Proceedings of the World Congress on Engineering 2007 Vol I
WCE 2007, July 2 - 4, 2007, London, U.K.

Modelling BLUE Active Queue Management using Discrete-time Queue

H. Abdel-jaber
Department of Computing, University of Bradford, UK
habdelja@brad.ac.uk

M. Woodward
Department of Computing, University of Bradford, UK
m.woodward@brad.ac.uk

F. Thabtah
MIS, Department, Philadelphia University, Amman, Jordan
ffayez@philadelphia.edu.jo

M. Al-diabat
Philadelphia University, MIS Department, Amman, Jordan
maldiabat@philadelphia.edu.jo

Abstract—**This paper proposes a new discrete-time queue analytical model based on BLUE algorithm in order to determine the network congestion in preliminary stages. We compare the original BLUE, which has been implemented in Java, with our proposed analytical model with regards to different performance measures (average queue length, throughput, average queueing delay and packet loss probability). The comparison results show that the proposed discrete-time queue analytical model outperforms BLUE algorithm in terms of throughput and packet loss probability. Moreover, the proposed model maintains the throughput performance regardless whether the amount of the traffic load is light or heavy. Furthermore, we calculate the packet dropping probability function for our analytical model and the BLUE algorithm in order to decide which algorithm drops fewer packets.**

Index Terms— **Discrete-time Queue, Congestion Control, Performance Measures, Analytical Model**

I. INTRODUCTION

A network becomes congested when the incoming packets exceed the available network resources (bandwidth allocation and buffer spaces) [1]. Congestion plays a main role in the deterioration of network performance and can cause low throughput, high delay for packets and high packet loss rate. In order to manage congestion in computer networks, one must utilise a control method such as drop-tail (DT) [2, 3], which drops packets from the buffers tails solely after the router buffer becomes overflow. The DT algorithm gives poor performance within Transport Control Protocol (TCP) networks [4, 5, 6, 7] since the TCP sources adjust their sending rates only after the DT router buffer becomes overflow.

Several researchers [8, 9, 10, 11, 17, 18] have developed Active Queue Management (AQM) congestion control techniques such as Random Early Detection (RED) [8], Adaptive RED [9], Gentle RED [10], Stabilize Random Early Drop (SRED) [11], Dynamic Random Early Drop (DRED) [12], Random Exponential Marking (REM)[13, 14, 15, 16], BLUE [17, 18] and others. The goals for most of the above AQM algorithms are: to achieve high throughput performance, low packet loss rate, low queueing delay for packets, and maintain an average queue size as small as possible.

In this paper, we propose a new discrete-time queue analytical model based on BLUE algorithm [17, 18] in order to control congestions in internet and cellular networks. Moreover, we conduct a computer simulation using the BLUE algorithm in order to obtain its performance measures. Specifically, we record the average queue length (aql), throughput (*T*), average queueing delay (D), packet loss rate (P_{loss}) and packet dropping probability (D_p) results for the BLUE and compare them with those of our proposed analytical model. The main reason of the comparison is to decide which of the two algorithms offers better Quality of Service (QoS).

The paper is organised as follows: The classic BLUE algorithm is presented in Section 2, and our proposed discrete-time queue analytical model is introduced in Section 3. The comparison results of the BLUE and our proposed BLUE-based analytical model are given in Section 4, and finally Section 5 gives the conclusions and future works.

II. THE CLASSIC BLUE ALGORITHM

BLUE is one of the known AQM algorithms, which was

On losing of the packets or $(th < B_{length})$

$if((current\_time - last\_adjustment) > freeze)$

{

$D_p = D_p + P_{inc}$;

$last\_adjustment = current\_time$;}

When the buffer is empty (or $B_{length} = 0$)

$if((current\_time - last\_adjustment) > freeze)$

{ $D_p = D_p - P_{dec}$;

$last\_adjustment = current\_time$; }

Fig. 1 BLUE pseudocode.

primarily developed to enhance the performance of the well-known RED algorithm [17, 18]. BLUE depends upon a single packet dropping probability parameter (D_p) and a certain threshold (th). If the buffer length of the BLUE router becomes larger than th position, BLUE increases D_p value to alleviate the congestion. Whereas, if the buffer is empty or the link is idle, the D_p value will be decreased. BLUE also relies on other parameters as congestion metrics, including, packet loss, link utilisation and buffer length. The pseudocode of the BLUE algorithm is shown in Figure 1.

