Fuzzy Logic Control for SFB Active Queue Management Mechanism

Nguyen Kim Quoc[1(✉)], Vo Thanh Tu[2], and Nguyen Thuc Hai[3]

[1] Faculty of IT, Nguyen Tat Thanh University, Ho Chi Minh, Vietnam
nkquoc@ntt.edu.vn
[2] Faculty of IT, Hue University of Science, Hue, Vietnam
vothanhtu_hue@yahoo.com
[3] Institute of IT, Ha Noi University of Technology, Hanoi, Vietnam
haint.fit@gmail.com

Abstract. Internet is growing not only in the number of connected devices but also the diversity of the application layers. Therefore, the bottleneck problem in the router is a pressing issue in congestion control. So using the mechanisms of active queue management for congestion control at routers is playing an important role for the reliable and effective Internet network operation for users. Mechanism of active queue management SFB works well in the router, but not highly effective. Therefore, we propose to incorporate intelligent computation through fuzzy logic control system into the mechanism SFB which can operate more efficiently to improve service quality and network performance.

Keywords: Congestion control · Active queue management · Fuzzy logic controller

1 Introduction

Engineering active queue management (AQM: Active Queue Management) is mechanism controlling queue and loading operations at the routers. AQM controls the number of packets in the router queues by scheduling, removing proactively a coming packet or notifying blockage to regulate traffic on the network [2, 4, 5, 12]. In recent years, researchers have proposed a number of queue management mechanisms in routers based on the size of the queue (such as RED…) [19], packet loss factor and performance airtime usage (such as BLUE…)[7–9]. However, these mechanisms do not ensure good fairness for flows [6, 11, 22]. The mechanism SFB (Stochastic Fair BLUE) activities based on BLUE mechanisms ensure fairness for the flows, but do not achieve high throughput, haven't got low packet loss rate and small queue used space yet, so latency is still high [24]. Therefore, in this paper we propose to make active queue management mechanism FSFB (Fuzzy SFB) by using fuzzy logic controller (FLC) integrated active queue management mechanism SFB to proactively detect and control congestion better [10].

The results of the analysis and evaluation of simulation experiments based on NS2 software [13, 18] installed show that: queue management mechanism FSFB actively works well at each router, reduces packet drop and the latency and increases

P.C. Vinh et al. (Eds.): ICCASA 2013, LNICST 128, pp. 102–114, 2014.
DOI: 10.1007/978-3-319-05939-6_11,

throughput of the flows. Therefore, the new queue management mechanism FSFB has improved network performance and responded quickly to the changes of network traffic of packets on the transmission line, so the quality of network services enhanced.

This article consists of five parts. The first part points out the necessity of queue management and proposed idea of new queue management mechanism FSFB. The second part discuses new queue management mechanism FSFB and relating issues. The third part focuses on the new queue management FSFB of the authors. The fourth part shows the process of simulation installation, test of experimental results with the process of theoretical study. The last part compares the performance of proposed queue management mechanism with the current queue management mechanism to make judgments and conclusions.

2 Related Works

2.1 Operation of SFB Mechanism

SFB divided queue into calculation bins, each bucket maintains a packet marking probability p_m. This probability increased /decreased linearly depending on the packet drop rate or extent of use of the transmission line. If at queue, there is a continuousness of packet cancellation because large transduction overflows queue, it will increase p_m, increases severity of obstructive message that it will sends back to the source. Conversely, if the queue becomes empty due to weak transduce or idle transmission line, then packet marking probability p_m reduces. Packet marking probability of each bin is determined as follows [23, 24]:

Based on the packet loss: if ((now-last_update) > freeze_time) then

$$p_m = p_m + \delta_1 \text{ and } Last\_update = now \tag{1}$$

Based on the idle connection: if ((now-last_update) > freeze_time) then

$$p_m = p_m - \delta_2 \text{ and } Last\_update = now \tag{2}$$

Where, p_m packet marking probability, δ_1 the increasing amount of p_m, δ_2 the reducing amount of p_m, *now* current time, *last_update* last time when p_m changed, *freeze_time* amount of time between successful changes.

