Fuzzy Three Time Scale Congestion Controller

Mehdi Mohtashamzadeh and Mohsen Soryani

Computer Engineering Department, Iran University of Science & Technology, Tehran, Iran
Mohtashamzadeh@gmail.com, soryani@iust.ac.ir

Abstract. Although loss rate, length of the queues in the routers and throughput are affected by self similar property of traffic, but classic congestion control algorithms work in short time scales and do not consider self similarity and long range (time) dependency phenomenon of data. To profit these properties, researchers have proposed several methods. Multi time scale congestion control is one of the successful ways to adapt with self similar traffic and predict network status. In this research, a three part structure has been implemented in which second and third parts take advantage of fuzzy engines. Results show throughput improvement in case of using fuzzy three time scale controller instead of a two time scale controller or a classic controller such as New Reno.

Keywords: Self similarity, multi time scale, long range dependency, fuzzy congestion control.

1 Introduction

Most network protocols have been designed according to Poisson probability distribution for network traffic, but researchers found that network traffic imitate Heavy-tail distribution [1-3] and have self similarity property. Erramilli et al. [4] proved that Long Range Dependency (LRD) which is the most important property of self similar traffic, strongly affects network performance. Also author of [5] showed undesirable effect of self similarity on loss rate and queue length. To diminish these effects, 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. Multi time scale congestion control [5-7] is a technique which belongs to the second solution. Controllers of this type have multiple sub-controllers inside, in which first part is a classic controller which acts in short time scales (Reno, Tahoe …) and controllers in other two parts make decisions based on prediction of traffic levels in future and decision of the first part. False predictions may lead to network failure, so controllers of different parts must be precise enough to predict accurate traffic levels in future.

2 Fuzzy Three Time Scale Congestion Controller

Three time scale controller performs congestion control in three stages, short time scales (20-200 ms), mid time scales (3 seconds) and at the longer intervals (12 seconds). The part of algorithm which works in the short time scales is a classic congestion controller

(New Reno). This type of controller increases size of sending window in linear form, after achieving a threshold. Assume the data traffic which is running on the network is burst; it means that sometimes there are a lot of data on the network and mostly often network traffic is very low. Since classic congestion control algorithms perform window size increase in the linear way, most of available bandwidth would be wasted in case there is no burst traffic. Therefore to take advantage of bandwidth, the second and third parts of controller (mid time and long time controllers) should be active and enter more data to the network, by augmenting the increase factor of window. In fact the second and third parts do not change decision which is taken by the first part, but they make mid and long time decisions compatible with decision of the first part. For example if the first part decides to increase the size of window, second and third parts make increasing factor strengthening or weakening, but do not lead to decrease in window size. To have a steady congestion controller, two fuzzy controllers have been used beside a classic short time scale controller. The three time scale controller functions as follows. If second and third parts of the controller sense an increase in bit rate by the first part, increasing factor would be (α+β) in which α and β are figured out by the second and third parts of controller, respectively. Considering future status of network is essential to fuzzy controllers. If fuzzy controllers predict low level traffic for future, they can consider α and β large enough to profit bandwidth. Each fuzzy controller presents a predicted traffic level at the end of its time scale. α and β can be acquired by reversing these predicted values. Predicting large traffic level results to small values for α and β and consequently a small increasing factor. Before describing the controller, let's take a look at some essential prerequisites such as specifying traffic levels and selecting time scales.

2.1 Specifying Traffic Level

To specify traffic level at previous time scale, quantity of sent bits should be mapped to one of eight congestion levels (assume there are eight traffic levels). The controller forms a time series and saves quantity of sent bits in that every 200 milliseconds. Then average (μ) and standard deviation (σ) is computed. Each level stands for a range of bit rate. These ranges (which are in [x,y) format in this paper) can be figured out using following equations.

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

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

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

In accordance with the above equations, having eight traffic levels results to have eight ranges of bit rates which are listed below:

$(-\infty, \mu-3\alpha)$ $[\mu-3\alpha, \mu-2\alpha)$ $[\mu-2\alpha, \mu-\alpha)$ …. $[\mu+2\alpha, \mu+3\alpha)$ $[\mu+3\alpha, \infty)$

2.2 Selection of the Second Time Scale

To design an efficient multi time scale congestion controller, an important issue is to specify time scales. As in Fig. 1 probability $\Pr\{L2|L1=l\}$ will be more concentrated if the large time scale increases. This concentration means having more precise predictability.

