

On the Performance of a Simple Packet Rate Estimator

K. Salah

F. Haidari

Department of Information and Computer Science

King Fahd University of Petroleum and Minerals

Dhahran 31261, Saudi Arabia

Email: {salah,haidari}@kfupm.edu.sa

Abstract—This paper proposes a simple packet rate estimator that can be very useful in predicting the rate of network traffic. The quality and performance of the estimator is evaluated and compared with three popular rate estimators that were originally designed for estimating bit rate. The proposed estimator is highly cost effective as its computation is not carried out upon the arrival of each incoming packet. In addition, the computation is simple and does not depend on the measurement of interarrival times of packets. We evaluate and compare the quality and performance in terms of agility, stability, accuracy and cost. The performance evaluation is conducted using discrete-event simulation that produces synthesized bursty traffic with empirical packet sizes.

KEYWORDS: *Traffic Rate Estimation, Exponential Averaging, Simulation, Bursty Traffic, Performance Evaluation*

I. INTRODUCTION

Estimating the incoming arrival rate of network traffic is an integral component of network traffic management, monitoring and call-admission control. In Call Admission Control (CAC), the rate of the flows are monitored at the access points of the network in order to make decisions on admitting new flows [1]. For providing QoS-enabled Internet, the rates of current flows get also monitored for traffic policing to ensure senders comply to their SLAs (Service Level Agreement). SLA defines the traffic characteristics (especially the traffic rate) expected from the sender. Other than call admission control and policing, rate estimation is also an integral part of link-sharing [2] and fair scheduling algorithms [3] which are utilized in DiffServ and RSVP protocols [4].

Devising a proper rate estimator is not a trivial task. The quality of any estimator depends on multiple factors that may include agility, stability, accuracy, and overhead cost [3]. Agility defines the quickness of the estimator in following the changes in the actual data rate of the traffic. Stability (a.k.a. smoothing) refers to the ability of the estimator to ignore short-term changes or peaks of traffic. Accuracy refers to the difference between the estimated and the actual data rates of the traffic. Finally, the cost refers to the complexity of the algorithm and the required computational time and memory. In other words, a cost-effective estimator has to be simple, fast, and does not need to store much data.

All of the estimators utilized in [1-4] were designed originally for estimating bit arrival rate. However, little research has been done on estimating packet arrival rate. Estimators of packet arrival rate can be very useful and attractive in a variety of applications in networking and communications. The computation of packet rate is much simpler than bit rate as it does not depend on accumulation of packet sizes. In addition, a packet rate estimator can be designed to perform estimation at a periodic rate, as the case for our proposed estimator. Our proposed estimator is simple and does not depend on interarrival time measurements of packets and running estimation at every packet arrival, but utilizes a fixed time window. This can be an

indispensable design option for network elements of sensor networks nodes as well as PC-based routers and servers. Nodes in sensor networks are typically limited in computational power and energy. For PC-based routers and typical workstations and servers, avoiding summing up packet sizes and computation at every incoming packet will result in better overall performance. And more importantly, the NIC of these systems typically does not timestamp the arrival of each packet, and timestamping has to be performed by the kernel at interrupt handling of packet reception. This requires considerable overhead and is not doable in the default interrupt handling schemes of FreeBSD polling and Linux NAPI where interrupts of received packets are disabled.

Recently, the use of packet arrival rate estimators has been increasing. In [5], optimal routes through sensor network elements and optimal wakeup rate to minimize energy consumption are determined based on the incoming packet arrival rate to sensor nodes. In [6], an adaptive rate-scheduling based on the estimation and prediction of packet arrival rate has been proposed for minimizing QoS degradations in next generation CDMA wireless mobile systems. In [7], a queue management technique based on packet arrival rate was introduced for routers and switches to provide end-to-end high performance DiffServ QoS. In [8], the packet arrival rate of incoming traffic is utilized in a dynamic weighted fair queueing scheduler of routers. The weight gets adjusted dynamically based on the estimated packet arrival rate of each flow. In [9], an enhanced scheduling scheme of time varying fair queueing (TVFQ) was proposed using dynamic weights that get set according to estimated packet arrival rate. In [10], estimated packet rate is used to identify abnormal activities and attacks. In [11], estimated packet rate is used to reduce power consumption of network devices such as LAN switches. An interface on a switch can go to sleep (and thereby saving power) if the estimated next arrival time is long enough.

