Video application.

Figure 6-8 illustrates the cluster centres for 50 seconds time intervals during transmission of VoIP. The FCM algorithm classified the QoS parameters of VoIP into three levels at each time interval. The Figure shows that VoIP started its transmission approximately at 170s of the simulation time. During that time, there were two other VoIP applications using the channel since their transmission started during the first third of the simulation (i.e. 0s-170s). Therefore, during the time interval 170s-185s, the values of delay, jitter, and packet loss ratio of VoIP were at medium range of QoS parameters. As the number of transmitted applications increased to three VoIP, two video applications, two best effort traffics, and background traffic during the time interval 215s - 275s, the load became heavy. This in turn made the network incapable of meeting minimum QoS requirements for a VoIP application. The values of QoS parameters for cluster 3 were in the high range, 500 ms for delay, 5 ms for jitter, and 6% for packet loss ratio, indicating a poor quality VoIP. However, once the two video applications were transmitted during 240s - 260s, the network load became light. This allowed the network to meet the QoS requirements of the VoIP application during the time interval 275s - 500s. The values of delay, jitter, and packet loss ratio were in the low range of QoS parameters.

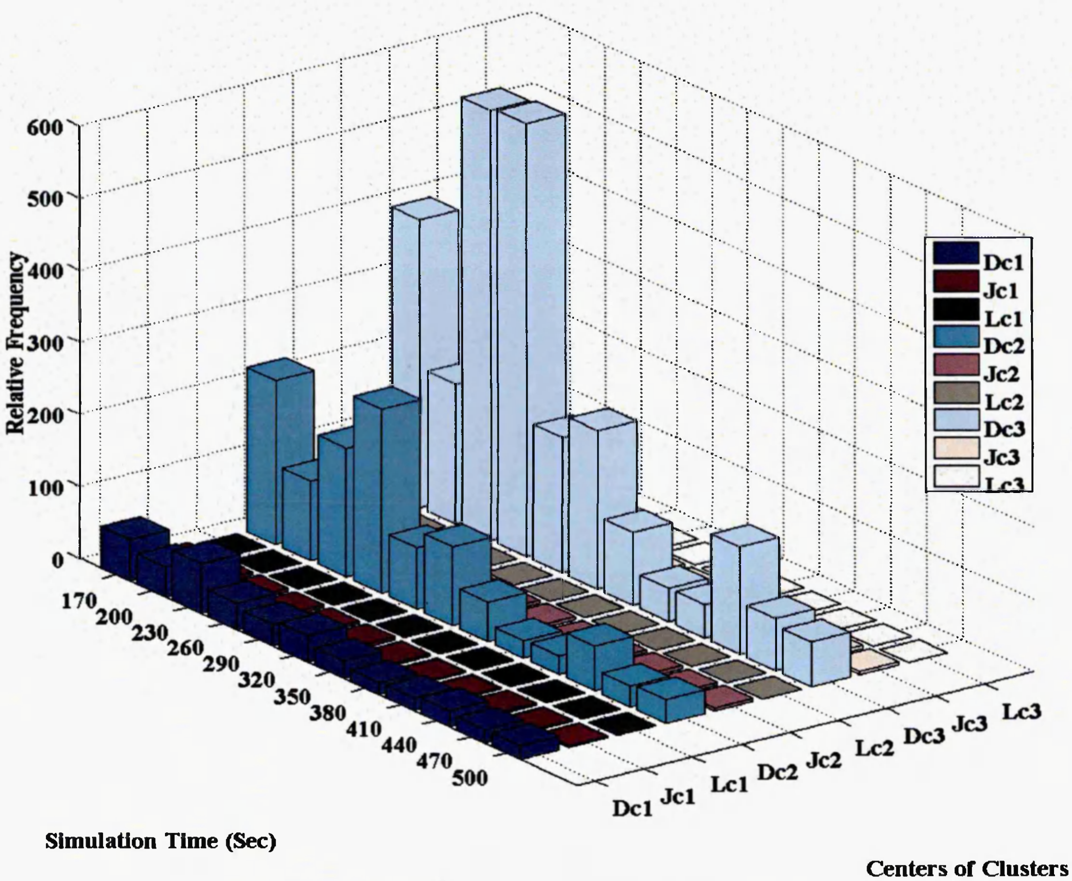


Figure 6-8. Clustering QoS parameters of VoIP.

The FCM algorithm classified the QoS parameters of video application into three levels for each time interval (i.e. 5 seconds) as shown in Figure 6-9. Video application transmission started approximately at 170s and terminated at around 240s. Figure 6-9 shows that the majority of QoS parameters values for video application were in the high range, particularly in cluster 3, around 550 ms for delay, 30 ms for jitter, and 3% for packet loss ratio during the simulation time 170s - 230s. The reasons for this were: (i) the allocation of the video application to an access category which had a lower priority than the access category that VoIP was assigned (ii) the network load was heavy due to the transmission of multiple applications during the same simulation time interval. Consequently, the network was incapable of meeting the QoS requirements for the video application.

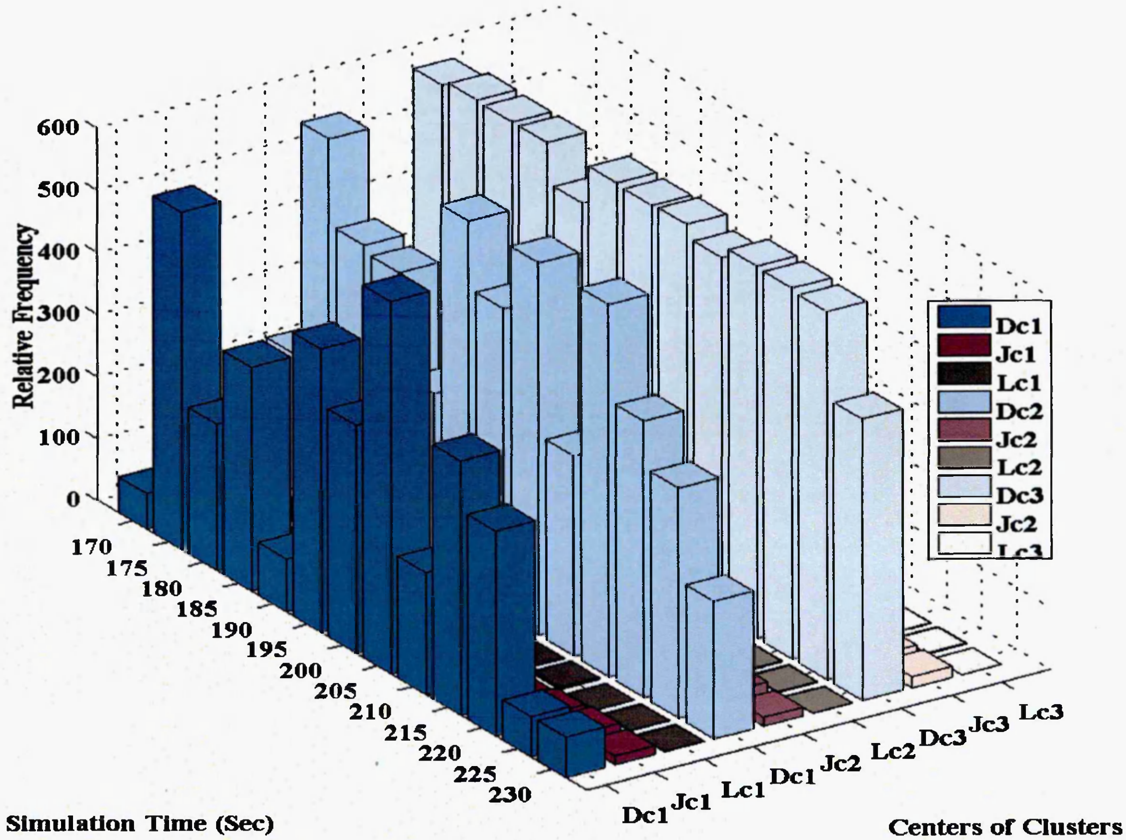


Figure 6-9. Clustering QoS parameters of video.

6.4.2 QoS Assessment using Regression Model:

In this section, the developed regression model was used to combine the QoS parameters (i.e. delay, jitter, and packet loss ratio) for each centre of the generated clusters by FCM in order to estimate the overall QoS. Figure 6-10 shows the results from the devised regression model. The results show the predicted QoS of VoIP for the centres of generated clusters at each 50 seconds time interval. Figure 6-10 shows that the QoS values reflected the corresponding QoS parameters indicated in Figure 6-8. In other words, as the values of QoS parameters decreased, the values of overall QoS increased accordingly. When VoIP started its transmission at 170s, its overall quality was poor (i.e. below 30%). This is because of its contention to access the channel which carried other VoIP applications with the same priority. As the number of transmitted applications increased to three VoIP, two videos, two best effort traffics, and background traffic during the time interval 215s - 335s, the VoIP quality degraded by overall 51.83% as compared with its quality of 55.97% during the period 185 - 215. The heavy load on the network made its performance incapable of meeting minimum QoS requirements for VoIP application. However, during the last third of simulation time, the network load became light due to the termination of two video applications. The VoIP quality increased sharply at certain time interval. For example, VoIP quality reached 87.9% at time interval 365s - 395s.

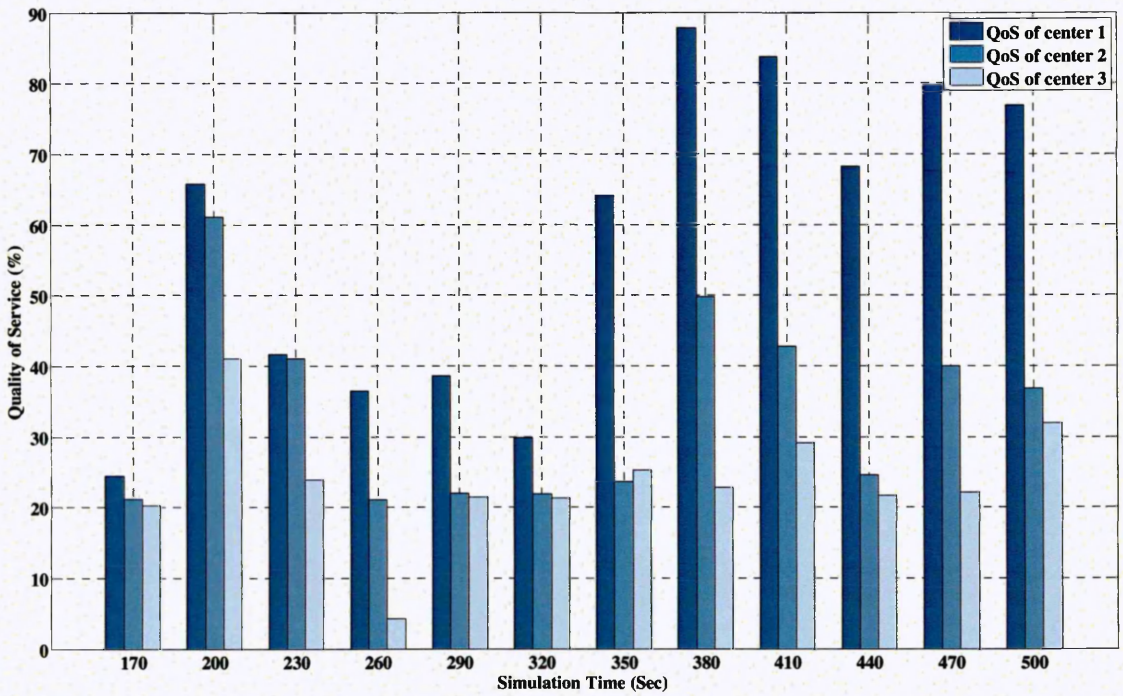


Figure 6-10. The QoS of VoIP using regression model

The developed regression model was also used to assess the QoS of video application for each centre of the FCM generated clusters shown in Figure 6-9 by combining the QoS parameters: delay, jitter, and packet loss ratio. Figure 6-11 shows the predicted QoS of the video application for the centres of generated clusters at each 5 seconds time interval. The Figure shows that the video quality was good at 170s. This is because the transmission of video started prior to other traffic such as VoIP, other video application, and background traffic during the second third of simulation time (170s - 340s). However, due to the low priority of video as compared with VoIP and the load of the network became heavy, the quality of video degraded sharply and its QoS range remained between poor and average level (i.e. 3.04% - 73.05%).

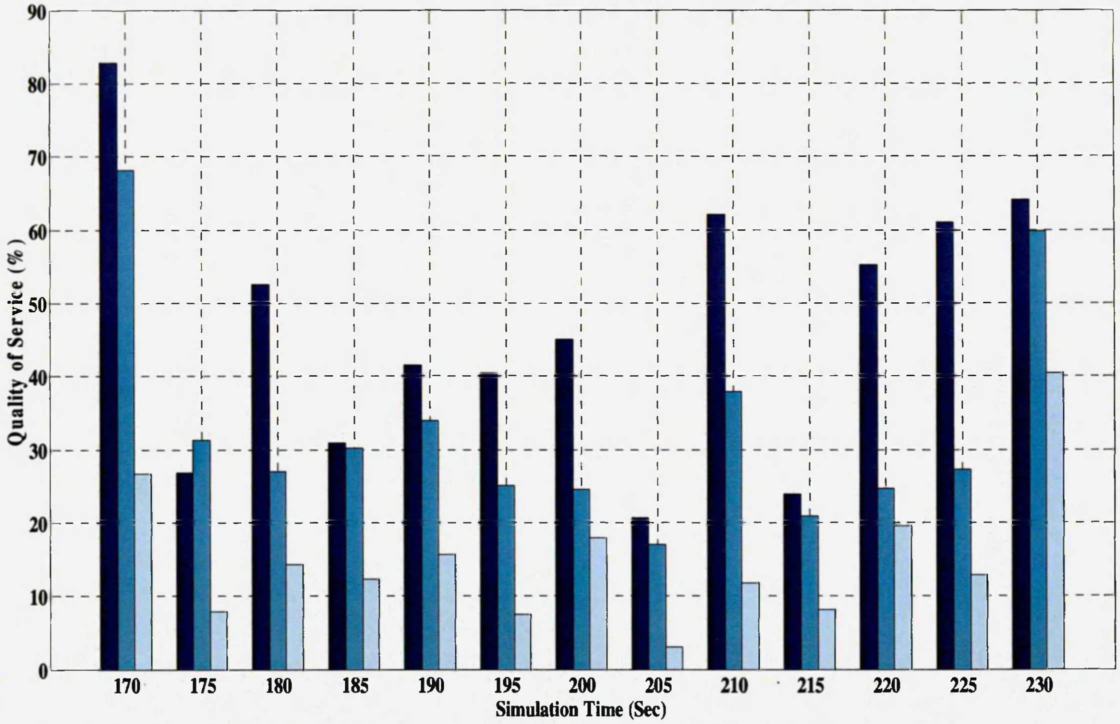


Figure 6-11. The QoS of video using regression model

The results obtained from the devised regression model were compared using other QoS assessment methods. These were: fuzzy inference system mechanism, and distance measurement evaluation system reported in (Al-Sbou, $2010_{(a)}$) and (Al-Sbou, $2006_{(b)}$). Tables 6-1 and 6-2 show respectively the comparison between the aforementioned assessment methods used to quantify the overall QoS of VoIP, and video application. From the sampled values of evaluated QoS provided in Tables 6-1 and 6-2, it is indicated that the three assessment methods provided results which were comparable. The values of the correlation coefficients between the QoS determined using regression model, and the other QoS assessment methods (i.e. fuzzy inference system mechanism, and distance measurement evaluation system) were respectively: 0.95 and, 0.97 for VoIP application, and 0.89, and 0.93 for video application. Although some outputs were slightly different, they were still in the same region (i.e. poor, average, or good). The discrepancies were due to the fact that each method followed a different operation.

However, the values of QoS obtained from devised regression model spanned between (0%-100%), whereas the range of QoS values produced by FIS was between (10%-90%). This indicates that the devised regression model provides more accurate results.

Table 6-1. QoS parameters of VoIP and expected QoS using: fuzzy inference system, distance measurement, and regression model.

QoS parameters			Assessment of overall QoS using:		
Delay (ms)	Jitter (ms)	Packet loss ratio (%)	Fuzzy inference system	Distance measurement	Regression model
36.69	4.62	0.00	9.64	29.69	29.78
108.84	5.00	0.00	9.28	24.13	21.92
599.94	5.00	6.00	9.28	0.01	4.31
24.21	2.87	0.00	71.50	55.79	64.14
54.76	4.89	0.18	9.31	25.71	23.68
101.20	4.83	0.00	9.34	26.49	25.29
16.14	1.66	0.00	81.97	95.40	87.90
24.01	3.60	0.00	36.49	44.45	49.84
46.32	4.97	0.00	9.28	24.72	22.87
17.48	1.87	0.00	82.96	93.93	83.77
25.38	3.96	0.00	18.60	39.18	42.78
46.24	4.65	0.00	9.57	29.20	29.14

Table 6-2. QoS parameters of video and expected QoS using: fuzzy inference system, distance measurement, and regression model.

QoS parameters			Assessment of overall QoS using:		
Delay (ms)	Jitter (ms)	Packet loss ratio (%)	Fuzzy inference system	Distance measurement	Regression model
57.15	13.03	0.00	80.42	92.69	82.29
85.37	24.52	2.61	13.81	24.90	40.72
289.30	25.64	1.38	12.22	32.19	33.59
534.18	24.00	1.06	10.11	20.25	15.69
600.00	29.55	1.16	9.29	8.30	3.04
599.95	9.84	0.87	9.39	17.88	25.44
599.98	29.10	0.78	9.32	8.87	7.51
367.18	17.90	0.34	42.56	43.11	44.92
492.40	24.09	0.59	11.91	23.97	24.49
597.01	16.90	0.93	9.40	17.70	18.16
600.00	5.89	1.65	9.39	17.54	21.04
600.00	17.27	0.96	9.39	17.28	17.20

6.4.3 QoS Analysis using Kohonen Neural Network

Kohonen neural network (i.e. Self-Organising Map SOM) processed the QoS parameters (i.e. delay, jitter, and packet loss ratio) of VoIP, and video applications into correlated groups. The values of delay, jitter, and packet loss ratio of transmitted VoIP and video were grouped into three classes representing Low, Medium, and High values as shown in Figures 6-12 and 6-13. Low values of delay, jitter, and packet loss ratio are represented in blue, Medium values are represented in green, and High values are represented in red.

Figures 6-12 (a), (c) show that low delay, and packet loss ratio activate most neurons in SOM. This was due to VoIP having the highest priority to access the channel amongst other transmitted traffic such as video, and best effort traffic. However, the heavy load on the network and the contention with other VoIP applications having the same priority caused high fluctuation to the delay. This in turn generated high values of jitter as shown in Figure 6-12 (b). The high values of jitter would have a negative effect on the overall VoIP quality.

The labels of the QoS parameters in Figure 6-12 (d) provided a differentiation between the three generated QoS classes. Label L identified the active neurons which represent Low values of QoS parameters; label M covered the region where Medium values of QoS parameters were located; whereas High QoS parameters were represented by other neurons with label H. The groupings produced by the Kohonen network provided information about the relationships between the different QoS parameters of transmitted VoIP traffic and subsequently discovered how VoIP was treated as high priority traffic transmitted over heavy loaded network.

Figure 6-12 (e) shows the response of a Kohonen neural network to the QoS parameters of VoIP. SOM differentiated delay, jitter, and packet loss ratio into almost three clusters. The Figure shows that Low and Medium QoS parameters have well defined regions. The winning neurons for these regions were characterised by small number of QoS values. In contrast, the neighbourhood region of high QoS parameters was large. This was because most jitter values for VoIP were high .This in turn had an impact on VoIP quality, although other values such as packet loss ratio were 0% during most of VoIP transmission period. The winning neuron for high QoS parameters region was activated by 69 values.

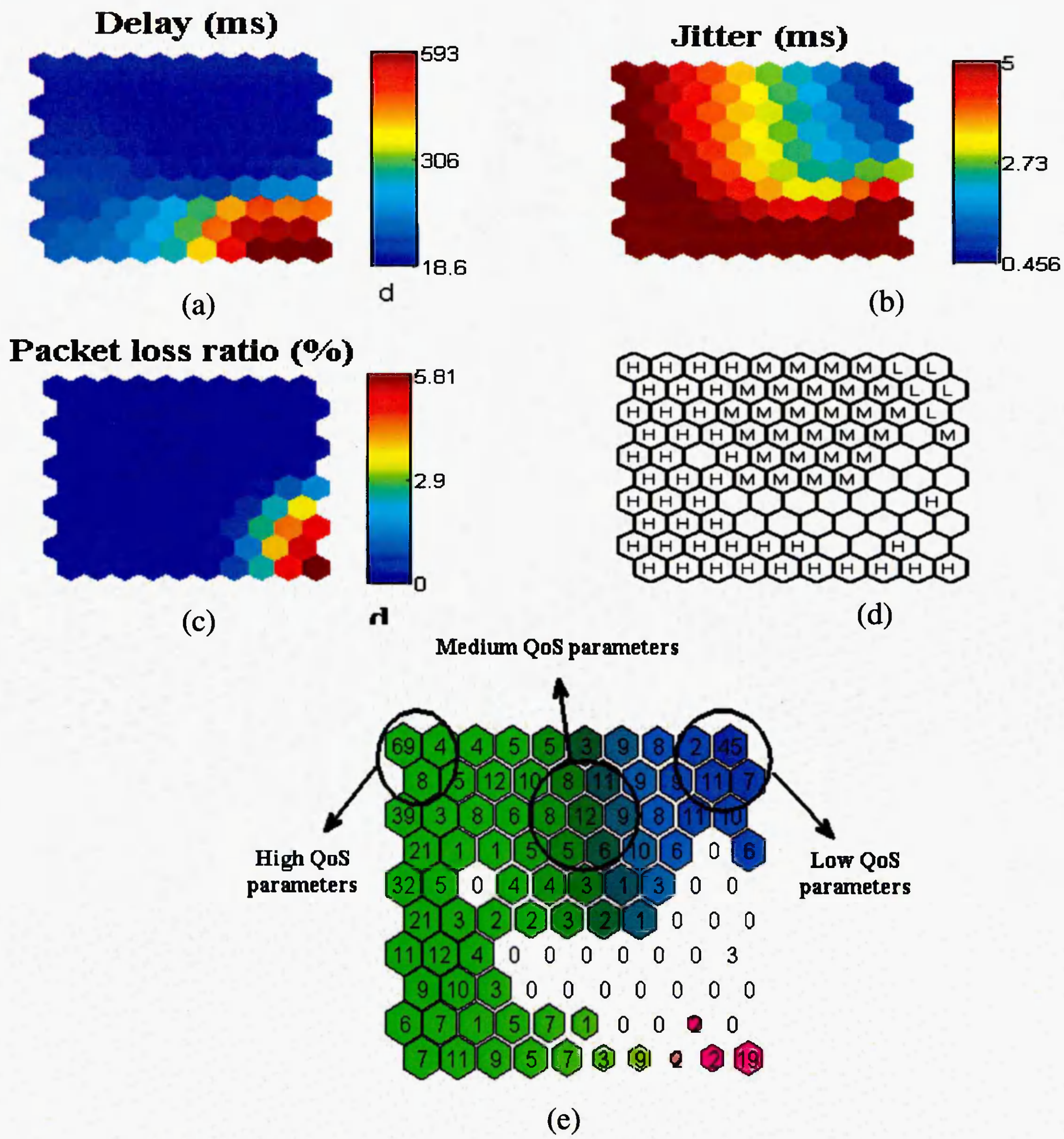


Figure 6-12. Classifying QoS parameters of VoIP using kohonen network: (a) Delay, (b) Jitter, (c) Packet loss ratio, (d) QoS parameters labels, (e) SOM sample hits.