According to Figure 1, BLUE uses several parameters in order to adjust the D_p value like the $freeze$ parameter, which is utilised to determine the least time period between two successive adjustments. $freeze$ is often set to a fix value according to [17, 18], however, it can also be given an arbitrary value in order to avoid global synchronisation [19]. Other parameters associated with D_p are P_{inc} and P_{dec} that are usually used to determine the increasing or decreasing amount of D_p. Generally, P_{inc} parameter is given a larger value than P_{dec} in order to prevent underutilisation [17, 18]. It should be noted that BLUE algorithm drops packets at the router buffer arbitrarily.

III. THE PROPOSED DISCRETE-TIME QUEUE ANALYTICAL MODEL

In this section, we introduce our discrete-time queue analytical model, which is based on BLUE algorithm. This system depends on a particular time unit named slot [20], where each slot could occur in single or multiple events. An example of a single event is packets arrival or departure in a slot, whereas packets arrival and departure in the same slot is an example of multiple events. The proposed discrete-time queue system model has a finite (K packets) capacity including packets currently in service. Moreover, the arrival process utilised is the Identical Independent Distribution (I.I.D) Bernoulli process, $a_n \in \{0,1\}$, $n = 0,1,2,...$, where a_n denotes the packet arrival at slot n. The proposed queuing system model uses the classic BLUE algorithm as a congestion control method, and thus, it relies on a BLUE single threshold (th) [18].

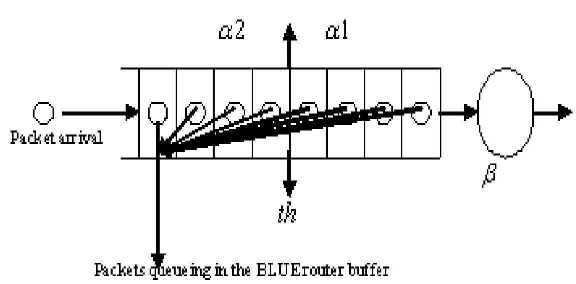

Fig. 2 The queuing system model

Our queuing system model is shown in Figure 2 where, the queuing system sources (or connections) start sending their packets at $\alpha 1$ rate, and that occurs only when the queue length at the BLUE router buffer is equal or smaller than th, and thus, no packets are dropped $(Dp = 0)$. However, when the queue length at the router buffer becomes above th, then (Dp) value increases from 0 to $\left(\frac{\alpha 1 - \alpha 2}{\alpha 1}\right)$. $\alpha 1$ and $\alpha 2$ denote the average sending rates for the sources in the queuing system before and after reaching th, respectively. β denotes the average departure rate at the BLUE router buffer, the queuing discipline which we use here is first come first serve (FCFS). We assume that the probability for packets arrival in a slot is $\alpha 1$ solely if the current queue length is equal or less than th, and $\alpha 2$ when the queue length exceeds th. We also assume that the queuing system model is equilibrium, and the queue length process is a Markov chain with finite state spaces. These state spaces are: $\{0,1,2,3,\ldots,th, th+1,\ldots,K-1,K\}$. Finally, we assume that $\alpha 1 > \alpha 2$ and $\beta > \alpha 1$, and therefore $\beta > \alpha 2$. The state transition diagram for the BLUEdiscrete time queuing model is given in Figure 3.

From Figure 3, we can obtain the balance equations for the presented discrete-time queue analytical model, the balance equations are defined in the eleven equations below,

$$\Pi_0 = (1-\alpha 1)\Pi_0 + [\beta(1-\alpha 1)]\Pi_1 \quad \ldots\ldots\ldots\ldots (1)$$

$$\Pi_1 = \alpha 1 \Pi_0 + [\alpha 1 \beta + (1-\alpha 1)(1-\beta)]\Pi_1 + [\beta(1-\alpha 1)]\Pi_2 \quad \ldots\ldots\ldots\ldots (2)$$

$$\Pi_2 = [\alpha 1(1-\beta)]\Pi_1 + [\alpha 1 \beta + (1-\alpha 1)(1-\beta)]\Pi_2 + [\beta(1-\alpha 1)]\Pi_3 \quad \ldots\ldots\ldots\ldots (3)$$

$$\Pi_{th-1} = [\alpha 1(1-\beta)]\Pi_{th-2} + [\alpha 1 \beta + (1-\alpha 1)(1-\beta)]\Pi_{th-1} + [\beta(1-\alpha 1)]\Pi_{th} \quad \ldots\ldots\ldots\ldots (4)$$

