International Journal of Control, Automation, and Systems, vol. 5, no. 3, pp. 317-323, June 2007

Lyapunov-based Fuzzy Queue Scheduling for Internet Routers

Hyun Cheol Cho, M. Sami Fadali, Jin Woo Lee, Young Jin Lee, and Kwon Soon Lee*

Abstract: Quality of Service (QoS) in the Internet depends on queuing and sophisticated scheduling in routers. In this paper, we address the issue of managing traffic flows with different priorities. In our reference model, incoming packets are first classified based on their priority, placed into different queues with different capacities, and then multiplexed onto one router link. The fuzzy nature of the information on Internet traffic makes this problem particularly suited to fuzzy methodologies. We propose a new solution that employs a fuzzy inference system to dynamically and efficiently schedule these priority queues. The fuzzy rules are derived to minimize the selected Lyapunov function. Simulation experiments show that the proposed fuzzy scheduling algorithm outperforms the popular Weighted Round Robin (WRR) queue scheduling mechanism.

Keywords: Fuzzy inference, internet routers, Lyapunov stability, queue scheduling.

1. INTRODUCTION

The Internet is used for many different applications with diverse QoS requirements ranging from the best-effort (no guaranteed QoS) to deterministic guaranteed QoS. Consequently, there is a growing demand for more effective network resource allocation to achieve desirable service levels while improving network performance. For example, a variety of Active Queue Management (AQM) schemes [1] have been proposed to prevent incipient network congestion. Typically, congestion is avoided by discarding an incoming packet to a router according to a loss probability determined by an active queuing controller.

Differentiated services (Diff-Serv) architecture has also been developed to address such QoS issues in IP networks. Diff-Serv identifies traffic flows as classes and provides each class with a different service level. A separate queue is maintained for each differentiated class and an efficient queue scheduling algorithm is required. A queue scheduler effectively allocates network resources by selecting a packet in the classified queue to access a single physical link with fixed capacity. Since there are no industry standards [2], various scheduling mechanisms can be freely selected depending on the network parameters (such as link bandwidth, time delay, etc.).

Queue scheduling is based on Priority Queuing (PQ), which has a relatively simple structure. Packets are first classified and assigned to the appropriate queue based on their priorities. The packets in the higher priority queue are first scheduled to be transmitted and a First-In-First-Out (FIFO) policy is followed within each queue. PQ requires low computation complexity and is easy to implement but has several limitations when applied to complex networks [2].

Fair Queuing (FQ), proposed by Nagle [3], ensures that traffic flows have fair access to network resources and prevent a burst in flow from overflowing its fair share of link capacity. In [4], several advantages of FQ are addressed including fair allocation of bandwidth, lower delay for sources, and protection from ill-behaved sources. A major limitation, however, is that FQ allocates the same bandwidth to each flow over time. FQ does not support flows with different bandwidth requirement.

To address the limitations of PQ and FQ, WRR or Class-based Queuing (CBQ) was developed for the next generation of multi-service IP networks [5]. Unlike PQ and FQ, WRR allocates bandwidth based on a weight assigned to each queue.

In [5], the authors evaluated the ALTQ/CBQ

Manuscript received September 13, 2006; accepted March 14, 2007. Recommended by Editorial Board member Poo Gyeon Park under the direction of Editor Jin Young Choi. This work was supported by the Korea Science and Engineering Foundation under grant KOSEF 2003-02-25. This work was supported by the Korea Science and Engineering Foundation (KOSEF) through the National Research Lab. Program funded by the Ministry of Science and Technology (No. M103000 0030306J000030310).

Hyun Cheol Cho, Jin Woo Lee, and Kwon Soon Lee are with the Dept. of Electrical Engineering, Dong-A University, 840, Hadan 2-dong, Saha-gu, Pusan 604-714, Korea (e-mails: hyunccho@gmail.com, jinwoo@donga.ac.kr, kslee@dau.ac.kr).

M. Sami Fadali is with the Dept. of Electrical Engineering, Univ. of Nevada-Reno, Reno, NV, 89557, USA (e-mail: fadali@unr.edu).

Young J. Lee with the Dept. of Avionics Electrical Eng., Korea Aviation Polytechnic, 438, Egeum-dong, Sacheon, Korea (e-mail: airlee011@hotmail.com).

