ISSN 1823-626X

Malaysian Journal of Fundamental and Applied Sciences

available online at http://mjfas.ibnusina.utm.my

# Network Quality of Service Assessment Implementation Using a Microcontroller Board

Aboagela Dogman, Reza Saatchi and Samir Al-Khayatt

*Faculty of Art, Computing, Engineering and Sciences, Sheffield Hallam University, Sheffield, United Kingdom*

Received 6 December 2012, Revised 25 February 2013, Accepted 28 February 2013, Available online 1 March 2013

## ABSTRACT

Network Quality of Service (QoS) assessment plays an important role in managing network resources and ensuring that various applications receive an appropriate prioritisation or sufficient resources. Therefore, developing a portable hand-held device that accurately assesses QoS for different applications is very valuable. In this study, a QoS monitoring system was designed and evaluated. The QoS assessment was based on regression modelling that was implemented on the MCB2300 KEIL ARM microcontroller board. The system analysed the QoS requirements (delay, jitter and packet loss ratio) of multimedia applications to determine their overall QoS. The QoS assessment results were compared with other assessment methods (e.g. Fuzzy Inference System and Neural Network based QoS monitoring). The study indicated the effectiveness of the developed system in monitoring multimedia QoS accurately.

| Quality of Service Assessment | Multimedia Networks | KEIL ARM MCB2300 |

http://dx.doi.org/10.11113/mjfas.v9n2.83

## 1. INTRODUCTION

The growth in multimedia networks has led to the need for finding efficient ways to monitor Quality of Service (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. The latter approach guarantees QoS.

In this study, a mechanism that assesses QoS is implemented on a portable microprocessor board. The use of embedded systems in recent years has increased in such a way that impinged into various mobile and multimedia applications. An efficient environment for microcontroller applications is KEIL MDK-ARM [1]. 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.

The contribution of this paper is the implementation of the regression modelling to assess QoS that was reported in [2] on the MCB2300 KEIL ARM microcontroller board.

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 [3].

In [4], a new algorithm was proposed to reduce the KEIL ARM memory requirement needed by an application program. The proposed algorithm enhanced ARM microprocessor performance.

The remainder of this paper is organized as follows : in Section 2 the theory of the regression model that was used to assess QoS is explained. Section 3 outlines how the QoS assessment technique was devised based on regression modelling and its implementation on the MCB2300 KEIL ARM microcontroller board. Section 4 describes the process of modelling the transmitted traffics over the simulated network. The results are discussed in section 5. Finally, the conclusions are presented in Section 6.

## 2. THEORETICAL BACKGROUND

### 2.1 Regression Models

Regression models provide an exploration of the relationships between several variables. One variable is the dependent or response variable and the others are independent or descriptive variables [5]. Equation (1) defines the regression model between the dependent variable ($y$) and independent variables ($x_1, x_2, \ldots., x_n$),

$$y = b_0 + b_1x_1 + b_2x_2 + \cdots + b_nx_n + e \qquad (1)$$

In matrix notation, the regression model can be

*Corresponding author. E-mail: aadogman@my.shu.ac.uk
Tel: (+44)1142256289 , Fax: (+44)1142256289

written as in equation (2).

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_{11} & \cdots & x_{k1} \\ 1 & x_{12} & & x_{k2} \\ \vdots & & \ddots & \vdots \\ 1 & x_{1n} & \cdots & x_{kn} \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_n \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{bmatrix} \qquad (2)$$

Or $\mathbf{Y=Xb + e}$

where $\boldsymbol{b} = \{b_0, b_1, b_2, \ldots b_n\}$ is a vector of the regression coefficients that are determined from recorded data and $\boldsymbol{e}$ is a vector of $n$ error terms. The regression coefficients are calculated using equation (3).

$$\boldsymbol{b} = (\boldsymbol{X^T X})^{-1}\boldsymbol{X^T Y} \qquad (3)$$

The vector of residual (i.e. error terms) is given by equation (4).

$$\boldsymbol{e} = \boldsymbol{Y} - \boldsymbol{Xb} \qquad (4)$$

In order for the regression model to be valid as an accurate predictor, there are some assumptions that need to be met[6]. These are: $\boldsymbol{X}$ and $\boldsymbol{Y}$ need to be linearly related, $\boldsymbol{X}$ needs to be non-stochastic and measure without error, the error needs to be independent and normally distributed.