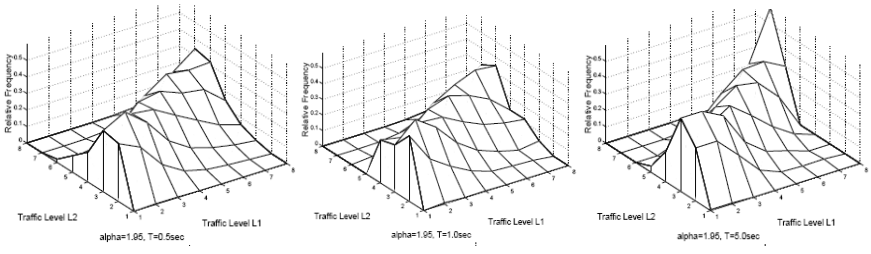

Fig. 1. Probability density used to identify L2 according to recent traffic level L1 [5]

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

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

$$S(P_i) = \sum_i p_i \log 1/p_i \tag{3}$$

where Pi is the probability density function. So in our case entropy is,

$$S_l = -\sum_{l'} \Pr\{L2 = l' | L1 = l\} \log \Pr\{L2 = l' | L1 = l\} \tag{4}$$

Authors of [5] 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.

2.3 Selection of the Third Time Scale

Choosing a proper time scale for the third part of controller depends on several issues such as types of first and second parts (short-time and mid-time controllers), second time scale, self similarity of traffic and average time of connections.

Hagivara et.al [8] have shown that in case of having self-similar traffic (e.g. H=0.9) burst property of throughput would be evident in 10-seconds time scales, and

for larger time scales traffic will be more smooth. The authors have mentioned that if the self similarity is to some extent low (e.g. H=0.5) burst property would not be apparent enough, so shorter time scales should be chosen.

2.4 Mid Time Fuzzy Controller [9]

To make decisions, controller should be aware of recent traffic level. Traffic level can be attained according to bit rate at last mid time scale, using described intervals in section "2.1". The controller has two input linguistic variables and an output. Input variables refer to traffic level and throughput experienced at the last mid time scale.
The controller is of Mamdani type and has 40 rules. Performance of the controller is severely affected by selected rules.

2.5 Long Time Fuzzy Controller

Third part of the design takes advantage of a Mamdani fuzzy controller with three input linguistic variables namely "Recent traffic", "Predicted traffic" and "Effective throughput" which refer to traffic level in the last mid time scale, predicted traffic level by mid time controller and effective throughput in the last long time scale, respectively. This controller has also an output linguistic variable namely "Predicted traffic2" which its membership functions are the same as membership functions of "predicted traffic" input.

Table 1 presents some of the rules used for long time scale controller.

Table 1. Fuzzy rules used in long time scale part of the fuzzy controller

If (Predicted-traffic1 is VL) and (Recent-traffic is VL) and (Effective-throughput is VH) then (Predicted-traffic2 is L)
If (Predicted-traffic1 is VL) and (Recent-traffic is VL) and (Effective-throughput is M) then (Predicted-traffic2 is H)
If (Predicted-traffic1 is VL) and (Recent-traffic is M) and (Effective-throughput is M) then (Predicted-traffic2 is M)
If (Predicted-traffic1 is VL) and (Recent-traffic is M) and (Effective-throughput is H) then (Predicted-traffic2 is H)
If (Predicted-traffic1 is VL) and (Recent-traffic is L) and (Effective-throughput is H) then (Predicted-traffic2 is H)

3 Simulation and Results

3.1 Producing Self-similar Traffic Using ON/OFF Traffic Generators

Simulations have been done using Omnet++ version 3.3 [10]. To have self-similar traffic during simulations, workstations have taken advantage of Pareto distribution in

Omnet++ simulator to generate Heavy-tailed ON/OFF traffic, in which both on and off times conform Pareto distribution. In this research, 30 ON/OFF traffic generators have been used. Fig. 2 presents generated traffic for 1000 seconds of simulation for various values of α (α is shape parameter in Pareto distribution) and bit rate. Network consists of 30 workstations. Each node runs fuzzy three time scale congestion control algorithm to detect congestion and make proper decisions to utilize network resources in case of low traffic. As expected, in accordance with equation $H=(3-\alpha)/2$ having a smaller value of α leads to a greater Hurst parameter (H) and a traffic which is more self-similar, as in Fig. 2 (Down-right).