In general we obtain,

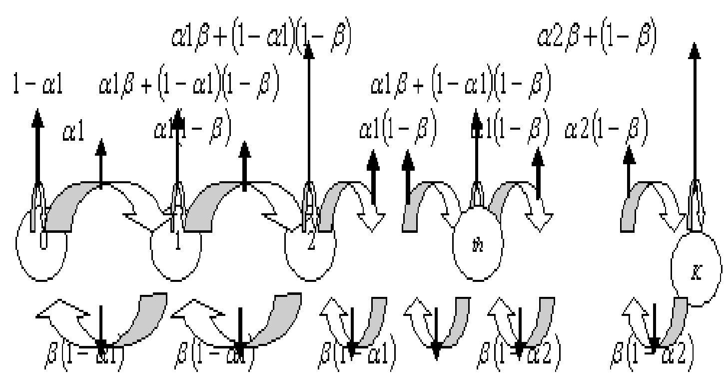

Fig. 3 The state transition diagram for the proposed analytical model

$$\Pi_i = [\alpha 1(1-\beta)]\Pi_{i-1} + [\alpha 1\beta + (1-\alpha 1)(1-\beta)]\Pi_i + [\beta(1-\alpha 1)]\Pi_{i+1}$$,

where

$i = 2,3,4,\ldots,th-1$ …………………………………...….. (5)

$$\Pi_{th} = [\alpha 1(1-\beta)]\Pi_{th-1} + [\alpha 1\beta + (1-\alpha 1)(1-\beta)]\Pi_{th} + [\beta(1-\alpha 2)]\Pi_{th+1}$$ …

…………………..…. (6)

$$\Pi_{th+1} = [\alpha 1(1-\beta)]\Pi_{th} + [\alpha 2\beta + (1-\alpha 2)(1-\beta)]\Pi_{th+1} + [\beta(1-\alpha 2)]\Pi_{th+2}$$

………………….. (7)

$$\Pi_{th+2} = [\alpha 2(1-\beta)]\Pi_{th+1} + [\alpha 2\beta + (1-\alpha 2)(1-\beta)]\Pi_{th+2} + [\beta(1-\alpha 2)]\Pi_{th+3}$$

………………….. (8)

$$\Pi_{K-1} = [\alpha 2(1-\beta)]\Pi_{K-2} + [\alpha 2\beta + (1-\alpha 2)(1-\beta)]\Pi_{K-1} + [\beta(1-\alpha 2)]\Pi_{K}$$

………………….. (9)

In general we obtain,

$$\Pi_i = [\alpha 2(1-\beta)]\Pi_{i-1} + [\alpha 2\beta + (1-\alpha 2)(1-\beta)]\Pi_i + [\beta(1-\alpha 2)]\Pi_{i+1}$$,…

…………………………….… (10)

Where $i = th+2, th+3, th+4,\ldots,K-1$

Finally,

$$\Pi_K = [\alpha 2(1-\beta)]\Pi_{K-1} + [\alpha 2\beta + (1-\beta)]\Pi_K$$,

Where $K = th + I$ ……....... (11)

Let $\gamma 1 = \dfrac{\alpha 1(1-\beta)}{\beta(1-\alpha 1)}$ ……... (12)

and $\gamma 2 = \dfrac{\alpha 2(1-\beta)}{\beta(1-\alpha 2)}$ …..... (13)

