Linear Analysis for a BLUE Congestion Control Algorithm using A Discrete-time Queue

Hussein Abdel-jaber
Department of Computing, University of Bradford, Bradford BD7 1DP, UK
habdelja@ Bradford.ac.uk

Fadi Thabtah
Department Management Information Systems, Philadelphia University, Amman, Jordan
ffayez@philadelphia.edu.jo

Mike Woodward
Department of Computing, University of Bradford, Bradford BD7 1DP, UK
m.e.woodward @ Bradford.ac.uk

Wa'el Musa Hadi
Department of Computer Information Systems
Arab Academy for Banking and Financial Sciences, Amman, Jordan
whadi81@students.aabfs.org

ABSTRACT

With the increasing demands on network resources by network connections, congestion becomes one of the critical problems, which deteriorates network performance. In order to control the congestion, we propose a discrete-time queue analytical model based on BLUE that increases packet dropping probability linearly in order to manage the congestion incident. We compare the presented analytical model with the original BLUE algorithm with respect to different performance measures, including, average queue length, throughput ratio, average queueing delay and packet loss rate. The experimental results show that the original BLUE outperforms our BLUE-based analytical model in terms of average queue length and average queueing delay. In addition, BLUE and our analytical model give similar results with respects to throughput ratio and packet loss rate at certain level of traffic load. However, when the traffic load increases at certain levels, BLUE results degrade with reference to throughput ratio, and packet loss rate results increases. Whereas the proposed analytical model maintains better throughput performance than BLUE regardless whether the traffic load is heavy or light and consequently, the packet loss rate decreases.

Key Words**:** BLUE, Discrete-time Queue, Analytical Model, Congestion Control, Performance Measures.

1. Introduction

Congestion is one of the major problems that might occur in the internet when the number of injected packets exceeds the network capacity such as: when the number of injected packets becomes larger than the available bandwidth and/or the buffer spaces [15]. Congestion plays a main role in the degradation of the network performance causing low throughput, high packet loss rate and packets queueing delay. To control congestion in the internet, common methods such as drop-tail (DT) [1, 2] were proposed and used for several years. The DT method has drawbacks, including, lockout phenomenon, full queues, global synchronisation and bias versus bursty traffic [1]. In order to overcome DT limitations, several researchers have presented active queue management (AQM) mechanisms for the purpose of controlling and managing congestions in computer networks. AQM techniques have many goals such as:

- Achieving high throughput
- Achieving low queueing delay and packet loss rate

- Maintaining the average queue length as small as possible

Random Early Detection (RED) [3], Adaptive RED [4], Gentle RED [5], Random Exponentially Marking (REM) [6, 7, 8, 9], Dynamic Random Early Drop (DRED) [10], Stabilized Random Early Drop (SRED) [11] and BLUE [12, 13] are examples of AQM algorithms.

In this paper, we propose a discrete-time queue analytical model based on BLUE algorithm. The ultimate goal of the proposed analytical model is to discover the congestion incident on the router buffer as early as possible. This paper also presents an experimental comparison between our BLUE-based analytical model and the classic BLUE algorithm according to different performance measures, including, average queue length, throughput, average queueing delay and packet loss rate. The aim of this comparison is to decide which of the two techniques gives better performance. The paper is organised as follows, Section 2 presents the original BLUE method. The proposed discrete-time queue analytical model is given in Section 3. A comparison between our analytical model and BLUE with regards to the performance measures is given in Section 4. Finally, Section 5 exhibits the conclusions and future works.

2. The Original BLUE Mechanism

BLUE is one of the known AQM algorithms, which was primarily developed to enhance the performance of the well-known RED algorithm [12, 13]. 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 [12, 13], however, it can also be given an arbitrary value in order to avoid global synchronisation. 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 [12, 13]. It should be noted that BLUE algorithm drops packets at the router buffer arbitrarily.

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 $B_{length} = 0$)

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

{

$D_p = D_p - P_{dec}$;

$last\_adjustment = current\_time$;

}

Figure 1: BLUE pseudocode.