The prediction accuracy of regression model can be measured using correlation coefficient. The magnitude of correlation coefficient ($R$) is between 0 and 1. The values close to 1 indicate a high correlation, whereas values less than 0.5 indicate a low correlation. The correlation between the actual values ($y_A$) and predicted values ($y_P$) can be calculated using equation (5).

$$R_{(y_A\ ,\ y_P)} = \frac{\sum(y_{Ai} - \overline{y_A})(y_{Pi} - \overline{y_P})}{\sqrt{\sum(y_{Ai} - \overline{y_A})^2 \sum(y_{Pi} - \overline{y_P})^2}} \qquad (5)$$

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.

## 2.2 KEIL MCB2300 Evaluation Board

The hardware components and interfaces of the MCB2300 board are shown in Figure (1). The connectors on the MCB2300 board provide an easy access to the on-chip peripherals [1]. The MCB2300 Evaluation Kit uses the µVision IDE software tool.

The MCB2300 Evaluation Board is a USB-powered device. The MCB2300 Evaluation Board can be connected directly to the KEIL µVision Debugger installed on the PC using the KEIL ULINK USB-JTAG Adapter, and no additional software is required on the board. 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.

The advantage of KEIL MCB2300 Evaluation Board is the ease of generating, testing, and redesigning application programs.

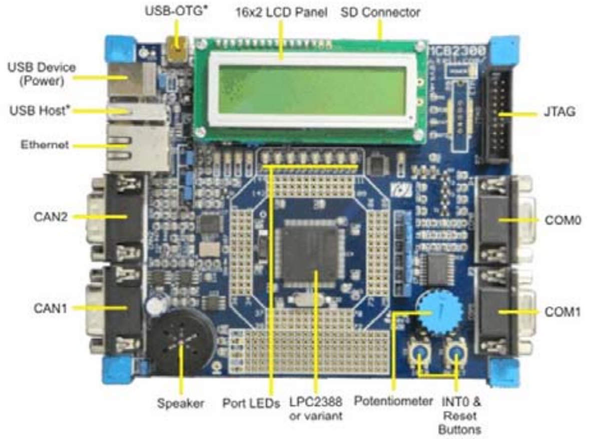

**Fig. 1.** KEIL MCB2300 Evaluation Board [1].

## 2.3 An Overview of QoS Parameters

In this study, VoIP application was considered to assess its overall QoS. VoIP is sensitive to network QoS parameters such as delay, jitter, and packet loss ratio. Table 1 indicates the QoS requirements of VoIP, where delay, jitter, and packet loss ratio requirements are listed [7].

**Table 1.** VoIP QoS requirements.

| Range | Delay(ms) | Jitter(ms) | Loss ratio (%) |
|---|---|---|---|
| Low (good quality) | 0-150 | 0-1 | 0-2 |
| Medium(average quality) | 150-400 | 1-3 | 2-4 |
| High (poor quality) | >400 | >3 | >4 |

## 3. QOS ASSESSMENT IMPLEMENTATION USING KEIL MCB2300 ARM

A schematic diagram of the proposed regression model based computer network QoS monitoring system using the KEIL ARM microcontroller board is shown in Figure 2.

The QoS parameters (i.e. delay, jitter, and packet loss ratio) were obtained from the generated trace files of the simulated network. The extracted QoS parameters were then used as inputs to KEIL MCB2300 microcontroller to quantify the overall QoS. The following subsections explain the QoS assessment technique using regression model and the digital implementation of the devised QoS assessment technique using MCB2300 KEIL ARM board microcontroller

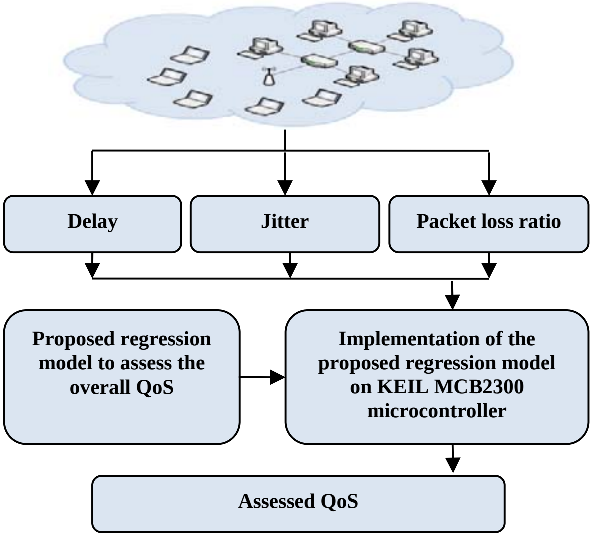
Delay
Jitter
Packet loss ratio
Proposed regression model to assess the overall QoS
Implementation of the proposed regression model on KEIL MCB2300 microcontroller
Assessed QoS