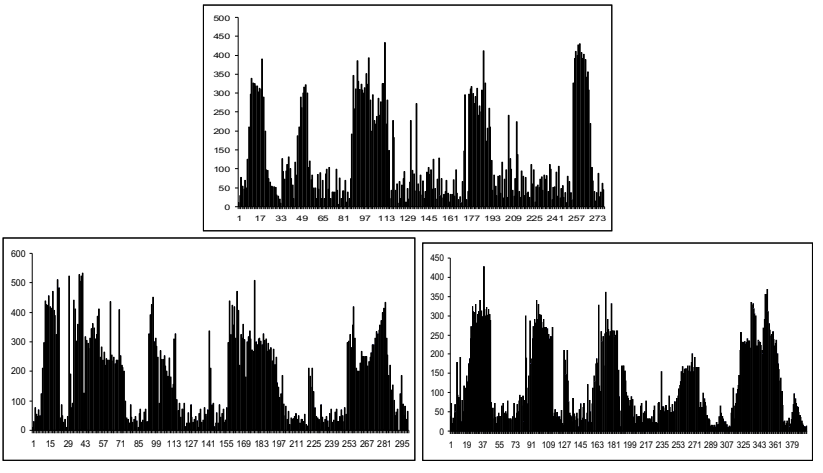

Fig. 2. Generated traffic, simulation time 1000 seconds, α=1.5 and bit rate 22 Mbps (up) Generated traffic, simulation time 1000 seconds, α=1.5 and bit rate 27 Mbps (Down-left) Generated traffic, simulation time 1000 seconds, α=1.05 and bit rate 27 Mbps (Down-right)

3.2 Second Time Scale

Results show that in case of using larger time scales, entropy is lower. For example with T2=1sec (T2 is the second time scale in our design) Entropy is approximately 0.71 and for T2=3sec it is 0.65. According to [5] entropy does not change for time scales larger than 3 seconds (it shows negligible increase). So T2 has been considered to be 3sec.

3.3 Third Time Scale

The third time scale (T3) has been considered to be 10, 12 and 15 seconds in simulations. Table 2, 3 and Table 4 show results of simulations for 1000 seconds. Tian et al. [11] have shown that enlarging time scales leads to more smooth traffics.

Also authors of [8] have shown that in case of a self similar traffic with large value of H (e.g. 0.9) burst property of throughput rate is clear enough in 10 seconds time scales and as time scale increases, clarity of burst property becomes less.

Table 2. Results for 1000 seconds of simulation and T3=10 seconds

L1/L2	1	2	3	4	5	6	7	8	E(L2|L1=…)
1	0.23	0.35	0.3	0.05	0.03	0	0.01	0.03	2.49
2	0	0.22	0.1	0.2	0.1	0.23	0	0.15	4.62
3	0.11	0	0.12	0	0.27	0.28	0.22	0	5.04
4	0.01	0	0.25	0.28	0.3	0.14	0.01	0	4.29
5	0.13	0	0.17	0.26	0.1	0.32	0.02	0	4.24
6	0	0.14	0	0.01	0.11	0.21	0.22	0.31	6.17
7	0	0.08	0	0.11	0.06	0.31	0.14	0.3	6.14
8	0	0	0.09	0	0.22	0.26	0.32	0.11	6.05

Table 3. Results for 1000 seconds of simulation and T3=12 seconds

L1/L2	1	2	3	4	5	6	7	8	E(L2|L1=…)
1	0.33	0.36	0.22	0.05	0.04	0	0	0	2.11
2	0.29	0.29	0.31	0.02	0.07	0	0.02	0	2.37
3	0.11	0.15	0.28	0.27	0.18	0	0	0.01	3.31
4	0.13	0.08	0.08	0.19	0.17	0.12	0.13	0.1	4.57
5	0.02	0	0.22	0.21	0.18	0.27	0.09	0.08	5.31
6	0	0.01	0	0.1	0.24	0.21	0.21	0.23	6.19
7	0	0.02	0.07	0	0.06	0.22	0.24	0.41	6.83
8	0	0.02	0	0.12	0.02	0.31	0.28	0.25	6.44