3. The Proposed Linear Decreasing Discrete-time Queue Analytical Model Based on BLUE

In this section, we present a new discrete-time queue analytical model shown in Figure 2 in order to drop the arrival packets as early as possible and to avoid the congestion incident in computer networks. The proposed analytical model has been developed using the discrete-time queue [14]. The model relies on a particular time unit named slot, where single or multiple event(s) might occur in each slot. The capacity of the proposed queueing system is K packets, including packets currently in service. Moreover, the arrival process, which the given model uses is the identical independent distribution (i.i.d) Bernoulli process, ($a_n \in \{0,1\}, n = 0,1,2,...,$),

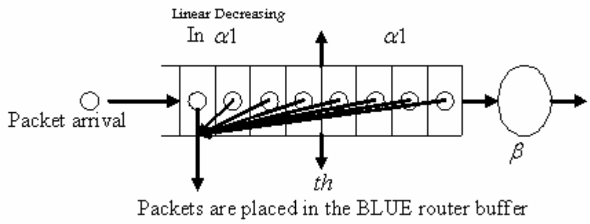

Figure 2: The proposed queuing system.

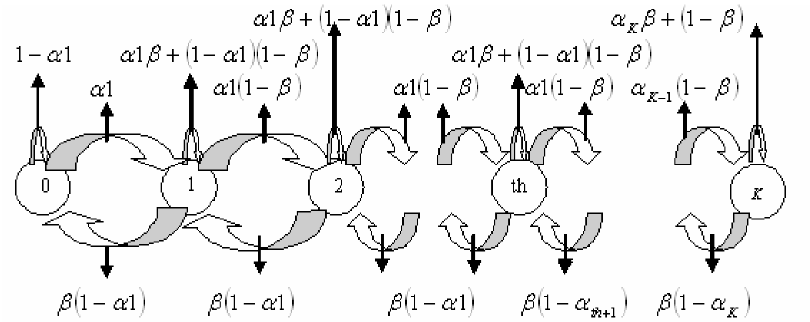

Figure 3: The state transition diagram for the BLUE discrete-time queue analytical model.

where a_n represents packet arrival at slot n. Since the presented queuing model is based on BLUE, it depends on BLUE single threshold (th), where th denotes the threshold index at the BLUE router buffer.

Linear decreasing

According to Figure 2, the queueing model connections begin transmitting at $a1$ rates when the queue length is less than or equal to the th at the BLUE router buffer, thereby no packets are dropped $(Dp = 0)$, where Dp represents the packet dropping probability. Whereas, if the queue length index at the BLUE router buffer is greater than th position, the connections decrease their transmission rates linearly from $a1$ to a_i to alleviate the congestion. Moreover, the Dp value increases linearly from 0 to $\left(\frac{a1 - a_i}{a1}\right)$ as the queue length increases from $th+1$ to the maximum capacity of the BLUE router buffer (K), where

$$a_i = a1 - (1 + i - th)\frac{a1}{(1 + K - th)}, \text{ if } i > th .$$

$a1$ and a_i represent the average arrival rates (transmission rates) for the connections in the proposed queuing model before reaching $th+1$ index and after reaching $th+1$ index at the BLUE router buffer, respectively. b represents the average departure rate at the BLUE router buffer and the queuing discipline used in this queueing model is first come first serve (FCFS). The probability for packet arrival in a slot is assumed to be $a1$ solely if the current queue length at the BLUE router buffer is below or equal to the th index. Also, assume that a_i is the probability for the arrival packet in a slot if the current queue length is above th index. Furthermore, let b be the probability of packet departure from a slot. We also consider that the given queuing system is equilibrium, and the queue length process is a Markov chain with finite state spaces. These state spaces are $\{0,1,2,3,\ldots,th-1,th,th+1,\ldots,K-1,K\}$. Finally, we assume that $a1 > a_i$ and $b > a1$, thus $b > a_i$. Figure 3 displays the state transition diagram for the given queueing system.

From Figure 3, we can derive the balance equations for the proposed discrete-time queue analytical model, where the balance equations are expressed in the seven equations shown below,

$$\Pi_0 = (1 - a1)\Pi_0 + [b(1 - a1)]\Pi_1 \ldots (1)$$

$$\Pi_1 = a1\Pi_0 + [a1b + (1 - a1)(1 - b)]\Pi_1 + [b(1 - a1)]\Pi_2$$
……………………………... (2)

