KALMAN FILTER BASED CONGESTION CONTROLLER

Mehdi Mohtashamzadeh

Department of Computer Engineering, Soosangerd Branch, Islamic Azad University, Soosangerd, IRAN
Mohtashamzadeh@gmail.com

ABSTRACT

Facing burst traffic, TCP congestion control algorithms severely decrease window size neglecting the fact that such burst traffics are temporal. In the increase phase sending window experiences a linear rise which may lead to waste in hefty proportion of available bandwidth. If congestion control mechanisms be able to estimate future state of network traffic they can cope with different circumstances and efficiently use bandwidth. Since data traffic which is running on networks is mostly self-similar, algorithms can take advantage of self-similarity property and repetitive traffic patterns to have accurate estimations and predictions in large time scales.

In this research a two-stage controller is presented. In fact the first part is a RED congestion controller which acts in short time scales (200 milliseconds) and the second is a Kalman filter estimator which do RTT and window size estimations in large time scales (every two seconds). If the RED mechanism decides to increase the window size, the magnitude of this increase is controlled by Kalman filter. To be more precise, if the Kalman filter indicates a non-congested situation in the next large time scale, a magnitude factor is calculated and given to RED algorithm to strengthen the amount of increase.

KEYWORDS

Kalman filter, Self similarity, congestion control

1. INTRODUCTION

Ever increasing amount of data in networks may lead to congested networks, routers with full queues, increase of loss rate, and end-to-end latency. As a result effective throughput in networks would be plummeted. To deal with such challenges routers have to route packets faster and more precise. Towards this end congestion control algorithms have developed to enable the dramatic growth of networks, especially Internet. The original idea ([1]) was to adjust the transmission rate based on the loss probability. The first implementation of this mechanism was named TCP Tahoe and later refined into TCP Reno, New Reno, RED, REM and etc. These algorithms are now the dominating transport protocols on the Internet.

Almost all congestion control mechanisms cannot utilize available bandwidth since they react late and harsh to traffic bursts and cannot effectively recover network as they increase sending window size in a slow conservative linear manner. Admittedly, such manner may protect network resources from high level contentions. However, if mechanisms be able to get aware of network status in near future, they can control the magnitude of window size increase or decrease. Since network traffic imitate Heavy-tail distribution [2-4] and have Long Range Dependency (LRD) and self-similarity, network parameters can be predicted. Unfortunately, most network protocols

DOI : 10.5121/ijdps.2013.4307

have been designed according to Poisson probability distribution of network traffic. Researchers have proposed two main solutions, first to overwrite network protocols and other to modify existing control mechanisms to gain from self-similarity and LRD property. Current work belongs to the second solution and tries to improve the performance of TCP RED mechanism [5] using mathematical concept of Kalman filters to estimate Round Trip Time (RTT) and window size parameters in large time scales considering the fact that traffic is self-similar and repeated in time. As a light and precise estimation tool, Kalman filter has widely been deployed by researchers to control congestion in data networks. Authors of [6] have studied RTT estimation as a suitable parameter to control congestion using network state information. Investigating the characteristic of RTT and the application of the estimation, they concluded that the estimation objective is to keep the estimate smooth but still able to capture sudden mean changes. As a result a Kalman filter combined with CUSUM change detection was proposed and shown to perform well based on results of experiments on real data.Hoon Yang et al. [7] have proposed an LQ-Servo control mechanism with observer for TCP/AQM system. The controller's structure is composed of a traditional servo mechanism based on Linear Quadric approach. In order to estimate the state of TCP window packet size, they facilitate the control structure with a Kalman filter based observer.Last but not least, in [8] an Extended Kalman Filter based bandwidth estimation module which is capable of providing highly accurate link capacity estimations. Authors claim that their protocol-independent non-packet-probing module is distinct from other bandwidth estimators in two ways; firstly, rather than directly measuring link capacity, the module uses the persistent queue size and associated parameters. Second, their proposed module is claimed to overcome the degradation of the network utilization problem resulted by an increasing bandwidth value. They demonstrated that both XCP and VCP protocols with EKF-based Bandwidth Estimator could overcome the limitations of the original protocols and achieve significant performance improvements in network utilization by eliminating the protocol oscillatory behaviour.