**Fig. 2.** QoS monitoring System

## 3.1 Regression Model Based QoS Assessment

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 values of QoS parameters of VoIP (i.e. delay, jitter, and packet loss ratio) were normalised in order to have the same contribution for measuring overall QoS. The normalisation process considered the limits for QoS parameters, where the overall QoS is poor. For VoIP, these limits are 600 ms for delay, 5 ms for jitter, and 6% for packet loss ratio.

The regression formula was produced based on QoS requirements listed in Table 1 in order to provide the outputs that reflected the overall QoS. The QoS parameters shown in Table I were categorized 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. The overall QoS spans between (0%-100%). Low QoS parameters (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 < delay ≤ 400 ms, 1 < jitter ≤3 ms, and 2% < packet loss ratio ≤4%) corresponded to average QoS (i.e.33% < QoS ≤ 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 ≤ 33%).

The QoS parameters and the overall QoS were then arranged in matrices 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 $D_i,\ J_i,\ PLR_i\ ,\ QoS_i,\ i = 1,2,\ldots,n$ 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 (3). The vector of residual (i.e. error terms) was then calculated using equation (4). The calculated errors were normally distributed indicating that the mean of error terms $e_i\ , i = 1,2,\ldots,n$ was zero. This implies that the estimated regression model determined was as in equation (6):

$$QoS_i = b_0 + b_1 \times D_i + b_2 \times J_i + b_3 \times PLR_i \qquad (6)$$

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

## 3.2 Implementation of QoS Assessment Method using the KEIL MCB2300 ARM Microcontroller

The code for implementing the developed QoS assessment technique using regression model was written in the C language so that it could 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 program operation was successfully verified, the MCB2300 board worked as a standalone device.

Figure 3 shows the pseudo code of the program implemented on the MCB2300 board. The QoS parameters (i.e. delay, jitter, and packet loss ratio) were inputted into the QoS monitoring device (i.e. programmed MCB2300 board) through the SD connector. The liquid crystal display of the QoS monitoring device indicated when that the SD card was inserted and data processing was initiated. The device processed the QoS parameters file, read the values of delay, jitter, and packet loss ratio, and then assessed the QoS using equation (6).

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 network QoS was displayed on the display screen of the device.

```
If (SD card is not inserted) then
{
LCD output ("No SD-Card used")
}
Else
{
LCD output ("SD-Card used")
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
Calculate QoS using equation (6)
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)
}
```

**Fig. 3.** QoS assessment code

## 4. TOPOLOY AND TRAFFIC SCENARIO CHARACTERISTICS

To validate the performance of QoS monitoring system, a wireless-cum-wired network topology illustrated in Figure 4 was simulated using the Network Simulator- 2 (ns-2) [8]. The network topology consisted of 8 wireless nodes, 2 wired nodes, and 2 base stations. 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- video-conferencing 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.

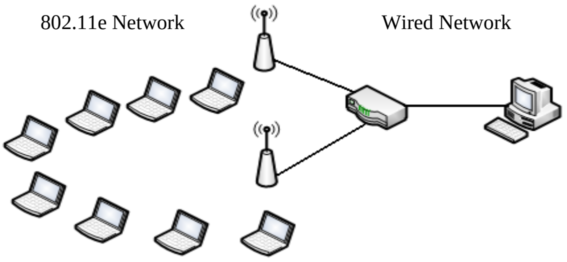

**Fig. 4.** Simulated network.

The transmitted traffic (VoIP, video-conferencing, and best effort traffic) were mapped into three access categories (ACs) to represent the three priorities levels as shown in Table 2. VoIP had the highest priority, whereas best effort traffic had the lowest priority [9]. However, due to the high sensitivity of VoIP quality to the QoS parameters as shown in Table 1, in this study, the quality of VoIP traffic was evaluated.

**Table 2.** 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 |

## 5. RESULTS AND DISCUSSION

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 KEIL MCB2300 microcontroller to quantify the overall QoS.