It is worth noting that, the packet arrival rate estimators reported in [5-9] employ typically an exponentially weighted moving average, described and named later in the paper as the static EWMA as it has a constant weight. In this paper, we describe and study a total of four different packet arrival rate estimators, including the static EWMA. One of these estimators, later called Time-Window-based EWMA, is a novel estimator that we propose. The performance and quality of these estimators are evaluated and compared. The performance is studied in terms of key estimation metrics which include agility, stability, accuracy, and cost. The performance is studied using DES (Discrete Event Simulation). Bursty traffic with empirical packet sizes is used to introduce realistic and highly variable network traffic.

The rest of the paper is organized as follows. Section 2 describes the algorithms of four packet rate estimators. Three of these estimators are modified versions of popular bit rate estimators. The fourth estimator is a proposed one by the authors. Section 3 compares the performance of the four

estimators in terms of stability, agility, accuracy and cost. Finally, Section 4 concludes the study and identifies future work.

II. PACKET RATE ESTIMATORS

In this section, we describe the algorithms of four estimators that can potentially be utilized for estimating the average packet arrival rate. Three of the popular algorithms that were originally used in estimating *bit* rate are modified to estimate *packet* arrival rate. A fourth algorithm is proposed. The proposed algorithm is simple and does not depend on the measurement of the interarrival times or computation upon the arrival of every packet, and yields acceptable performance in terms of accuracy, stability and agility.

A. TSW

Algorithm 1: TSW

Initialization:
 $Win_length = \text{a constant}$
 $Avg_rate = 0$
 $T_front = 0$

For each packet arrival do:
 $Pkts_in_TSW \leftarrow Avg_rate \times Win_length$
 $New_pkts \leftarrow Pkts_in_TSW + 1$
 $Avg_rate \leftarrow New_pkts / (now - T_front + Win_length)$
 $T_front \leftarrow now$
return Avg_rate

End

TSW (Time Sliding Window) is a time-based estimator that employs a rectangular data weighting function based on a fixed size time window. It essentially estimates the rate upon packet arrival and decays, or forgets, the past history over time. Algorithm 1 shows the modified version of the estimator presented in [12,13] to estimate packet rate. The algorithm in [12,13] was modified to exclude packet size calculations. The agility and stability of TSW depend on the size of the time window Win_length . If Win_length is small, the estimator will be more agile as the past takes less weight so that it can follow the new changes in the traffic rate quickly. On the other hand, if Win_length is large, the estimator will be more stable as the past takes more weight so that it can forget the short peaks which are very brief and only temporary. now is the time of the current packet arrival. T_front is the time of the last packet arrival. It is worth noting that computation of TSW estimator is performed on each packet arrival and measurements of interarrival times are required, and therefore may result in a considerable overhead, especially at high rate. In Ethernet of Gigabit link, the rate can go up to 1.4 million pps!

Algorithm 2: Static EWMA

Initialization:
 $\alpha = \text{a constant weight factor}$
 $Avg_rate = 0$
 $T_front = 0$

For each packet arrival do:
 $interarrival \leftarrow now - T_front$
 $Avg_rate \leftarrow (1 - \alpha) \times Avg_rate + \alpha \times interarrival$
 $T_front \leftarrow now$
return Avg_rate

End

B. Static EWMA

The static EWMA (Exponentially Weighted Moving Average) is a popular exponentially weighted moving average that is utilized in [2,11,12]. It has a constant weight. Algorithm 2 estimates the packet arrival rate. However, this estimator uses interarrival times and computation at every incoming packet, and thus requires more computational power and measurements, as is the case in TSW.