After solving the balance equations (1 - 11) and by substituting equations (12 and 13) we obtain,

$$\Pi_1 = \frac{\alpha 1}{\beta(1-\alpha 1)}\Pi_0 = \frac{\gamma 1}{(1-\beta)}\Pi_0$$ ………... (14)

$$\Pi_2 = \frac{\alpha 1^2(1-\beta)}{\beta^2(1-\alpha 1)^2}\Pi_0 = \frac{\gamma 1^2}{(1-\beta)}\Pi_0$$ ….…. (15)

$$\Pi_3 = \frac{\alpha 1^3(1-\beta)^2}{\beta^3(1-\alpha 1)^3}\Pi_0 = \frac{\gamma 1^3}{(1-\beta)}\Pi_0$$ ……. (16)

$$\Pi_{th} = \frac{\alpha 1^{th}(1-\beta)^{th-1}}{\beta^{th}(1-\alpha 1)^{th}}\Pi_0 = \frac{\gamma 1^{th}}{(1-\beta)}\Pi_0$$ … (17)

In general we obtain,

$$\Pi_i = \frac{\alpha 1^i(1-\beta)^{i-1}}{\beta^i(1-\alpha 1)^i}\Pi_0 = \frac{\gamma 1^i}{(1-\beta)}\Pi_0$$,where

$i = 1,2,3,\ldots,th$ ……………………..……. (18)

$$\Pi_{th+1} = \frac{\alpha 1^{th+1}(1-\beta)^{th}}{\beta^{th+1}(1-\alpha 1)^{th}(1-\alpha 2)}\Pi_0 = \frac{\gamma 1^{th+1}(1-\alpha 1)}{(1-\alpha 2)(1-\beta)}\Pi_0$$ …. (19)

$$\Pi_{th+2} = \frac{\alpha 1^{th+1}\alpha 2\ (1-\beta)^{th+1}}{\beta^{th+2}(1-\alpha 1)^{th}(1-\alpha 2)^2}\Pi_0 = \frac{\gamma 1^{th+1}\gamma 2\ (1-\alpha 1)}{(1-\alpha 2)(1-\beta)}\Pi_0$$ ...(20)

$$\Pi_{K=th+I} = \frac{\alpha 1^{th+1}\alpha 2^{I-1}(1-\beta)^{K-1}}{\beta^K(1-\alpha 1)^{th}(1-\alpha 2)^I}\Pi_0 = \frac{\gamma 1^{th+1}\gamma 2^{I-1}(1-\alpha 1)}{(1-\alpha 2)(1-\beta)}\Pi_0$$ .. (21)

In general we obtain,

$$\Pi_{th+i} = \frac{\alpha 1^{th+1}\alpha 2^{i-1}(1-\beta)^{th+i-1}}{\beta^{th+i}(1-\alpha 1)^{th}(1-\alpha 2)^i}\Pi_0 = \frac{\gamma 1^{th+1}\gamma 2^{i-1}(1-\alpha 1)}{(1-\alpha 2)(1-\beta)}\Pi_0$$ ….. (22)

Where $i = 1,2,3,\ldots,I$.

After the probabilities of the queuing system states are computed, we need to calculate the probability where there is no packets in the queue (also called the probability when the queuing system is idle (Π_0)), where Π_0 is calculated by applying the normalized equation expressed as in equation (23),

$$\sum_{i=0}^{K}\Pi_i = 1$$ …………………..………….……….. (23)

Then by substituting equations (14-21) in equation (23), we get,

$$\Pi_0 = \left[\frac{1-\gamma 1^{th+1} - \beta(1-\gamma 1)}{(1-\beta)(1-\gamma 1)} + \frac{\gamma 1^{th+1}(1-\alpha 1)(1-\gamma 2^I)}{(1-\beta)(1-\alpha 2)(1-\gamma 2)}\right]^{-1}$$ ………... (24)

After Π_0 is estimated, we utilise it to calculate the queuing system performance metrics (aql , T , D , P_{loss}). Firstly, we calculate aql by applying the generating function $P(z)$ expressed in equation (25),