Table 3 shows the results from the devised QoS monitoring system. From the table, 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.

The results obtained were compared with other QoS assessment methods (i.e. the developed fuzzy inference system FIS [10], distance measure [11], and neural network [12]). The values of the correlation coefficients between QoS determined using the monitoring tool, and the other QoS assessment methods were: 0.97 (for FIS), 0.99 (for distance measure), and 0.99 (for neural network).

**Table 3. Q**oS Parameters, and the evaluated QoS using Neural Networks (NN), Fuzzy Inference System (FIS), Distance Measurement (DM), and QoS Monitoring Tool (QoS_RM).

| **QoS Parameters** | | | **Evaluated QoS** | | | |
|---|---|---|---|---|---|---|
| ***Delay (ms)*** | ***Jitter (ms)*** | ***PLR (%)*** | ***QoS_ MT*** | ***FIS*** | ***DM*** | ***NN*** |
| 11.5 | 5.0 | 0.0 | 28.1 | 9.3 | 24.6 | 26.0 |
| 16.9 | 0.8 | 0.0 | 92.5 | 89.6 | 96.2 | 96.9 |
| 16.1 | 3.1 | 2.0 | 50.7 | 47.7 | 50.5 | 46.8 |
| 17.1 | 2.9 | 2.0 | 53.7 | 51.1 | 54.3 | 52.9 |
| 117.4 | 5.0 | 5.7 | 7.7 | 9.3 | 10.6 | 11.4 |
| 286.9 | 5.0 | 6.0 | 4.0 | 9.3 | 7.9 | 7.3 |
| 24.3 | 2.5 | 2.0 | 59.7 | 54.1 | 59.0 | 61.5 |
| 105.8 | 5.0 | 6.0 | 6.9 | 9.3 | 8.2 | 8.7 |
| 422.3 | 5.0 | 6.0 | 1.8 | 9.3 | 6.2 | 4.2 |
| 600.0 | 5.0 | 6.0 | 0.01 | 9.3 | 0.0 | 1.1 |
| **Overall QoS** | | | 30.51 | 29.83 | 31.8 | 31.7 |

From the values of evaluated QoS provided in Table 3, 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 are due to the fact that each method follows a different scheme to determine QoS. However, the values of QoS obtained from the proposed monitoring device spanned between (1%-100%), whereas the range of QoS values produced by FIS was between (10%-90%). This indicates that the QoS monitoring tool could be more effective.

## 6. CONCLUSION

In this study, a portable hand-held device to assess the QoS for multimedia applications was designed and evaluated. Our developed QoS assessment technique which was based regression model published in previous work was implemented on the MCB2300 KEIL ARM microcontroller board. The proposed device 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 effectiveness of the developed system in monitoring QoS.

## REFERENCES

[1] MDK-ARM Microcontroller Development Kit [Online] last access on 15 November 2012 at URL: http://www.keil.com/arm/mdk.asp.

[2] A. Dogman, R. Saatchi and S. Al-Khayatt, International Journal of Electronics and Electrical Engineering 6 (2012) 58-65.

[3] T. Pal, C. Shekhar and H. D. Sharma, IEEE International Conference on Advances in Computing, Control, and Telecommunication Technologies, (2009) 406 - 410.

[4] S. Thangaraj, S. Gummadi, and S. Radhakrishnan, IEEE, Advanced Computing and Communications, (2006) 483 - 486.

[5] S. Chatterjee, and A. Hadi, 4th ed. John Wiley & Sons, Inc. (2006) 53-84.

[6] R. Jain, 2nd ed. John Wiley & Sons, Inc. (1991) 245-270.

[7] ITU Recommendation, G.1010 *Telecommunication Standardization Sector of ITU*. (2001) 1-18.

[8] NS Network Simulator 2. 2009 [Online] last access at 31 July 2012 at URL: http://www.isi.edu/nsnam/ns

[9] Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY)Specifications Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements, IEEE Std. 802.11e, 2005.

[10] Y. Al-Sbou, International Journal of QoS Issues in Networking (IJQoSIN), 1(2010) 1-9.

[11] Y. Al-Sbou PhD Thesis, Sheffield Hallam University, UK, (2006).

[12] A. Dogman, R. Saatchi and S. Al-Khayatt, IEEE International Symposium on Business, Engineering and Industrial Applications (2012) 217-222.