2. TCP-RTT

In random early detection (RED) [5], packets are dropped before the buffer is full, with a probability that increases with the queue length. RED can thus be seen as a way of indirectly signalling the queue length to the source. RED is window-based which means that each sender has a window that determines how many packets are allowed to be sent at any given time. The transmission rate is regulated by adjusting this window.

During congestion avoidance phase the congestion window size (W) of senders grows linearly which cause linear increase in queue length. This growth continues as long as weighted average queue length (X) remains belowq_{min}; if X exceeds this threshold system should go through the delayed drop-notification phase. This phase is governed by a count-down timer T_k which expires in exactly one round trip time (m_k). During this some queued packets are dropped base on their calculated drop probability. This phase ends with the expiration of the $T_k$timer, after which the congestion window W is cut in half and a new timerT_k, is used which is initialized to the value of the current round trip time. Third phase is called recovery in which senders are not permitted to transmit packets, the congestion window W is kept fixed and the queue is going to be empty. This phase lasts for $0.5m_k$.In the last phase which also lasts $0.5m_k$ senders are allowed to transmit packets but the congestion window W is still kept fixed resulting in no change in queue length. The controller returns to the congestion avoidance phase after the expiration of $m_k$timer.

$$cwnd=\begin{cases} cwnd+\dfrac{1}{cwnd} & congestion\ avoidance \\ cwnd+\dfrac{1}{cwnd} & delayed\ notification \\ cwnd & recovery\ (not\ sending) \\ cwnd & recovery\ (sending) \end{cases}$$

As it can be seen in the above equation window size increase in “congestion avoidance” and “Delayed notification” phases is done in a linear manner which may reduce performance according to bandwidth waste. This shortcoming is challenged in this research and details are presented in the following section.

3. SELF-SIMILARITY OF TRAFFIC IN LONG TIME SCALE

Crovella et al. [9] studied 500 thousand of web requests. Senders were assumed as on/off traffic generators.They found that overall generated traffic is self-similar. In addition, authors of [10] have shown that in most cases Ethernet local network’s traffic is burst and self-similar. Moreover, TCP, Telnet, and FTP traffic types have been studied by authors of [11]. They understood that Poisson model which has been the dominant model of network traffic for plenty of years is not compatible with burst property of traffic in different time scales. Last but not least, research results of [12] depict that VBR traffic can easily be modelled by Heavy-Tail distribution, thus it is self-similar. In accordance with these demonstrations, self-similarity can be considered to improve network throughput via predicting network status.

4. KALMAN FILTER BASED CONGESTION CONTROLLER

The proposed design of current research is a two-stage controller (Fig. 1). The first part is a RED controller which function in short time scales and the second part is a Kalman filter which tries do estimate RTT and window size parameters every three seconds (long time scale) keeping in mind that traffic is self-similar and traffic patterns are repeated in long time scales.

The mathematical model of classic congestion controllers may serve their purpose well in most cases but the assumptions of the mathematical model are not correct in all circumstances, for example, presence of a very large number of network nodes and packet-flows may affect the calculations up to a fractional level which can accumulate into a high level of measurement uncertainty when the results are used as feedback control [13]. Moreover, a RED algorithm which operates on a router at some congested link and uses the exponentially weighted-average-queue-length to predict packet losses and impose flow control may use wrong parameters (weights) and can consequently lead to uncertainties. The aim of State Estimation is to separate the signal of interest from the measurement uncertainties and filter out the noise. Kalman filter State Estimator is used here to achieve these goals. The reason of Kalman filter selection is the well-known property of feedback systems. Feedback can compensate for model uncertainty. This is one of the properties which explain the broad usage of Kalman filters.

The proposed controller functions as follows. If the estimated value of window size shows a rise and the estimated RTT depicts a fall in comparison with the last long time scale, aggressiveness factor (α) is calculated and sent to the first part of the controller. The RED algorithm can use this factor to strengthen the amount of increase in “congestion avoidance” and “Delayed notification” phases considering the fact that in the next long time scale network may experience low level of contention (based on Kalman filter estimations). Although the long time scale controller can be implemented just based on RTT estimations, controller accuracy would be increased in case of using window size estimations along with RTT estimations.