ISBN:978-988-98671-5-7

$$P(z) = \sum_{i=0}^{K} z^i \prod_i \text{ .... } (25)$$

We calculate aql by taking the first derivative of $P(z)$ at $z = 1$ as appeared in equation (26),

$$aql = P^{(1)}(1) \text{ .... } (26)$$

Then,

$$aql = P^{(1)}(1) = \frac{\prod_0}{(1-\beta)}\left[\begin{array}{l}\dfrac{\gamma 1 - \gamma 1^{th+1}[1+th(1-\gamma 1)]}{(1-\gamma 1)^2} + \\ \gamma 1^{th+1}\dfrac{(1-\alpha 1)}{(1-\alpha 2)}\left[\dfrac{th(1-\gamma 2)(1-\gamma 2^I)+1-\gamma 2^I[1+I(1-\gamma 2)]}{(1-\gamma 2)^2}\right]\end{array}\right]$$

$$\text{.. } (27)$$

It should be noted that there is another way to calculate aql using $\sum_{i=0}^{K} i \prod_i$ equation.

After aql is computed, we calculate (T), which represents the number of packets that have been passed through the queuing system successfully. We use equation (28) to generate T result. $T = \beta \sum_{i=1}^{K} \prod_i = \beta(1 - \prod_0)$ Packets/slot…(28)

Then by applying equation (24) in equation (28), we get the final form of T as shown below,

$$T = \beta\left(1 - \left[\begin{array}{l}\dfrac{1-\gamma 1^{th+1} - \beta(1-\gamma 1)}{(1-\beta)(1-\gamma 1)} + \\ \dfrac{\gamma 1^{th+1}(1-\alpha 1)(1-\gamma 2^I)}{(1-\beta)(1-\alpha 2)(1-\gamma 2)}\end{array}\right]^{-1}\right) packets/slot.$$

$$\text{............ } (29)$$

Based on aql and T results obtained from equations (27) and (29), respectively, we can estimate D using littles law as shown in equation (30).

$$D = \frac{aql}{T} slots = \frac{P^{(1)}(1)}{T} slots = \frac{\sum_{i=0}^{K} i \prod_i}{T} slots \text{ ...... } (30)$$

Then by substituting equations (27 and 29) in equation (30) we get,

$$D = \frac{\dfrac{\prod_0}{(1-\beta)}\left[\begin{array}{l}\dfrac{\gamma 1 - \gamma 1^{th+1}[1+th(1-\gamma 1)]}{(1-\gamma 1)^2} + \\ \gamma 1^{th+1}\dfrac{(1-\alpha 1)}{(1-\alpha 2)}\left[\dfrac{th(1-\gamma 2)(1-\gamma 2^I)+1-\gamma 2^I[1+I(1-\gamma 2)]}{(1-\gamma 2)^2}\right]\end{array}\right]}{\beta\left(1 - \left[\dfrac{1-\gamma 1^{th+1} - \beta(1-\gamma 1)}{(1-\beta)(1-\gamma 1)} + \dfrac{\gamma 1^{th+1}(1-\alpha 1)(1-\gamma 2^I)}{(1-\beta)(1-\alpha 2)(1-\gamma 2)}\right]^{-1}\right)} slot$$

$$\text{.. } (31)$$

Finally, we calculate P_{loss}, which corresponds to the proportion of packets that lost the service at the BLUE router buffer. The BLUE router buffer starts dropping packets solely when the queue length exceeds th position. Equation (32) is used to compute the P_{loss}.

$$P_{loss} = \sum_{i=th+1}^{K} \prod_i \text{ .... } (32)$$

Then by applying equations (19 – 21) in equation (32) we obtain P_{loss} final form.

$$P_{loss} = \prod_0 \gamma 1^{th+1} \frac{(1-\alpha 1)(1-\gamma 2^I)}{(1-\beta)(1-\alpha 2)(1-\gamma 2)} \text{ .... } (33)$$

IV. SIMULATION AND PERFORMANCE EVALUATION

In this section, we present a comparison between the original BLUE algorithm [17, 18] and the proposed BLUE-based analytical model with reference to different performance measures, including, (aql , T , D and P_{loss}). Both the original BLUE and our proposed BLUE discrete-time queue analytical model are implemented in java on a 1.7 Mhz processor machine with 512 RAM. The parameters of the classic BLUE algorithm ($\alpha 1$, $\alpha 2$, β , K , $threshold$, $freeze$, P_{inc} , P_{dec} , D_{init} and Number of slots) are set to [0.83 – 0.85], 0.8, 0.9, 30, 18,0.01, 0.00025, 0.000025, 0.05 and 1000000, respectively. Moreover, the proposed analytical model parameters, $\alpha 1$, β , th and K , are set to values similar to the those of the classic BLUE's parameters. Figures 4, 5, 6 and 7 show the performance measures results of the original BLUE and our proposed BLUE analytical model. Specifically, these figures display $\alpha 1$ results against aql , T , D , and P_{loss} results, respectively.