C. Dynamic EWMA

Dynamic EWMA is an EWMA estimator with dynamic weight that gets adjusted based on the temporary or persistent changes in the traffic over time [14]. Dynamic EWMA typically gives better performance than the conventional static EWMA with constant weight α . In Dynamic EWMA, temporary and brief traffic changes (peaks) are treated with lower weight; however, persistent and more permanent changes are treated with higher weight. These trends of changes are determined using the gradient of the occupied traffic. The complete Dynamic EWMA is described in Algorithm 3. The algorithm is devised such that it time-window based, meaning estimation is performed periodically and does not depend on the interarrival times of packets. α_{max} is typically chosen to be twice the value of the initial value of α_{opt} [14].

Algorithm 3: Dynamic EWMA

Initialization:
 $\alpha_{opt} = \text{a constant to represent the optimal weight}$
 $\alpha_{max} = \text{a constant to represent the maximum weight}$
 $Avg_rate = 0$
 $m_{norm} = \text{a constant which set to } (Link\ Capacity\ in\ pps) / (2 \times Time_Window)$
 $Time_Window = \text{a constant to represent the size of the jumping window}$

For each window do:
 $N \leftarrow \text{number of packets during this window}$
 $M_rate \leftarrow N / Time_Window$
 $m \leftarrow (M_rate - Avg_rate) / Time_window$
 $\alpha \leftarrow \alpha_{max} / (1 + |m| / m_{norm})$
 $Avg_rate \leftarrow (1 - \alpha) \times Avg_rate + \alpha \times M_rate$
return Avg_rate

End

D. Proposed EWMA

Our proposed EWMA algorithm is similar to the static EWMA but with a major difference. The proposed algorithm is time-window based, which means it does not require measurements of interarrival times of packets or computation on every packet arrival. Considerable overhead can be encountered if estimation is performed upon the reception of each incoming packet, especially at traffic arrival of high rate. In Gigabit Ethernet, a link may carry up to 1.4 million pps, and with 10 Gigabit links the problem can be significantly exacerbated. We propose a time-window-based EWMA estimator to alleviate such a constraint. Algorithm 4 describes the proposed algorithm of a time-window-based EWMA estimator. The purpose of the time window is two fold: First, it mitigates the computation at each incoming packets. Second, it is used to smooth out the burstiness of traffic and short peaks. In addition, there is no need for measuring interarrival times of packets. The window in the proposed EWMA is a jumping window (as opposed to TSW) where the window gets restarted after each window trigger to run the algorithm.

Algorithm4: Proposed EWMA

Initialization: α = a constant weight factor $Avg_rate = 0$ $Time_Window$ = a constant to represent the size of the jumping window**For each window do:** $N \leftarrow$ number of packets during this window $M_rate \leftarrow N / Time_Window$ $Avg_rate \leftarrow (1 - \alpha) \times Avg_rate + \alpha \times M_rate$ **return** Avg_rate **End**

III. EVALUATION AND COMPARISON

The performance of the four algorithms is evaluated using a DES simulation developed in C language and followed closely the guidelines in [15]. In order to introduce realistic and highly variable network traffic, bursty traffic with empirical packet sizes is used. To generate a synthesized bursty traffic, we implemented the method described in [16]. This method follows fractional Gaussian noise such as the resulting self-similar traffic is obtained by aggregating multiple streams (one stream per source) each consisting of alternating Pareto-distributed ON/OFF periods. In our simulation we used 8 sources, which are enough for producing realistic bursty aggregated traffic according to our own experimentation and findings, which are also in line with the observations in [17]. For the ON and OFF periods, we used the commonly-used shape parameters of $\alpha_{ON} = 1.3$ and $\alpha_{OFF} = 1.5$, respectively. During the ON period we generated packets with sizes that follow an empirical distribution, which are real measurements of packet sizes from MCI backbone. The measurements are reported in [18] and available online at <http://www.caida.org>.

For all of simulation runs (unless stated otherwise), we fix the weight factor α to 0.3 as used in [19], the maximum weight α_{max} of Dynamic EWMA to 0.6 as used in [14], and the window length of TWS to 10 ms. For simulation results of stability and agility, we record the average reference traffic rate every 10 ms (as opposed to say 1 ms) in order to reduce the fluctuation of the instantaneous data and thus plots become more readable. For the results of estimators, their output is recorded at the end of 10 ms interval (which indicates the predicted or estimated rate of the next interarrival time).