* Corresponding author.

implementation and discussed parameter sensitivities and effective deployment in real-time networks. ALQT (Alternate queuing) is a queuing framework for UNIX systems, which offers a range of queuing schemes to realize resource sharing and quality of service. An adaptive WRR was addressed in [6], in which the weight parameters are updated by using revenue as a target function. Due to the adaptation of the mechanism, the proposed adaptive WRR could perform even in a nonstationary network environment. Nevertheless, the primary limitation of WRR is the difficulty of dynamically assigning an appropriate weight to each service class.

Recently, fuzzy machine scheduling algorithms were proposed for flexible manufacturing systems [7,8]. They were shown to be more sophisticated strategies than conventional scheduling policies such as CLB, CAF, and CPK [9]. In [8], fuzzy rule bases are determined based on a Lyapunov function candidate to stabilize the scheduler whenever possible. Queue selection decreases the Lyapunov function change but no priority is specified.

We propose a new dynamic and adaptive queue scheduling mechanism based on [8]. We modify their fuzzy scheduling by considering priority, which is an important factor for class-based queue scheduling of Internet routers. The inputs of the fuzzy queue scheduler are classified queues, their queue levels, and priorities. Accordingly, the proposed fuzzy scheduler selects a queue to be served and transmits the first packet in the selected queue. We evaluate the effectiveness of the fuzzy scheduler and compare it to WRR through simulation experiments.

The paper is organized as follows. In Section 2, we present the new fuzzy scheduling strategy. Simulation and discussion of the queue scheduler are given in Section 3. Section 4 provides conclusions and suggestion for future work.

2. FUZZY SCHEDULING STRATEGY

We consider the general class-based queue schedule for Internet routers illustrated in Fig. 1.

Let $x(t)$ denote the aggregated traffic flow. As shown in Fig. 1, each flow within $x(t)$ is first classified into various priorities or service classes (for example, FTP, Real-time, e-mail, etc.), and then placed in the corresponding queue. In Fig. 1, we assume M queue types with queue 1 having the highest priority and queue M the lowest. In PQ theory packets are scheduled to be transmitted in the order of their priority. This can lead to packet loss, performance degradation, and unfairness if a lower priority queue is never served and becomes full. Thus, it is important that a queue scheduler prevent network congestion by keeping each queue length smaller than the queue capacity. We propose to dynamically schedule packet transmission to minimize the total queue levels of all the queue types in the router. Thus, our objective function is expressed by

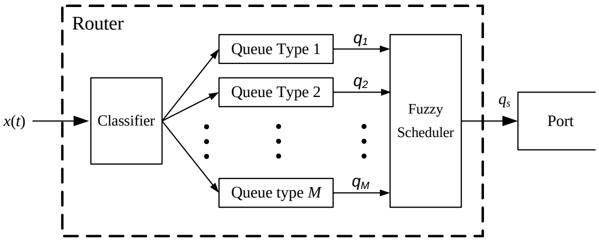


Fig. 1. Network environment with fuzzy queue scheduler.

$$J = \min \sum_{k=0}^{\infty} \sum_{i=1}^{M} q_i(k), \tag{1}$$

where q_i (k) is the dynamic queue level in the ith queue at discrete time k. Intuitively, an objective function is rarely converged at equilibrium state since $q_i(k) > 0$. But, our scheduling task is minimization of Equation (1) which is attained by selecting queue type q_s with the largest queue length, where

$$q_s = \arg\max_i \left(q_i(k)\right). \tag{2}$$

However, queue priority should also be taken into account by the scheduler. For instance, if two queue types with different priorities have large but unequal queue lengths, the scheduler should select the higher priority queue type even if its queue length is slightly smaller. Hence, proper queue scheduling should dynamically consider priorities as well as queue lengths. Conventional dynamic schedulers that consider these constraints can result in high computational complexity and may not be practicable for complex networks.

Fuzzy theory can provide simple but effective queue scheduling. Fuzzy systems utilize human expertise with fuzzy rules in the form of *If-Then* statements to solve practical problems. A general fuzzy rule for our scheduler is in the form