The bins are organized in L levels, each level has N bins. In addition, SFB uses L independent hash functions, each function corresponding to a level. Each hash function maps a flow into one of the coming bins at this level. The bins are used to track and capture the statistics of queue occupation of the packet in that bin. When a packet comes to queue, it is hashed into L bins, each level is a bin. If the number of packets mapped in a bin exceeds a certain threshold (e.g., the size of the bin), the probability p_m in that bin increases. If the number of packets in the bin is reduced until the end, p_m reduces. Figure 1 below shows the operation model of the active queue management mechanism SFB:

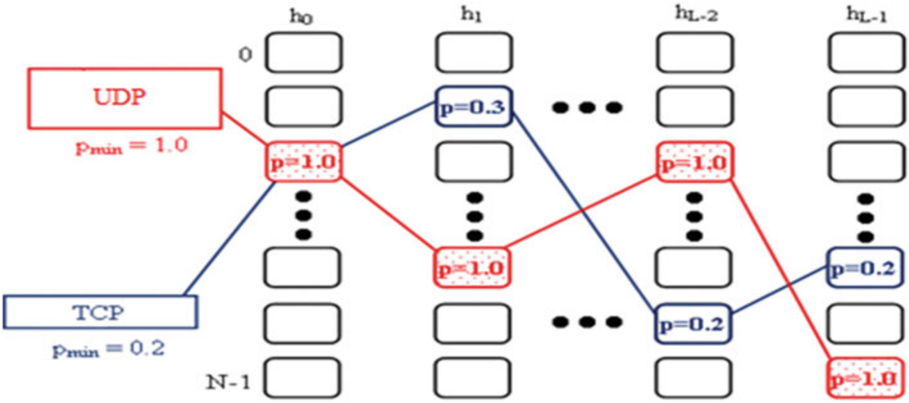


Fig. 1. The operation model of the mechanism SFB

Observation shows that with an unresponsive flow hashed into L bins, probability p_m at the bins rapidly rises to 1. The responsive flows can share one or two bins with unresponsive flows. However, if the number of unresponsive flows is not too large as compared to the number of bins, the responsive flow is able to be hashed at least into one bin without unresponsive flow, thus there is normal value p_m. The marking decision of a package based on p_{min} that is the minimum value of p_m of the mapped bins. If p_{min} is 1, the packet is defined as unresponsive flow and limited transmission speed of flow.

Here, the flows are defined as limited and unresponsive flows to save bandwidth. This strategy is done by limiting the speed of packet flowing in queue for flows with p_{min} value of 1. Picture above is an example that shows how SFB works. An unresponsive flow mapped into the bins, marking probability in these bins is 1. While TCP flows can be mapped into the same bin with the unresponsive flows at a particular level, it can also be mapped into the bins at other levels. Thus, the lowest marking probability of TCP flows is less than 1, so it is not defined as unresponsive flows. On the other hand, when the marking probability of unresponsive flows is 1, it will be limited transmission speed.

2.2 Effect of Boxtime Parameter

In SFB, all unresponsive flows are processed as a whole. How many the bandwidth used for the unresponsive flows has? It depends on the key parameter Boxtime. Boxtime is the interval without packet of such unresponsive flow coming into the queue. When a packet of UDP flow comes, if it is detected as packet of unresponsive flow, SFB will compare the current time with the nearest time when a packet of any unresponsive flow comes to the queue. If the period of these two events is greater than Boxtime, the packet will be in the queue, otherwise it will fall. If it is in the queue, the current time is updated for the next comparison. By this way, Boxtime indirectly controls how bandwidth is used for unresponsive flows. The large parameter Boxtime means that unresponsive flows can only achieve a low throughput, average queue length of the UDP flows is small. Conversely, if the value of Boxtime is small, the average queue size of the UDP flow is large. It is reasonable when the value of small Boxtime results throughput for large unresponsive flows. From the Boxtime is a static parameter, it can only be set by hand and hard to configure automatic adaptation, Boxtime value in a case cannot be applied to other case. This is a main drawback of SFB that should be addressed.

To improve fairness among UDP flows, we propose a method to create Boxtime as a random bit. By this way, the fairness among UDP flows is improved. However, this method only improves the fairness of unresponsive flows when they are limited the speed to create stability of bandwidth through the bottleneck transmission line. The high bandwidth streams will have higher mark probability as compared to the low bandwidth streams.

2.3 SFB Algorithm

Step 1: Calculate the hash functions (h_0, h_1,.., h_{L-1}).

Step 2: Check at each level. If the bin size is larger than allowed limit, then goes through Step 3. Conversely, if the bin is empty, goes through Step 4, if not goes through Step 5.

Step 3: Check if the interval from last update of the bin to the present time is greater than the allowed threshold, the increase of packet dropping probability (p) appears, goes through Step 5.

Step 4: Check if the interval from last update of the bin to the present time is less than the allowed threshold, the reduction of packet dropping probability (p) appears, goes through Step 5.

Step 5: Check if the smallest probability at the bins of packets mapped is of 1, the transmission speed of the flows is limited, in contrast coming packet is marked with probability p. Figure 2 shows the SFB algorithm.