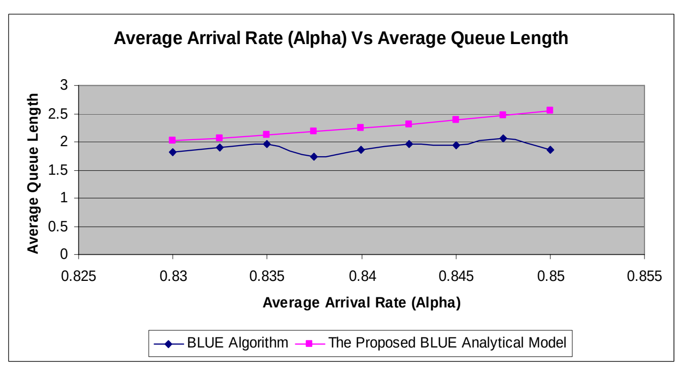

Fig. 4 $\alpha 1$ Vs aql

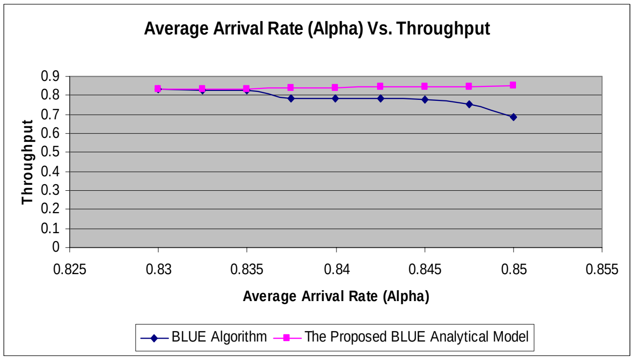

Figure 5: $\alpha 1$ Vs *T*.

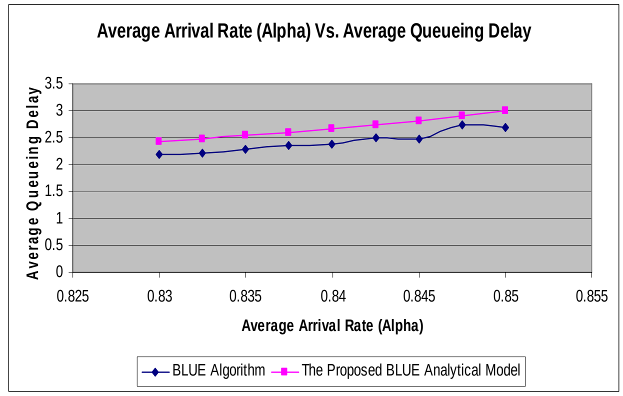

Fig. 6 $\alpha 1$ Vs aql.

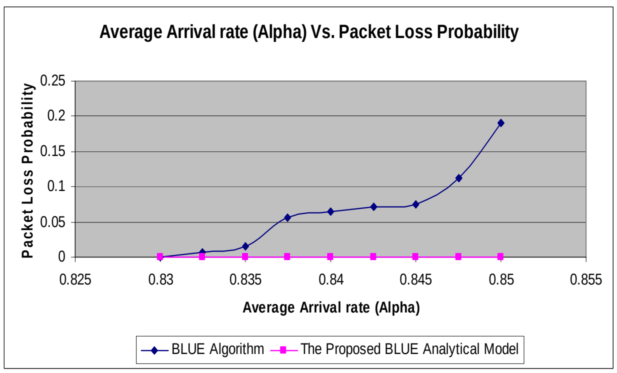

Fig. 7: $\alpha 1$ Vs P_{loss}.