Table 4. Results for 1000 seconds of simulation and T3=15 seconds

L1/L2	1	2	3	4	5	6	7	8	E(L2|L1=…)
1	0.17	0.08	0	0.24	0.06	0.13	0.11	0.21	4.82
2	0.24	0.12	0	0	0.1	0.43	0.05	0.06	4.39
3	0.34	0.04	0.02	0.03	0.24	0	0	0.33	4.44
4	0.11	0	0.2	0.18	0.23	0.14	0	0.24	5.34
5	0	0.26	0	0.11	0.07	0.05	0.07	0.44	5.62
6	0.31	0.02	0.01	0.18	0.11	0	0.01	0.36	4.6
7	0.21	0.18	0.21	0	0.09	0.02	0.06	0.23	3.76
8	0.22	0.1	0.1	0.2	0	0	0.1	0.28	4.46

So among these time scales 10 seconds would prepare better predictability. As the above tables show, for T3=15 seconds results are non-concentrated and Long Range Dependency (LRD) is not obvious. In contrast, for 10 and 12 seconds, tables' values show LRD property well. Results for T3=12 seconds seem to be better, so T3 has been considered to be 12 seconds in further simulations.

3.4 Number of Traffic Levels

Increasing traffic levels may augment controller performance. For example in case of having eight traffic levels, values μ+3α, μ+4α and μ+5α are all related to traffic level 8, whereas in case of having twelve levels, the above mentioned values are related to levels 10, 11 and 12 respectively. Facing various traffic levels, fuzzy controller would respond in different ways.

Although level increase may cause performance improvement, it has some drawbacks. To have a reliable controller, average and deviation should be computed every 200 milliseconds. Hence at the end of each long time scale, time periods' bounds should be computed because of changes in average and deviation values. So, expanding the number of traffic levels leads to computation overheads.

In this research, 8 levels of traffic have been considered. Fig. 3 presents results of having different number of traffic levels. Simulations have shown that few changes in quantity of levels make it necessary to have a new set of fuzzy rules.

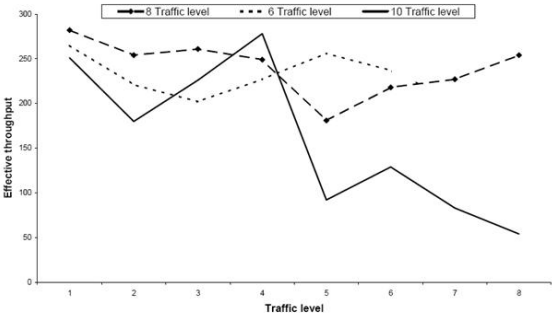

Fig. 3. Fuzzy three time scale controller in case of having 6, 8 and 10 traffic levels

3.5 Comparisons

In this section, performance of the fuzzy three time-scale congestion controller is compared to the performance of a fuzzy two time-scale controller. This comparison is presented in Fig. 4 in which, throughput related to each traffic level is the average of throughput values corresponding to that level during simulation. One can deduce that in most cases fuzzy three time-scale controller performs better than two time-scale controller. Also Fig. 5 presents performance improvement in case of replacing a classic controller with fuzzy two and three time-scale controllers based on different values of shape parameters (α). For example for $\alpha = 1.05$ replacing the classic controller (New Reno in our case) by a three time-scale controller would increase the throughput 15.5% and by a two time-scale controller throughput would be 14% greater.

Fig. 5 demonstrates that using fuzzy three time-scale controller leads to achieve more throughput except for $\alpha = 1.95$. In such a case modifying fuzzy rules may remove this drawback. As mentioned in section "3.2", increasing α would decrease Hurst parameter and self-similarity. Also it is possible to slow down the speed of performance decrease for shape parameters larger than 1.5 by improving rules.