In general we get,

$$\Pi_i = [a1(1 - b)]\Pi_{i-1} + [a1b + (1 - a1)(1 - b)]\Pi_i + [b(1 - a1)]\Pi_{i+1}$$
, where $i = 2,3,4,\ldots,th-1 \ldots$ (3)

$$\Pi_{th} = [a1(1 - b)]\Pi_{th-1} + [a1b + (1 - a1)(1 - b)]\Pi_{th} + [b(1 - a_{th+1})]\Pi_{th+1}$$
……………………………….. (4)

$$\Pi_{th+1} = [a1(1 - b)]\Pi_{th} + [a_{th+1}b + (1 - a_{th+1})(1 - b)]\Pi_{th+1} + [b(1 - a_{th+2})]\Pi_{th+2}$$
……………………………….. (5)

In general we achieve,

$$\prod_i = [a_{i-1}(1-b)]\prod_{i-1} + [a_i b + (1-a_i)(1-b)]\prod_i + [b(1-a_{i+1})]\prod_{i+1}$$

, Where $i = th+2, th+3, th+4, \ldots, K-1 \ldots$ (6)

Finally,

$$\prod_K = [a_{K-1}(1-b)]\prod_{K-1} + [a_K b + (1-b)]\prod_K,$$

Where $K = th + I$ .... (7)

Let $g = \dfrac{a1(1-b)}{b(1-a1)}$ .... (8)

And $g_i = \dfrac{a_i(1-b)}{b(1-a_i)}$,where $i > th$ .... (9)

After the balance equations (1-7) are computed, we apply equations (8 and 9) on them to achieve,

$$\prod_i = \frac{a1^i(1-b)^{i-1}}{b^i(1-a)1^i}\prod_0 = \frac{g^i}{(1-b)}\prod_0 ,\text{where}$$

$i = 1,2,3,\ldots,th$ .... (10)

$$\prod_i = \frac{a1^{th+1}(1-b)^{i-1}\prod_{j=th+1}^{i-1} a_j}{b^i(1-a1)^{th}\prod_{l=th+1}^{i}(1-a_l)}\prod_0 = \frac{g^{th+1}(1-a1)\prod_{j=th+1}^{i-1} g_j}{(1-a_i)(1-b)}\prod_0$$

Where $i = th+1, th+2, \ldots, th+I$ and $g_j = 1,\ j \le th$ ..... (11)

Beyond computing the probabilities of the given queuing system states, we derive the probability when the queuing system is idle $(\prod_0)$, where $\prod_0$ can be derived using the normalised equation shown in equation (12),

$$\sum_{i=0}^{K}\prod_i = 1 \text{ .... (12)}$$

Then,

$$\Pi_0 = \left[\frac{1-\gamma^{th+1}-\beta(1-\gamma)}{(1-\beta)(1-\gamma)} + \gamma^{th+1}\frac{(1-\alpha 1)}{(1-\beta)}\sum_{i=th+1}^{K}\prod_{j=th+1}^{i-1}(\gamma_j)\frac{1}{(1-\alpha_i)}\right]^{-1}$$

.... (13)

After $\prod_0$ is derived, we evaluate the performance measures for the given queuing system, where the performance measures are the average queue length (aql), throughput fraction (T) ,the average queuing delay (D) and the packets loss probability (P_{loss}). Firstly, we compute the aql utilising the generating function $P(z)$, where $P(z)$ is shown in equation (14) below,

$$P(z) = \sum_{i=0}^{K} z^i \prod_i \text{ .... (14)}$$

The aql is equal to $P(z)$ first derivative at $z = 1$ as given in equation (15) below,

$$P^{(1)}(1) = \frac{\Pi_0}{(1-b)}\left[\begin{array}{l}\dfrac{g - g^{th+1}[1+th(1-g)]}{(1-g)^2} + g^{th+1}(1-a1) \\ \displaystyle\sum_{i=th+1}^{K}\prod_{j=th+1}^{i-1}(g_j) * \frac{i}{(1-a_i)}\end{array}\right],$$

where $g_j = 1,\ j \le th$ .... (15)

We can also evaluate aql using the $\sum_{i=0}^{K} i\prod_i$ equation, which gives the same result as equation (15). Secondly, we calculate the T as the number of packets that have passed through the queuing system successfully divided by the packets that have arrived at the queuing system. T can also be defined as the fraction of time when the router is busy. We compute T using equation (16). $T = b\sum_{i=1}^{K}\prod_i = b(1-\prod_0)$ Packets/slot….. (16)

Depending on aql and T results obtained from equations (15) and (16), respectively, we estimate D using the littles law, as shown in equation (17).