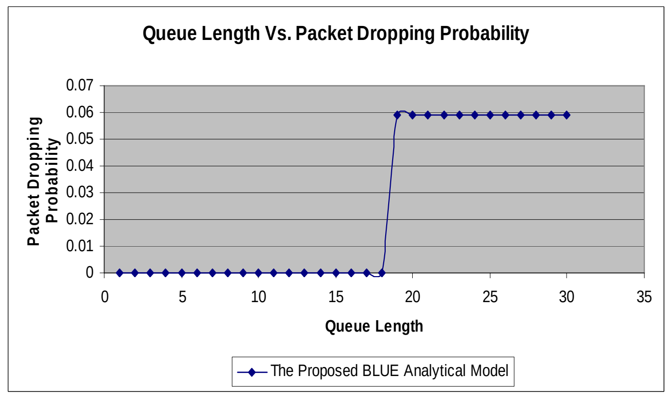

Fig. 9: Queue length versus D_n for the proposed

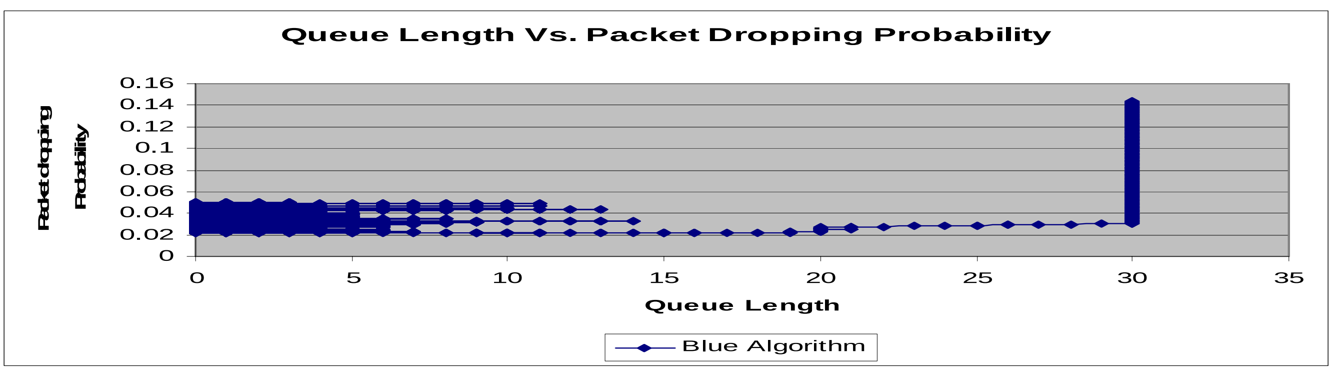

Fig. 8: Queue length against D_p for DRED simulation software.

We observe from Figure 4 that the original BLUE produces results in terms of aql smaller than those of our analytical model, and therefore BLUE maintains aql better than the proposed analytical model. Furthermore, Figure 6 indicates that the original BLUE gives smaller results of the average waiting time for the packets (D) in the queueing system if compared with those of the proposed analytical model. However, according to the throughput performance results in Figure 5, our proposed analytical model outperformed the original BLUE since it produces higher numbers with regards to *T*. Finally, Figure 7 displays the packet loss probability results for both algorithms. The results indicate that the proposed analytical model drops fewer packets than BLUE, and thus, it controls the throughput performance more than the original BLUE.

The D_p results produced by the classic BLUE and the proposed BLUE-based analytical model are shown in Figures 8 and 9, respectively. BLUE parameters were set in the experiments to the same values mentioned at the beginning of this section. Moreover, the proposed analytical model parameters have been set to the same parameters values of BLUE. The only exception is that the number of slots for BLUE algorithm is set 20000. Figures 8 and 9 clearly show that our analytical model drops fewer packets than the original BLUE algorithm, and therefore, it maintains better throughput performance than BLUE (Figure 5 gives further details).

V. CONCLUSIONS

A new discrete-time queue analytical model based on BLUE algorithm is presented in this paper. The proposed analytical model reduces the arrival rate from $\alpha 1$ to $\alpha 2$ when the congestion appeared at the BLUE router buffer. We performed a comparison between our proposed analytical model and the original BLUE with reference to aql , T , D and P_{loss} performance measures. The experimental results indicated that BLUE algorithm outperformed our discrete-time queue analytical model in terms of aql and D . However, our analytical model maintains better T performance than BLUE since it drops fewer packets regardless of the traffic load. In near future, we will apply the presented analytical model in queueing network system with multiple nodes. Further, we plan to apply our analytical model in the internet and cellular networks as a congestion control method.