3 Proposed Fuzzy Approach

3.1 Fuzzy Logic Controller

Fuzzy logic controllers, such as expert systems, can be used to model human experiences and decision making human behavior. FLC in the input-output relationship is expressed by using a set of linguistic rules or relational expressions. A FLC basically consists of four important parts including a fuzzifier, a defuzzifier, an inference engine and a rule base. In many fuzzy control applications, the input data are often clear, therefore, a fuzzification is necessary to convert the input crisp data into an appropriate value set with linguistic that is needed in inference engine. Singleton fuzzifier is the general fuzzification method used to map the crisp input to a singleton fuzzy set. In the rule base of a FLC, a set of fuzzy control rules, which characterizes the dynamic behavior of system, is defined. The inference engine is used to form inferences and draw conclusions from the fuzzy control rules. Figure 3 shows the fuzzy logic

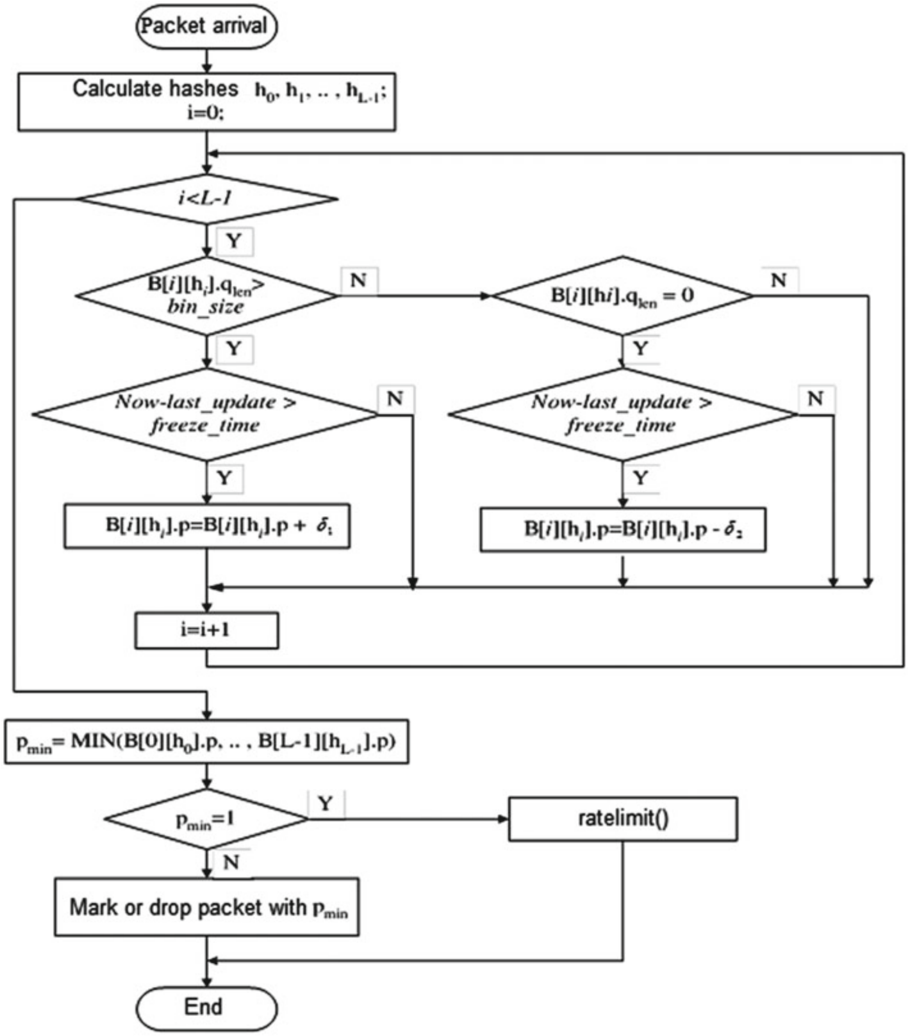


Fig. 2. Flowchart of algorithm of SFB mechanism

controller architecture. The output of inference engine is sent to defuzzification unit. Defuzzification is a mapping from a space of fuzzy control actions into a space of crisp control actions [26].