The analysis of SOM for QoS parameters of video application is shown in Figure 6-13. During its transmission, the video application experienced high delay, jitter, and packet loss ratio as indicated in Figures 6-13 (a) - (c). The network was incapable of meeting the QoS requirements for video application. This was because of the heavy load on the network as multiple applications were being transmitted at the same simulation time interval. Moreover, the video application was allocated to a lower priority access category compared with VoIP allocation.

Although some individual values of delay, jitter, and packet loss ratio were within an acceptable level of quality, high values of one the aforementioned parameters would have a negative effect on the overall quality of video. The labels of QoS parameters and the response of SOM to the QoS parameters of video application shown in Figures 6-13

(d) and (e) respectively provided a differentiation between the generated QoS classes. Both figures indicated that Low QoS parameters identified by label L covered a very small region of SOM. Two neurons were activated by low QoS values and the winning neuron contained 16 values. The region of Medium values of QoS parameters which had label M was large as compared with Low QoS parameters region. The neighbourhood region of high QoS parameters represented by label H covered most of the Kohonen map. Figure 6-13 (e) shows that high QoS parameters had four sub-regions. The number of values activated the winning neurons for these regions were relatively large (i.e. 121, 56, 19, and 8). It can be concluded from Figure 6-13 that high value of QoS parameters of video application: delay, jitter, or packet loss ratio could have negative impact on its overall quality. The analysis of SOM demonstrated how the simulated network could manage and treat different applications with different priorities during light and heavy loads.

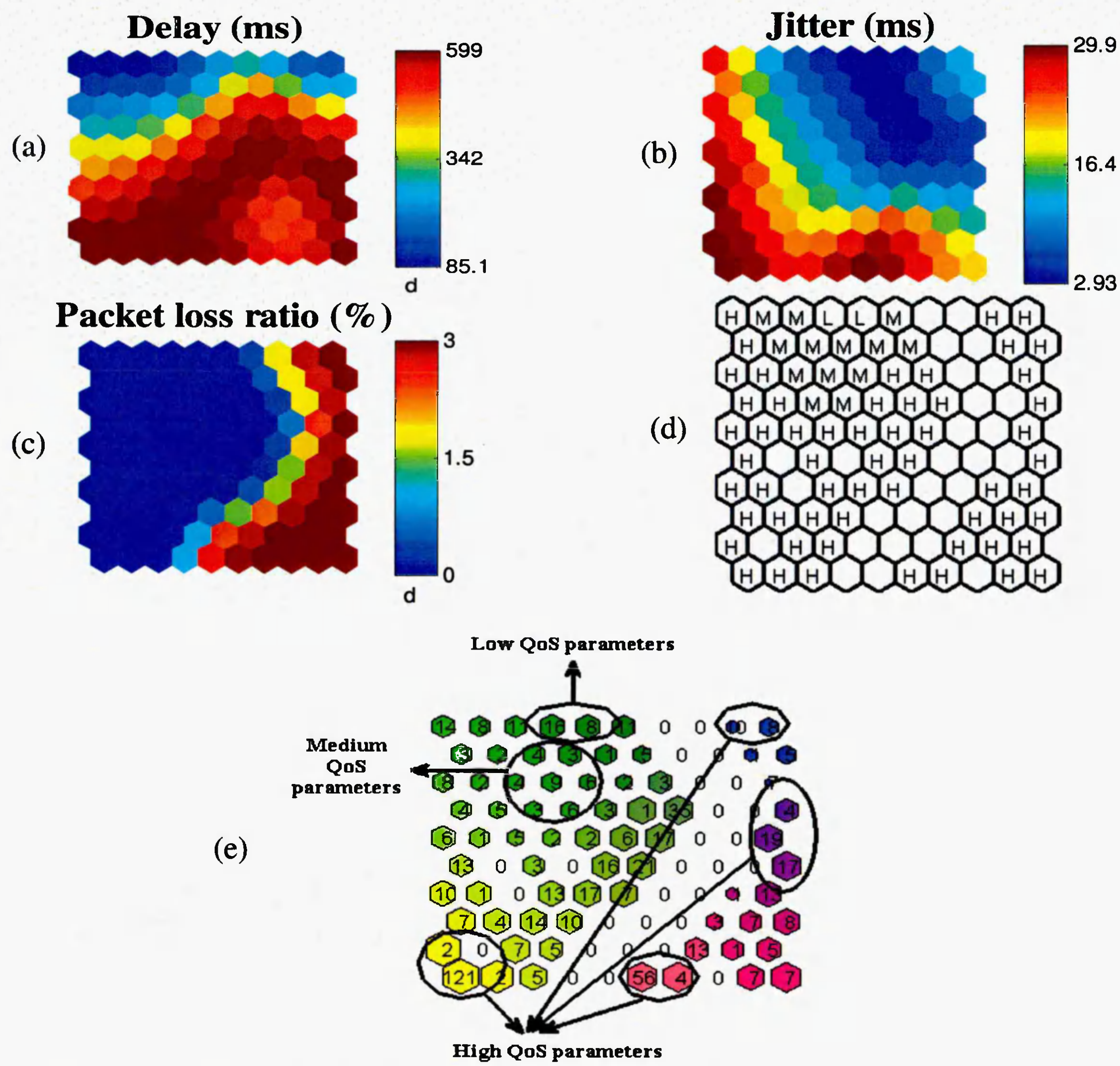
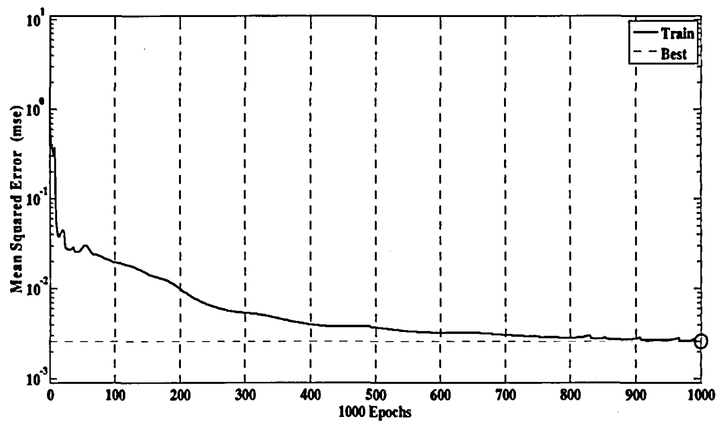


Figure 6-13. Classifying QoS parameters of video application using kohonen network: (a) Delay, (b) Jitter, (c) Packet loss ratio, (d) QoS parameters labels, (e) SOM sample hits.

6.4.4 QoS Assessment using Multi Layer Perceptron

After the QoS parameters of VoIP and video application were processed by Kohonen neural network, the average of QoS parameters (i.e. delay, jitter, and packet loss ratio) activated a winning neuron and a number of neurons in the neighbourhood region as shown in Figures 6-12 and 6-13 were combined by MLP to assess the overall QoS. The MLP was required to be trained correctly in order to assess the QoS in an effective manner. Figures 6-14 (a) and (b) show respectively the training process of the MLP to assess the QoS of VoIP and video application. The training process of MLP was terminated either when the maximum number of iterations was reached or when there was insignificant error (i.e. 0.001) between MLP outputs and desired results. In this study, the training process of MLP in case of QoS assessment of VoIP or video application was terminated when the maximum number of training iterations (i.e. 1000) was reached. The value of maximum number of training iterations was chosen experimentally, i.e. different values were tested and the MLP response was monitored. It is also indicated from the Figures that the Mean Squared Error (MSE) values for MLP when the training process terminated to assess the QoS of VoIP and video were respectively 0.003 and 0.005. These values indicated an insignificant error between MLP outputs and desired results.

(a)



(b)

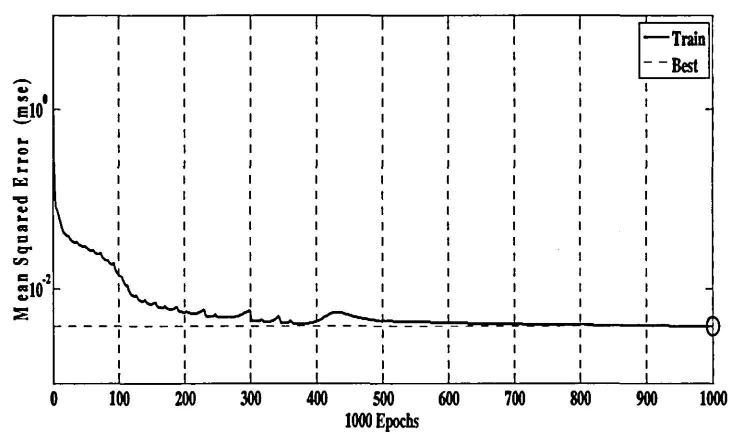
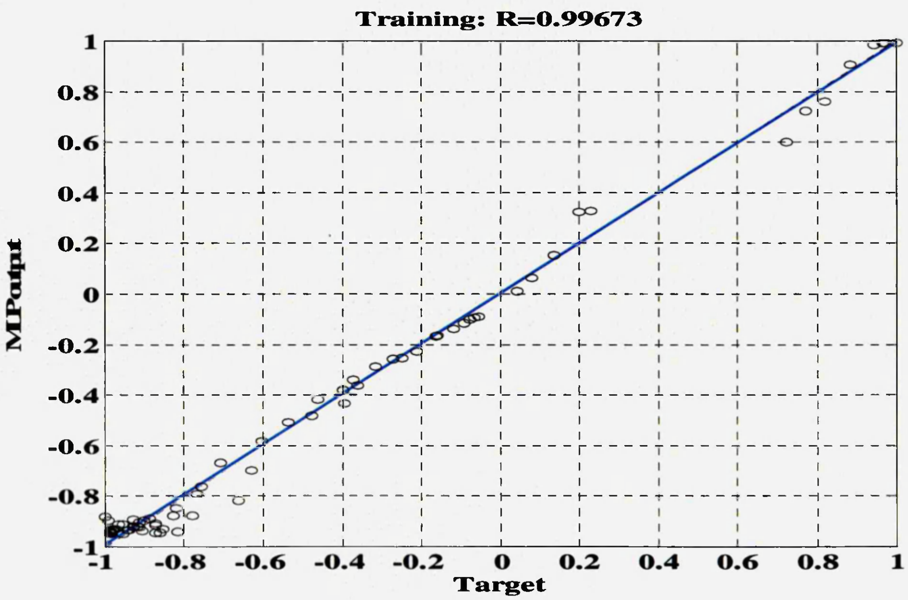
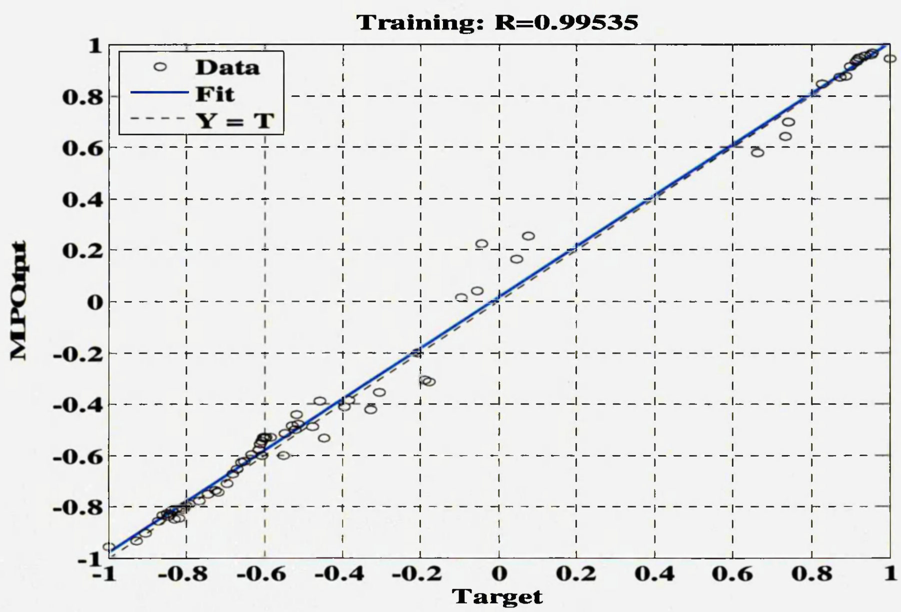


Figure 6-14. The progress of training MLP to assess the QoS for: (a) VoIP, (b) Video.

Figures 6-15 (a) and (b) show respectively the comparison between normalised actual QoS and calculated outputs from MLP during its training to assess QoS of VoIP and video. It can be perceived from the Figures that the actual output values used to train MLP and the output values following the termination of its training were strongly correlated. The correlation coefficients were 0.997 when MLP trained to assess QoS of VoIP and 0.996 when MLP trained to assess QoS of video application. The correlation coefficient values indicated that the MLP had been trained effectively.



(a)



(b)

Figure 6-15. Comparison between normalised actual QoS and calculated outputs from MLP in case of: (a) VoIP QoS assessment, and (b) Video QoS assessment.

The results in Figures 6-16 (a) and (b) show respectively the predicted QoS of VoIP and video applications obtained from MLP during the test phase. From the Figures, it can be observed that the range of QoS values reflected the QoS parameters regions shown in Figures 6-12 (e) and 6-13 (e). In other words, Good QoS reflects the region of Low QoS parameters, whereas High QoS parameters region produced Poor QoS. Due to its priority over video application, the average estimated QoS of VoIP for Low, Medium, and High QoS parameters regions was elevated by 56.01% as compared with the average estimated QoS of video application. However, the heavy load on the network at certain time intervals made its performance incapable of meeting the minimum QoS requirements for VoIP and video applications.

(a)

(b)

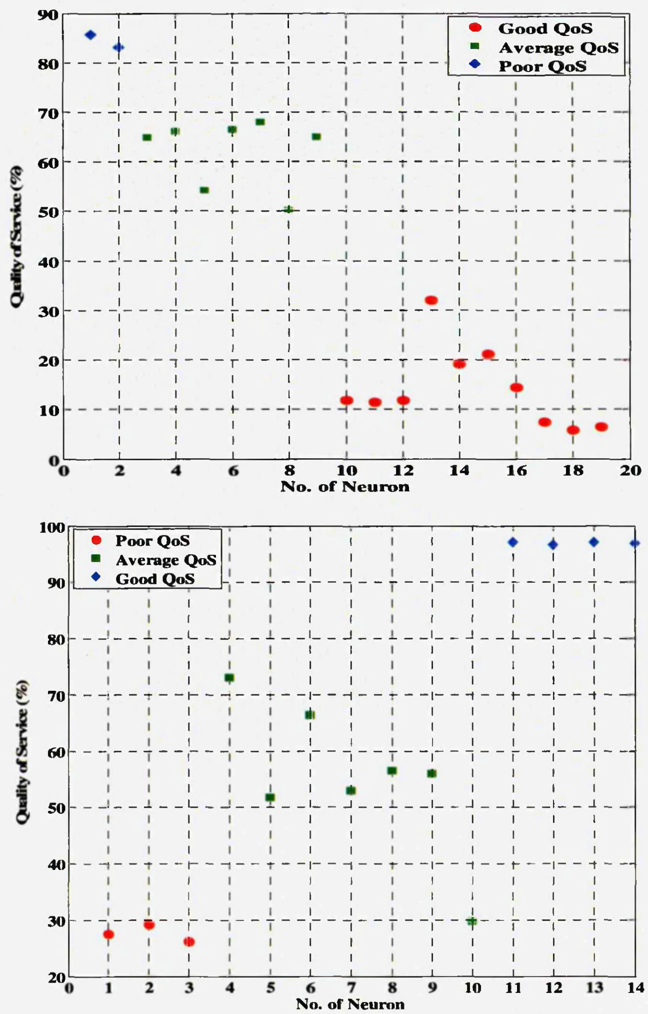


Figure 6-16. The QoS assessment of: (a) VoIP, and (b) Video application using Multi-Layer Perceptron.

The results of expected QoS for VoIP and video obtained using MLP were compared with other QoS assessment methods such as the developed fuzzy inference system FIS (Al-Sbou, $2010_{(a)}$), Euclidean distance measurement system (Al-Sbou, $2006_{(b)}$), and regression model (Dogman et al, 2012)) as shown in Tables 6-3 and 6-4 respectively. The values of correlation coefficients between the QoS of VoIP determined using multi-layer perceptron neural networks, and the other QoS assessment methods were: 0.90 for FIS, 0.99 for distance measure, and 0.98 for the regression model. Whereas correlation coefficients between the QoS of video determined using MLP and other QoS assessment techniques were: 0.97 for FIS, 0.97 for distance measure, and 0.99 for regression model. From the values of correlation coefficient and the values of evaluated QoS provided in Table 6-3 and 6-4, it was concluded that the aforementioned QoS assessment techniques for VoIP and video application provided results which were closely comparable. Although some outputs were slightly different, they were in the same QoS region. The discrepancies were due to the fact that each method followed a different process. However, the values of QoS obtained from the MLP architecture ranged from (1% - 100%), whereas the range of QoS values produced by the FIS was between (10%-90%). This indicates that the MLP was more effective.

Table 6-3. QoS parameters of VoIP and expected QoS using: Fuzzy Inference System (FIS), Distance Measurement (DM), Regression Model (RM), Multi-Layer Perceptron (MLP)

QoS parameters			Assessment of overall QoS using:			
Delay (ms)	Jitter (ms)	Packet Loss ratio (%)	FIS	DM	RM	MLP
55.44	4.59	0.00	9.73	29.98	29.29	27.41
35.36	4.58	0.00	9.76	30.26	30.63	29.16
180.85	5.00	1.25	9.28	23.94	21.90	26.12
35.23	2.26	0.00	81.98	67.77	69.95	72.98
37.96	3.18	0.00	60.30	50.42	54.20	51.68
20.28	2.41	0.00	80.39	65.44	68.27	66.29
71.67	2.99	0.00	67.51	52.66	55.46	52.89
24.24	2.76	0.00	74.18	57.69	62.11	56.44
52.51	2.78	0.00	73.58	56.26	60.13	55.92
77.99	4.06	0.00	16.91	37.29	36.97	29.67
27.64	0.81	0.00	89.33	95.29	94.96	97.05
18.77	1.70	0.00	82.21	94.94	80.39	96.54
21.63	0.24	0.25	90.31	96.11	98.53	97.02
24.57	1.56	0.00	83.64	94.95	82.43	96.76

Table 6-4. QoS parameters of video and expected QoS using: Fuzzy Inference System (FIS), Distance Measurement (DM), Regression Model (RM), Multi-Layer Perceptron (MLP)

QoS parameters			Assessment of overall QoS using:			
Delay (ms)	Jitter (ms)	Packet loss ratio (%)	FIS	DM	RM	MLP
55.37	4.92	0.00	88.41	93.82	81.95	85.58
142.73	4.02	0.00	87.94	93.64	73.94	83.10
264.02	12.10	0.00	54.99	90.51	57.03	64.93
320.64	14.77	0.00	48.92	83.58	49.82	54.23
249.58	11.04	0.00	57.66	91.14	59.10	68.02
402.18	10.48	0.00	42.80	46.27	44.48	50.17
304.88	8.28	0.00	62.54	87.78	55.39	65.02
600.00	29.04	0.00	9.32	10.01	13.56	11.79
336.61	14.00	3.00	10.47	21.77	27.37	31.97
380.12	16.05	3.00	10.47	20.27	21.83	19.13
516.10	5.56	3.00	10.22	13.98	14.99	21.14
595.00	2.49	3.00	9.41	8.57	9.16	14.39
600.00	7.79	3.00	9.39	7.75	5.38	7.34
599.61	29.66	3.00	9.29	0.51	8.14	5.70

6.5 Summary

This chapter introduced two novel Quality of Service (QoS) assessment systems. The first system included a combination of fuzzy C-means (FCM) and regression model to analyse and measure the QoS of VoIP and video traffic transmitted over a simulated network. Whereas the other system used a combination of supervised and unsupervised neural networks (i.e. Kohonen network and multi-layer perceptron (MLP)) to evaluate the QoS of VoIP and video applications.

The QoS parameters of VoIP and video traffic were analysed by FCM and Kohonen network. The capability and robustness of these techniques to cope with imprecise QoS patterns made them effective clustering mechanisms for QoS analysis. FCM and Kohonen network classified the values of QoS parameters of transmitted VoIP and video into clusters representing Low, Medium, and High values of QoS. The regression model and MLP in turn combined the QoS parameters (i.e. delay, jitter, and packet loss ratio) for each centre of generated clusters and produced a single value that represented the overall QoS. The overall QoS was a good indication of network performance. The overall QoS can be used to monitor the network and to avoid congestion. The values of assessed QoS were strongly correlated to a number of previously studied QoS assessment methods.

Chapter 7 Improvements in Quality of Service in Computer Networks

7.1 Introduction

In this chapter, a new Quality of Service (QoS) enhancement scheme for WLAN-wired networks is developed and its performance is evaluated. The proposed scheme consists of an adaptive Access Category (AC) traffic allocation algorithm that is incorporated into the network's wireless side to improve the performance of IEEE 802.11e Enhanced Distributed Channel Access (EDCA) protocol, and a Weighted Round Robin (WRR) queuing scheduling mechanism that is incorporated into the wired side of the network. The adaptive traffic allocation algorithm determines the Packet Arrival Rate (PAR) of up-link and down-link traffic for each AC. It then dynamically allocates the traffic of the lower priority AC to the next higher AC, when the higher AC is not receiving traffic at the time. On the wired side of the network, the aim of WRR is to share the network resources based on the traffic's quality of service (QoS) requirements. The performance of the proposed scheme was compared with the standard IEEE 802.11e EDCA and FIFO queuing mechanisms (i.e. WLAN-wired network legacy scheme). The incorporation of the scheme improved the performance of the WLAN-wired network, thus enhancing the QoS for transmitted applications. The scheme allowed an end-to-end QoS to be set up which in turn provided improved delivery of a variety of applications in the context of wired-cum-wireless networks.