REFERENCES

[1]Welzl, M., “Network Congestion Control: Managing Internet Traffic”, 282 pages, July, 2005.
[2] Braden, R., Clark, D., Crowcroft, J., Davie, B., Deering, S., Estrin, D., Floyd, S., Jacobson, V., Minshall, G., Partridge, C., Peterson, L., Ramakrishnan, K., Shenker, S., wroclawski, J., and Zhang, L., “Recommendations on Queue Management and Congestion Avoidance in the Internet,” RFC 2309, April 1998.
[3 Brandauer, C., Iannaccone, G., Diot, C., Ziegler, T., Fdida, S., and May, M., “Comparison of Tail Drop and Active Queue Management Performance for bulk-data and Web-like Internet Traffic,” In Proceeding. ISCC, pp. 122-129. IEEE, July 2001.
[4]Postel, J., B., Transmission Control Protocol. RFC, Information Sciences Institute, Marina del Rey, CA, September 1981, RFC 793.
[5]Stevens, W., R., “TCP/IP Illustrated, Volume 1, “. Addison-Wesley, Reading, MA, November 1994.
[6]Postel, J., “Transmission Control Protocol – DARPA Internet Program Protocol Specification,” DARPA, September 1981. RFC 793.
[7]Wright, G., R., and Stevens, W., R., "TCP/IP Illustrated, Volume 2 (The Implementation),” Addison Wesley, January 1995.
[8]Floyd, S., and Jacobson V., Random Early Detection Gateways for Congestion Avoidance. IEEE/ACM Transactions on Networking, 1(4):397-413, Aug 1993.
[9]Floyd, S., Ramakrishna, G., and Shenker, S., “Adaptive RED: An Algorithm for Increasing the Robustness of RED’s Active Queue Management,” Technical report, ICSI, August 1, 2001.
[10]Floyd, S.,”Recommendations on using the gentle variant of RED, “May 2000. available at
http://www.aciri.org/floyd/red/gentle.html.
[11]Ott, T., Lakshman, T., and Wong, L., “SRED: Stabilized RED,” in Proc. IEEE INFOCOM, Mar. 1999, pp. 1346-1355.
[12]Aweya, J., Ouellette, M., and Montuno, D., Y., “A Control Theoretic Approach to Active Queue Management,” Comp. Net., vol. 36, issue 2-3, July 2001, pp. 203-35.
[13]Athuraliya, S., Li, V., H., Low, S., H., and Yin, Q., “REM: Active Queue Management”, IEEE Network, 15(3), 48-53. May, 2001.
[14]Lapsley, D., and Low, S., “Random Early Marking: An Optimisation Approach to Internet Congestion Control,” in Proceedings of IEEE ICON '99.
[15]Lapsley, D., and Low, S., “Random Early Marking for Internet Congestion Control,” Proceeding of GlobeCom’99, pp. 1747-1752, 1999.
[16]Athuraliya, S., Lapsley, D., and Low, S., “An Enhanced Random Early Marking Algorithm for Internet Flow Control, “INFOCOM’ 2000, Telaviv, Israel, pp. 1425-1434.
[17]Feng, W., kandlur, D.,Saha, D., and Shin, K.G., “Blue: A new class of active queue management algorithms,”Univ. Michigan, Ann Arbor,MI,Tech. Rep.UM CSE-TR-387-99,Apr.1999.
[18]Feng, W., Shin, K.G., and kandlur, D., “The Blue Active Queue Management Algorithms,” IEEE/ACM Transactions on Networking, Volume 10. Issue 4, August 2002.
[19]Floyd, S., and Jacobson V., On Traffic Phase Effects in Packet-Switched Gateways, Internetworking: Research and Experience, V.3 N.3, September 1992, p.115-156.
[20]Woodward, M., E., “Communication and Computer Networks: Modelling with discrete-time queues, Pentech Press, London, 1993.