A. Stability

In this example we reduced the weight α to 0.125 in order to obtain clear graphs for stability. Fig. 1 compares the stability results of the four algorithms when subjected to an average aggregated bursty traffic of 150 kpps. It is shown that the conventional static EWMA performs poorly again, as shown on the left of Fig. 1. A zoom-in version of the figure is also shown on the right of Fig. 1. The zoom-in version excludes the results of the conventional static EWMA. It is clear that for the most part our proposed algorithm is more stable (not reacting much to short peaks) than all others.

B. Agility

Fig. 2 compares the agility results of the four algorithms when subjected to an average aggregated bursty traffic of different rates. To study agility, we introduce fluctuation in the offered traffic load, as shown. An average offered rate of bursty traffic of 150 kpps is used for 200 ms and then reduced to 50 kpps, and then increased after 200ms to 150 kpps again. It is shown that the

conventional EWMA overreacts and performs poorly. For the zoom-in version which excludes the static EWMA, it is obvious that our proposed algorithm TW-based EWMA reacts slowly to permanent traffic changes, unlike TSW and dynamic EWMA.

C. Accuracy and Cost

To measure accuracy we offered a constant load of 70% (corresponding to an average of 150 kpps) for 60 seconds and we measured the results of the relative estimation error. The estimators were evaluated in terms of accuracy based on the relative error between average estimated and average reference. Our simulation results showed that the Dynamic EWMA was the most accurate, and second to it was the proposed EWMA, outperforming TSW. The conventional static EWMA performed poorly.

TSW and static EWMA estimators are packet-based estimators as they require computation upon the arrival of each packet arrival. This can result in a significant overhead especially at high arrival rate of packets, which can go up to 1.4 million pps in Gigabit links. The Dynamic EWMA and our proposed EWMA estimators are window-based estimators, whereby computation is not performed on each packet arrival, but triggered periodically based on the configured time window size. When comparing the least overhead of the Dynamic EWMA and proposed EWMA, it is clear that our proposed EWMA requires the least overhead and processing. The Dynamic EWMA algorithm requires far more computational steps which involves significant division and floating point operations.

Summarizing, our TW-based EWMA outperforms the other three algorithms in terms of cost and stability and compares well in terms of accuracy and agility. For accuracy, the proposed algorithm is comparable to TSW and Dynamic EWMA. However, for agility, the algorithms of TSW and dynamic EWMA give relatively better performance.

IV. CONCLUSION

We presented three algorithms that can be potentially used for estimating packet rate: TSW, conventional static EWMA, and dynamic EWMA. We also proposed a novel fourth algorithm time-window-based EWMA that requires the least computational overhead. We used simulation with empirical packet sizes to study the performance of the estimators. As was demonstrated, our proposed TW-based EWMA estimator showed adequate performance in terms of accuracy and agility. However in terms of stability and overhead cost, the proposed estimator is superior. As further study, we are in the process of implementing the proposed estimator on Linux in order to measure its performance experimentally

ACKNOWLEDGMENTS

We acknowledge the support of King Fahd University of Petroleum and Minerals in completion of this work. This work has been funded under Project #ICS/Linux/383.

REFERENCES

- [1] M. Grossglauser and D. Tse, "A Framework for Robust Measurement Based Admission Control," IEEE/ACM Trans. Networking, vol.7(3), pp.293-309, June 1999.
- [2] S. Floyd and V. Jacobson, "Link-Sharing and Resource Management Models for Packet Networks," IEEE/ACM Trans. Networking, vol. 3, pp.365-386, Aug. 1995.
- [3] F. Agharebparast and V. Leung, "A New Traffic Rate Estimation and Monitoring Algorithm for the QoS-Enabled Internet," IEEE GLOBECOM, VOL. 7, pp. 3883-3887, 2003.