This chapter is organised as follows: the relevant studies are discussed in section 7.2. Section 7.3 introduces a detailed description of the proposed QoS enhancement scheme. The simulated network and traffic models are presented in section 7.4. The results are presented and discussed in section 7.5. The conclusions are provided in section 7.6.

7.2 Related Work

The growth in the transmission of multimedia applications with different transmission time sensitivities has raised the challenge of facilitating their QoS. Time-sensitive applications such as videoconferencing and Voice over IP (VoIP) are susceptible to communication parameters such as packet delay, jitter and throughput. On the other hand, some time-insensitive applications, such as File Transfer, are more affected by packet loss and reliability (this includes bits error rate), but are unaffected by jitter (Kurose and Ross, 2005). The interconnection between wired and wireless networks also requires that QoS of traffic being exchanged to be appropriately realised.

Most previous studies explored QoS support either in wireless local area networks (WLANs) or in wired networks as discussed in section 3.4 of chapter 3. However, there were only few studies to enable end-to-end QoS for wired and wireless networks. For instance, Skyrianoglou et al. (2002) proposed a Wireless Adaption Layer (WAL) to provide an integrated QoS between WLAN and IP infrastructure. The WAL was located between the MAC layer and the IP layer. The function of WAL was to provide service differentiation to the IP traffic transmitted between WLAN and a fixed IP network to improve the performance in a wireless IP networks. An architecture to map IP layer Differentiated Services Code Point (DSCP) QoS to MAC layer EDCA Access categories (ACs) was proposed by Park et al. (2003). The integrated scheme used DSCP of the IP packets to allocate packets to an appropriate AC. Another mapping scheme of Enhanced Distributed Channel Access (EDCA) access categories to IP traffic class in wired network was proposed by Senkindu and Chan (2008). Their mapping aim was to ensure that end-to-end service guarantees were provided for multimedia applications.

However, most previous studies either required modifications of wireless stations, which in turn complicated the WLAN operation, or supported QoS of high priority traffic which in turn starved other transmitted traffic. The limitation of WAL proposed by Skyrianoglou et al. (2002) is that an intermediate layer was introduced between the MAC and IP layers in wireless stations. This in turn resulted in more complexity to WLAN management. Also, the modification of wireless stations to support DiffServ functionality was a disadvantage of the method introduced by Park et al. (2003).

The scheme by Senkindu and Chan in (2008) was simple to implement, but the low priority traffic's packet drop rate was high due to link congestions and QoS prioritisation. Therefore, the challenge in end-to-end QoS is that both wired and wireless parts of the network provide suitable treatment for each class of traffic and to efficiently use network resources.

The main contribution of this study is the development of a new QoS enhancement scheme that improves delivery of a variety of applications in both wired and wireless sides of the network. The scheme provides an integrated MAC layer QoS for the wireless side of the network and network layer QoS controls in the wired side. A novel aspect of the approach is that an adaptive AC traffic allocation algorithm was devised and incorporated into wireless access point (AP) in order to improve QoS in the IEEE 802.11e EDCA protocol. Also, Weighted Round Robin (WRR) queuing scheduling mechanism was implemented into the congestion point (i.e. router) in wired networks to support a fair distribution of bandwidth among different traffic types.

7.3 Description of QoS Enhancement Approach

The aim was to introduce an enhanced integrated QoS in the wired and wireless sides of the network. The approach consists of: (i) an adaptive traffic allocation algorithm at the wireless side of the Access Point (AP), and (ii) a Weighted Round Robin (WRR) queuing scheduling mechanism implemented at the congestion points of the network's wired side. The proposed QoS enhancement scheme was designed to allow an end-to-end QoS to be set up. This would then provide an effective delivery of a variety of applications in the context of wired-cum-wireless networks. The next subsections explain the mechanism of the proposed adaptive traffic allocation algorithm and the manner in which WRR queue scheduling was adapted to enhance QoS.

7.3.1 Adaptive Traffic Allocation Algorithm

The adaptive traffic allocation algorithm was located in the access point in the wireless side of the network. The algorithm is outlined in the flow chart shown in Figure 7-1.

The algorithm monitors the Packet Arrival Rate (PAR) of the uplink and downlink traffic for each AC, in order to allocate traffic to an appropriate AC. The PAR parameter was chosen because it signify how well real-time and non-real-time applications are delivered. Also, it can be easily computed in real-time. This in turn facilitates quick allocation of the arrived traffic to the most appropriate AC.

PAR is defined as the number of successfully received packets in a given time interval (Wang et al., 2000). Equation (7.1) was used to calculate PAR:

$$PAR_i\,(t) = \frac{\sum P_i(t)}{t_i} \tag{7.1}$$

where, $PAR_i\,(t)$ is packet arrival rate during i^{th} time interval. $\sum P_i(t)$ is the total number of packet received during the i^{th} interval, and t_i is the time duration of the i^{th} interval.

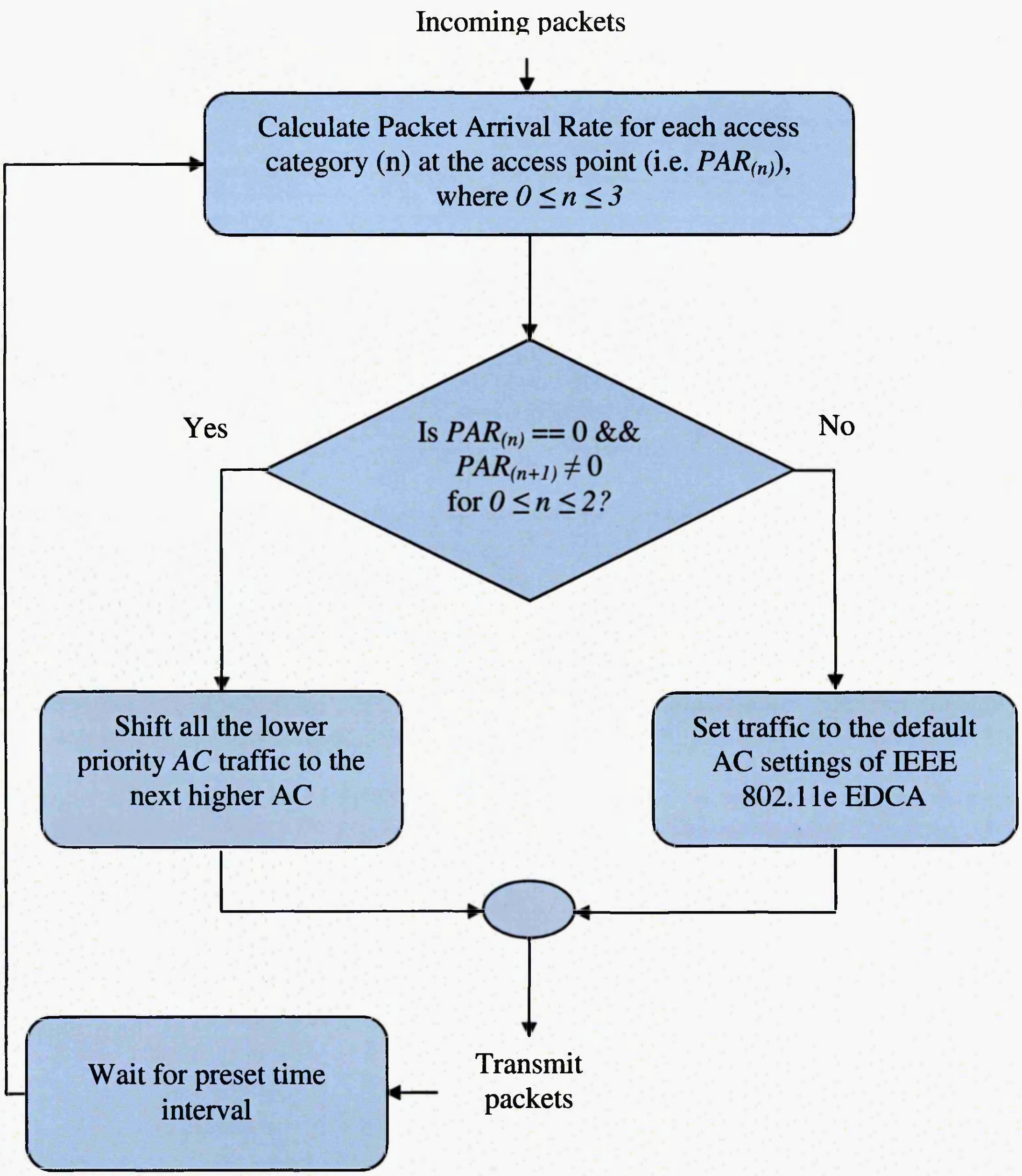


Figure 7-1. Adaptive traffic allocation algorithm flow chart.

During transmission, the algorithm calculates at each time interval, the PAR for each AC (i.e., AC_0 to AC_3). AC_0 receives the highest priority traffic, while AC_3 is for the lowest. When the PAR at a higher AC is zero, the algorithm allocates the incoming traffic to a lower priority AC to the next higher priority AC. This operation increases the lower priority traffic's ability to access the channel and thus it transmits its packets at a faster rate. This also ensures the network resources do not remain idle.

The algorithm waits for a pre-set time interval and re-calculates the PAR associated with traffic for each AC. If there is an application for transmission at a higher priority AC, the lower priority traffic previously diverted to the higher AC, would be moved back to its original default AC. However, if higher priority traffic started its transmission during the pre-set time interval, it would be directed to its original AC and the lower priority traffic would be moved back to its original AC after the current interval is terminated. Therefore, the pre-set time interval must be chosen carefully to accommodate the process. This is because in case of a large time interval, the higher priority traffic would be negatively affected, whereas in case of short time interval, the high computational load would be experienced (that is an issue for real-time operations).

7.3.1 Integration of Weighted Round Robin Queuing Mechanism

In the wired side of the network, WRR scheduling mechanism was integrated between the router and the AP in order for the transmitted traffic to share the network resources, based on QoS requirements. The WRR takes the incoming packets from a number of traffic types and schedules them according to their allocated respective weights. The allocation of weights considers the traffic priority, application's packet size, and its transmission rate. In this study, four types of traffic were considered: VoIP and video as time-sensitive applications and best effort traffic and background traffic as time-insensitive applications. Time-sensitive applications were high priority and were assigned 60% of the bandwidth, whereas best effort and background traffic were treated as low priority and were allocated 40% of the bandwidth.

The respective weights for the aforementioned applications are shown in Table 7-1. However, when one type of application was not transmitted at a particular time interval, WRR divided the bandwidth allocated to that application to other applications being transmitted according to their respective weights (Semeria, 2001).

Table 7-1. WRR queue weights.

WRR \ Application	VoIP	Video	best effort traffic	background traffic
Weights	3	3	2	2

7.1 Network Modelling and Simulation

To validate the performance of the QoS enhancement scheme, wireless-cum-wired network topologies with different network sizes (small, medium, and large) were simulated using the Network Simulator- 2 (NS2). A small network was simulated using 8 unidirectional connections. Networks with a medium size were simulated using 16 and 24 unidirectional connections. A large network was simulated with 32 unidirectional connections. In all networks, half of the connections, transmitted traffic from wireless to wired network, whereas the other half transmitted traffic from wired to wireless. A network topology that illustrates wireless-cum-wired network is shown in Figure 4-3 (See chapter 4).

The traffic transmitted over the simulated networks was: VoIP, video streaming, best effort traffic, and background traffic. Constant Bit Rate (CBR) traffic was adapted to model VoIP. VoIP was modelled with the G.711 voice encoding scheme. The video streaming source was YUV QCIF (176 × 144) Foreman with 400 frames (YUV QCIF, 2012). Prior to transmission, each video frame was fragmented into packets that in turn had a maximum length of 1024 bytes. The best-effort traffic was modelled using CBR with different packet sizes and generation rates that corresponded to non- VoIP usage. File Transfer Protocol (FTP) application was used for the background traffic. FTP was transmitted over TCP (to ensure reliability), whereas other traffics were transmitted using UDP (to ensure high transmission rate) transport protocol. Table 7-1 shows the traffic characteristics of the aforementioned applications.

Table 7-2. Traffic characteristics.

Parameters \ Traffic Type	VoIP	Video	Best effort	Back-ground traffic
Packet size (Byte)	160	~692	1000	1000
Traffic type	CBR	VBR	CBR	FTP
Transmission rate (kbps)	64	-	125	-

Two different scenarios were considered for each simulated network. The first scenario excluded the presence of QoS enhancement scheme. The WLAN was based on IEEE 802.11e EDCA, The main parameters that modelled the wireless channel were the default settings for IEEE 802.11e. These parameters are shown in Table 4-1. The transmitted traffic (i.e. VoIP, video streaming, best effort traffic, and background traffic) were mapped to the ACs to represent different levels of priority as shown in Table 4-2. VoIP had highest priority, while the priority of the background traffic was lowest. At the wired side of the network, First-in-First out (FIFO) queue scheduling mechanism (queue size=50) was implemented between the router and the access point. FIFO was chosen due to its simplicity that facilitates real-time operations.

However, in the second scenario where the QoS enhancement scheme was included, the adaptive traffic allocation algorithm was operated at the wireless side of the AP. The main parameter of adaptive allocation algorithm was the pre-set time interval which was set up to be 2.5s. This value was chosen experimentally to provide best results. In the wired side of the network, the queue scheduling mechanism implemented between the router and the access point was Weighted Round Robin (WRR). WRR classified traffic based on their QoS requirements. Time-sensitive applications had a higher priority than time-insensitive applications. The main parameters of WRR are shown in Table 7-3.

Table 7-3. WRR Parameters.

WRR queue No. / Parameters	1	2	3	4
Application type	VoIP	Video	Best effort	Background traffic
WRR weights	3	3	2	2
Queue length	25	25	25	25
Allocated bandwidth	0.6 Mbps	0.6 Mbps	0.4 Mbps	0.4 Mbps

The network topology covered an area of 500m × 500m and the stations were positioned randomly within this area. This setup remained unchanged during the simulations. Simulations were repeated 10 times. Each time a different initial seed value was used to randomly position the stations and to control which node transmitted first. The randomness introduced using the different seeds avoided the bias of random number generation. The results of the 10 simulations were then averaged. Simulation duration was 300 seconds. These periods were considered sufficient to examine the behaviour of IEEE 802.11e protocol.

7.2 Results and Discussion

This section is to demonstrate the effectiveness of the proposed QoS enhancement scheme. The results obtained using the proposed QoS enhancement scheme (i.e. adaptive allocation algorithm and WRR queuing scheduling mechanisms) were compared with the results obtained using the legacy scheme (i.e. standard IEEE 802.11e EDCA and FIFO queuing mechanisms). The following subsections respectively explain the comparison of delay, jitter, packet loss ratio, and the overall assessed QoS for the transmitted traffic with and without the proposed QoS enhancement scheme.

7.2.1 Delay

In this study, the measurement of delay includes various types of delays such as queuing delay, transmission delay, processing delay, and propagation delay. Figure 7-2 (a) - (d) show respectively the average values of delay for VoIP, video streaming, best effort traffic, and background traffic under different network loads. It is indicated from the Figures that the average values of delay using legacy and QoS enhancement scheme were increased as the number of connections increased. This is because increased probability of packets collisions, which in turn caused the nodes to retransmit the collided packets. In this case, the delay between the consecutive packets would be further increased.

In addition, Figures 7-2 (a) - (d) indicate that both schemes treated the transmitted traffic based on their QoS requirements. VoIP gained the lowest delay because it was assigned by default to the highest priority access category (i.e. AC_0), whereas the background traffic experienced the highest delay among other traffics as it was assigned by default to the lowest priority access category (i.e. AC_3).

The VoIP that was assigned to AC_0 had a small value for AIFS (i.e. equivalent to 2µs) allowing AC_0 to start its backoff procedure after detecting the channel was idle for an AIFS period. Also, small values for CW_{min}, and CW_{max} which were respectively 7 slots, and 15 slots for AC_0, made VoIP traffic to have a small waiting duration before accessing the channel. In addition, large value of TXOP (i.e. equivalent to 3.008 ms) allowed AC_0 to transmit multiple data frames continuously during that particular time interval defined by TXOP. The values of aforementioned EDCA parameters gave VoIP traffic the highest priority to access the channel. Therefore, the expected packet delay would be lower than other transmitted traffic.

In contrast, the background traffic that was assigned to AC_3 experienced highest delay among other transmitted traffics. For example, the delay values of the background traffic at medium network loads (i.e. 16 connections) using legacy and QoS enhancement scheme were receptively 186.6 ms, and 130.5 ms. These values were much higher as compared with delay values of other traffic under the same network conditions. The delay values of video streaming for instance when using legacy and QoS enhancement scheme were respectively 17.1 ms, and 48.5 ms. This is because of the values of EDCA parameters assigned to AC_3. Large values of AIFS, CW_{min}, and CW_{max} which were respectively 7 μs, 31 slots, and 1023 slots made the traffic assigned to AC_3 to wait for long period of time before accessing the channel and then to start its backoff procedure after detecting the channel was idle for an AIFS period.

However, Figures 7-2 (a) - (d) also indicate that the traffic transmitted using the proposed QoS enhancement scheme experienced a lower delay for all ACs as compared with the legacy network scheme. The average delay for video streaming, best effort traffic, and background traffic were decreased by 57.6%, 63.6%, and 31.5% respectively when the QoS enhancement scheme was introduced. The reductions in delay values were due to the allocation of lower priority traffic to the next higher AC in cases of Packet Arrival Rate (PAR) for the higher priority traffic was zero. For instance, the packets of video streaming were transmitted by default using AC_1 which its EDCA parameters value were 2 μs, 15 slots, and 31 slots for AIFS, CW_{min}, and CW_{max} respectively. However, when QoS enhancement scheme was applied, the packets of video streaming were transmitted for particular time interval using AC_0 when VoIP traffic did not transmit at that particular time interval. The allocation of video streaming traffic's to AC_0 (which its EDCA parameters value were 2 μs, 7 slots, and 15 slots for AIFS, CW_{min}, and CW_{max} respectively) increased its ability to access the channel and thus reduced the average value of delay.

Also, when QoS enhancement scheme was applied, the fairness in treating traffic with different QoS requirements introduced by WRR at the wired side of the network had been a factor in improving network performance. For example, the reduction of 66.2% in delay for VoIP was due to the implementation of WRR, as the adaptive allocation algorithm did not shift the traffic transmitted by the highest priority AC.

(a)

(b)

(c)

(d)

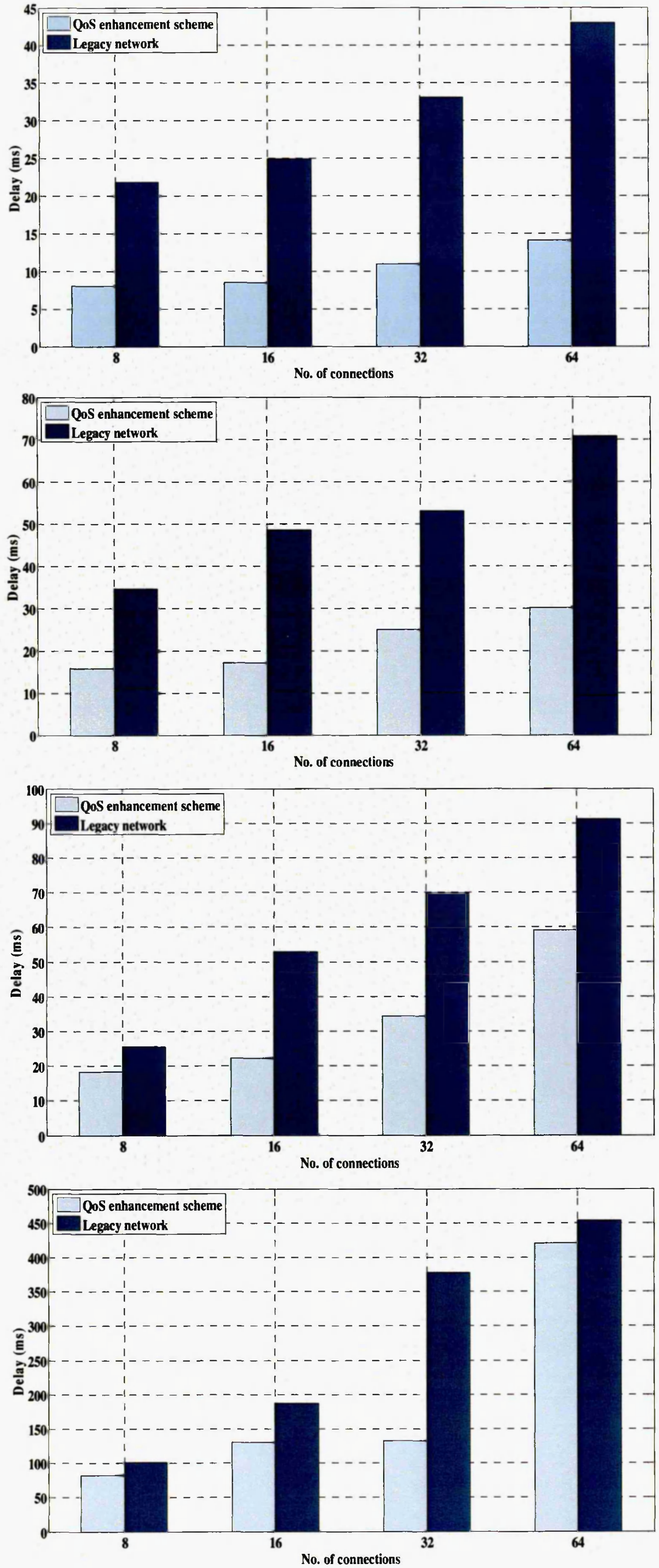


Figure 7-2. Bar chart representation of average delay with and without QoS enhancement scheme for: (a) VoIP, (b) Video, (c) Best effort traffic, (d) Background traffic.

7.2.2 Jitter

The average values of jitter for VoIP, video streaming, best effort traffic, and background traffic under different network loads are shown respectively in Figures 7-3 (a) - (d). Each Figure provides a comparison between the jitter's values obtained using the proposed QoS enhancement scheme and the legacy scheme under a light network load (i.e. 8 connections), a medium network load (i.e. 16-24 connections), and a heavy network load (i.e. 32 connections) respectively.

It is shown from the Figures that the values of jitter for all transmitted traffic gradually increased as the network became more congested for both legacy and QoS enhancement schemes. For example, the values of jitter for VoIP traffic under a light network load were respectively 1.7 5ms, and 3.3 ms using QoS enhancement scheme and the legacy scheme. Whereas, the jitter's values for VoIP traffic using QoS enhancement scheme and the legacy scheme under a heavy network load were respectively 4.9 ms, and 5.7 ms.