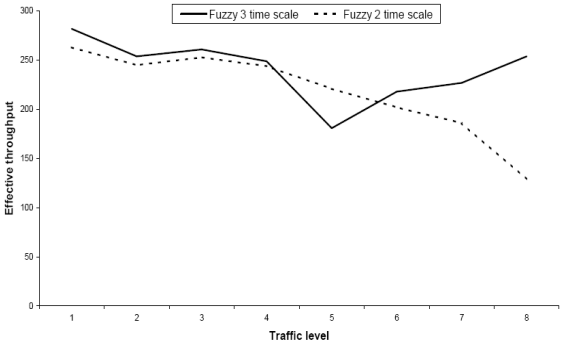

Fig. 4. Comparing performance of fuzzy two and three time-scale controllers

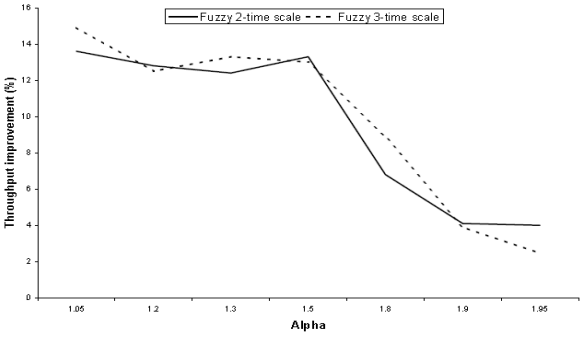

Fig. 5. Percentage of performance improvement in case of replacing New Reno controller by fuzzy two and three time-scale controllers

4 Conclusions

Self similarity causes kind of disorders in performance of classic controllers. To have congestion control algorithms which consider self similarity, researchers have proposed two methods; first to rewrite all protocols to get on with self similarity and second to modify existing protocols and standards. Our implementation has been based on the second solution. In addition to the New Reno controller which acts in short time scales (200-500 milliseconds), two other parts have been used to predict network status in larger time scales (up to 12 seconds). Results have shown that the proposed structure would improve network throughput.

Network traffic has been categorized into 8 levels. The long time scale controller predicts Future traffic level based on prediction of the second part, latest traffic level and throughput of the network. According to the predicted level, an increase factor will be obtained by which amount of bit rate increase (decision of first part) is controlled. For future works, data mining techniques can be used instead of fuzzy logic. Accuracy of a fuzzy engine depends on programmer experience of extracting

useful items from data and preparing rules; in contrast, data mining techniques use various statistical methods and train data (which can be related to a real network, not a simulated one) to predict.

References

1. Paxon, V., Floyd, S.: Wide-area traffic: The failure of poison modeling. IEEE/ACM Transactions on Networking (TON) 3(3), 226–244 (1995)
2. Crovella, M., Bestavors, A.: Self similarity in world wide web traffic: Evidence and possible causes. In: International Conference on Measurement and Modeling of Computer Systems (ACM SIGMETRICS). ACM, Philadelphia (1996)
3. Willinger, W., Taqqu, M., Sherman, R., Wilson, D.: Self-similarity through high-variability: statistical analysis of Ethernet LAN traffic at source level. IEEE/ACM Transactions on Networking (TON) 5(1), 71–86 (1997)
4. Erramilli, A., Narayan, O., Willinger, W.: Experimental Queuing analysis with long range dependence packet traffic. IEEE/ACM Transactions on Networking (TON) 4(2), 209–223 (1996)
5. Park, K., Tuan, T.: Performance evaluation of multiple time scale TCP under self similar traffic condition. ACM Transactions on Modeling and Computer Simulation 10(2), 152–177 (2000)
6. Lu, J., Ruan, Q., Ni, R.: Fractal-based multiple time scale TCP-friendly congestion control for multimedia Streaming. In: 18th Canadian Conference on Electrical and Computer Engineering, Saskatchewan (2005)
7. Lu, J., Ni, R.: Media Streaming TCP-Friendly Congestion Control Using Multiple Time Scale Prediction. In: Second International Conference on Innovative Computing, Information and Control, Kumamoto, vol. 1(1), pp. 535–540 (2007)
8. Hagivara, T., Majima, H., Matsuda, T., Yamamoto, M.: Impact of Round Trip self-similarity on TCP performance. In: 10th International Conference on Computer Communications and Networks (2001)
9. Mohtashamzadeh, M., Soryani, M., Fathy, M.: Fuzzy two time-scale congestion control algorithm. In: International Conference on Computational Intelligence, Communication Systems and Networks. IEEE, Indore (2009)
10. Omnetpp V3.3 simulator, `http://www.omnetpp.com`
11. Tian, X., Wu, H., Ji, C.: A unified framework for understanding network traffic using independent wavelet models. In: Proceedings of IEEE Infocom 2002, New York (2002)