Suppose the FLC has *n* input variables x_1, x_2,.., x_n. Furthermore, suppose the rule base consists of *K* rules with the following general form: IF (x_1 is A_1)$\wedge \ldots \wedge$ (x_i is A_i) $\wedge \ldots \wedge$(x_n is A_n) THEN y is B. Where in the A_i and B are fuzzy sets of linguistic variables x_1, x_2,.., x_n and *y* respectively. The output function *f(X)* of this fuzzy controller with singleton fuzzifier, inference engine of result and center-average defuzzification method can be calculated as follows:

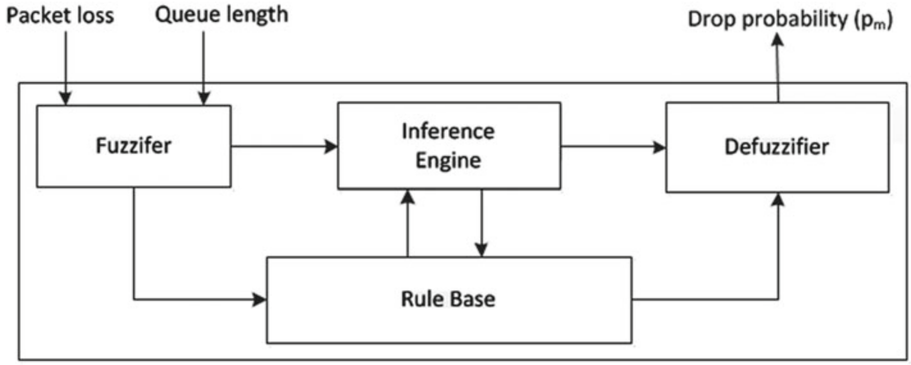


Fig. 3. Architecture of fuzzy inference system

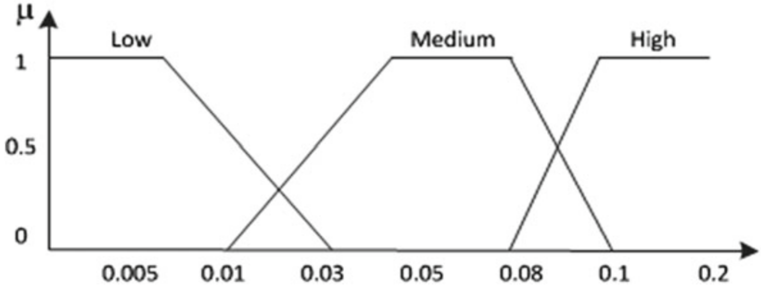


Fig. 4. Membership function of packet loss rate

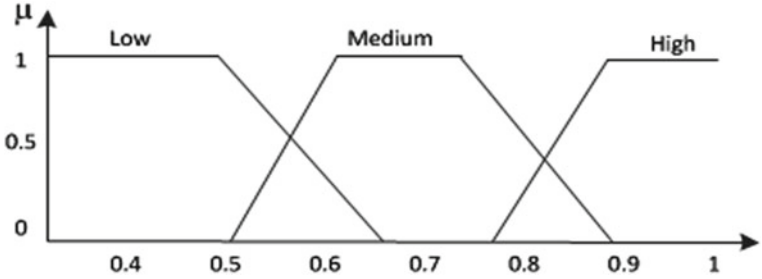


Fig. 5. Member function of level of using the size of the queue

$$f(x) = \frac{\sum_{j=1}^{k} y_0^j \cdot \prod_{i=1}^{n} \mu_i^j(x_i)}{\sum_{j=1}^{k} \prod_{i=1}^{n} \mu_i^j(x_i)} \tag{3}$$

Where y_0 is the center value of the output fuzzy set b, $\mu\ (x)$ is the membership function for fuzzy sets. In our proposed model, we use two input variables for fuzzy controller which shows the current congestion including the packet loss rate and current queue length and the output will be the packet making probability value.