In other words, as the number of active stations increased; the probability of collisions increased accordingly due to increased competition between the stations. This forced the MAC protocol to retransmit the collided packets. When the collided packets were successfully received; the time between any two consecutive packets that were successfully received at the destinations were increased, and subsequently increasing the jitter.

However, Figures 7-3 (a) - (d), also indicates that the implementation of QoS enhancement scheme reduced the jitter values of all transmitted traffic as compared when the legacy scheme was applied. The average jitter for VoIP, video streaming, best effort traffic, and background traffic were reduced by 36.3%, 36%, 20.5%, and 35.1% respectively.

The reduction of jitter for traffic transmitting by lower priority access category (i.e. AC_1 - AC_3) was due to the adaptive allocation algorithm at the wireless network, and WRR implemented at wired side of the network.

The jitter's reduction of best effort traffic for instance was due to the dynamic allocation for CBR packets from the lower priority AC (i.e. AC_2) to the next higher AC (i.e. AC_1) when the AC_1 was not transmitting video traffic at that time. In other words, best effort traffic were transmitted by default using AC_2 which its EDCA parameters were 3 μs, 31 slots, and 1023 slots for AIFS, CW_{min}, and CW_{max} respectively. However, when QoS enhancement scheme was applied, the CBR packets were transmitted for a particular time interval using AC_1 when video traffic did not transmit at that time interval. The allocation of CBR traffic to AC_1 which its EDCA parameters were 2 μs, 15 slots, and 31 slots for AIFS, CW_{min}, and CW_{max} respectively increased its ability to access the channel and thus reduced the average jitter.

Also, the fairness of WRR to share network resources among transmitted traffic based on their QoS requirements decreased the values of jitter. WRR allocated 20% of the bandwidth to CBR traffic making the packet collision to decrease accordingly. This is unlike the FIFO mechanism in legacy scheme where high and low priority traffic were treated in the same manner. The arrived packets dropped regardless of their priorities when the queue of FIFO became full. As a result, the implementation of QoS enhancement scheme reduced the possibility of CBR packets to collide and subsequently reduced the variations in delays between any two consecutive CBR packets.

The jitter for VoIP traffic which was reduced by 36.3%, when QoS enhancement scheme was implemented was due to the implementation of WRR only. This is because the adaptive allocation algorithm does not shift the traffic transmitted by default using the highest priority AC (i.e. AC_0).

As the highest priority traffic, 30% of the bandwidth was allocated to VoIP when WRR was implemented. This is unlike FIFO mechanism in legacy scheme which made VoIP as high priority traffic to be treated in the same manner as low priority traffic. Subsequently, the arrived packets were dropped regardless of their priorities when the buffer at the router became full.

The implementation of WRR in QoS enhancement scheme reduced the possibility of competition between VoIP and other traffic to be transmitted throughout the router. This reduced packets drop rate and subsequently reduced the variations in delays between any two consecutive VoIP packets.

(a)

(b)

(c)

(d)

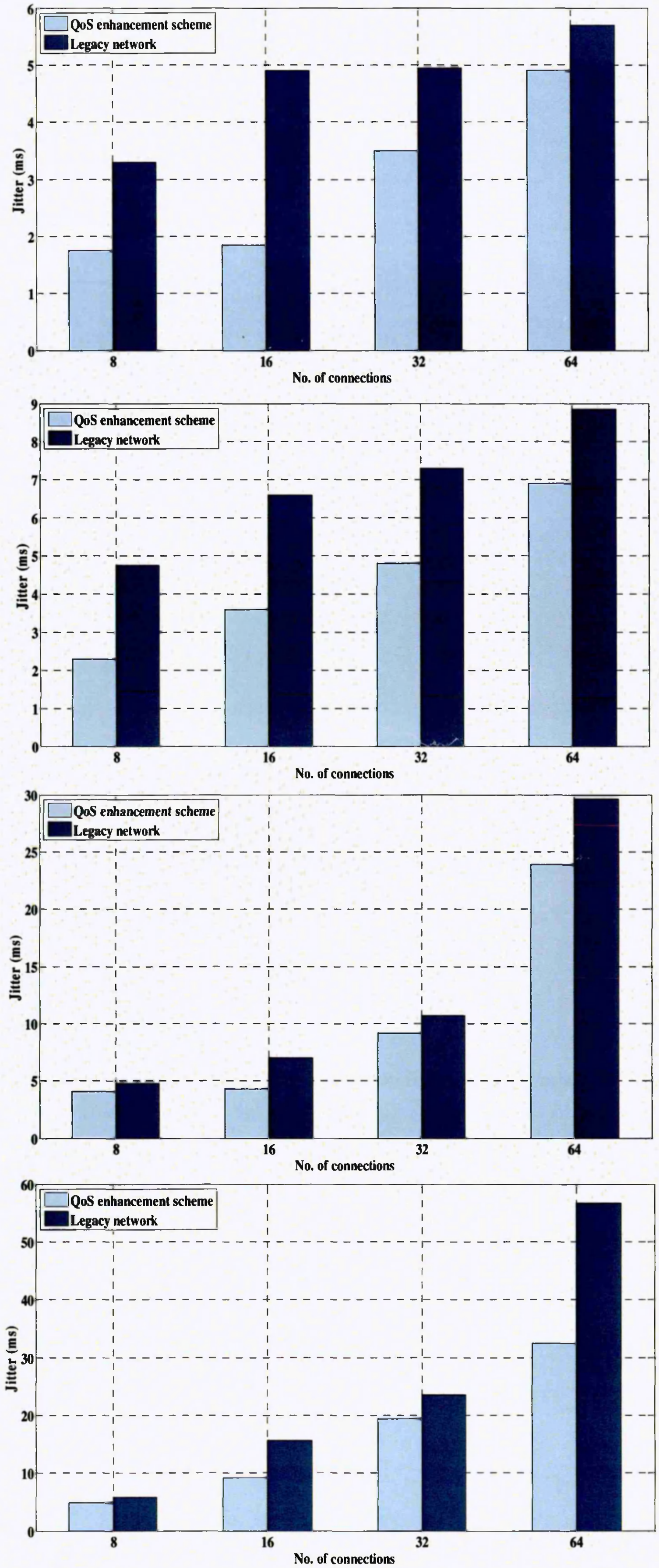


Figure 7-3. Bar chart representation of average jitter with and without QoS enhancement scheme for: (a) VoIP, (b) Video, (c) Best effort traffic, (d) Background traffic.

7.2.3 Packet loss ratio

Figures 7-4 (a) - (d) depict comparison of packet loss ratio for VoIP, video streaming, best effort traffic, and background traffic obtained using QoS enhancement scheme and the legacy scheme.

The results indicate, the increase in packet loss ratio was based on the number of connections, and the priority of the traffic for both schemes. For example, packet loss ratio for video streaming increased as the number of connections increased as shown in Figure 7-4 (b). Packet loss ratios for video streaming under a light load (i.e. 8 connections) were respectively 0.001%, and 0.1% for QoS enhancement scheme and the legacy scheme. Whereas, packet loss ratios for video traffic using QoS enhancement scheme and the legacy scheme under a heavy network load (i.e. 32 connections) were respectively 1.9%, and 3.5%. This was caused by a high degree of competition between the transmitting stations, which in turn increased the probability of packets colliding.

The priority of the traffic also influenced packet loss. For example, Figure 7-4 (a) shows the packet loss ratios for VoIP under a medium network load (i.e. 32 connections) which were respectively 0.1%, and 0.25% using the QoS enhancement scheme and the legacy scheme. Whereas, the packet loss ratios for the video application under the same load were respectively 0.51%, and 1.71% using the QoS enhancement scheme and the legacy scheme as shown in Figure 7-4 (b). This is because; VoIP traffic was assigned to the highest priority AC (i.e. AC_0) with small values of AIFS, CWmin, and CWmax. Whereas video traffic was assigned to the lower priority AC (i.e. AC_1) with large values of AIFS, CWmin, and CWmax as compared with AC_0. Small values of AIFS, CWmin, and CWmax increased the traffic's ability to access the channel and thus the AC transmitted its packets faster.

Figure 7-4 (d) shows that the packet drop rate for FTP which was transmitted using the lowest priority AC (i.e. AC_3) was lower than the packet loss ratio for CBR shown in Figure 7-4 (c). For example, the packet loss ratios for FTP under a medium load (i.e. 16 connections) were respectively 0.21%, and 1.68% using QoS enhancement scheme and the legacy scheme. Whereas, the packet loss ratios for CBR traffic under the same network load were respectively 0.7%, and 2.05% using QoS enhancement scheme and the legacy scheme.

Although the CBR traffic was located at a higher priority AC (i.e. AC_2), FTP was transmitted using Transmission Control Protocol (TCP) whereas CBR traffic transmitted using User Datagram Protocol (UDP). TCP as a connection-oriented protocol provides a reliable, ordered, and error-checked delivery of packets.

Figures 7-4 (a) - (d) also indicate the proposed QoS enhancement scheme reduced packet loss ratio for all ACs as compared with legacy scheme. The overall packet loss ratios for the VoIP, Video streaming, CBR traffic, and FTP application decreased by 72.2%, 68%, 26.1%, and 81.9% respectively.

The reductions in packet loss ratios were due to two factors: the adaptive traffic allocation algorithm at the wireless side of the AP, and the WRR queuing scheduling mechanism which was implemented between the AP and the router at the wired side of the network.

The former factor affected the video, CBR, and FTP traffic as it shifted traffic from a lower AC to the next higher AC, whereas the latter factor reduced the packet loss for all transmitted traffic due to its fairness distribution of network resources. As an example, the packet loss ratio for video streaming under a heavy network load (i.e. 32 connections) was 3.5% using the legacy scheme, whereas its packet loss ratio under the same condition using QoS enhancement scheme was 1.9%. This reduction was due to the dynamic allocation for the video packets from a lower priority AC (i.e. AC_1) to the highest AC (i.e. AC_0) when the AC_0 was not transmitting VoIP traffic at that time. The values of EDCA parameters for AC_1 were 2 μs, 15 slots, and 31 slots for AIFS, CW_{min}, and CW_{max} respectively. Whereas, the values of AIFS, CW_{min}, and CW_{max} for AC_0 were 2μs, 7 slots, and 15 slots respectively. This operation increased video traffic's ability to access the wireless channel and transmit its packets faster.

The 20% of the bandwidth allocated by WRR to video traffic made the probability of packet collision to be decreased. WRR implemented in QoS enhancement scheme treated traffic based on their QoS requirements. This is unlike FIFO mechanism in the legacy scheme where high and low priority traffic were treated in the same manner.

(a)

(b)

(c)

(d)

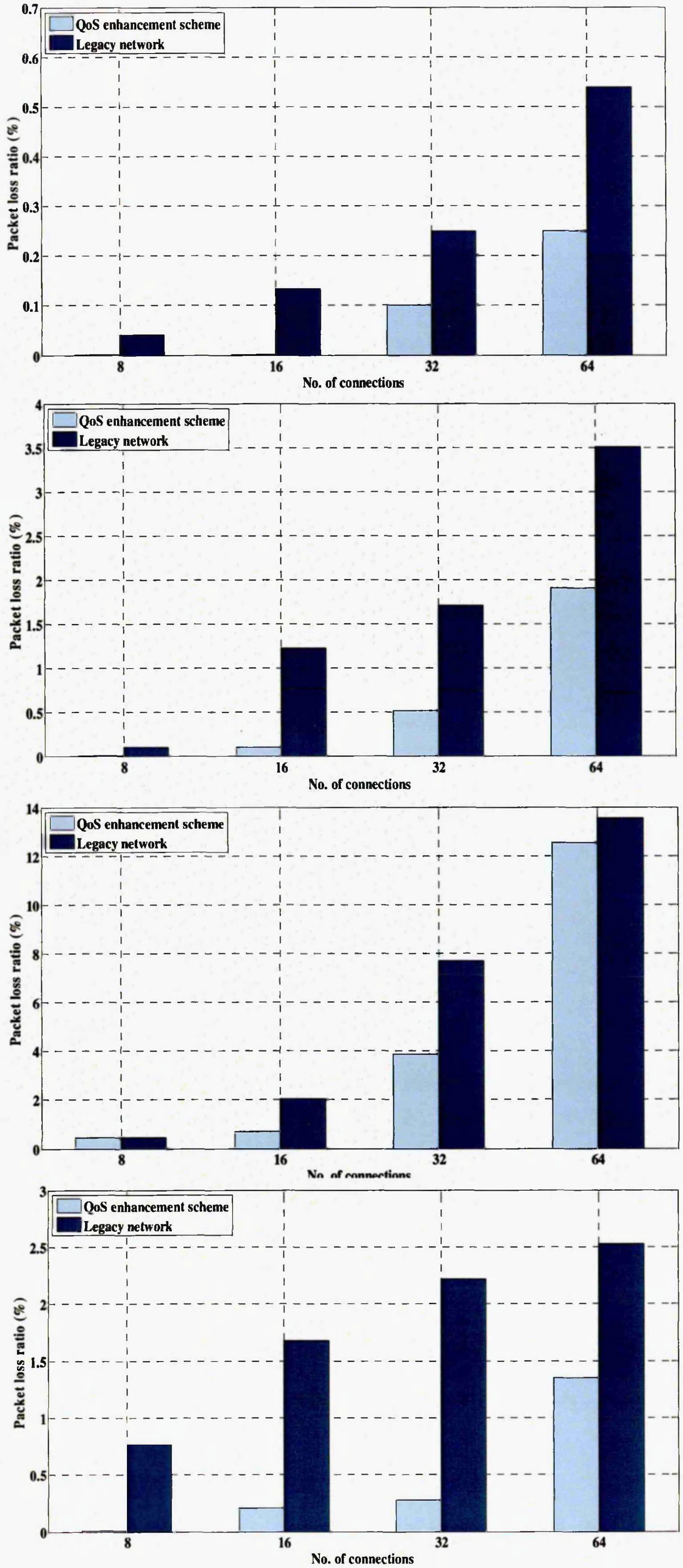


Figure 7-4. Bar chart representation of average packet loss with and without QoS enhancement scheme for: (a) VoIP, (b) Video, (c) Best effort traffic, (d) Background traffic.

7.2.4 Overall QoS

Figures 7-5 (a) - (d) show the overall QoS for VoIP, video streaming, best effort traffic, and background traffic respectively when legacy and QoS enhancement schemes were implemented.

The overall QoS was assessed using a supervised learning Multi-Layer Perceptron (MLP) neural network (Dogman et al, 2012). In this technique, the QoS parameters (i.e. delay, jitter, and packet loss ratio) for the transmitted applications were used as inputs to the trained MLP. The MLP then quantified the overall QoS based on the values of delay, jitter, and packet loss ratio taking into the account the QoS requirements for each application. The overall QoS spanned between (0%-100%). A poor QoS did not exceed 33%, whereas a Good QoS could not be below 67%. More details about QoS assessment using MLP can be found in section 6.3.4 (see chapter 6).

From Figures 7-5 (a) - (d), the overall QoS for all transmitted traffic were decreased as long as the number of connections in the network was increased for both the legacy and QoS enhancement schemes. For example, the values of overall QoS for the video streaming under a light network load (i.e. 8 connections) were respectively 82.18%, and 42.46% using the QoS enhancement scheme and the legacy scheme. Whereas, the overall QoS values for the video application using QoS enhancement scheme and the legacy scheme under a heavy network load (i.e. 64 connections) were respectively 22.1%, and 15.01%. This was due to the gradual increase of delay, jitter, and packet loss ratio for all transmitted traffic when the load on the network was increased. In both schemes, as the number of active stations increased; the probability of collisions increased due to a high degree of competition between stations. This is in turn forced the MAC protocol to retransmit the collided packets, and subsequently increased the values of delay, jitter, and packet loss.

Also indicated from Figures 7-5 (a) - (d), is that the priority of transmitted applications could affect their overall QoS. For instance, the values of overall QoS for the VoIP application under a medium network load (i.e. 32 connections) were respectively 75.46%, and 46.62% using the QoS enhancement scheme and the legacy scheme. Whereas, the overall QoS values for CBR traffic using the QoS enhancement scheme and the legacy scheme under the same network load were respectively 35.86%, and 21.55%. This is because VoIP was transmitted using the highest priority AC (i.e. AC_0) while the best effort traffic was transmitted by default using the lower priority AC (i.e.

AC_2). The values of EDCA parameters for AC_0 facilitated its traffic to access the channel and transmitted at faster rate. The small values of AIFS, CW_{min}, and CW_{max} which were respectively 2 μs, 7 slots, and 15 slots enabled VoIP transmitted by AC_0 to have a small waiting period before accessing the medium. Also, the large value of TXOP, which was equivalent to 3.01 ms, allowed AC_0 to transmit multiple data frames continuously during that particular time interval defined by TXOP.

However, the transmission mechanism, and the QoS requirements for a transmitted application could have an impact on its overall QoS rather than the traffic priority. Although the background traffic was transmitted by default using the lowest priority AC (i.e. AC_3), its overall QoS was outperformed on the overall QoS of the video traffic which was transmitted using AC_1. For instance, the values of the overall QoS for background traffic under heavy network load (i.e. 64 connections) were respectively 49.18%, and 33.48% using the QoS enhancement scheme and the legacy scheme. Whereas, the overall QoS values for the video traffic using the QoS enhancement scheme and the legacy scheme under the same network load were respectively 22.1%, and 15.01%. This is because the background traffic was transmitted using TCP, whereas the video traffic was transmitted using UDP. TCP is a connection-oriented protocol that provides a reliable, ordered, and error-checked delivery of packets transmitted between stations, while UDP is a connectionless Internet protocol with no acknowledgment of packet delivery. Also, the background traffic is tolerant of the QoS parameters such as delay, and jitter as compared with the video traffic. The acceptable QoS requirements for video application in terms of delay, jitter, and packet loss ratio should not exceed 150 ms, 10 ms, and 3% respectively.

Figures 7-5 (a) - (d) indicate that the overall QoS of traffic transmitted for all ACs using the proposed QoS enhancement scheme was higher as compared with the legacy network. For example, the average QoS for the VoIP, video, best effort traffic, and the background traffic, when the network load was light (i.e. 8 connections), were improved by 80.9%, 93.5%, 7.8%, and 29.4% respectively. The improvements in QoS were due to allocating the lower priority traffic to the next higher AC, in cases of zero PAR for the higher priority AC and the fairness introduced by WRR in treating the traffic based on their QoS requirements.

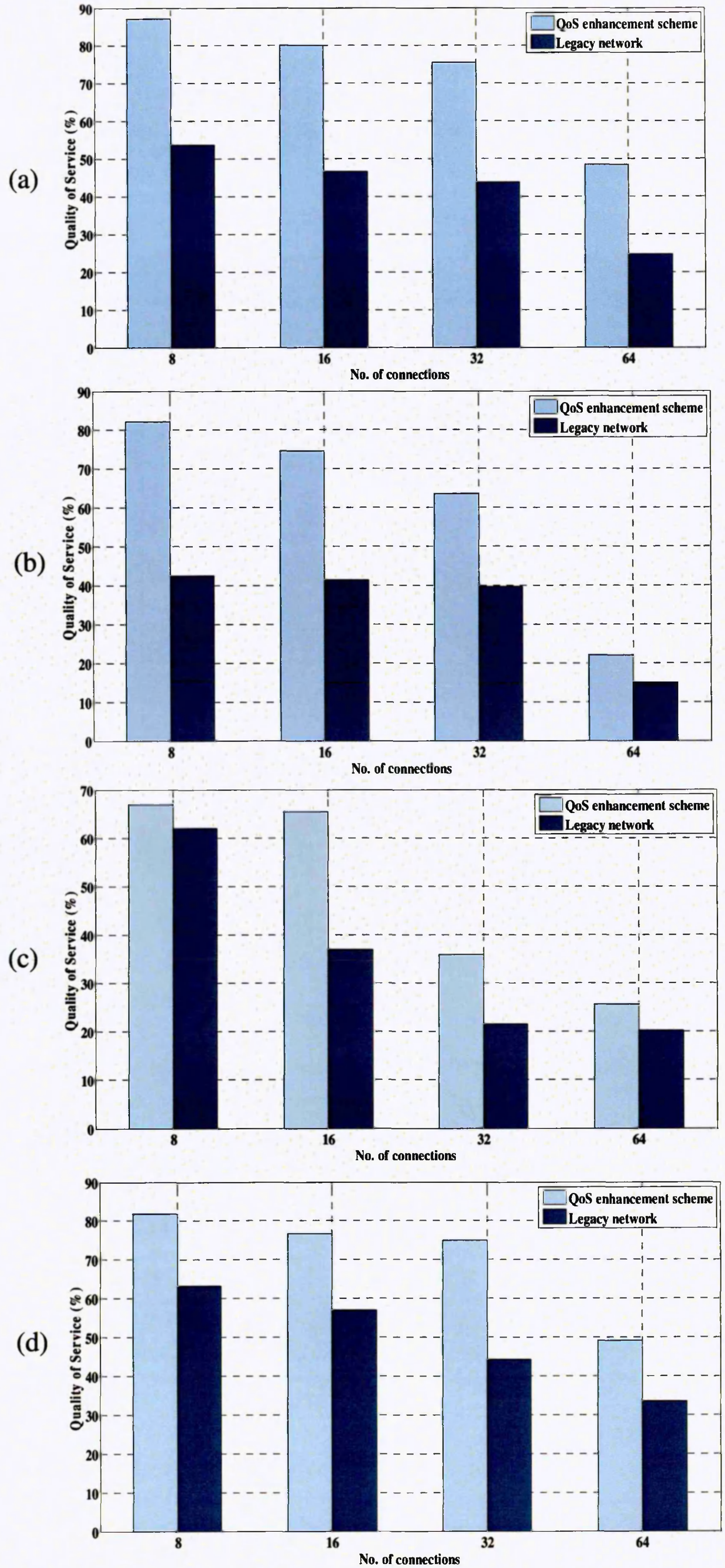
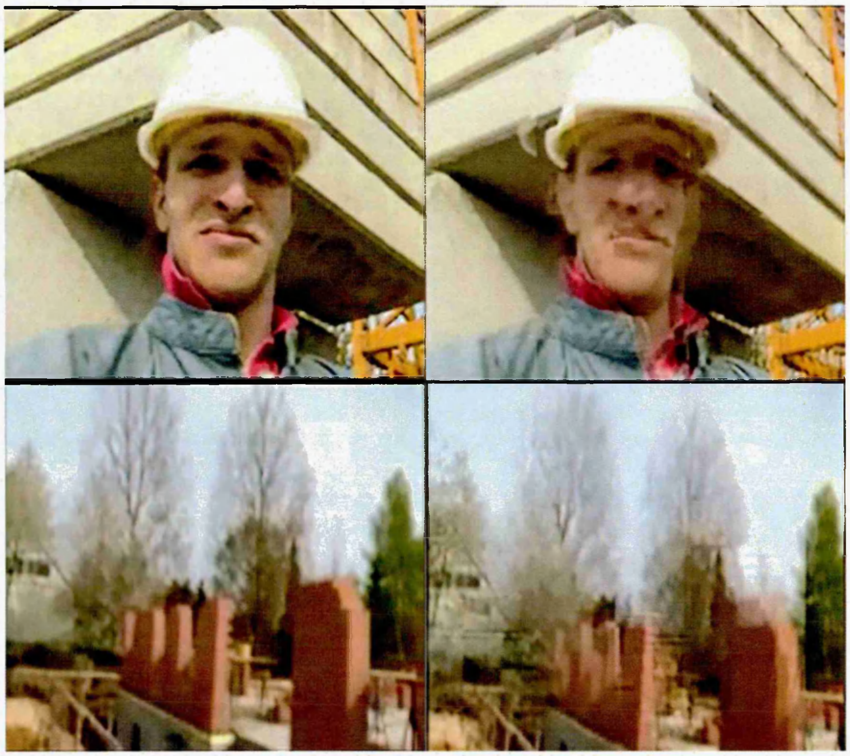


Figure 7-5. Bar chart representation of overall assessed QoS with and without QoS enhancement scheme for: (a) VoIP, (b) Video, (c) Best effort traffic, (d) Background traffic.