If q_1 is *large* and q_2 is *small* and $\cdots$ and q_m is *small* Then q_s is q_1, (3)

where q_s is q_1 implies that q_1 is selected. A fuzzy queue scheduler can be heuristically designed based on (3), but it is then difficult to prove its stability. We use the fuzzy Lyapunov synthesis of [8] in our scheduler to investigate its stability. Fuzzy Lyapunov synthesis is similar to classical Lyapunov synthesis [10], but only requires linguistic information to achieve its objectives. A Lyapunov function candidate is selected and conditions are determined for its derivatives along the trajectories of the system to be negative definite. If the scheduler can be designed to satisfy the stability conditions using the available linguistic information, a suitable Lyapunov function is identified and stability is guaranteed. Otherwise, another Lyapunov function is selected and the process is repeated until a positive result is obtained. To minimize the performance measure of (1) and investigate the stability of the scheduler we select the Lyapunov function candidate

$$V(k) = \sum_{i=1}^{m} q_i^2(k). \tag{4}$$

The change in the Lyapunov function along the trajectories of the system is

$$\begin{aligned} \Delta V(k) &= \sum_{i=1}^{m} \left[q_i^2(k) - q_i^2(k-1) \right] \\ &= \sum_{i=1}^{m} \Delta q_i(k) \left[q_i(k) + q_i(k-1) \right], \end{aligned} \tag{5}$$

where $\Delta q_i(k) = q_i(k) - q_i(k-1)$. To minimize the performance measure we process a message from the longest queue. If q_j is selected we rewrite (5) as

$$\begin{aligned} \Delta V(k) = &\left\{ \sum_{i \neq j}^{m} \Delta q_i(k) \left[q_i(k) + q_i(k-1) \right] \right\} \\ &- q_j^2(k-1). \end{aligned} \tag{6}$$

For scheduler stability, ΔV must be negative. Although the selection of q_j is the best decision for $\Delta V < 0$, during a burst of packet arrivals ΔV can be positive and transient instability may result. Packet loss due to insufficient queue space is then inevitable. However, in the steady state, queue levels remain bounded as long as the link capacity (i.e., processing rate) is larger than the packet arrival rate. This is a well-known necessary stability condition in the queuing theory literature [13].

3. FUZZY SCHEDULER DESIGN AND SIMULATION

We designed a fuzzy queue scheduler for an Internet router with multiple queue types using the methodology of Section II. We conducted a series of MATLAB simulations to evaluate the performance of our scheduler. For comparison, we also included a WRR scheduler in our simulation study.

3.1 Network environment

The network environment for our computer simulations is as follows: three queue types in an Internet router with different priorities and with maximum queue capacity, Q_1, Q_2, and Q_3, respectively, a link bandwidth with 15 Mb/sec, and a total queue length of 900 packets (i.e., Q_1+Q_2+Q_3=900). In these simulations, we assume constant-length packets of 512 bytes, i.e., 15 Mb/sec equals 3,840packets/sec. The packet arrivals for each queue type are assumed to have a Poisson distribution with three different Poisson parameters, λ = 110, 90, and 70, respectively. Fig. 2 shows each incoming packet level with the average level of queue 1 the largest and queue 3 the smallest. To mathematically model queue dynamic behavior, we use a discrete simplified version of the nonlinear continuous time model of [11].

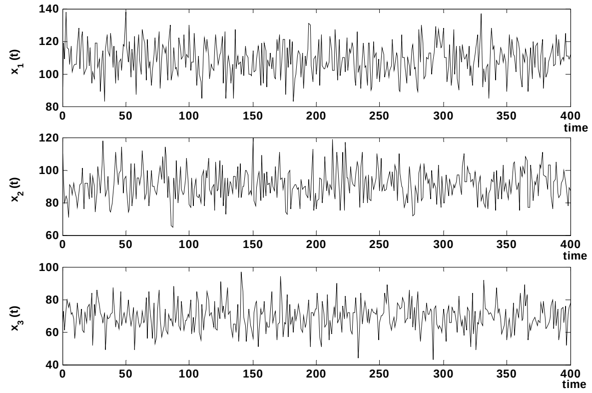


Fig. 2. Packet arrivals for three queue types.

$$\begin{aligned} &q_i(k+1) = q_i(k) + \alpha x_i(k) - C, \ \ if\ q_s = q_i \\ &q_i(k+1) = q_i(k) + \alpha x_i(k), \qquad otherwise. \end{aligned} \tag{7}$$

Here, α is the number of incoming traffic flows (α= 14), each with rate x_i, and C is the link capacity, where both x_i and C are measured in packets/sec. Each queue length is measured in number of packets. In (7), the first equation is for queue i selected by the scheduler and accounts for a decrease in queue length. The second equation applies if queue i is not selected and the queue length is increased by incoming packets.

3.2 Fuzzy inference

We use a simple fuzzy model to construct the proposed scheduling algorithm, which is adequate for the relatively simple network environment used in our simulations. We use a Mamdani fuzzy model [12] with three inputs and a single output. A block diagram of this fuzzy system is shown in Fig. 3.