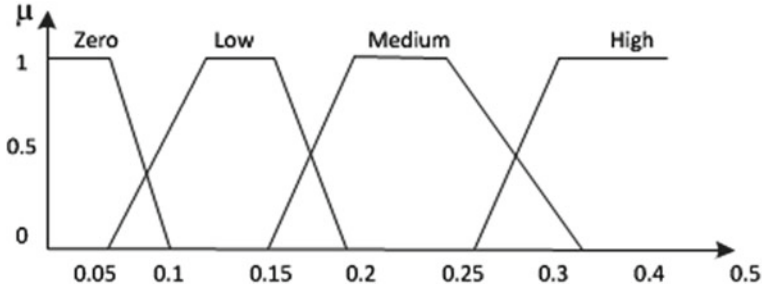


Fig. 6. Membership function of packet loss probability

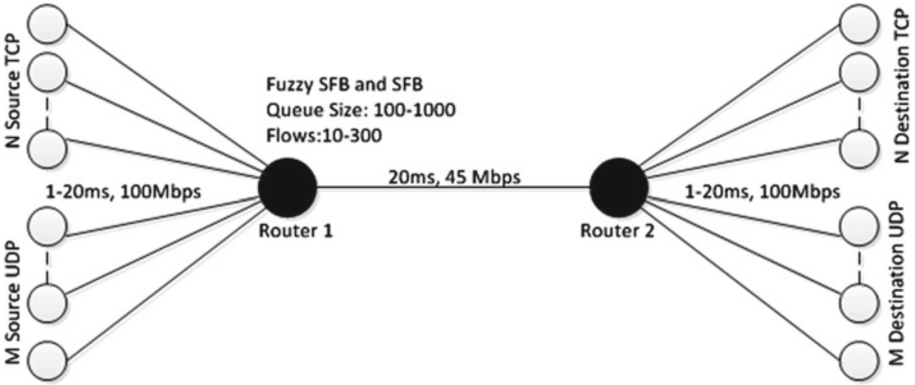


Fig. 7. Network simulation model

3.2 Linguistic Variables and Membership Functions

Input linguistic variables are variables representing the main affecting factors on the operation mechanism SFB. Here, we select packet loss factor, level of queue use to make input linguistic variables and the probability of packet loss used as output linguistic variables. Because the method of fuzzy triangular /trapezoidal is simple and effective noise reduction, so we choose this method to construct the membership function for the linguistic variable input and output (Figs. 4, 5, 6).

3.3 Construction of Fuzzy Rules

The fuzzy IF-THEN rules are built on experience from the experimental results and the value of the membership functions of the linguistic variables. There are two approaching methods: trial and error approaching method based on the knowledge gained from the experiment, a set of rule base based on IF-THEN rules and then system is tested. If the experimental results are deduced from the unsatisfactory laws, the laws will be amended. This process is continued until the function of the controller is satisfied. Based on functions of experiment and theory, we build rules in the rule base as follows:

Rule 1: **if** packet loss is low **and** queue length is low **then** p_m is zero;
Rule 2: **if** packet loss is low **and** queue length is medium **then** p_m is zero;
Rule 3: **if** packet loss is low **and** queue length is high **then** p_m is zero;
Rule 4: **if** packet loss is medium **and** queue length is low **then** p_m is zero;
Rule 5: **if** packet loss is medium **and** queue length is medium **then** p_m is zero;
Rule 6: **if** packet loss is medium **and** queue length is high **then** p_m is medium;
Rule 7: **if** packet loss is high **and** queue length is low **then** p_m is zero;
Rule 8: **if** packet loss is high **and** queue length is medium **then** p_m is low;
Rule 9: **if** packet loss is high **and** queue length is high **then** p_m is high;

4 Simulation and Results

4.1 Simulation Settings

During experimental process, network model is described according by following model: in simulation, we use N flows TCP and M unresponsive flows UDP responses flows. The transmission lines from source TCP and UDP to bottleneck and from bottleneck to destinations has a 100 Mbps bandwidth, latency is changed from 1 to 20 ms. Transmission line in the script is the link between two routers. We put the transmission bandwidth is 45 Mbps and the latency is 20 ms. Router at bottleneck is installed algorithms to evaluate and queue size at bottleneck changes in each circumstance [14, 21, 22] (Fig. 7).

In addition, parameters such as packet size of all TCP and UDP flows are set to 1000 bytes, TCP window size is 2000 packets, transmission speed of UDP flows changes in the simulation as an evaluation basis. Selected simulation time is 60 s.

Parameters for the mechanisms: δ_1, δ_2, freeze_time, N, L, bin_size, Boxtime. In particular, δ_1 is set large enough as compared to δ_2. We have chosen the following values: δ_1 is 0.0025, δ_2 is 0.00025 and freeze_time is 10 ms. N, L depends on the amount of flows to the bottleneck, if the number of unresponsive flows is large while N and L are small, the TCP flows are easy to be classified error layer as unresponsive flows. In our simulations, set as its default value is N = 6 and L = 2. Bin_size is set to equal of (1.5 / N)* queue size. We set the value for Boxtime as its default is 50 ms. However, this parameter must be calculated for each specific network model. So maybe it is ideal for a case but cannot be good for other case.

4.2 Evaluation Metrics

The performance evaluation of congestion control mechanisms is usually through criteria such as packet loss probability at place where congestion occurs, achieved network throughput, transmission line utilization level, the level of fairness of transmission line when the together connection to the transmission bottleneck and the queue utilization level at bottleneck [1, 3, 15–17, 20, 25].