$$D = \frac{P^{(1)}(1)}{T}\, slots \text{ .... (17)}$$

Finally, we evaluate the P_{loss} as the proportion of packets that have lost the service at the BLUE router buffer from all packets that have arrived. The P_{loss} can be obtained from equation (18).

$$P_{loss} = \sum_{i=th+1}^{K}\prod_i \text{ .... (18)}$$

4. Experimental Results

This section introduces a comparison between the proposed BLUE discrete-time queue analytical model and the original BLUE with reference to the following performance measures: (aql), (T), (D), (P_{loss}). Both the BLUE and the proposed analytical model were implemented in Java on 1.8 Mhz Pentium machine with 512 RAM. The main goal of this comparison is to detect which algorithm offers better quality of service (QoS). The proposed discrete-time queue analytical

$a1$	b	K	th	$freeze\_time$	P_{inc}	P_{dec}	D_{init} (initial probability)	Number of slots
0.81	0.9	40	17	0.01	0.00025	0.000025	0.05	1000000
0.8125	0.9	40	17	0.01	0.00025	0.000025	0.05	1000000
0.815	0.9	40	17	0.01	0.00025	0.000025	0.05	1000000
0.8175	0.9	40	17	0.01	0.00025	0.000025	0.05	1000000
0.82	0.9	40	17	0.01	0.00025	0.000025	0.05	1000000
0.8225	0.9	40	17	0.01	0.00025	0.000025	0.05	1000000
0.825	0.9	40	17	0.01	0.00025	0.000025	0.05	1000000
0.8275	0.9	40	17	0.01	0.00025	0.000025	0.05	1000000
0.83	0.9	40	17	0.01	0.00025	0.000025	0.05	1000000
0.8325	0.9	40	17	0.01	0.00025	0.000025	0.05	1000000
0.835	0.9	40	17	0.01	0.00025	0.000025	0.05	100000
0.8375	0.9	40	17	0.01	0.00025	0.000025	0.05	1000000

Table1: BLUE algorithm parameters.

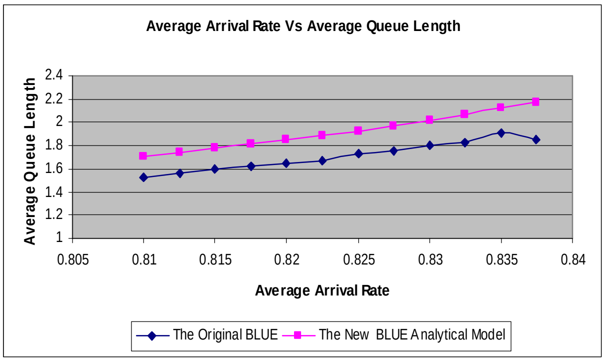

Figure 4: $a1$ Vs. aql.

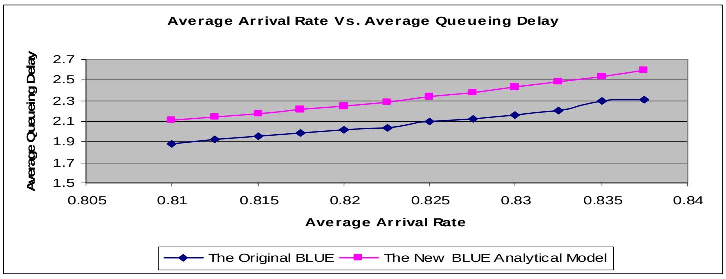

Figure 6: $a1$ Vs. D.

model parameters ($a1$, b, th and K) were set to [0.81 – 0.8375], 0.9, 17 and 40 packets, respectively. On the other hand, BLUE parameters were set to the values shown in Table 1.

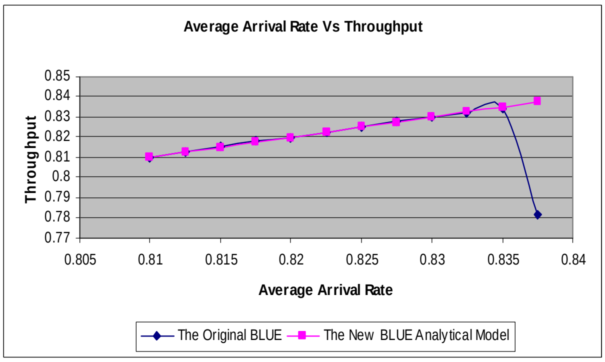

Figure 5: $a1$ Vs. T.