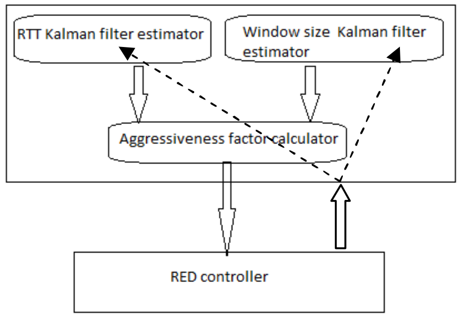

Figure 1.Two-stage congestion controller. The first part has a short time view of network, while the second part makes long time predictions according to self-similarity property of traffic patterns

Six congestion levels have been considered for the simulated network based on average (μ) and deviation (б) of RTT estimates. Each level can be mapped to a period [x,y) of RTT in which x and y are computed according to the following equations. It is worth noting that periods are recomputed at the end of every long time scale.

$$x=\begin{cases}-\infty & k=1\\ \mu-\left(\frac{m}{2}-k+1\right)\sigma & 1<k<m\end{cases}$$

$$y=\begin{cases}\mu-\frac{(m-2)}{2}\sigma & k=1\\ x+\sigma & 1<k<m\\ +\infty & k=m\end{cases}$$

"m" stands for quantity of levels in these equations.

In accordance with the above equations, six traffic levels results to six ranges of RTTs which are listed below:

(-∞, μ-2б) [μ-2 б, μ- б) [μ- б, μ) [μ, μ+ б) [μ+ б, μ+2 б) [μ+2 б, ∞)

Aggressiveness factor can then be calculated considering predicted traffic level (L).

$$\alpha = (1/L)\,A$$

"A" is maximum value of Aggressiveness factor. As a result, increase equation of RED algorithm can be represented as follow,

$$cwnd=\begin{cases}cwnd+\frac{\alpha}{cwnd} & congestion\ avoidance\\ cwnd+\frac{\alpha}{cwnd} & delayed\ notification\\ cwnd & recovery\ (not\ sending)\\ cwnd & recovery\ (sending)\end{cases}$$

The more the quantity of levels are, the more precise the controller will be. However, having more levels means more computational overheads.

4.1 Kalman filter estimators

There are two estimators functioning in the second part of the controller, one to estimate RTT value and the other one to provide window size estimations. RTT estimation model's equations are as follows,

$$RTT_k = A * RTT_{k-1} + B * U_k + Y_{k-1} \begin{cases} Y_{k-1} = \delta m_{k-1} \ (\delta \in \{0,1\}) \\ A = 1 \\ B = 0 \end{cases} \tag{1}$$

$$Z_k = RTT_k + V_k \tag{2}$$

Furthermore, time update and measurement update equations are listed in Table 1.

Table 1. Time update and measurement update equations of RTT estimator

time update	measurement update
$\dddot{\acute{RTT}}_k = R\acute{T}T_{k-1} + \delta_{k-1} * m_{k-1}$ $\overline{P_k} = P_{k-1} + Q$	$K_k = \frac{\overline{P_k}}{\overline{P_k} + R}$ $, R = var(V_k)$ $R\acute{T}T_k = R\dddot{\acute{T}}T_k + K_k \left(Z_k - R\dddot{\acute{T}}T_k\right)$ $P_k = (1 - K_k)\overline{P_k} + \delta_{t-1}Q$ $, Q = var(m_{k-1})$

The difference between the sampled RTT value and estimated RTT depicts the amount of measurement noise in each long time scale.

$$V_k = Z_k - R\acute{T}T_k \tag{3}$$

If sum of measurement noise till the current time scale exceeds a predefined threshold (0.05 in simulations), this noise should be considered in calculations ($Y_{k-1} > 0$). Otherwise, measurement noise is considered to be zero.

$$S_t = S_{t-1} + V_k$$

$$\begin{aligned} &\mathit{if}\ S_t > \mathit{threshold\ then} \\ &\quad \delta_t = 1, S_t = 0 \\ &\mathit{Else} \\ &\quad \delta_t = 0 \end{aligned}$$

The window size estimator has similar equations as follows,

$$w_k = A * w_{k-1} + B * U_k + Y_{k-1} \quad , \quad U_k = \delta n_{k-1} \tag{4}$$

$$Z_k = w_k + V_k \tag{5}$$

In which A=1 and B=0 (there is no control input in the system while controlling window size). As a result the last equation can be rewritten as

$$w_k = w_{k-1} + \delta n_{k-1} \tag{6}$$

Time update and measurement update equations are presented in Table 2. Measurement noise considerations are the same RTT.