Packet loss rate: The ratio of the number of loss packet and the total sending package. For stability network, the rate is low, whereas this rate is very high. Packet loss rate is determined by the formula:

$$\textit{packet loss percentage} = \frac{\sum_{i=1}^{N} \textit{packet loss}}{\sum_{1}^{N} \textit{packet sent}} \tag{4}$$

Transmission line utilization level: As the ability to take advantage of network traffic that said the index's ability to communicate through the network connection is strong or weak and is calculated by the following formula:

$$\textit{utilization} = \frac{\textit{byte\_departures}_t}{\textit{bandwidth} \times t} \tag{5}$$

Where utilization is the level of using transmission lines, byte_departurest is the number of bytes transmitted in t seconds, the bandwidth is the bandwidth of the transmission line and t is time of transmission.

Fairness level: is level of flows in network with ensuring fairness of connections when network has many other throughput types. Level of fairness is 1 when throughput of flows is equal, unless when throughput of flows is unequal, this value is less than 1. This value demonstrates greater, assurance of the congestion control algorithms is well. Fairness level is calculated as following formula:

$$\textit{fairness} = \frac{\left(\sum_{i=1}^{N} x_i\right)^2}{N \times \sum_{i=1}^{N} x_i^2} \tag{6}$$

In particular, Fairness is fair level of flows, Fairness $\in [0, 1]$, x_i: is the throughput of flow i and N is the number of flows.

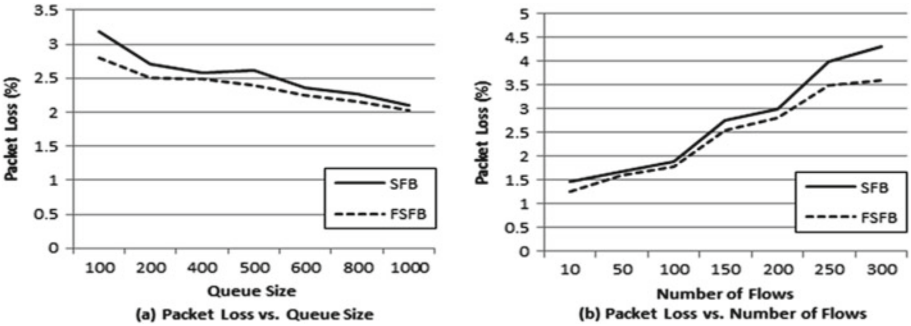
Average Queue Size: The index indicates directly the level of resource use at router. This index is defined as the ratio of the average queue size to the actual size of the queue. Mechanism with this small ratio will have small latency at the queue and risk of overflew queue is low. In contrast, the mechanism will make large latency and high risk of overflew queue. We use the quadratic average of control deviation to be index of queue utilization level and it is defined as:

$$S_e = \sqrt{\frac{1}{M+1}\sum_{i=0}^{M} e_i^2} = \sqrt{\frac{1}{M+1}\sum_{i=0}^{M} \left(Q_i - Q_{ref}\right)^2} \tag{7}$$

In particular, Q_{ref} is the queue size, Q_i is the queue size at the ith sampling time and M is the number of samples.

4.3 Evaluation of Packet Loss Rate

From the graph Fig. 8, we see that the queue size in the router increases, the packet loss rate of mechanisms reduces and when the number of connections to the router increases, the packet loss rate increases. In all cases, SFB always has the highest packet loss rate and the FSFB always have the lowest packet loss, when the queue size of 400 or more and the number of connections is less than 100, the packet loss rate of FSFB less than 2.5 %.



(a) Packet Loss vs. Queue Size

(b) Packet Loss vs. Number of Flows

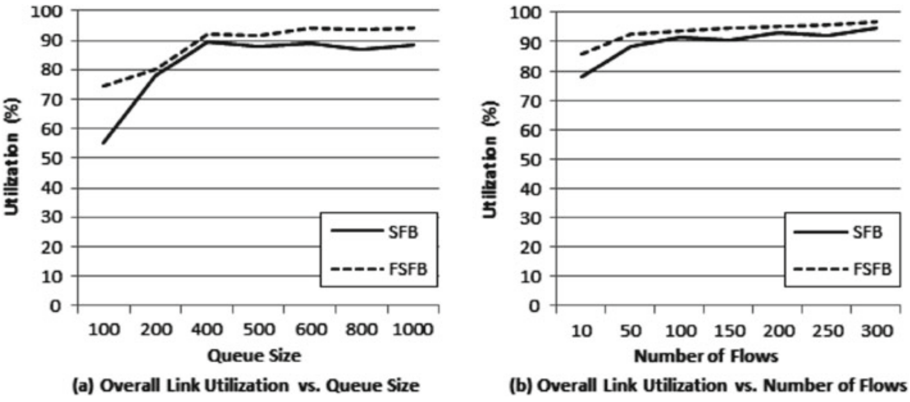
Fig. 8. Packet loss rate of the mechanisms of active queue management

4.4 Evaluation of Transmission Utilization Level

The graph in Fig. 9 shows the level of transmission line utilization of the mechanisms. The ability to take advantage of the transmission line utilization of the mechanisms increases, when the queue size and loading (number of connection flows) increases. When the queue size from 400 and over or the number of connections into router from 100 and over, mechanism FSFB uses better of transmission line, transmission rate used is over 90 %, and is always higher than the mechanism SFB.