Figures 4, 5, 6 and 7 exhibit the performance measures results for the proposed BLUE analytical model and the classic BLUE algorithm. Specifically, Figure 4 indicates the results of $a1$ versus aql, Figure 5 shows $a1$ versus T results, Figure 6 displays $a1$ versus D results, and finally Figure 7 illustrates $a1$ versus P_{loss} results.

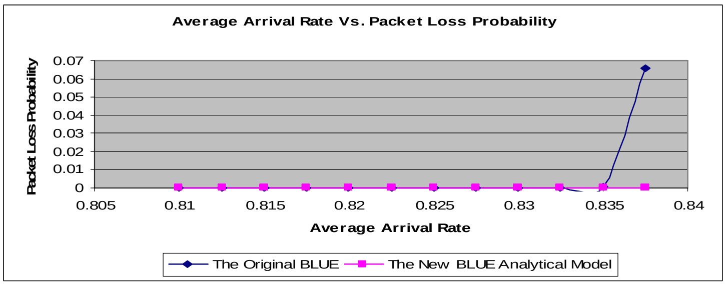

Figure 7: $a1$ Vs. P_{loss} .

BLUE results in Figures 4 and 6 are better than the proposed discrete-time queue analytical model in terms of aql and D. Furthermore, Figures 5 and 7 indicate that both algorithms present similar results in terms of T and P_{loss}. However, this similarity appeared only when the traffic load is between 0.81 and 0.835 (or less than 0.835). When the traffic load exceeds 0.835, BLUE drops higher packets than our analytical model, which leads to the deterioration of T performance for BLUE algorithm. Figure 5 also demonstrates that our proposed BLUE analytical model maintains better T performance than the original BLUE regardless whether the traffic load is light or heavy, and thus P_{loss} performance is maintained better for the proposed model than BLUE (Figure 7 gives further details).

5. Conclusions and Future Works

In this paper, we introduced a new BLUE based discrete-time queue analytical model, in which its sources decrease their sending rates linearly from $a1$ to a_i in order to control congestion. We compared our analytical model with the original BLUE in regards to different performance measures (aql, T, D, P_{loss}), and we concluded the following:

- The original BLUE gives better results than the proposed analytical model in terms of (aql, D).
- Blue and the introduced analytical model generate similar results with respect to (T, P_{loss}) when the traffic load is equal or less than 0.835. However, when the traffic load becomes greater than 0.835, our model derives better results with reference to T and therefore, drops fewer packets.
- The proposed analytical model does not get impacted when the traffic load changes from heavy to light and vice versa. On the other hand, BLUE performance measures results depend heavily on the traffic load, and thus, BLUE performance deteriorates when the traffic load is heavy.

In near future, we intend to apply the proposed discrete-time queue analytical model on single and multiple arrivals in each slot rather than just a single arrival slot.

References

[1] 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.

[2] 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.

[3] Floyd, S., and Jacobson V., Random Early Detection Gateways for Congestion Avoidance. IEEE/ACM Transactions on Networking, 1(4):397-413, Aug 1993.

[4] 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.

[5] Floyd, S.,"Recommendations on using the gentle variant of RED, "May 2000.

Available at http://www.aciri.org/floyd/red/gentle.html.
[6] Athuraliya, S., Li, V., H., Low, S., H., and Yin, Q., "REM: Active Queue Management", IEEE Network, 15(3), 48-53. May, 2001.
[7] Lapsley, D., and Low, S., "Random Early Marking: An Optimisation Approach to Internet Congestion Control," in Proceedings of IEEE ICON '99.
[8] Lapsley, D., and Low, S., "Random Early Marking for Internet Congestion Control," Proceeding of GlobeCom'99, pp. 1747-1752, 1999.
[9] Athuraliya, S., Lapsley, D., and Low, S., "An Enhanced Random Early Marking Algorithm for Internet Flow Control, "INFOCOM' 2000, Telaviv, Israel, pp. 1425-1434.
[10] 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.
[11] Ott, T., Lakshman, T., and Wong, L., "SRED: Stabilized RED," in Proc. IEEE INFOCOM, Mar. 1999, pp. 1346-1355.
[12] 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.
[13] 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.
[14] Woodward, M., E., "Communication and Computer Networks: Modelling with discrete-time queues, Pentech Press, London, 1993.
[15] Welzl, M., "Network Congestion Control: Managing Internet Traffic", 282 pages, July, 2005.