Figure 7-6 shows a visual comparison of sample images from the Foreman video transmitted over a medium network load (i.e. 24 connections) using the proposed QoS enhancement scheme. It was observed that the image quality with the QoS enhancement scheme was higher than the image quality without it. The number of received *I, P,* and *B* frames for the transmitted Foreman video was greater when using the QoS enhancement scheme as shown in Table 7-4. This improvement was due to the shift of the video packets from AC_1 to AC_0 in the absence of VoIP and the fairness of distributing network resources by WRR.



(a) (b)

Figure 7-6. Visual comparison of reconstructed Foreman video using: (a) QoS enhancement scheme, (b) legacy network scheme.

Table 7-4. The Amount of sent, received, and lost Foreman video frames.

Frame Type / No. of frames	I	P	B	Total
Sent	45	89	266	400
Received with QoS enhancement scheme	45	89	266	400
Received without QoS enhancement scheme	42	73	228	343
Lost with QoS enhancement scheme	0	0	0	0
Lost without QoS enhancement scheme	3	16	38	57

7.3 Summary

A new QoS enhancement scheme that consisted of a combination of an adaptive access category (AC) traffic allocation algorithm implemented on the wireless side of a simulated network, and the Weighted Round Robin (WRR) scheduling implemented on its wired side was devised and its performance was evaluated. The traffic allocation algorithm assigned traffic of a lower priority AC to the next higher priority AC in the absence of any higher priority traffic to further improve the performance of IEEE 802.11e EDCA standard, whereas WRR managed to fairly allocate network resources among transmitted traffic types based on their QoS requirements. The performance of the proposed scheme was compared with the standard IEEE 802.11e EDCA and FIFO queuing mechanisms (i.e. WLAN-wired network legacy scheme). The proposed scheme significantly improved the QoS for transmitted applications. The average QoS for VoIP, video, best effort traffic, and background traffic increased from their original values by 72.5%, 70.3%, 44.5%, and 45.2% respectively. The QoS proposed scheme allowed an end-to-end QoS to be set up which in turn provided an improved delivery for a variety of applications in the context of wired-cum-wireless networks.

Chapter 8 Microcontroller Board Implementation of Quality of Service Assessment System

8.1 Introduction

Network QoS assessment plays an important role in managing network resources and ensuring that various applications receive an appropriate priority or sufficient resources. Therefore, developing a hand-held system that accurately assesses QoS for different applications is very valuable.

In this chapter, a network QoS monitoring system is designed and evaluated. It incorporated the QoS assessment approach developed by (Dogman et al, 2012_a) that was based on regression model. More details about regression modelling based QoS assessment is provided in Section 6.3.3 of Chapter 6. The microcontroller board MCB2300 KEIL ARM was used for the purpose of this study.

Following the measurement of the QoS parameters: delay, jitter and packet loss ratio of multimedia applications, they are fed into the microcontroller board. The board then analysed the parameters based on their transmission requirements and produced the corresponding overall QoS. The performance of the system device is compared with other QoS assessment methods (e.g. QoS assessment using Fuzzy Inference System introduced by (Al-Sbou et al, 2006), and Neural Network QoS monitoring approach proposed by (Dogman et al, 2012_b). The results indicated that the developed system is capable of accurately assessing QoS.

This chapter is organised as follows: the relevant studies about QoS monitoring tools are discussed in section 8.2. Section 8.3 explains the MCB2300 KEIL ARM microcontroller board. Section 8.4 outlines how the QoS assessment technique using regression modelling was devised, and implemented on the MCB2300 KEIL ARM microcontroller board. The experimental procedures are described in section 8.5. The results are discussed in section 8.6. The conclusion is presented in Section 8.7.

8.2 Related Work

The growth in multimedia networks has led to the need for finding efficient ways to monitor QoS accurately. QoS can be achieved by either (i) prioritising time sensitive applications (such as video conferencing and voice over internet protocol) over time insensitive applications (such as file transfer), or (ii) reserving network resources (such as bandwidth) for the time sensitive applications prior to their transmission.

There are a number of QoS monitoring tools that have been proposed to monitor network performance as in (Graham et al, 1998), (Zseby and Scheiner 2004), and (Carvalho et al, 2009). A critical analysis of these tools can be found in Section 3.5, Chapter 3. The existing network QoS monitoring tools have a number of shortcomings. Some methods cannot determine directly the overall network QoS as in (Graham et al, 1998). Network managers have to do a variety of operations to assess the overall network QoS. Other QoS monitoring tools are not stand-alone devices as in (Zseby and Scheiner, 2004) and (Carvalho et al, 2009). From these limitations, the process of monitoring QoS can be complicated, expensive, and time consuming.

Therefore, in this study, a hand-held system that accurately determines the overall network QoS for multimedia applications was designed. The proposed device assessed network QoS for multimedia applications taking into the account the QoS requirements of these applications. A novel aspect of this study is that a microcontroller board with integrated QoS monitoring tool is used.

The use of embedded systems in recent years has increased in such a way that it is used in various mobile and multimedia applications. An efficient environment for microcontroller applications is KEIL Microcontroller Development Kit (MDK-ARM) (MDK-ARM, 2013). The features of KEIL MDK-ARM are its ease of use, and the manner it can be redesigned based on application's requirements without incurring major Non-Recurring Engineering (NRE) costs.

MDK-ARM Microcontroller board has been used in a number of studies to implement different applications. For instance, the digital implementation of a prototype DC motor control system was performed based on MCB 2300 KEIL microcontroller (Pal et al, 2009). The study showed that MCB 2300 KEIL microcontroller had facilities for concurrent programming and real-time control for fast handling of events in micro manufacturing applications.

The algorithm which was proposed by (Thangaraj et al, 2006) enhanced MCB 2300 KEIL microcontroller performance. Most applications that run on embedded platform were constrained by limited built in memory available for storing the application programs. Therefore, the proposed algorithm reduced the KEIL ARM memory requirement needed by an application program.

The contribution of this study is the implementation of the regression modelling used to assess QoS which was reported by (Dogman et al, 2012a) on the KEIL ARM MCB2300 microcontroller board. The proposed system could work independently to assess the QoS of multimedia applications based on their transmission requirements (Dogman et al, 2012_d), and (Dogman et al, 2013).

8.3 KEIL ARM MCB2300 Evaluation Board

The Keil MCB2300 Evaluation Board is designed to be a very flexible evaluation board for the NXP LPC2300 family of microprocessors. Due to its facilities to create, test, and run application programs, the MCB2300 evaluation board can be expanded to build hardware prototypes. MCB2300 evaluation board operation can be described in terms of its hardware and Development Kit.

8.3.1 MCB2300 Hardware Components

The hardware components and interfaces of the Keil MCB2300 evaluation board are shown in Figure 8-1 (MDK-ARM, 2013). The interfaces on the Keil MCB2300 evaluation board provide an easy access to the on-chip peripherals.

8.3.2 MDK-ARM Microcontroller Development Kit

The MDK-ARM is a complete software development environment for ARM processor-based devices. MDK-ARM is designed for microcontroller applications. Its simplicity to learn and use makes it powerful software for most demanding embedded applications. MDK-ARM Microcontroller Development Kit includes many components as shown in Figure 8-2 (MDK-ARM, 2013). These components are:

- **ARM C/C++ Compiler:** this component allows the user to write ARM applications in C or C++ and then compiles C/C++ source files to have the efficiency and speed of assembly language. The ARM Compiler translates C/C++ source files into re-locatable object modules which contain full symbolic information for debugging with the µVision Debugger.

- **µVision Project Manager:** µVision Integrated Development Environment (IDE) is a complete simulation in one powerful environment. It provides facilities such as source code editing, and program debugging. The µVision IDE software is used to create, compile, download, debug, and run a program on the MCB2300 board. When the program is downloaded and ran successfully, the MCB2300 board could work independently as a standalone device.

- **RTX Real-Time Operating System:** the Keil RTX Real-Time Operating System is designed for ARM devices. It allows the user to create programs that simultaneously perform multiple functions. RTX Real-Time Operating System aids to create good structured and easy maintained applications.

- **CAN Driver:** MDK tool kit includes a CAN interface layer which provides an easy and quick approach to implement a CAN network. It also provides a standard programming API for all supported microcontrollers.

- **Flash File System:** the Flash File System allows the embedded applications to create, save, read, and modify files in standard storage devices such as ROM, RAM, Flash ROM, and SD/MMC/SDHC Memory Cards.

- **USB Host and Device Interface:** MDK-Professional provides USB Host and USB Device support for embedded systems. The USB Host library is an embedded USB stack supporting USB MSC (Mass Storage Class) and HID (Human Interface Device) classes. The USB Device Interface uses standard device driver classes that are available with all Windows PCs.

- **TCP/IP Networking Suite:** the full TCP/IP Networking Suite is designed for ARM processor-based microcontrollers. It is highly optimized, has a small code footprint, and gives an excellent performance.

- **Graphic User Interface Library (GUI):** the GUI Library is a fully featured graphics suite that makes it possible to add graphical user interfaces to embedded applications. It supports a large number of displays such as monochrome, grayscale and colour LCDs.

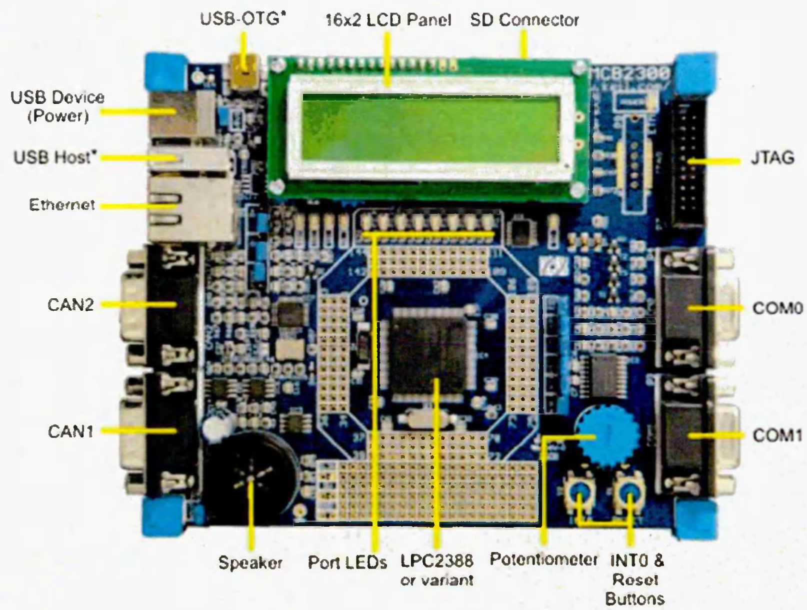


Figure 8-1. KEIL MCB2300 Evaluation Board.

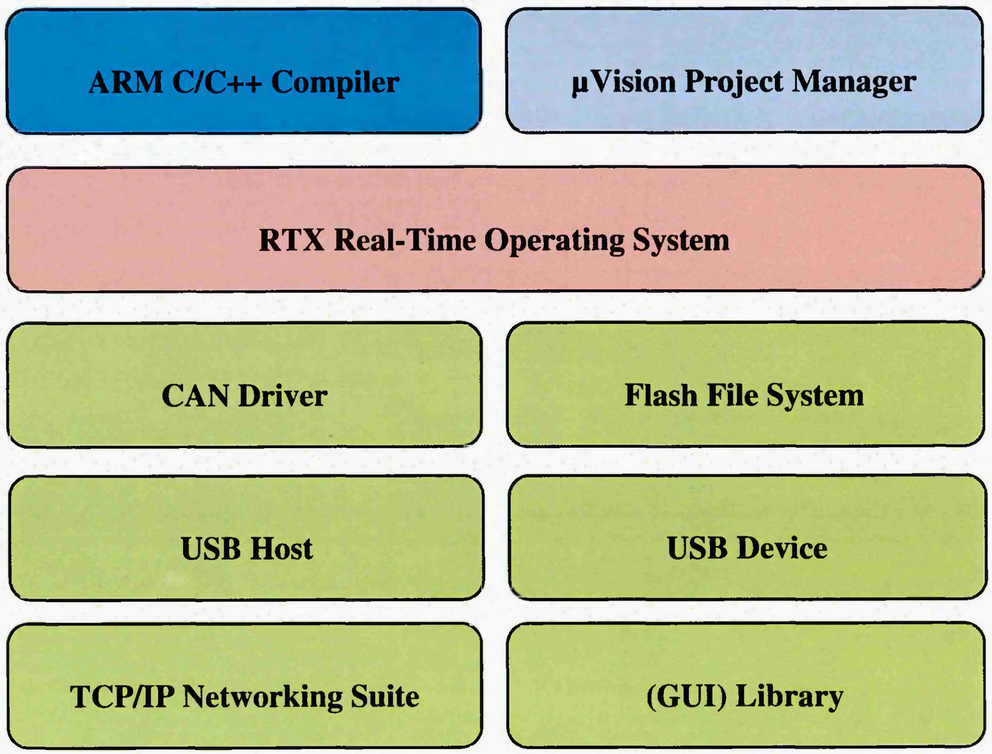


Figure 8-2. MDK-ARM Microcontroller Development Kit.

8.4 QoS Assessment Implementation Using KEIL MCB2300 ARM

A schematic diagram of computer network QoS monitoring system is shown in Figure

8-3. This used the regression model (explained in Section 6.3.3, Chapter 6) on a KEIL MCB2300 microcontroller board.

The QoS parameters of transmitted multimedia applications (i.e. delay, jitter, and packet loss ratio) were obtained from the generated trace files of the simulated network. These were used as inputs to KEIL MCB2300 microcontroller in order to quantify the overall QoS. The following subsections explain the QoS assessment technique using regression model and its implementation on the MCB2300 KEIL ARM board microcontroller.

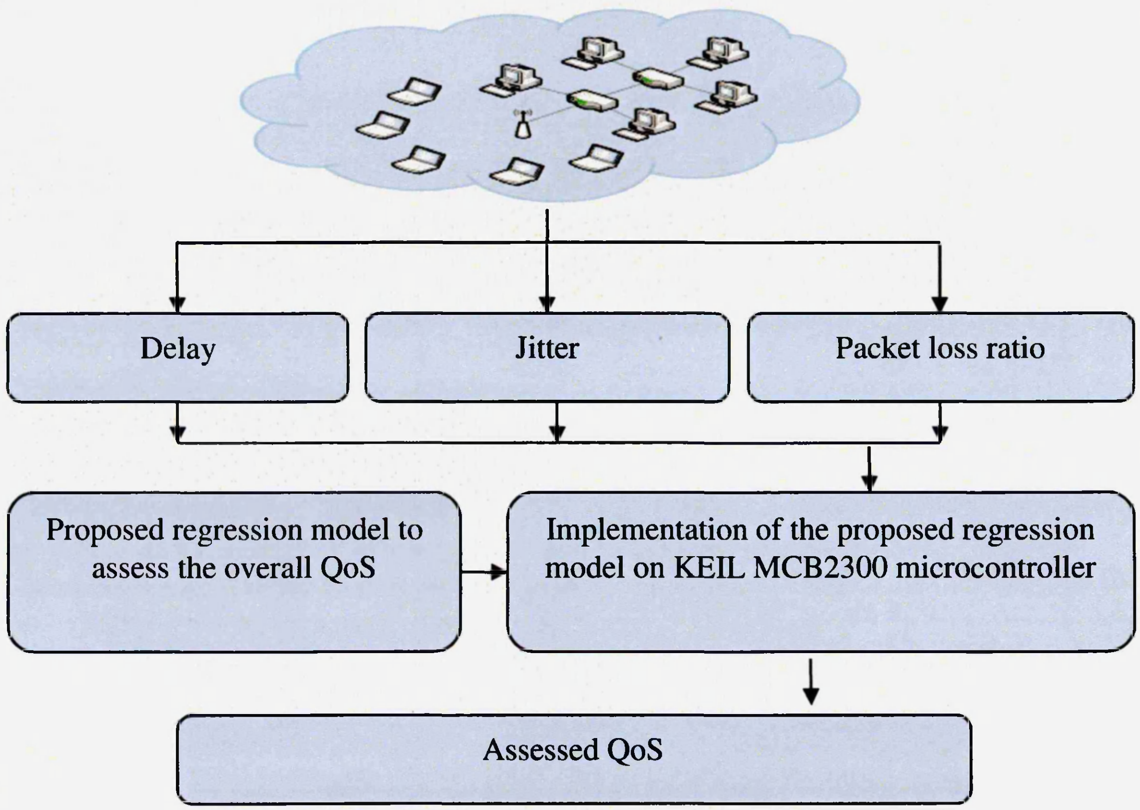


Figure 8-3. QoS monitoring System.

8.4.1 Proposed Regression Model to Assess QoS

In Section 6.3.3, Chapter 3, the QoS assessment technique using regression model was proposed and evaluated.

In brief, in the devised regression model, the values of independent variables (x_1, x_2, x_3) were represented by delay, jitter, and packet loss ratio respectively, while the values of dependent variable (y) were represented by the overall QoS.

The regression expression was developed based on QoS requirements listed in Table 2-1 in order to provide the outputs that reflected the overall QoS. The QoS parameters and the overall QoS were then arranged in matrices in order to feed them to the regression model as follows:

$$\begin{bmatrix} QoS_1 \\ QoS_2 \\ \vdots \\ QoS_n \end{bmatrix} = \begin{bmatrix} 1 & D_1 & J_1 & PLR_1 \\ 1 & D_2 & J_2 & PLR_2 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & D_n & J_n & PLR_n \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{bmatrix}$$

where $D_i,\ J_i,\ PLR_i,\ QoS_i,\ i = 1,2,\dots,n$ are delay, jitter, packet loss ratio, and overall QoS respectively. The regression coefficients b_0, b_1, b_2, b_3 were determined from the recorded data using equation (2.12). The vector of residual (i.e. error terms) was then calculated using equation (2.13). In this study, the means of vector of residual produced from regression formula for VoIP and video traffics were zeros (i.e. mean of error terms $e_i, i = 1,2,\dots,n$ was zero). This implied that the estimated regression model determined was as expressed in equation (8.1):

$$QoS_i = b_0 + b_1 * D_i + b_2 * J_i + b_3 * PLR_i \tag{8.1}$$

where $QoS_i, D_i, J_i,\ PLR_i \quad i = 1,2,\dots,n$ are the overall QoS delay, jitter, packet loss ratio for i^{th} packet respectively.

8.4.2 Implementation of QoS Assessment Technique using KEIL ARM Microcontroller

In this section, the QoS assessment technique using regression model is implemented on Keil MCB2300 microcontroller. The code for developed QoS assessment technique using regression model was written in C language in order to be implemented on MCB2300 board.

The KEIL µVision was used to create, compile, download, debug, and run the C program on the MCB2300 board. Once the C program operation was successfully verified, the KEIL µVision downloaded the C code on the MCB2300 microcontroller. The KEIL ARM MCB2300 microcontroller then was able to work as a standalone QoS monitoring device.

Figure 8-4 shows the pseudo code of the C program implemented on the MCB2300 microcontroller. First, the QoS parameters of multimedia applications (i.e. delay, jitter, and packet loss ratio) were fed into the QoS monitoring device (i.e. programmed

MCB2300 board) through the SD connector. The Liquid Crystal Display (LCD) of the QoS monitoring device indicated when the SD card was inserted and then the data processing was initiated. The device then processed the QoS parameters file, read the values of delay, jitter, and packet loss ratio, and assessed the overall QoS using equation (8.1).

Afterward, the device created an output file and recorded the values of delay, jitter, and packet loss ratio, and their corresponding QoS value.

As soon as the QoS monitoring device completed the data processing, the overall QoS was displayed on the LCD of the device.

```
// Process of inserting SD card
If (SD card is not inserted) then
{ LCD output ("No SD-Card used") }
Else
{ LCD output ("SD-Card used")
// Process of reading QoS parameters
LCD output ("Processing Data")
Total QoS=0
Counter=0
Overall QoS=0
Open QoS parameters file
Create overall QoS file
While (Not an end-of-file)
{ Read delay, jitter, and packet loss ratio from QoS parameters file
// Process of assessing QoS
Calculate QoS using equation (8.1)
// The output process
Write delay, jitter, packet loss ratio, and assessed QoS on overall QoS file
Total QoS= Total QoS + QoS
Counter is incremented by 1
}
LCD output ("Processing End")
Overall QoS= Total QoS / Counter
LCD output ("Overall QoS=", Overall QoS)
}
```

Figure 8-4. QoS Assessment Code.

8.5 Experimental procedure

The aim of this section is to validate the performance of QoS monitoring system. This section includes network modelling and simulation, and hardware and software for experimental setup.

8.5.1 Modelling and Simulation

A wireless-cum-wired network topology was simulated using the Network Simulator- 2 (NS-2) in order to validate the performance of QoS monitoring system. The network topology consisted of 8 wireless nodes, 2 wired nodes, and 2 base stations as illustrated in Figure 8-5. The bandwidth of wired connections was 5 Mbps and 2 ms propagation delay. The Wireless Local Area Network (WLAN) was based on IEEE 802.11e standard and implemented Enhanced Distributed Channel Access (EDCA) technique.