4.5 Evaluation of Fairness

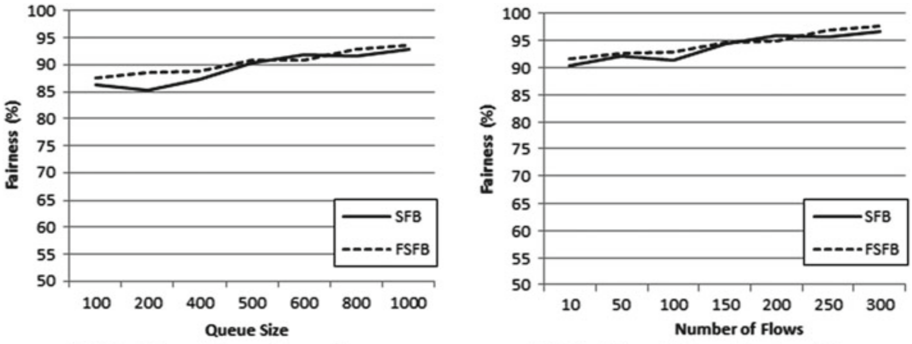
Based on the graph of Fig. 10 shows the fairness of the algorithm, we found that the fairness level of the algorithm by SFB and FSFB is very high at over 80 % for all cases. Particularly, mechanism FSFB always balance over 90 % in the cases which there are the changed number of connection flows.



(a) Overall Link Utilization vs. Queue Size

(b) Overall Link Utilization vs. Number of Flows

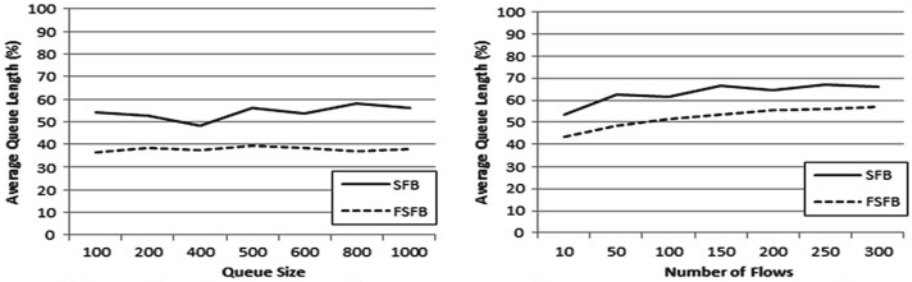
Fig. 9. The usage level of the transmission line of mechanisms of active queue management



(a) Jain's Fairness Index vs. Queue Size

(b) Jain's Fairness Index vs. Number of Flows

Fig. 10. The balance of the mechanism of active queue management



(a) Average Queue Length vs. Queue Size

(b) Average Queue Length vs. Number of Flows

Fig. 11. The usage of the active queue management mechanisms

4.6 Evaluation of medium queue size

Based on the simulation results and graph demonstrating usage rate of the queue size of algorithm in Fig. 11, we found that FSFB usage level is always lower than SFB, in cases of the changing queue size, this figure is less than 40 %, and less than 60 % for all cases having changed flows. This matter makes the latency and the ability to overflow queue at routers of low mechanism FSFB.

5 Conclusion

Internet facing boom in connectivity, applications and services based on it. The congestion control by mechanisms of the active queue management in routers is essential. However, putting intelligent computing factors, fuzzy control into mechanisms of the active queue management, so these mechanisms operate more efficient, to improve quality of service and network performance. In this paper, we have changed the mechanism SFB of queue management by introducing fuzzy logic controllers involved in the process of calculating the probability of packet mark based on the level of packet loss and queue use level at the router. Experimental simulation based on software NS2 to the traditional mechanisms SFB and SFB with fuzzy controller (FSFB) in the same network model, showed FSFB has low packet loss rate, the use of high transmission and small latency at router queue. So FSFB controls and conducts congestion control better than the FSB. Results of the study group would contribute to the study of the world to improve network performance, enhance network quality of service.