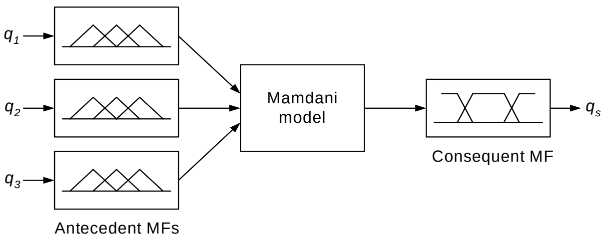


Fig. 3. Fuzzy queue scheduling.

In Fig. 3, the three inputs of the fuzzy system are the queue levels q_1, q_2, and q_3 and the output is the scheduled queue type q_s. We assume three fuzzy rule sets: *small*, *medium*, and *large*, and use Gaussian antecedent membership functions for the three queue inputs. Parameter values in these membership functions must be selected for best performance. The range of the horizontal axis in each antecedent membership function is from zero to the maximum queue length. The maximum queue length Q_i is first determined in the design procedure and then other parameter values of the membership function are specified. Thus, redesign of the fuzzy inference systems is required as Q_i changes. We designed and simulated the fuzzy scheduler for two different schemes as described in *Examples* I and II. We selected trapezoidal consequent membership functions with a range of 1 to 3 to denote the three queue types.

Fuzzy rules for our problems are derived as discussed in Section II and are summarized in Table 1. The table shows that some rules include indices of fuzzy sets of several queues which have equal values. In these cases, one of these queue types is selected based on priority.

Table 1. Rule bases for a fuzzy scheduler.

(S: Small, M: Medium, L: Large)

Rule No.	q_1	q_2	q_3	q_s
1	S	S	S	1
2	S	S	M	3
3	S	S	L	3
4	S	M	S	2
5	S	M	M	2
6	S	M	L	3
7	S	L	S	2
8	S	L	M	2
9	S	L	L	2
10	M	S	S	1
11	M	S	M	1
12	M	S	L	3
13	M	M	S	1
14	M	M	M	1
15	M	M	L	3
16	M	L	S	2
17	M	L	M	2
18	M	L	L	2
19	L	S	S	1
20	L	S	M	1
21	L	S	L	1
22	L	M	S	1
23	L	M	M	1
24	L	M	L	1
25	L	L	S	1
26	L	L	M	1
27	L	L	L	1

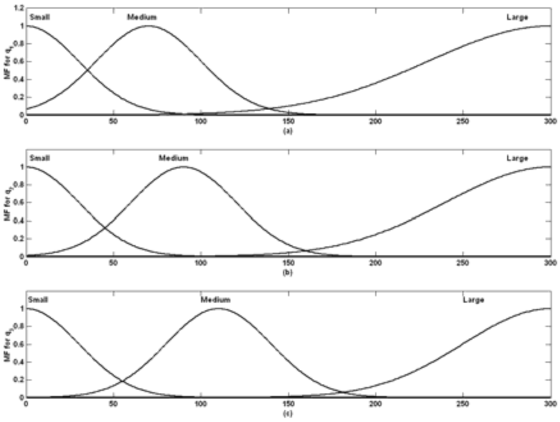


Fig. 4. Membership functions for Example I.

Defuzzification is the final computation stage in the fuzzy scheduling algorithm, in which a certain queue type is determined. We use Centroid of Area (COA) [12], the most popular defuzzifier, in our fuzzy systems. Defuzzification yields a representative crisp value from a fuzzy set which is generally not an integer number in this case. Since queue type is an integer 1, 2, or 3, we assign a defuzzified integer with a range of [1,3] using

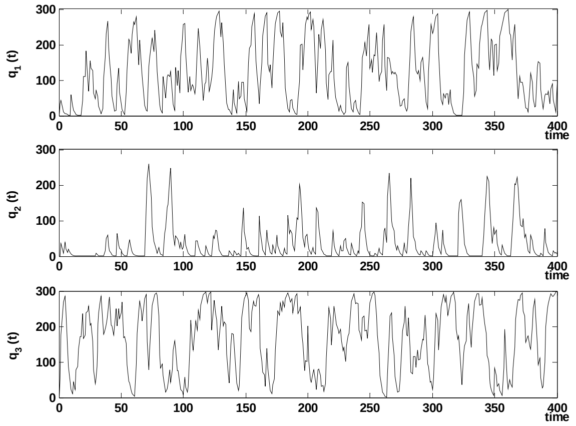
$$\begin{aligned} q_s &= 1, \text{ if } 1.0 \le z < 1.7, \\ q_s &= 2, \text{ if } 1.7 \le z < 2.3, \\ q_s &= 3, \text{ elsewhere,} \end{aligned} \tag{8}$$

where z indicates a defuzzified number. For WRR simulations, we assumed that the percentage of link bandwidth for q_1, q_2, and q_3 to be 50%, and 30%, and 20%, respectively. We simulated these two methods for two scenarios: equal queue capacities and unequal queue capacities.