The queue management mechanism was Drop-Tail and the queue size was 50 packets. A number of traffic types were transmitted over the simulated network. These were; VoIP, video-conferencing, and best effort traffic. Constant Bit Rate (CBR) traffic was adapted to model VoIP, videoconferencing, and data. VoIP modelled as G.711 voice encoding scheme with 160 packet size and 64 kbps generation rate. The packet size of the video traffic was 512 bytes and the inter-packet interval was 10 ms. This generated a packet transmission rate of 384 kbps. The best effort traffic was modelled using different packet sizes and the generation rates that corresponded to non-videoconferencing or VoIP usage. All traffic were transmitted using UDP.

The simulation time was 500 seconds and was repeated 10 times for each experiment. Each time a different initial seed was used in order to randomly manage which node transmitted first as all the nodes were requested to transmit at a given time. The randomness introduced using the different seeds avoided the bias of random number generation.

The transmitted traffic (VoIP, video-conferencing, and best effort traffic) were mapped into three access categories (ACs) to represent the three priority levels as shown in Table 8-1. VoIP had the highest priority, whereas best effort traffic had the lowest priority (IEEE Computer Society, 2005).

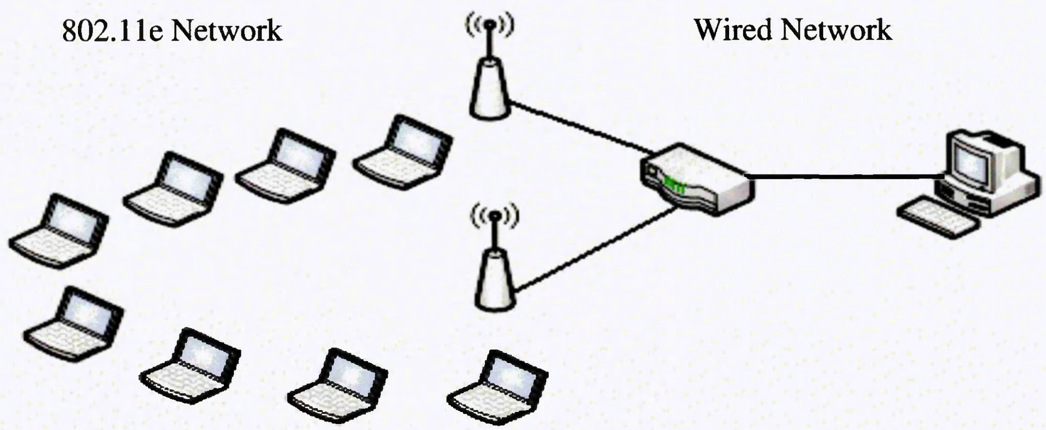


Figure 8-5. The Simulated Network.

Table 8-1. IEEE 802.11e Access Categories.

Parameters	AC_0 VoIP	AC_1 Video-conferencing	AC_2 best effort traffic
Arbitration Inter-Frame Space (AIFS)	2	2	3
Minimum Contension Window value (CW_{min})	7	15	31
Maximum Contension Window value (CW_{max})	15	31	1023
Transmit Opportunity TXOP (ms)	3.01	6.02	0

8.5.2 Hardware and Software Setup

This section explains the setup for the MCB2300 microcontroller board. This includes the details about connecting and configuring procedures for the MCB2300 evaluation board.

8.5.2.1 Hardware Setup

The following components were needed in this experiment in order to use the MCB2300 Evaluation Kit:

- The MCB2300 microcontroller board.
- Compatible PC with at least one unused USB port in order to supply power to the board and for downloading and debugging purpose.
- The Keil ULINK-ME USB-JTAG Adapter to run the Keil debugger using JTAG emulation.

- One USB cable.

The Keil ULINK-ME USB-JTAG Adapter which is shown in Figure 8-6 was first connected to the MCB2300 board via its JTAG plug. Then the ULINK-ME USB-JTAG Adapter was connected to the PC's USB port using a standard USB cable. This allows the user to power to board, and download programs on the MCB2300 evaluation board.

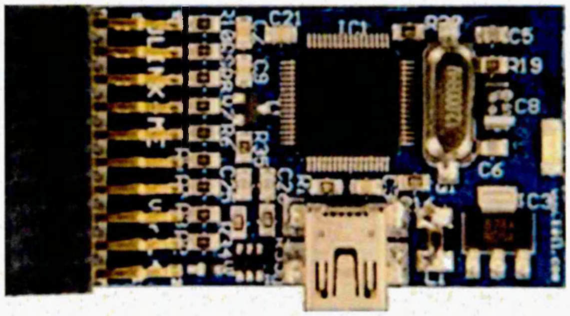


Figure 8-6. The ULINK-ME Adapter.

Figure 8-7 shows the connection between the PC and the MCB2300 evaluation broad using the Keil ULINK-ME USB-JTAG Adapter.

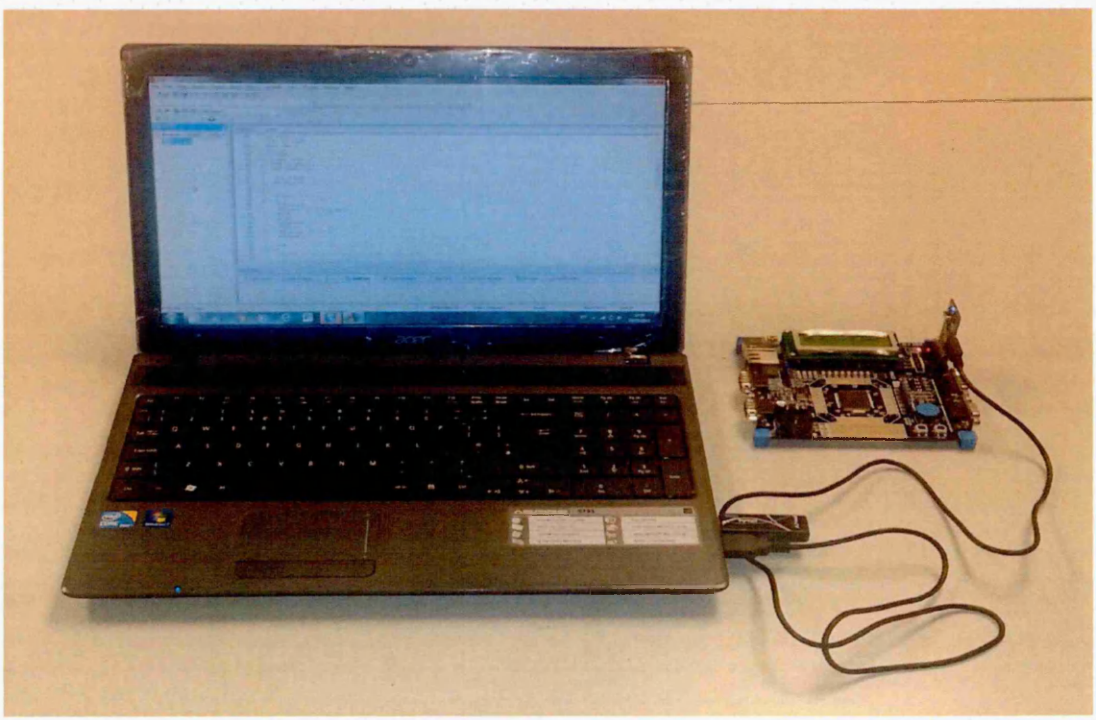


Figure 8-7. The connection between the PC and the MCB2300 using ULINK-ME JTAG Adapter.

8.5.2.2 Software Setup

The MCB2300 Evaluation Kit uses the µVision IDE software tool. In this experiment, the MCB2300 Evaluation Board was connected directly to the KEIL µVision IDE software which was installed on the PC via the KEIL ULINK-ME USB-JTAG Adapter.

The Keil µVision IDE software was the front-end used with ULINK-ME adapter to create, download, and test the embedded application on the MCB2300 microcontroller board. Therefore, no additional software was required to run the board. Figure 8-8 shows the snapshot of µVision IDE software tool (MDK-ARM, 2013).

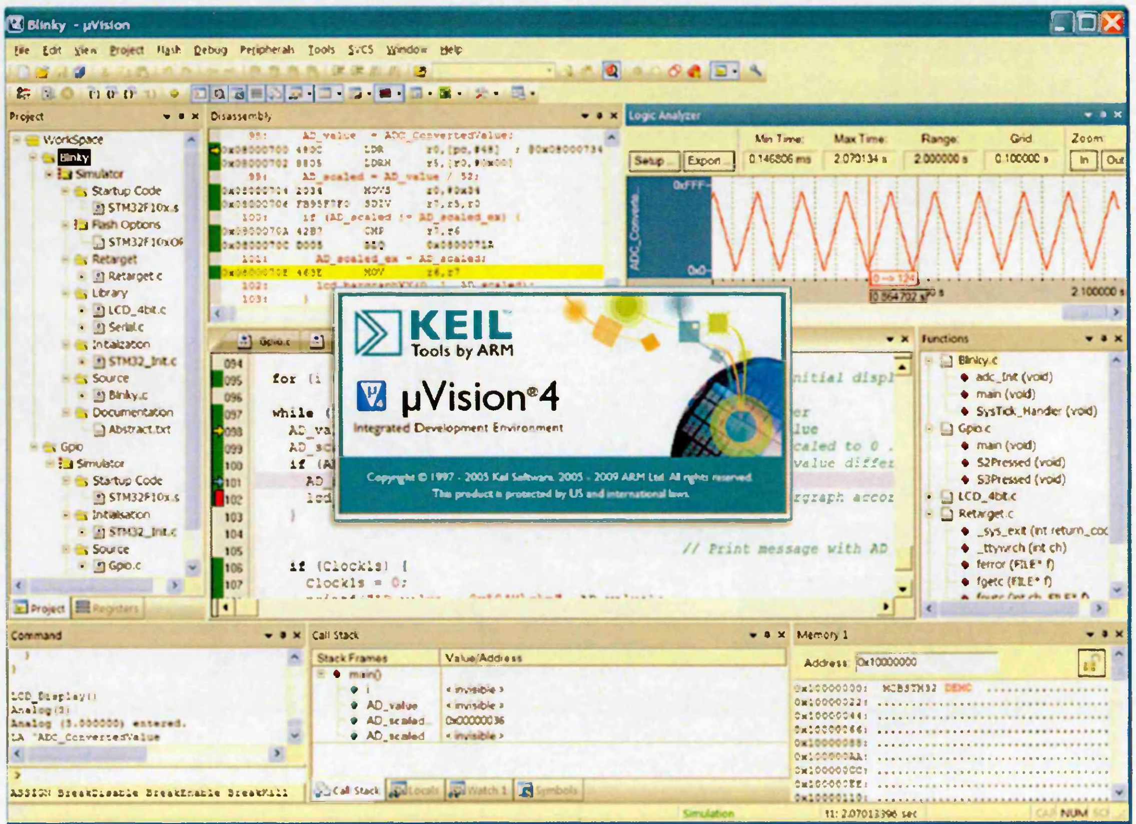


Figure 8-8. µVision IDE software tool.

The project development cycle of the Keil µVision IDE is similar to any other software development project. The development cycle started with creating a project for an embedded application until downloading the application on target hardware. In this experiment, the following stages were implemented to create an embedded application with µVision IDE software:

1. Create a project, select the target device from the Device Data base, and configure the tool settings: in this study, a new project file called "**SD-File**" was created from

"**New Project**" option in "**Project**" Menu. μVision then created a new, empty project file with the specified name. After a file name for the project was selected, a target microcontroller was chosen. μVision had an option to select a target microcontroller. The "**Select Device for Target option**" from "**Project**" Menu lists all the devices from the μVision Device Database. The target device was MCB2300 microcontroller. This step is very important, since μVision customizes the tool settings, peripherals, and dialogs for a chosen device.

2. *Create source files in C/C++ or Assembly:* the "**New**" option from "**File**" Menu was used to create a source code written in C language. The source code was saved in "**main.c**" file. This file contained the pseudo code of QoS assessment shown in Figure 8-4. After a source file was created and saved, it was added to the project. The "**SD-File**" project had a single source file called "**main.c**". This file contained a C function which was used to read the QoS parameters of multimedia applications and assess the overall QoS using regression model for QoS assessment.

3. *Building the "SD-File" project with the Project Manager:* There are several commands were used from the "**Project**" Menu to compile and link the files in the project. These were: "**Translate File**" command to compile the selected file in the Project Workspace, "**Build Target**" command to compile files that have changed since the last build and link them, and "**Rebuild All Target Files**" command to compile and link all files in the project.

4. *Downloading the "SD-File" program to MCB2300 board and test the linked application:* the "**Configure Flash Tools**" command from "**Flash**" Menu was used to configure the target driver for flash programming. After the μVision IDE was configured, the "**Download**" command from "**Flash**" Menu used the specified adapter for Flash programming in order to flash the "**SD-File**" application program to the target hardware (i.e. MCB2300 board). Blinking LED indicated successful download of the program into the target hardware.

5. *Debug the "SD-File" program application:* the aim of debugging was to verify and optimize the "**SD-File**" program application. The **Options for Target – Debug** dialog was used to verify the configuration settings for the Debugger. The "**Start/Stop Debug Session**" command from "**Debug**" Menu was used to debug "**SD-File**" program application.

8.6 Results and Discussion

In this section, the feasibility and the functionality of the proposed QoS monitoring device is demonstrated. The measured QoS results obtained using the proposed device were compared with the results obtained using other QoS assessment techniques. These techniques were QoS assessment using Fuzzy Inference System (FIS) introduced by (Al-Sbou et al, 2006), and neural network QoS monitoring approach proposed by (Dogman et al, 2012_b).

Due to the high sensitivity of multimedia applications to the QoS parameters as shown in Table 2-1 (See Chapter 2), in this chapter, the QoS for VoIP, and video traffic were measured and evaluated. The following subsections respectively evaluate the assessed QoS for the transmitted traffic (i.e. VoIP, and video traffic).

8.6.1 VoIP Traffic

After the traffic was configured and the network topology was simulated, the QoS parameters (i.e. delay, jitter, and packet loss ratio) of VoIP were extracted from the generated trace files of the simulated network. The QoS parameters were then used as inputs to the KEIL MCB2300 microcontroller in order to quantify the QoS for VoIP.

Table 8-2 shows the QoS parameters of VoIP, and the evaluated QoS obtained from the QoS Monitoring Device (QoS_MD), Fuzzy Inference System technique (FIS), and Multi-Layer Perceptron neural network (MLP). The QoS assessment techniques as well as the QoS monitoring device quantified the overall QoS based on the values of delay, jitter, and Packet Loss Ratio (PLR) taking into the account the QoS requirements for VoIP application. The overall QoS spanned between (0%-100%). Poor QoS did not exceed 33%, whereas Good QoS could not be below 67%. More details about the aforementioned QoS assessment techniques can be found in (Al-Sbou et al, 2006), (Dogman et al, 2012_b), (Dogman et al, 2012_d), and (Dogman et al, 2013).

From Table 8-2, it can be observed that the QoS values reflected the corresponding QoS parameters. In other words, as the values of QoS parameters increased, the values of overall QoS decreased accordingly. For example, when the values of delay, jitter, and packet loss ratio were 16.9ms, 0.8ms, and 0% respectively, the values of QoS for VoIP were in Good region, i.e. 92.5% for QoS monitoring system, 89.6% for FIS based technique and 96.9% for MLP neural network based technique.

In contrast, when the values of QoS parameters were high (i.e. 422.3ms for delay, 5.0 ms for jitter, and 6% for packet loss ratio), the corresponding QoS were at the Poor level, i.e. 1.8% for QoS monitoring system, 9.3% for FIS based technique, and 4.2% using MLP neural network. This was due to the gradual increase of delay, jitter, and packet loss ratio for VoIP traffic when the load on the network was increased. As the number of active stations increased; the probability of collisions increased due to the high degree of competition between stations. This is in turn forced the MAC protocol to retransmit the collided packets, and subsequently increased the values of delay, jitter, and packet loss.

Table 8-2. QoS Parameters, and the evaluated QoS of VoIP application using QoS Monitoring Device (QoS_MD), Fuzzy Inference System (FIS), and Multi-Layer Perceptron (MLP).

QoS Parameters			Evaluated QoS		
Delay(ms)	*Jitter (ms)*	*PLR (%)*	*QoS_MT*	*FIS*	*MLP*
11.5	5.0	0.0	28.1	9.3	26.0
16.9	0.8	0.0	92.5	89.6	96.9
16.1	3.1	2.0	50.7	47.7	46.8
17.1	2.9	2.0	53.7	51.1	52.9
117.4	5.0	5.7	7.7	9.3	11.4
286.9	5.0	6.0	4.0	9.3	7.3
24.3	2.5	2.0	59.7	54.1	61.5
105.8	5.0	6.0	6.9	9.3	8.7
422.3	5.0	6.0	1.8	9.3	4.2
600.0	5.0	6.0	0.01	9.3	1.1

As shown in Table 8-2, the results obtained from QoS monitoring device were compared with other QoS assessment methods (i.e. QoS assessment using FIS technique introduced by (Al-Sbou et al, 2006), and MLP neural network QoS monitoring approach proposed by (Dogman et al, 2012b)). The correlation coefficient (*R*) was used to evaluate the accuracy of the proposed device compared with the other QoS assessment techniques. When using the correlation coefficient, the magnitude of *R* is between 0 and 1. The magnitude closest to 1 indicates a perfect correlation, whereas a correlation less than 0.5 would be described as weak correlation. More details about measuring accuracy using correlation coefficient can be found in Section 6.3.5 (see Chapter 6)

In this study, the values of the correlation coefficients between QoS determined using the monitoring tool, and the other QoS assessment methods were: 0.97 for FIS technique, and 0.99 for MLP neural network.

From the values of evaluated QoS provided in Table 8-2, it can be concluded that all approaches provided results which were closely comparable. Although some outputs were slightly different, they were in the same QoS region (i.e. Poor, Average, or Good). The discrepancies were due to the fact that each method followed a different scheme to determine QoS. However, the values of QoS obtained from the proposed QoS monitoring device spanned between 1-100%, whereas the range of QoS values produced by FIS was between 10-90%. These indicate that the QoS monitoring system was accurately in its operation.

8.6.2 Video Traffic

The QoS parameters of video traffic were also fed to the QoS monitoring device (i.e. programed KEIL MCB2300 microcontroller board) to assess the overall QoS. The extracted delay, jitter, and packet loss ratio of video traffic were combined by the QoS monitoring device which in turn produced its overall QoS.

As shown in Table 8-3, the QoS parameters of video application were processed using QoS monitoring system, Fuzzy Inference System technique (FIS), and Multi-Layer Perceptron neural network (MLP). The QoS assessment processes for these techniques including the QoS monitoring system were based on the QoS requirements for video application. A good overall QoS which ranged between 67-100%) corresponded to low value of QoS parameters (i.e. delay $\leq$ 150 ms, jitter $\leq$ 10 ms, and packet loss ratio $\leq$ 1%), An average QoS (i.e.33% $<$ QoS $\leq$ 67%) corresponded to medium QoS parameters (i.e. 150 $<$ delay $\leq$ 400 ms, 10 $<$ jitter $\leq$ 20 ms, and 1% $<$ packet loss ratio $\leq$ 2%), and high QoS parameters (i.e. delay $>$ 400 ms, jitter $>$ 20 ms, and packet loss ratio $>$ 2%) corresponded to a poor QoS (i.e. QoS $\leq$ 33%).

Therefore, it can be observed from Table 8-3 that the values of evaluated QoS are reflecting the corresponding QoS parameters. Low values of QoS parameters produce high values of overall QoS and vice versa. For example, when the values of delay, jitter, and packet loss ratio were 55.2ms, 11.32ms, and 0% respectively, the values of expected QoS for video application were 77.14% when QoS monitoring device was used, 80.69% when FIS technique was used, and 81.73% when using MLP neural network. The expected QoS values were all in Good QoS region. In contrast, when the values of QoS parameters were high (i.e. 533.4ms for delay, 24.5ms for jitter, and 3% for packet loss ratio), the corresponding QoS were in poor region. The values of QoS were 7.81% using QoS monitoring device, 9.81% using FIS technique, and 5.27% using MLP neural network.

Table 8-3. QoS Parameters, and the evaluated QoS of video application using QoS Monitoring Device (QoS_MD), Fuzzy Inference System (FIS), and Multi-Layer Perceptron (MLP).

QoS Parameters			Evaluated QoS		
Delay(ms)	*Jitter (ms)*	*PLR (%)*	*QoS_MT*	*FIS*	*MLP*
55.20	11.32	0.00	77.14	80.69	81.73
215.34	1.41	0.00	82.42	84.41	79.37
41.46	19.58	0.00	57.35	54.81	58.21
235.44	6.68	0.00	66.08	78.52	73.36
81.65	29.60	0.00	16.03	9.29	12.30
420.45	5.49	0.00	45.39	30.23	42.15
452.07	12.20	0.00	23.81	21.02	32.81
344.27	30.00	0.00	8.79	9.28	8.67
600.00	9.74	0.00	11.21	9.39	11.65
533.35	24.54	3.00	7.81	9.81	5.27

From Table 8-3, the results obtained from the QoS monitoring device, QoS assessment using FIS technique, and MLP neural network QoS monitoring approach were comparable. The correlation coefficient (R) was used to evaluate the accuracy of the proposed QoS device as compared with the other QoS assessment techniques.

In this study, the values of the correlation coefficients between QoS determined using the monitoring system, and the other QoS assessment methods were: 0.98 for FIS technique, and 0.99 for MLP neural network.

From the values of evaluated QoS obtained from the QoS monitoring system, QoS assessment using FIS technique, and MLP neural network QoS monitoring approach as provided in Table 8-3, it can be concluded that all approaches provided results which were closely comparable. Nevertheless, some outputs were slightly different. These divergences were because each method followed a different scheme to determine QoS. However, the values of QoS obtained from the proposed QoS monitoring system spanned between 1-100%, whereas the range of QoS values produced by FIS was between 10-90%. This indicates that the QoS monitoring device could be more accurate.

8.7 Summary

In this study, a portable hand-held system to assess the QoS for multimedia applications was designed and evaluated. Our developed QoS assessment technique which was based on the regression model was implemented on the MCB2300 KEIL ARM microcontroller board. The proposed system analysed the QoS parameters for multimedia applications to measure the overall QoS. The QoS parameters (i.e. delay, jitter, and packet loss ratio) were fed into the proposed device which in turn produced a single value that represented the overall QoS.

The QoS assessment results were highly correlated with results obtined from a number of previously developed QoS assessment methods. This indicated the correctness of the developed system in monitoring QoS.

Further discussion about the results obtained from the developed approaches in this thesis will be presented in the next chapter. Chapter 9 will conclude the thesis and provide recommendations for future work.

Chapter 9 Conclusions, and Future Work

9.1 Conclusions

In this research, network QoS management referred to evaluation and improvement of QoS provided by wired and wireless computer networks. Therefore, the main focus was on development of techniques to evaluate QoS in multimedia networks and the use of this information as part of network management to improve its performance.