Table 2. Time update and measurement update equations of window size estimator

time update	measurement update
$\dddot{W}_k = \acute{W}_{k-1} + \delta_{k-1} * n_{k-1}$ $\overline{P_k} = P_{k-1} + Q$	$K_k = \frac{\overline{P_k}}{\overline{P_k} + R} \quad , R = var(V_k)$ $\acute{W}_k = \dddot{W}_k + K_k \left(Z_k - \dddot{W}_k \right)$ $P_k = (1 - K_k)\overline{P_k} + \delta_{t-1} Q \quad , Q = var(n_{k-1})$

4.2. Selection of the long time scale

To design an efficient two time scale congestion controller an important issue is to specify time scales. As in Fig. 2 probability Pr{L2|L1=l} will be more concentrated if the large time scale increases. This concentration means having more precise predictability.

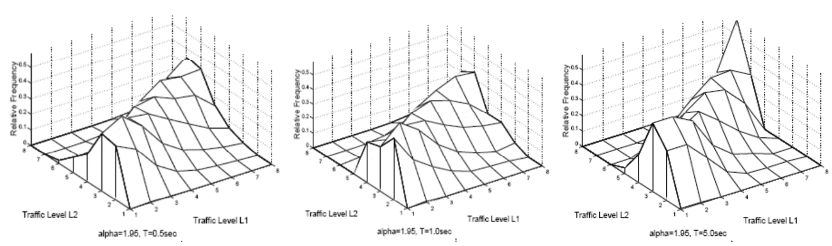

Figure 2. Probability density used to identify L2 according to recent traffic level L1 [14].

Entropy is a method by which we can discuss about distributions. Entropy is at maximum level when the probability distribution is not concentrated and otherwise entropy will be low.

For probability density function (Pi) entropy can be calculated as follow,

$$S(P_i) = \sum_i p_i \log 1/p_i \tag{7}$$

In this equation Pi is the probability density function. So in our case entropy is,

$$S_l = -\sum_{l'} \Pr\{L2 = l' | L1 = l\} \log \Pr\{L2 = l' | L1 = l\} \quad (8)$$

Authors of [14] have shown that increasing the time scale leads to reduction of entropy. Therefore larger time scales would be more adequate. Also it has been mentioned that decrease of entropy is negligible for time scales larger than 3 seconds.

5. RESULTS

Simulated network included 30 ON/OFF traffic generators to produce self-similar traffic. The average hop count for each packet in flight was set to be more than four. Furthermore, network parameter setting has considered as follows.

Table 3. Network parameter setting

	Variable	*Value*
RED Parameters	q_{min}	130
	q_{max}	270
	p_{max}	0.15
	weight	0.001
	Propagation delay (T_p)	0.1

According to measurements RTT estimation values and actual RTT sampling values show little differences (Fig. 3) and these differences become dispensable after 90 Seconds since Kalman filter corrects itself according to the past estimations and network states. A comparison between RTT results of pure RED congestion controller and Kalman controller depicts that the design with Kalman filter estimator leads to have a network with higher degree of throughput and low variant RTT.

Fast decreases of RTT in Fig. 3 at 50, 92, and 149 seconds are results of predicting low level of contention by the Kalman controller which enables the first part of the controller (RED) to increase the window size more aggressively.

In addition, RTT results related to pure RED congestion controller are presented in Fig. 4. By comparing the RTT results of Fig. 3 and Fig. 4, capability of Kalman filter based controller could be perceived, since the average RTT value for Kalman controller and RED controller are 0.190 and 0.223 respectively.