- [4] S. Shenker, P. Danzig, and S. Jamin, "Comparison of Measurement-based Admission Control Algorithms for Controlled-Load Service", Proceedings of IEEE INFOCOM '97, Kobe, Japan, pp. 973-980, April 1997.
- [5] R. C. Shah, D. Petrovic and J. Rabaey, "Energy Aware Routing and Data Funneling in Sensor Networks", book chapter in Handbook of Sensor Networks: Compact Wired and Wireless Sensing Systems, CRC Press, 2004.
- [6] Y. Oliver, E. Saric, and A. Li, "Adaptive Rate-Scheduling with Reactive Delay Control for Next Generation CDMA Wireless Mobile Systems," EURASIP Journal on Wireless Communications and Networking, vol. 06, pp. 1-15, 2006.
- [7] B. Wyrowski and M. Zukerman, "High Performance DiffServ Mechanism for Routers and Switches: Packet Arrival Rate based Queue Management for Class Based Scheduling," Proceeding of the 2nd International IFIP-TC6 Networking Conference on Networking Technologies, Services and Protocols, Springer-Verlag, 2002, pp. 62-73
- [8] C.-C. Li, S.-L. Tsao, M. C. Chen, Y. Sun, and Y.-M. Huang, "Proportional Delay Differentiation Service Based on Weighted Fair Queueing," in Proceedings of the 9th IEEE ICCCN'00, pp. 418-423, Las Vegas, October 2000.
- [9] A. Stamoulis, N. D. Sidiropoulos, and G. B. Giannakis, "Time-Varying Fair Queueing Scheduling for Multicode CDMA Based on Dynamic Programming," IEEE Transactions on Wireless Communications, vol. 3, no. 2, pp. 512-523, 2004.
- [10] J. Zimmerman, A. Clarck and G. Mohay, "The Use of Packet Interarrival Times for Investigating Unsolicited Internet Traffic," in Proceedings of the 1st International Workshop Systematic Approaches to Forensic Engineering, pp. 89-104, November 2005.
- [11] M. Gupta, S. Glover, and S. Singh, "A Feasibility Study for Power Management in LAN Switches," in Proceedings of the 12th IEEE International Conference on Network Protocols, pp. 361-371, October 2004.
- [12] D. Clark and W. Fang, "Explicit Allocation of Best-Effort Packet Delivery Service", IEEE/ACM Trans. Networking, vol. 6 (4), pp 362-373, August 1998.
- [13] W. Fang, N. Seddigh N, B. Nandy, "A Time Sliding Window Three Colour Marker (TSWTCM)", RFC 2859, June 2000.
- [14] L. Burgstahler and M. Neubauer, "New Modifications of the Exponential Moving Average Algorithm for Bandwidth Estimation", Proceedings of the 15th ITC Specialist Seminar, July 2002.
- [15] A. Law and W. Kelton, *Simulation Modeling and Analysis*, McGraw-Hill, 2nd Edition, 1991.
- [16] M. S. Taqqu, W. Willinger, and R. Sherman, "Proof of a Fundamental Result in Self-Similar Traffic Modeling" ACM/SIGCOMM Computer Communication Review, vol. 24, no. 2, 1997, pp. 5-23.
- [17] P. M. Wells, "Simulation of Self-Similar Network Traffic Using High Variance ON/OFF Sources, M.S. Thesis in Computer Science, May 2002.
- [18] K. C. Claffy, G. Miller, and K. Thompson, "The Nature of the Beast: Recent Traffic Measurements from an Internet Backbone," In the Proceedings of INET 1998, Geneva, Switzerland, July 1998.
- [19] C. Dovrolis, B. Thayer, and P. Ramanathan, "HIP: Hybrid Interrupt-Polling for the Network Interface," *ACM Operating Systems Reviews*, vol. 35, October 2001, pp. 50-60.