In chapter 5, statistical adaptive sampling techniques to adjust sampling rate based on traffic's statistics were developed and evaluated. Three adaptive statistical sampling techniques were proposed to sample multimedia traffic. The sampling rates of the three devised sampling techniques were controlled using three different mechanisms: simple linear adjustment mechanism, quarter adjustment mechanism, and fuzzy inference system. The proposed adaptive statistical sampling techniques decreased the sampling rate when the statistics of the traffic did not significantly change over time (i.e. steady traffic) and increased the sampling rate when the statistics of the traffic significantly changed with time (i.e. time varying traffic). A comparison of adaptive statistical sampling techniques versus conventional sampling techniques (i.e. systematic sampling, stratified sampling, and random sampling) was also carried out.

The main QoS parameters of VoIP traffic (i.e. throughput, delay, jitter, and packet loss ratio) with their sampled versions using adaptive and non-adaptive sampling techniques were discussed in chapter 5. It was concluded from the findings in chapter 5 that the sampled versions of throughput, delay, jitter, and packet loss ratio obtained using the adaptive statistical sampling approaches were closer to the actual population than the non-adaptive sampling approaches (i.e. systematic, stratified, and random sampling). For instance, the bias values of sampled jitter versions obtained from adaptive sampling based on fuzzy approach, linear adjustment approach, and quarter adjustment approach were closer to zero as compared with non-adaptive sampling approaches as shown in Figure 5-12 (See chapter 5).

This concludes that the developed adaptive statistical sampling methods were more effective than conventional sampling methods in representing the traffic. This is because the sample interval was adjusted during the sampling process in case of adaptive statistical sampling approaches based on linear adjustment mechanism, quarter adjustment mechanism, and FIS, whenever the calculated overall traffic statistic changed significantly over time. Conversely, the sampling rates of the conventional sampling techniques were either constant as in systematic sampling or changed randomly as in stratified and random samplings. The fixed and random sampling rates resulted in a significant discrepancy between the actual data and its sampled version. The advantages of the proposed adaptive statistical sampling techniques were the ease of implementation and their ability to be implemented in real time.

Chapter 6 of this research is about development of techniques to analyse and assess QoS parameters of multimedia networks accurately in real time. The multimedia QoS information collected by adaptive statistical sampling techniques was considered.

Two innovative QoS evaluation approaches that combine analysis and measurement techniques were developed. The first approach combined FCM and regression model to analyse and assess QoS of multimedia applications in a simulated network. The second approach analysed and assessed QoS in multimedia applications using a combination of supervised and unsupervised neural networks.

The contribution of chapter 6 was to analyse and classify network QoS parameters (i.e. delay, jitter, and packet loss ratio) using either Fuzzy C-means (FCM) or Self Organizing Maps (SOM). Another contribution was to propose QoS assessment techniques. The proposed techniques assess QoS in a manner similar to human subjects and quantified the QoS without the necessity for complex mathematical models. Also, the proposed QoS assessment techniques did not add significant extra load to the network. The proposed assessment techniques were based on the traffic generated from the proposed analysis techniques. A regression model was developed and a multi-layer perceptron (MLP) was trained to combine the QoS parameters (i.e. delay, jitter, and packet loss ratio) for each QoS class identified by SOM or FCM to estimate the overall QoS. Both regression model and MLP were capable of combining QoS parameters (i.e. delay, jitter, and packet loss ratio) to provide overall QoS.

The accuracy of QoS evaluation approaches was examined using typical values of QoS parameters for VoIP, and a video. The findings of chapter 6 showed that FCM and Kohonen network classified the values of QoS parameters of transmitted VoIP and video into clusters representing Low, Medium, and High values of QoS as illustrated in Figures 6-8, 6-9, 6-12, and 6-13. The regression model and MLP in turn combined the

QoS parameters (i.e. delay, jitter, and packet loss ratio) for each centre of generated clusters and produced a single value that represented the overall QoS. The overall QoS was an accurate indication of network performance as indicated in Tables 6-3, and 6-4.

Chapter 7 of this thesis considers the use of QoS information as part of multimedia network management to improve its performance. Therefore, another object of this study was about deployment network QoS enhancement which can be an effective solution for multimedia applications to be shared under finite network resources.

A new QoS enhancement scheme for WLAN-wired networks was proposed. The devised enhancement scheme consisted of an adaptive Access Category (AC) traffic allocation algorithm which was incorporated into the network's wireless side to improve the performance of IEEE 802.11e Enhanced Distributed Channel Access (EDCA) protocol, and a Weighted Round Robin (WRR) queuing scheduling mechanism that was incorporated into the wired side of the network.

The algorithm considered the Packet Arrival Rate (PAR) of the uplink and downlink traffic for each access category (AC), in order to allocate traffic to an appropriate AC. PAR was chosen because it is an important QoS parameter that affects most real-time and non-real-time applications. In addition, PAR can be easily computed while applications are being transmitted. This in turn facilitated quick allocation of the arrived traffic to the most appropriate AC. Once PAR values for the up-link and down-link traffic for each AC were determined, the algorithm dynamically allocated the traffic of a lower priority AC to the next higher AC, when the higher AC was not receiving traffic at the time. On the wired side of the network, a weighted round robin (WRR) shared the network resources, based on the traffic's quality of service requirements.

The performance of the proposed scheme was compared with the standard IEEE 802.11e EDCA and FIFO queuing mechanisms (i.e. WLAN-wired network legacy scheme). The incorporation of the scheme enhanced the performance of the WLAN-wired network and significantly improved the QoS for transmitted applications. The average QoS for VoIP, video, best effort traffic, and background traffic were increased from their original values by 72.5%, 70.3%, 44.5%, and 45.2% respectively.

The QoS scheme proposed allowed an end-to-end QoS to be set up. This in turn provided an improved delivery of a variety of applications in the context of wired-cum-wireless networks.

In chapter 8, a network QoS monitoring device was designed and evaluated. The existing QoS assessment devices do not determine directly the overall network QoS. Network managers have to carry out a variety of tasks to assess the overall network QoS. This makes the process complicated, expensive, and time consuming. Therefore, developing a hand-held system that accurately assesses QoS for multimedia applications is very valuable. The proposed QoS monitoring device used the QoS assessment approach which was based on regression model which is described in section 6.3.3, Chapter 6. The QoS assessment approach was implemented on the MCB2300 KEIL ARM microcontroller board to design a hand-held device that assessed QoS of multimedia applications.

The QoS parameters (delay, jitter and packet loss ratio) for multimedia applications were fed into the proposed device to determine their overall QoS. The QoS monitoring device analysed to QoS parameters of multimedia applications based on their QoS requirements, and then produced an output that reflected their overall QoS. The performance of the proposed device was compared with other QoS assessment methods. it was observed from the findings that the overall QoS values obtained from the proposed device were highly correlated with the QoS values obtained from QoS assessment using Fuzzy Inference System introduced by (Al-Sbou et al, 2006), and Neural Network QoS monitoring approach proposed by (Dogman et al, 2012_b). The obtained results indicated the effectiveness of the developed device in monitoring multimedia QoS accurately.

9.2 Future Work

Although many solutions to improve multimedia network operation were developed in this study, there remains several ongoing research and development follow ups. These include:

- **Implementation of the Proposed Approaches in Physical Networks:** The execution and validation of the proposed approaches were carried out by simulations in this study. The implementations these approaches in physical networks can further demonstrate their effectiveness.
- **Incorporating QoS into Call Admission Control (CAC):** The overall assessed QoS should be used to manage the utilisation of available network resources. The value of QoS can be integrated into CAC algorithm in future studies. The purpose of

CAC is to determine whether a new traffic should be admitted into the network. This operation depends on many factors. The factors could include the overall QoS, QoS requirements for new traffic, and the state of the network. Consequently, the QoS requirements of admitted traffic will be satisfied.

- **Proposed Adaptive Sampling Techniques and QoS Evaluation Methods over other Packet Networks:** Although the focus of this thesis was on wired-wireless networks, the sampling approaches and the QoS evaluation methods can be applied to other networks such as Mobile Ad hoc Networks (MANETs). This is because adaptive sampling and QoS evaluation approaches are based on the traffic QoS requirements, and measured QoS parameters.
- **Implementation of Adaptive Statistical Sampling Algorithm into Hardware:** Investigating how an adaptive statistical sampling approach can be implemented into hardware as a System-on-Chip (SoC) is another area of further development. The chip could be integrated into network monitoring points such as routers in order to sample transmitted traffic.
- **Extend the Investigation of Queuing Mechanism:** The impact of queuing mechanisms on application's QoS is a source of further studies. There are many queuing mechanisms. The advantages of these mechanisms, and their effects on traditional and real time applications should be studied further.

Finally, this thesis contributed significantly to the field of QoS management in multimedia computer networks. The developed techniques drew a realistic scenario about the process of managing QoS in multimedia networks. Also, they provided a firm basis and useful insights on how to effectively design future QoS management solutions.

References

Abdel-Hady, M. and Ward, R. (2007), A Framework for Evaluating Video Transmission over Wireless Ad Hoc Networks, *IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM)*, pp. 78 - 81.

Abdul Aziz, S. and Parthiban, J. (2006), Fuzzy Logic, [Online] Last access on 08 March 2013 at URL:
http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/sbaa/report.fuzzysets.html

Abeysekera, H., Matsuda, T., and Takine, T. (2009), Dynamic Contention Window Control Scheme in IEEE 802.11e Wireless LANs, *In IEEE 69th Vehicular Technology Conference, (VTC),* pp. 1-5.

Abraham, A. (2005) Artificial neural network, *Handbook of Measuring System Design*, John Wily and Sons, pp. 901-908.

Aho, A.V., Kernighan, B.W., and Weinberger, P.J. (1988), The AWK Programming Language, Addison-Wesley.

Akinaga, Y., Kaneda, S., Shinagawa, N., and Miura, A. (2005), A Proposal for a Mobile Communication Traffic Forecasting Method using Time Series Analysis for Multi-variant Data, *IEEE Global Telecommunications Conference (GLOBECOM)*, pp. 1119 - 1124.

Alahmadi, A. A., and Madkour, M. A. (2008), Performance Evaluation of the IEEE 802.11e EDCA Access Method, *IEEE International Conference on Innovations in Information Technology (IIT),* pp. 490-494.

Alkahtani, A., Woodward, M., and Al-Begain, K. (2003), An Overview of Quality of Service QoS and QoS Routing in Communication Networks, *In Proceedings of 4th PGNet Conference*, Liverpool, pp. 236-244.

Al-Kuwaiti, M., Kyriakopoulos, N., Hussein, S. (2007), QoS Mapping: A Framework Model for Mapping Network Loss to Application Loss, *IEEE International Conference on Signal Processing and Communications (ICSPC)*, pp. 1243-1246.

Al-Sbou, Y (2006), Quality of Service Assessment and Analysis of Wireless Multimedia Networks, PhD Thesis, Sheffield Hallam University, UK.

Al-Sbou, Y (2010), Fuzzy Logic Estimation System of Quality of Service for Multimedia Transmission, *International Journal of QoS Issues in Networking (IJQoSIN)*, Vol. 1, No. 1, pp. 1 - 9.

Al-Sbou, Y., Saatchi, R., Al-khayatt, S., and Strachan, R. (2006), Quality of Service Assessment of Multimedia Traffic over Wireless Ad hoc Networks, *International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP)*, pp. 129-133.

Al-Sbou, Y.A., Saatchi, R., Al-Khayatt, S., Strachan, R., Ayyash, M., and Saraireh, M., (2008), A Novel Quality of Service Assessment of Multimedia Traffic over Wireless Ad hoc Networks, *The Second International Conference on Next Generation Mobile Applications, Services and Technologies, (NGMAST)*, pp. 479-484.

Arshdeep, K., Amrit, K. (2012), Comparison of Mamdani-Type and Sugeno-Type Fuzzy Inference Systems for Air Conditioning System, *International Journal of Soft Computing and Engineering (IJSCE)*, Vol. 2, Issue 2, pp. 323 -325.

Bajaj, S., Breslau, L., Estrin, D., Fall, K., Floyd, S., Haldar, P., and Zappala, D. (1999), Improving Simulation for Network Research. *Defence Advanced Research Projects Agency (DARPA), Virtual Internetwork Testbed project (VINT)*, pp. 1-11.

Baldwin, R. (1999), Improving the Real-Time Performance of a Wireless Local Area Network, PhD Thesis, Virginia Polytechnic Institute and State University, Blacksburg, Virginia.

Balogh, T., and Medvecký M. (2011), Performance Evaluation of WFQ, WF2Q+ and WRR Queue Scheduling Algorithms, *The 34th International Conference on Telecommunications and Signal Processing (TSP)*, pp. 136-140.

Black, D., Blake, S., Carlson, M., Davies, E., Wang, Z., and Weiss, W. (1998), An Architecture for Differentiated Services. *Informational RFC 2475*, pp. 1-35

Box, G., and Jenkins, G, (1976) Time Series Analysis: Forecasting and Control, Holden-Day.

Braden, R., Clark, D., and Shenker, S. (1994), Integrated Services in the Internet Architecture: an Overview. *Informational RFC 1633*, pp. 1-32

Brase, C. H., and Brase, C. P. (2010), Understanding Basic Statistics, The 5th edition Brooks/Cole.

Brauer, F., Ehsan, M. S., and Kubin, G. (2008), Subjective Evaluation of Conversational Multimedia Quality in IP Networks, *IEEE 10th Workshop on Multimedia Signal Processing*, pp. 872-876.

Brekne, T., Clemetsen, M., Heegaard, P., Ingvaldsen, T., and Viken, B. (2002), State of the Art in Performance Monitoring and Measurements, pp. 1-96.

Čaky, P., Klimo, M., Paluch, P., and Škvarek, O. (2006), End-to-End VoIP Quality Measurement, *Acta Electrotechnica et Informatica*, Vol. 6, No. 1, pp. 1-5.

Caro, G. (2003), Analysis of Simulation Environments for Mobile Ad hoc Networks, *Microsoft Academic Search, Technical Report*. No. IDSIA-24-03.

Carvalho, P., Lima S. R., Ferreira, A., Freitas, E., and Leit˜ao, F. (2009), Providing Cost-Effective QoS Monitoring in Multiservice Networks, *Next Generation Internet Networks (NGI)*, pp. 1 - 8.

Chaabane, S., Sayadi, M., Fnaiech, F., and Brassart, E. (2008), Colour Image Segmentation using Automatic Threshold and the Fuzzy C-Means Techniques, *The 14th IEEE Mediterranean Electrotechnical Conference (MELECON)*, pp. 857 - 861.

Chan, A., Zeng, K., Mohapatra, P., Lee, S., and Banerjee, S. (2010), Metrics for Evaluating Video Streaming Quality in Lossy IEEE 802.11 Wireless Networks, *IEEE INFOCOM*, pp. 1 - 9

Chatterjee, S., and Hadi, A. (2006), Regression Analysis by Example, 4th edition. John Wiley & Sons, Inc, pp. 53-84.

Chen, B., Hu, J., Duan, L., and Gu, Y. (2009), Network Administrator Assistance System Based on Fuzzy C-Means Analysis, *Journal of Advanced Computational Intelligence and Intelligent Informatics,* pp. 91-96.

Cheong, F., and Lai, R. (1999), QoS Specfication and Mapping for Distributed Multimedia Systems: A Survey of Issues. *The Journal of Systems and Software*, Vol. 45, Issue. 2, pp. 127-139.

Chung and Claypool (2004) NS By Example, [Online] Last access at 15 February 2010 at URL: http://nile.wpi.edu/NS/

Cirstea, M., Dinu, A., Khor, J., and McCormick, M. (2002), Neural and Fuzzy Logic Control of Drives and Power Systems. Oxford.

Claffy, K.C., Polyzos, G.C., and Braun, H.W. (1993), Application of Sampling Methodologies to Network Traffic Characterization, *Conference Proceedings on Communications Architectures, Protocols and Applications*, pp. 194-203.

Cranley, N., and Davis, M. (2005), Performance Evaluation of Video Streaming with Background Traffic over IEEE 802.11 WLAN Networks, *In Proceedings of the 1st ACM Workshop on Wireless Multimedia Networking and Performance Modeling,* New York, NY, USA, ACM, pp. 131-139.

Crawley, E., Nair, R., Rajagopalan, B., and Sandick, H. (1998), A Framework for QoS-Based Routing in the Internet, *RFC2386,* pp. 1-37.

D'Antonio, S., Esposito, M., Gargiulo, M., Romano, S., and Ventre, G. (2003), A Component-Based Approach to SLA Monitoring in Premium IP Networks, *In First International Workshop on Inter-Domain Performance and Simulation IPS,* Salzburg, Austria, pp. 1-8.

Dogman, A., Saatchi, R., and Al-Khayatt, S. (2010_a), An Adaptive Statistical Sampling Technique for Computer Network Traffic, *In Proceedings of the 7^{th} IEEE, IET International Symposium on Communication Systems Networks and Digital Signal Processing (CSNDSP10)*, 21^{st} -23^{rd} July, University of Northumbria, Newcastle, UK, pp. 479 - 483.

Dogman, A., Saatchi, R., and Al-Khayatt, S. (2010_b), Evaluation of Adaptive Statistical Sampling versus Random Sampling for Video Traffic, *In Proceedings of the 11^{th} International Arab Conference on Information Technology (ACIT10),* 14^{th}-16^{th} December, University of Benghazi, Benghazi, Libya.

Dogman, A., Saatchi, R., Nwaizu, H., and Al-Khayatt, S. (2011), Adaptive Statistical Sampling of VoIP Traffic in WLAN and Wired Networks using Fuzzy Inference System, *In Proceedings of the 7^{th} IEEE International Wireless Communications and Mobile Computing Conference (IWCMC11),* 4^{th} - 8^{th} July, Bahcesehir University, Istanbul, Turkey, pp. 1731 - 1736.

Dogman, A., Saatchi, R., and Al-Khayatt, S. (2012_a), Quality of Service Evaluation using a Combination of Fuzzy C-Means and Regression Model, *International Journal of Electronics and Electrical Engineering, World Academy of Science Engineering and Technology (WASET12)*, Vol. 6, pp. 58 - 65.

Dogman, A., Saatchi, R., and Al-Khayatt, S. (2012_b), Evaluation of Computer Network Quality of Service Using Neural Networks, *In Proceedings of IEEE Symposium on Business, Engineering and Industrial Applications (ISBEIA12),* 23rd - 26th September, Bandung, Indonesia, pp. 217 - 222.

Dogman, A., Saatchi, R., and Al-Khayatt, S. (2012_c). Improving Quality of Service in IEEE 802.11e Enhanced Distributed Channel Access Protocol, *In Proceedings of the 8th IEEE International Symposium on Communication Systems Networks and Digital Signal Processing (CSNDSP12),* 18th - 20th July, Poznan University, Poznan, Poland, pp. 1 - 6.

Dogman, A., Saatchi, R., and Al-Khayatt, S. (2012_d), Computer Network Quality of Service Monitoring Using KEIL ARM Microcontroller, *In Proceedings of Regional Annual Fundamental Science Symposium (RAFSS12),* 10th - 13th December, Persada Johor International Convention Centre, Johor Bahru, Malaysia.

Dogman, A., Saatchi, R., and Al-Khayatt, S. (2013), Network Quality of Service Assessment Implementation Using a Microcontroller Board, *Malaysian Journal of Fundamental and Applied Sciences*, Vol.9, No.2, pp. 57-61.

Eberhart, R. C., and Dobbins, R. W., (1990) Neural Network PC Tools: A Practical Guide, *Academic Press*, ISBN 0-12-228640-5.

Elashheb, A. (2012) Performance Evaluation of AODV and DSDV Routing Protocol in wireless sensor network Environment, International Conference on Computer Networks and Communication Systems (CNCS), pp 55 -62

Epiphaniou, G., Maple, C., Sant, P., and Reeve, M. (2010), Affects of queuing mechanisms on RTP traffic: comparative analysis of jitter, end-to-end delay and packet loss. The 10th IEEE *International Conference on Availability, Reliability, and Security ARES'10,* pp. 33-40.

Fall, K., and Varadhan, K. (2011) The NS Manual, Virtual Internetwork Testbed project (VINT).

Farrel, A. (2008) *Network quality of service. know it all,* Pearson Education, London, UK.

Floyd, S., Allman, M. (2008) Comments on the Usefulness of Simple Best-Effort Traffic, Informational RFC 5290, pp 1 - 19

Fraleigh, C., Tobagi, F. and Diot, C. (2003) Provisioning IP backbone networks to support latency sensitive traffic, *In Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies IEEE INFOCOM.* pp. 375-385.

Fraley C, and Raftery, A. (1998). "How Many Clusters? Which Clustering Method? – Answers via Model-based Cluster Analysis." Computer Journal, 41, pp 578–588.

Frantti, T., and Jutila, M. (2009), Embedded fuzzy expert system for adaptive weighted fair queuing. *Expert Systems with Applications*, *36*(8), 11390-11397

Gan, Y., Zhang, Y. and Qian, D. (2009) Adaptive sampling measurement for high speed network traffic flow, *Proceedings of the 5th International Conference on Wireless Communications, Networking and Mobile Computing IEEE Press*, pp. 4085-4088.

Georgoulas, S., Trimintzios, P. and Pavlou, G. (2004) Admission Control Placement in Differentiated Services Networks, Ninth International Symposium on Computers and Communications (ISCC), pp 816 - 821

Giertl, J., Baca, J., Jakab, F. and Andoga, R. (2008) Adaptive sampling in measuring traffic parameters in a computer network using a fuzzy regulator and a neural network, *Cybernetics and Systems Analysis,* vol. 44, no. 3, pp. 348-356.

Giertl, J., Jakab, F., Baca, J., Andoga, R., and Mirilovic, M. (2006) Contribution to adaptive sampling of QoS parameters in computer networks, *Acta Electrotechnica et Informatica,* vol. 6, no. 1, pp. 1-8.

GloMoSim. GloMoSim Manual (2012) [Online] last access at 25 January 2012 at URL: http://pcl.cs.ucla.edu/projects/glomosim/GloMoSimManual.html

Gozdecki, J., Jajszczyk, A. and Stankiewicz, R. (2003) Quality of service terminology in IP networks, *IEEE Communications Magazine,* vol. 41, no. 3, pp. 153-159.