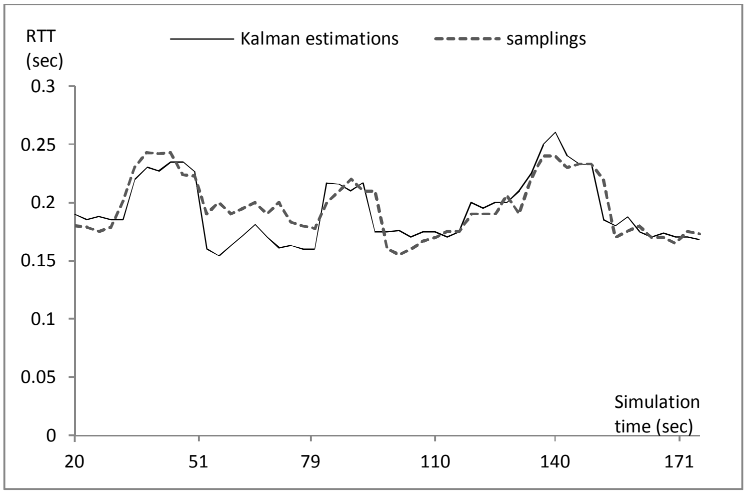

Figure 3. RTT estimation values made by Kalman filter vs. actual RTT samplings

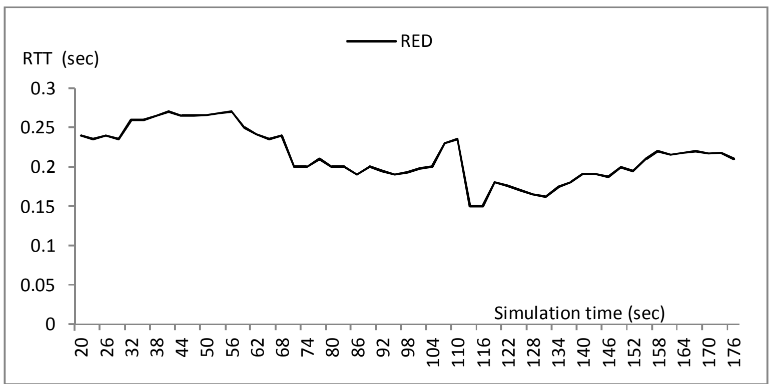

Figure 4. RTT results of RED controller

Moreover, queue length in routers was investigated. Overflowed router queues in a network can be interpreted as congested network. A comparison between the queue size of a particular node with the proposed mechanism versus the pure RED is presented in Fig. 5. it can be deduced that the Kalman filter approach has been more successful in controlling queue length parameter, which means that the network has experienced fewer congested situations. Fast falling slopes in this figure are due to improved window size increases performed by the first part of the controller based on precise estimations of the Kalman filter estimator functioning in the second part. In other words, detecting low level of contention, Kalman filter based controller triggers the first part to do window sizeincrease more aggressively. However, every controller has some uncertainties and measurement noises by which overall performance could be plummeted. Plenty of miss-predictions have been detected during simulations. A case in point is the fast rise of queue length in (538,542) period which is due to two false estimations. Apart from these anomalies, the proposed controller has outperformed the classic congestion controller in most cases.

In conclusion, there is no doubt that the measurement update and time update calculations which should be done in each long time scale impose computational overheads but according to the above discussed results the proposed Kalman filter approach may be an appropriate replacement for classic congestion controllers.

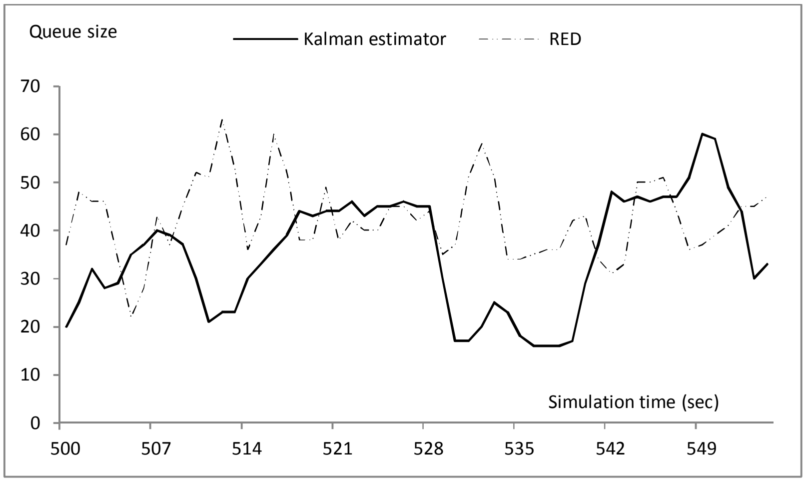

Figure 5. Queue length for the proposed structure vs. pure RED congestion controller