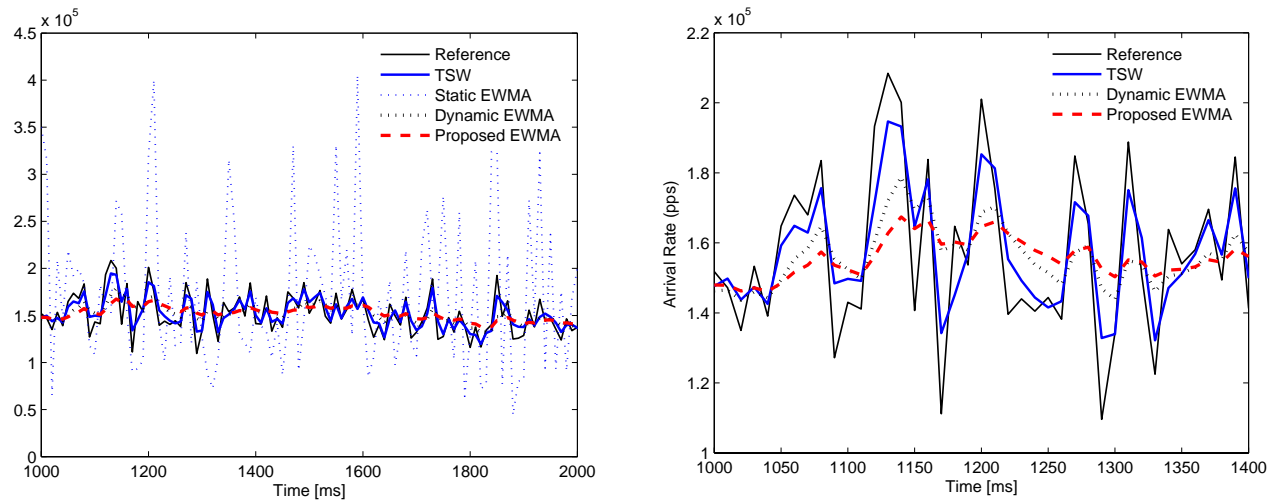


Figure 1. Stability comparison (left) with zoom-in (right)

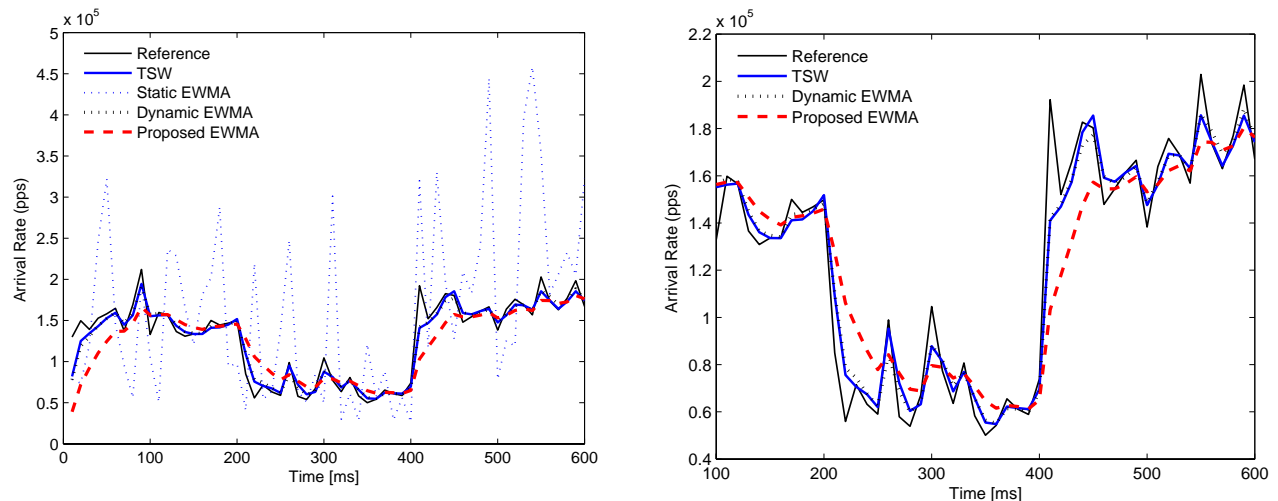


Figure 2. Agility comparison (left) with zoom-in (right)