References

1. Kapadia, A., Feng, W., Campbell, R.H.: GREEN: a TCP equation-based approach to active queue management. U.S Department of Energy through Los Alamos National Laboratory contract W-7045-ENG-36 (2011)
2. Dana, A., Malekloo, A.: Performance comparison between active and passive queue management. Int. J. Comput. Sci. Issues (IJCSI) **7**(3) No 5, 13–17 (2010)
3. Bitorika, A., Robin, M., Huggard, M., Goldrick, M.: A Comparative Study of Active Queue Management Schemes. Department of Computer Science Trinity College, Dublin (2011)
4. Wydrowski, B.P.: Techniques in Internet Congestion Control. Electrical and Electronic Engineering Department, The University of Melbourne (2003)
5. Wydrowski, B., Zukerman, M.: GREEN: an active queue management algorithm for a self. ARC Special Research Centre for Ultra-Broadband Information Networks, EEE Department, The University of Melbourne, Parkville, Vic. 3010, Australia (2010)
6. Ali Ahammed, G.F., Banu, R.: Analyzing the performance of active queue management algorithms. Int. J. Comput. Netw. Commun. (IJCNC) **2**(2), 1–19 (2010)
7. Thiruchelvi, G., Raja, J.: A survey on active queue management mechanisms. Int. J. Comput. Sci. Netw. Secur. (IJCSNS) **8**(12), 130–145 (2008)
8. Hasegawa, G., Murata, M.: Analysis of dynamic behaviors of many TCP connections sharing tail-drop – RED. CybermediaCenter, Osaka University, Japan (2003)
9. Chandra, H., Agarwal, A., Velmurugan, T.: Analysis of active queue management algorithms & their implementation for TCP/IP networks using OPNET simulation tool. Int. J. Computer Appl. (0975 – 8887) **6**(11), 12–15 (2010)
10. Tabash, I.K., Mamun, M.A.A., Negi, A.: A fuzzy logic based network congestion control using active queue management techniques. J. Sci. Res. **2**(2), 273–284 (2010)
11. Chung, J., Claypool, M.: Analysis of active queue management. Computer Science Department Worcester Polytechnic Institute Worcester, MA 01609, USA (2008)
12. Chitra, K., Padamavathi, G.: Classification and performance of AQM-based schemes for congestion avoidance. Int. J. Comput. Sci. Inform. Secur. (IJCSIS) **8**(1), 331–340 (2010)
13. Fall, K., Varadhan, K.: The ns Manual, A Collaboration between researchers at UC Berkeley, LBL, USC/ISI, and Xerox PARC (2010)
14. Le, L., Jeffay, K., Donelson Smith, F.: A loss and queuing-delay controller for router buffer management. In: Proceedings of the 26th IEEE International Conference on Distributed Computing Systems (ICDCS'06) (2006)
15. Le, L., Aikat, J., Jeffay, K., Donelson Smith, F.: The effects of active queue management and explicit congestion notification on web performance. IEEE/ACM Trans. Netw. **15**(6), 1217–1230 (2007)
16. Kwon, M., Fahmy, S.: A comparison of load-based and queue-based active queue management algorithms. Department Of computer Science, Purdue University, West Lafayette, IN 47906-1398, USA (2010)

17. Soyturk, M., Design and Analysis of TCP/IP Networks. G.Y.T.E., Spring (2011)
18. Ns-2.: Network Simulator. http://www.isi.edu/nsnam/ns
19. Floyd, S., Jacobson, V.: Random early detection gateways for congestion avoidance. Lawrence Berkeley Laboratory, University of California (1993)
20. Athuraliya, S., Low, S.H., Li, V.H., Yin, Q.: REM active queue management. IEEE Netw. **15**(3), 48–53 (2001)
21. Ahmad, S., Mustafa, A., Ahmad, B., Bano, A., Hosam, A.A.: Comparative study of congestion control techniques in high speed networks. IJCSIS **6**(2), 222–231 (2009)
22. Bhaskar Reddy, T., Ahammed, A., Banu, R.: Performance comparison of active queue management techniques. IJCSNS Int. J. Comput. Sci. Netw. Secur. 9(2), 405–408 (2009)
23. Feng, W., Kandlurz, D.D., Sahaz, D., Shin, K.G.: BLUE-a new class of active queue management algorithms. Department of EECS Network Systems, University of Michigan (2009)
24. Feng, W., et al.: Stochastic fair blue: a queue management algorithm for enforcing fairness. In: Proceedings of IEEE, April 2001
25. Stallings, W.: High-Speed Network: TCP/IP and ATM Design Principles. Prentice Hall, Upper Saddle River (1998). ISBN 0-13-525965
26. Zadeh, L.A.: Fuzzy sets. Inf. Control **8**, 333–353 (1965)