Graci`a, R. S., Aparicio, A. C., and Pascual, J. D. (2008) Network performance assessment using adaptive traffic Sampling, *In proceedings of the 7th International IFIP-TC6 Networking Conference on Ad-Hoc and Sensor Networks, Wireless Networks, and Next Generation Internet*, pp 252-263

Graham, I., Donnelly, S., Martin, S., Martens, J., and Cleary, J (1998) Nonintrusive and Accurate Measurement of Unidirectional Delay and Delay Variation on the Internet, Proceeding of INET, Switzerland

Gupta, S. and Saket, R. (2011) Peformance Metric Comparison of AODV AND DSDV Routing Protocols in MANETs using NS-2, *International Journal of Research and Reviews in Applied Sciences*, Vol. 7, Issue 3, pp 339 - 350

Haman, A, and Geogranas, N (2008) Comparison of Mamdani and Sugeno Fuzzy Inference Systems for Evaluating the Quality of Experienceof Hapto-Audio-Visual Applications, IEEE International Workshop on Haptic Audio Visual Environments and their Applications (HAVE), pp 87 - 92

Han, J and Kamber, M (2006) Data Mining: Concepts and Techniques, 2nd edition.

Hasegawa, G., Matsuo, T., Murata, M., and Miyahara, H. (2002) Comparisons of packet scheduling algorithms for fair service among connections on the Internet, *Journal of High Speed Networks*, *12*(1), 1-27.

Haykin, S. (1999) Neural Networks: A Comprehensive Foundation, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall.

He, Y., Ma, X., (2011). Deterministic backoff: Towards efficient polling for 802.11e HCCA in wireless home networks, *IEEE Transactions on Mobile Computing* 10 (12) pp 1726 - 1740.

Heckmann, O., Pandit, K., Schmitt, J., Hoffmann, M., and Jobmann, M. (2002) LETSQoS milestone 2, network QoS , *LETSQoS project, Darmstadt University of Technology and Technical University Munich,* pp. 1-106.

Hernandez, E.A., Chidester, M.C. and George, A.D. (2001) Adaptive sampling for network management, *Journal of Network and Systems Management,* vol. 9, no. 4, pp. 409-434.

Hoang, D., Kumar, R., and Panda, S. (2010) Fuzzy C-means clustering protocol for wireless sensor networks, *IEEE International Symposium on Industrial Electronics ISIE*, pp 3477-3482.

IEEE Computer Society LAN/MAN Standards Committee (1999), Wireless LAN medium access control (MAC) and physical layer (PHY) specifications.

IEEE Computer Society LAN/MAN Standards Committee (2005), Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements.

ITU-T E.800 (1994) Terms and definitions related to quality of service and network performance including dependability, *Telecommunication Standardization Sector of ITU,* pp. 1-57.

ITU-T P.910 (2008) Subjective video quality assessment methods for multimedia applications, *Telecommunication Standardization Sector of ITU.* pp. 1-42.

ITU-T P.911 (1998) Subjective audiovisual quality assessment methods for multimedia applications, *Telecommunication Standardization Sector of ITU.* pp. 1-27.

ITU-T, G.1010 (2001) End-user multimedia QoS categories, Telecommunication Standardization Sector of ITU. pp. 1-18.

Jain, R. (1991) The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modelling. Wiley- New York, ISBN: 0471503363.

Jain, R. (2008) A Survey of Network Simulation Tools: Current Status and Future Developments, [Online] at URL: http://www.cse.wustl.edu/~jain/cse567-08/ftp/simtools

Jantzen, J., (1998) Design of Fuzzy Controllers. Technical report, University of Denmark, Department of Automation, Vol: 236.

Karlik, B and Olgac A., (2010) Performance analysis of various activation functions in generalized mlp architectures of neural networks. *International Journal of Artificial Intelligence and Expert Systems*, Vol. 1, Issue 4., pp 111–122.

Ke,C.H., Shieh, C.K., Hwang, W.S., and Ziviani, A (2008) An Evaluation Framework for More Realistic Simulations of MPEG Video Transmission, *Journal of Information Science and Engineering*, Vol.24, No.1, pp 425-440

Khoukhi, L., and Cherkaoui, S. (2008) Experimenting with Fuzzy Logic for QoS Management in Mobile Ad Hoc Networks. IJCSNS International Journal of Computer Science and Network Security, VOL.8 No.8, P 372-386

Kiziloren, T., and Germen, E. (2007) Network traffic classification with self organizing maps, *International Symposium on Computer and Information Sciences ISCIS,* pp. 1-5.

Klir, J. and Yuan, B. (1995) Fuzzy Sets and Fuzzy Logic: Theory and Applications. Prentice Hall Inc. ISBN: 0131011715.

Kohonen, T. (1982) Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, pp: 59-69.

Koutsakis, P. (2009) Dynamic versus Static Traffic Policing: A New Approach for Videoconference Traffic over Wireless Cellular Network, IEEE TRANSACTIONS ON MOBILE COMPUTING, VOL. 8, NO. 9, pp 1153-1166

Kumar, D., Ryu, Y. and Jang, H. (2008) Quality of service (QoS) of voice over MAC protocol 802.11 using NS-2.ACM, International Multimedia Conference, P. 39-44. ISBN:978-1-60558-319-8.

Kurose, J. and Ross, K. (2005) *Computer networking a top-down approach,* Fifth edition, Addison-Wesley, Reading.

Lavington, S., Hagras, H., and Dewhurst, N. (1999) Using a MLP to predict packet loss during real-time video transmission, *Tech Report, University of Essex,* pp 1-10.

Lee, K. Y., Cho, K. S., and Ryu, W. (2011), Efficient QoS Scheduling Algorithm for Multimedia Services in IEEE 802.11e WLAN, *IEEE Vehicular Technology Conference (VTC Fall),* pp. 1-6).

Lei, X., Yifei, G., Sheng, L., Wei, H., and Di, X (2012) Research on the Improved FCM Cluster Method in the Hotspots Analysis on Web, The 2nd IEEE International Conference on Instrumentation, Measurement, Computer, Communication and Control (IMCCC), pp 478 - 481.

Lekcharoen S. (2007) Performance and Evaluation of Adaptive Backoff Schemes in Traffic Shaping over High Speed Network, Proceedings of Asia-Pacific Conference on Communications, pp 241 -245

Liang, H. M., Ke, C. H., Shieh, C. K., Hwang, W. S., and Chilamkurti, N. K. (2006) "Performance evaluation of 802.11e EDCF in infrastructure mode with real audio/video traffic", IEEE International Conference on Networking and Services ICNS, pp. 92-97.

Lin, C. H., Shieh, C. K., Ke, C. H., Chilamkurti, N. K., and Zeadally, S. (2009), An adaptive cross-layer mapping algorithm for MPEG-4 video transmission over IEEE 802.11 e WLAN. *Telecommunication Systems*, *42*(3-4), 223-234.

Lindfield, G. R., and Penny, J. E. T. (2012) Numerical methods: using MATLAB, *Elsevier Science & Technology.*

Liu, L., Jin, X., Min, G., and Xu, L. (2012), Real-Time Diagnosis of Network Anomaly Based on Statistical Traffic Analysis. In *IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom),* pp. 264-270.

Ljung, L. (1999) System Identification: Thoery for the User, 2nd edition, Prentice Hall.

Lucio et al (2003) OPNET Modeler and Ns-2: Comparing The Accuracy of Network Simulators for Packet-Level Analysis Using a Network Testbed. International Conference on Simulation, Modelling and Optimization (ICOSMO). P 1-8.

Ma, W., Huang, C. and Yan, J. (2004) Adaptive sampling for network performance measurement under voice traffic, *IEEE International Conference on Communications,* vol. 2, pp. 1129-1134.

Ma, W., Yan, J. and Huang, C. (2003) Adaptive sampling methods for network performance metrics measurement and evaluation in MPLS-based IP networks, *In Proceedings of the Canadian Conference on Electrical and Computer Engineering Citeseer*, pp. 1005–1008.

Malan, G., and Jahanian, F. (1998) An extensible probe architecture for network protocol performance measurement, *ACM SIGCOMM Computer Communication Review,* vol. 28, no. 4, pp. 215-227, 1998.

Malhotra, M., Varma, D. P., Pramod, P. J., and Jain, D. K. (2011), StreamTrack: A tool for QoS monitoring of multimedia sessions. The *6^{th} IEEE International Conference on* In *Telecommunication Systems, Services, and Applications (TSSA),* pp. 121-125.

Markopoulou, A., Tobagi, F., and Karam, M. (2003) Assessing the Quality of Voice Communications over Internet Backbones, IEEE/ACM TRANSACTIONS ON NETWORKING, VOL. 11, NO. 5, pp 747 - 760

Mathworks ($2012_{(a)}$) Fuzzy Logic Toolbox. [Online] last access on 18 February 2013 at URL: http://www.mathworks.com/access/helpdesk/help/toolbox/fuzzy

Mathworks ($2012_{(b)}$) Neural Netowrk Toolbox. [Online] last access on 08 March 2013 at URL: http://www.mathworks.co.uk/help/nnet/index.html

MATLAB, The Language of Technical Computing (2012) [Online] last access at 17 March 2012 at URL: http://www.mathworks.co.uk/products/matlab/

MDK-ARM Microcontroller Development Kit [Online] last access on 15 September 2013 at URL: http://www.keil.com/arm/mdk.asp.

Menth, M., Martin, R. and Charzinski, J. (2006) Capacity over provisioning for networks with resilience requirements, *In Proceedings of the 2006 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, New York, NY, USA, ACM*, pp. 87-98.

Miaji, Y., and Hassan, S. (2010), Comparative Simulation of Scheduling Mechanism in Packet Switching Network. The 2^{nd} IEEE *International Conference on Network Applications Protocols and Services (NETAPPS),* pp. 141-147.

Mishra, R., and Sharma, V. (2003) QoS routing in MPLS networks using active measurements, *IEEE Conference on Convergent Technologies for Asia-Pacific Region (TENCON),* Citeseer, pp. 323- 327.

Mishra, R., and Sharma, V. (2003) QoS routing in MPLS networks using active measurements, *IEEE Conference on Convergent Technologies for Asia-Pacific Region (TENCON),* Citeseer, pp. 323- 327.

Mohamed, S. (2002) A study of real-time packet video quality using random neural networks, *IEEE Transactions on Circuits and Systems for Video Technology,* Vol.12, No.12, pp 1071-1083.

Mohamed, S., Rubino, G., Cervantes, F. and Afifi, H. (2001) Real-time video quality assessment in packet networks: A neural network model, *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1-21

Molina-Jimenez, C., Shrivastava, S., Crowcroft, J. and Gevros, P. (2004) On the monitoring of contractual service level agreements, *In First IEEE International Workshop on Electronic Contracting*. pp.1-8.

Muyeen, S. M., and Al-Durra, A (2013) Modeling and Control Strategies of Fuzzy Logic Controlled Inverter System for Grid Interconnected Variable Speed Wind Generator, IEEE System Journal, issue 99, pp 1-8.

Naoum-Sawaya, J., Ghaddar, B. (2005) A Fuzzy Logic Approach for Adjusting The Contention Window Size in IEEE 802.11e Wireless Ad Hoc Networks. International Symposium on Communication, Control, and Single Processing (ISCCSP). P 1-4

Nascimento, S., Mirkin, B., and. Moura-Pires, F (2000) A fuzzy clustering model of data and fuzzy c-means, IEEE International Conference on Fuzzy Systems, vol. 1, pp. 302–307.

Nogueira, A., Salvador, P., and Valadas, R. (2006) Predicting the quality of service of wireless LANs using neural networks, Proceedings of the 9th ACM International Symposium on Modeling Analysis and Simulation of Wireless and Mobile Systems ACM, New York, NY, USA, pp. 52-60.

Nortal Networks (2003) Introduction to quality of service, White Paper, pp. 1-14.

NS Network Simulator 2 (2012) [Online] last access at 17 January 2012 at URL: http://www.isi.edu/nsnam/ns/

Ollos, G., and Vida, R. (2009) Adaptive regression algorithm for distributed dynamic clustering in wireless sensor networks, *IEEE International Federation for Information Processing IFIP*, pp 1-5.

OMNeT++ (2012) [Online] last access at 03 February 2012 at URL: http://www.omnetpp.org/

Pal,T., Shekhar, C., and Sharma, H. D. (2009) "Design and Implementation of Embedded Speed Controller on ARM for Micro-manufacturing Applications," *IEEE International Conference on Advances in Computing, Control, and Telecommunication Technologies*, pp. 406 - 410.

OPNET (2012) OPNET Technologies [Online] last access at 10 February 2012 at URL: http://www.opnet.com/

Palomar, D., Skehill, R., Rice, I., Picovici, D., and Mc Grath, S. (2008) Objective assessment of audio quality, *IET Signals and Systems Conference, ISSC.* Irish, pp. 37-42.

Park, S., Kim, K., Kim, D., Choi, S., and Hong, S. (2003) "Collaborative QoS architecture between DiffServ and 802.11e wireless LAN," The 57th IEEE Semi annual Vehicular Technology Conference VTC, Vol 2, pp 945-949.

Parker, J. and Hall, L. Bezdek, J. (2012) Comparison of Scalable Fuzzy Clustering Methods, IEEE World Congress on Computational Intelligence (WCCI), Brisbane, Australia, pp 1 - 9.

Patcha, A., and Park, J. M. (2006) An adaptive sampling algorithm with applications to denial-of-service attack detection, *15th IEEE International Conference on Computer Communications and Networks, ICCCN,* pp. 11-16

Patel, T., Ogale, V., Baek, S., Cui, N. and Park, R. (2003) Capacity estimation of VoIP channels on wireless networks, *Dept of Electrical and Computer Engineering, The University of Texas at Austin*, pp. 1-37.

Peng, F. Peng, B. and Qian, D. (2010) "Performance analysis of IEEE 802.11e enhanced distributed channel access", IET Communications, Vol. 4, Issue. 6, pp. 728–738.

Pias, M., and Wilbur, S. (2001) "EdgeMeter distributed network metering model", *IEEE Global Telecommunications Conference, GLOBECOM,* pp.2517-2521.

Politis, A. Mavridis, I. Manitsaris, A. and Hilas, C. (2011) "X-EDCA: A cross-layer MAC-centric mechanism for efficient multimedia transmission in congested IEEE 802.11e infrastructure networks" IEEE International Wireless Communications and Mobile Computing Conference IWCMC, pp. 1724-1730.

Priestly, M., (1988) Non Linear and Non Stationary Time Series Analysis, Academic Press.

Radhakrishnan, K., and Larijani, H. (2011) Evaluating perceived voice quality on packet networks using different random neural network architectures, *Performance Evaluation,* Vol. 68, pp 347-360.

Rashid, M. M., Hossain, E., and Bhargava, V. K. (2007), HCCA scheduler design for guaranteed QoS in IEEE 802.11e based WLANs, *IEEE Wireless Communications and Networking Conference, (WCNC),* pp. 1538-1543.

Rauf, B., Amjad, M.F. and Ahmed, K. (2009) "Performance evaluation of IEEE 802.11 DCF in comparison with IEEE 802.11e EDCA", IEEE International Conference for Internet Technology and Secured Transactions, ICITST, pp. 1-6.

Rokach, L. and Maimon, O (2005) Clustering Methods, Data Mining and knowledge discovery handbook, pp 321 - 352.

Ross (1995) Fuzzy Logic with Engineering Applications. McGraw-Hill, USA. ISBN 007-113637-1

S. Park, S. Kim, K., Kim, D., Choi, S., and Hong, S. (2003) "Collaborative QoS architecture between DiffServ and 802.11e wireless LAN", The 57th IEEE Semi annual Vehicular Technology Conference VTC, Vol 2, pp. 945-949.

Said, A., and R. Saatchi, R. (2009) A Performance Analusis of AODV AND DSR Routing Protocols in Mobile Wireless Networks, *The Mediterranean Journal of Computers and Networks,* Vol. 5, No. 1, pp 17 - 26

Saliba, A.J., Beresford, M.A., Ivanovich, M. and Fitzpatrick, P. (2005) User-perceived quality of service in wireless data networks, *Personal and Ubiquitous Computing,* vol. 9, no. 6, pp. 413-422.

Saraireh, M. (2006) Medium access control mechanisms for quality of service in wireless computer networks, *PhD Thesis, Sheffield Hallam University,* UK.

Saraireh, M. Saatchi, R. Al-Khayatt, S. and Strachan, R. (2008) "Assessment and improvement of quality of service using fuzzy logic and hybrid genetic-fuzzy approaches", *ACM.*, vol 27, issue 2-3, P.95 – 111, ISSN:0269-2821.

Saraireh, M., Saatchi, R., Al-Khayatt, S., and Strachan, R. (2007) Assessment and improvement of quality of service using fuzzy logic and hybrid genetic-fuzzy approaches, ACM., vol. 27, issue 2-3, pp. 95 – 111.

Sarkar, N, and Halim, S (2011) A Review of Simulation of Telecommunication Networks: Simulators, Classification, Comparison, Methodologies, and Recommendations, Journal of Selected Areas in Telecommunications (JSAT), pp 10 - 17

Semeria, C. (2001) "Supporting differentiated service classes: queue scheduling disciplines", White Paper Juniper Networks, pp. 1-27, 2001.

Senkindu, S. and Chan, H.A. (2008) "Enabling end-to-end quality of service in a WLAN-wired network," IEEE Military Communications Conference, MILCOM, pp.1-7.

Shah, I. (2001) Brining comprehensive quality of service capabilities to next generation networks, white paper, pp 1-20

Shah, S., Khandre, A., Shirole, M., and Bhole, G. (2008) Performance Evaluation of Ad Hoc Routing Protocols Using NS2 Simulation", Conference of Mobile and Pervasive Computing (CMPC), pp 167 - 171

Siraj, S., Gupta, A., and Badgujar, R (2012) Network Simulation Tools Survey, International Journal of Advanced Research in Computer and Communication Engineering, Vol. 1, Issue 4, pp 201 - 210

Skyrianoglou, D., Passas, N., Salkintzis, A. and Zervas, E. (2002) "A generic adaption layer for differentiated services and improved performance in wireless networks", The 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications PIMRC, Vol 3, pp. 1141-1145.

Sun, L. and Ifeachor, E. (2002) Perceived speech quality prediction for voice over IP-based networks, *In IEEE International Conference on Communications, ICC,* pp. 2573-2577.

Sweet, S. and Grace-Martin, K. (2010) Data Analysis with SPSS: A First Course in Applied Statistics, 4th edition.

Szigeti, T., and Hatting, C. (2005). End-to-end qos network design. Cisco Systems.

Teyeb, O., Sørensen, T., Mogensen, P., and Wigard, J. (2006) Subjective evaluation of packet service performance in UMTS and heterogeneous networks, *The 2nd ACM International Workshop on Quality of Service and Security for Wireless and Mobile Networks ACM*, New York, NY, USA, pp. 95-102.

Timo, L., Hannu, K., and Tapani, H. (2002) Profiling network applications with fuzzy C-means clustering and SOM, *International Conference on Fuzzy Systems and Knowledge Discovery*, pp 1-5.

Ting, B., Yong, W., and Xiaoling, T. (2010) Network traffic classification based on kernel self-organizing maps, *International Conference on Intelligent Computing and Integrated Systems ICISS*, pp. 310 - 314.

Thangaraj, S., Gummadi, S., and Radhakrishnan, S. (2006) "Enhancement in ARM Code Optimization for Memory Constrained Embedded Systems," *IEEE Advanced Computing and Communications*, pp. 483 - 486.

Vandaele, W., (1983) Applied Time Series and Box-Jenkins Models, Academic Press.

Vesanto, J., Himberg, J., Alhoniemi, E., and, Parhankangas, J. (1999) Self–organizing map in matlab: the SOM toolbox, *In Proceedings of Matlab DSP Conference*, Espoo, Finland, pp. 35–40.

Villalón, J., Cuenca, P., and Orozco-Barbosa, L. (2007), on the capabilities of IEEE 802.11e for multimedia communications over heterogeneous 802.11/802.11e WLANs, *Telecommunication Systems*, *36*(1-3), 27-38.

Wang, P.Y., Yemini, Y., Florissi, D., Zinky, J. and Florissi, P. (2000) Experimental QoS performances of multimedia applications, *IEEE INFOCOM, Citeseer,* pp. 970-979.

Wang, T., Liang, Z., and Zhao, C. (2009) A detection method for routing attacks of wireless sensor network based on fuzzy C-means clustering, I*EEE Sixth International Conference on Fuzzy Systems and Knowledge Discovery FSKD*, pp445-449.

Weingärtner, E., Lehn, H., and Wehrle, K. (2009) A performance comparison of recent network simulators, IEEE International Conference on Communications, ICC, pp 1 - 5

Xia W., and Wang R. (2010) Prediction Model of Network Security Situation Based on Regression Analysis, IEEE International Conference on Wireless Communications, Networking and Information Security (WCNIS). pp 616 - 619.

Xie, X. and Beni, G. G. (1991) A validity measure for fuzzy clustering, *IEEE Pattern Analysis and Machine Intelligence*, pp 841-847.

Xin, Q., Hong, L., and Fang, L. (2009). A modified FLC adaptive sampling method, *In Proceedings of IEEE International Conference on Communications and Mobile Computing (CMC09)*, 6th - 8th Januarey, China University of Petroleum, China, pp 515-520.

Yu, Z., and Meng, F. (2009) An improved algorithm for packet fair queuing scheduling, 2nd IEEE International Conference on Computer Science and Information Technology, ICCSIT, pp 153-157

YUV QCIF Reference Videos. (2012, January 7) [Online]. Available at: http://www.tkn.tu-berlin.de/research/evalvid/qcif.html

Zadeh, L. (1965) Fuzzy sets, in Information and Control, vol. 8, Pages: 338-353.

Zaknich, A. (2003) Neural networks for intelligent signal processing, World Scientific Pub Co Inc, New Jersey.

Zhai, H., Chen, X. and Fang, Y. (2005) How well can the IEEE 802.11 wireless LAN support quality of service?, *IEEE Transactions on Wireless Communications,* vol. 4, no. 6, pp. 3084-3094.

Zhang, L. and Zhang, T. (2007) Small packet threshold adaptive sampling for network management, *In International Conference on Mechatronics and Automation ICMA*. pp. 811-815.

Zhou, R. and Sik-Jang, K. (2008) Adaptive MPEG-4 Video Streaming over IP networks, the 23 International Technical Conference on Circuits, Systems, Computer, and Communication, (ITCCSCC), pp 637 - 640.

Zseby, T. (2002) Deployment of sampling methods for SLA validation with non-intrusive measurements, *Pass and Active Measurement workshop* (PAM), USA, pp.84-94.

Zseby, T. (2004) Comparison of sampling methods for non-intrusive SLA validation, *Second Workshop on End-to-End Monitoring Techniques and Services E2EMON,* pp. 1-8.

Zseby, T. and Scheiner, F. (2004) QoS Monitoring and Measurement Benchmarking, Information Society Technology. [Online] Last accessed on 06 April 2013 at URL: www.ist-mome.org/documents/M_and_M_benchmark.pdf