6. Conclusions

Accurate modelling of networks can efficiently help avoiding congestion. Although congestions can be eliminated completely when the model does not have any uncertainties and measurement noises, even the most accurate and fastest modelling and simulation techniques experience some extent of anomalies and uncertainties. Admittedly, these uncertainties can be negligible for a small scale network. However, studies have shown that a small fraction of noise may lead to disorders in networks with heavy traffic loads. A congestion avoidance modelling technique coupled with State Estimation can therefore be worthwhile as it is capable of predicting network status and eliminating congestions. Towards this end, in this research a RED congestion controller was combined with a Kalman filter estimator to do precise estimations of network parameters according to the self-similarity property of traffic patterns and to improve the functionality of classic controllers. Results have shown that the proposed structure improves network throughput.

References

1. Jacobson, V. (2004) “Congestion avoidance and control”, in symposium proceedings on Applications, Architecture and Protocols for Computer Communications, Stanford, California, US., 16-18 August, vol. 18, no. 4, p.314-329.
2. Paxon, V.& Floyd, S.(1995)“Wide-area traffic: The failure of poison modeling”, IEEE/ACM Transactions on Networking (TON), vol. 3, No 3, pp. 226--244. IEEE Press, New York.
3. Crovella, M.& Bestavors, A.(1996)“Self similarity in World Wide Web traffic: Evidence and possible causes”, International conference on measurement and modeling of computer systems (ACM SIGMETRICS). ACM, Philadelphia.
4. Willinger, W., Taqqu, M., Sherman, R. & Wilson, D.(1997)“Self-similarity through high-variability: statistical analysis of Ethernet LAN traffic at source level”, IEEE/ACM Transactions on Networking (TON), vol. 5, No 1, pp. 71--86. IEEE Press, New York.
5. Floyd, S. and Jacobson, V. (1993) “Random Early Detection Gateways for Congestion Avoidance”, IEEE Transactions on Networking, New York, United States, August 1993, Vol. 1, no. 4, p. 397-413.

6. Jacobsson, K., Hjalmarsson, H., Möller, N. & Johansson, K. H. (2004) “Estimation of RTT and Bandwidth for Congestion Control Applications in Communication Networks”, 43rdIEEE Conference on Decision and Control (CDC),Atlantis, Bahamas, December.
7. Yang, J. H., Lim, D. K., Lee, J.B. & Shin, S. J. (2012) “Robust Congestion Control based on Loop-Shaping Method with Kalman Filter for TCP/AQM System”, International Journal of Multimedia and Ubiquitous Engineering, Vol. 7, No. 2.
8. Huang, Z., Li, X. &Yousefizadeh, H. (2010) “Robust EKF-BasedWireless Congestion Control”, International Communications Conference (ICC2010), Cape Town, South Africa, May.
9. Crovella, M.& Bestavors, A.(1996)“Self similarity in World Wide Web traffic: Evidence and possible causes”, International conference on measurement and modeling of computer systems (ACM SIGMETRICS). ACM, Philadelphia.
10. Leland, W., Taqqu, M., Willinger, W.& Wilson, D. (1994) “On the self-similar nature of Ethernet traffic (extended version)”, IEEE/ACM Transactions on Networking (TON), Vol 2, No 1, pp. 1-15, February.
11. Paxon, V.& Floyd, S.(1995)“Wide-area traffic: The failure of poison modeling”, IEEE/ACM Transactions on Networking (TON), vol. 3, No 3, pp. 226--244. IEEE Press, New York.
12. Garret, W.&Willinger, W. (1994) “Analysis, Modelling and generation of self-similar VBR video traffic”, ACM SIGCOMM’94 Conference on Communications Architectures, Protocols and Applications, London, UK, September.
13. Nagar, A., Abbas, G.&Tawfik, H. (2006)“Online State Estimation of Communication Networks”, International Journal of Computer Science and Network Security (IJCSNS), vol. 6, No 5.
14. Park, K.& Tuan, T. (2000)“Performance evaluation of multiple time scale TCP under self-similar traffic condition”, ACM Transactions on modeling and computer simulation, vol. 10, No 2, pp. 152--177. ACM, New York.