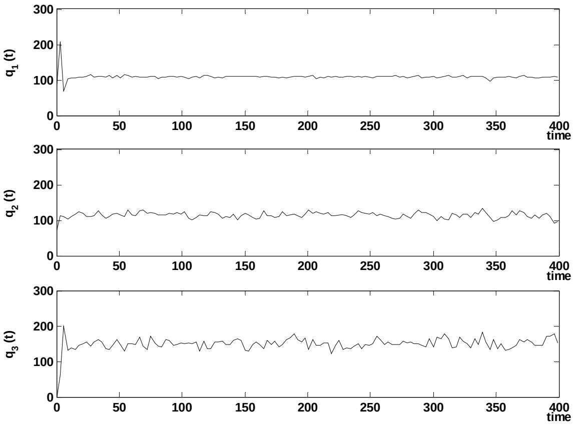
Example I: First, we simulated the system with the same queue capacity for each queue with $Q_1 = Q_2 = Q_3 = 300$ packets. We used Gaussian membership functions and tuned their parameter values. Membership functions for this case are plotted in Fig. 4.

The range of queue levels in Fig. 4 is 0 to 300, but the fuzzy sets have different centers for different queue types. The parameter values of the membership functions were selected based on our simulation results. Fig. 5 shows queue dynamic responses for WRR and for fuzzy scheduling. Fig. 5(a) shows three responses for WRR. The responses shown have nonperiodic random oscillations of amplitude between approximately 0 and 300, especially, in q_1 and q_3. All queue lengths can reach the capacity of the queue which can result in packet loss. By contrast, fuzzy scheduling provides stable queue behavior as shown in Fig. 5(b). All three queues converge to steady-state levels with almost no ripples and appear to offer significant improvements over WRR.

Table 2 gives simulation results for fuzzy scheduling and WRR. Fuzzy scheduling results in a smaller peak queue size and smaller oscillations



(a) WRR scheduling.



(b) Fuzzy scheduling.

Fig. 5. Queue dynamics for Example I.

Table 2. Simulation results for Example I [packets].

Scheduling Method	Queue Type	Max	Mean	Variance
WRR	q_1 q_2 q_3	299.6 281.3 299.8	143.9 52.3 143.9	9.0×10^3 4.4×10^3 6.8×10^3
Fuzzy	q_1 q_2 q_3	229.4 215.3 213.0	153.6 133.9 105.1	2.2×10^3 1.8×10^3 1.1×10^3

around its steady-state level. As shown in Table 2, with our fuzzy scheduling, queue type 1 shows the lowest average queue level followed by type 2 and then type 3. This is desirable because the highest priority queue has the lowest average queue level. Our results demonstrate that fuzzy scheduling is more effective than WRR.

Example II: We consider the case of different queue capacities: Q_1=400, Q_2=300, and Q_3=200. We redesign the fuzzy membership functions for these capacities but use the same fuzzy rule bases as in Example I. The membership functions for this example are shown in Fig. 6.

Each horizontal axis in Fig. 6 is a queue capacity and the parameter values are chosen based on our simulations. Fig. 7 illustrates queue responses for the two queuing methods. Nonperiodic random oscillations occur in WRR as in *Example* I, but fuzzy scheduling results in stable queue responses. We again conclude from this simulation that fuzzy scheduling is superior to WRR.

Table 3 shows simulation results of the two

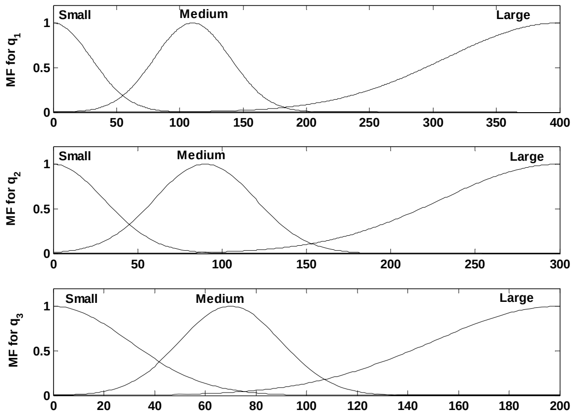
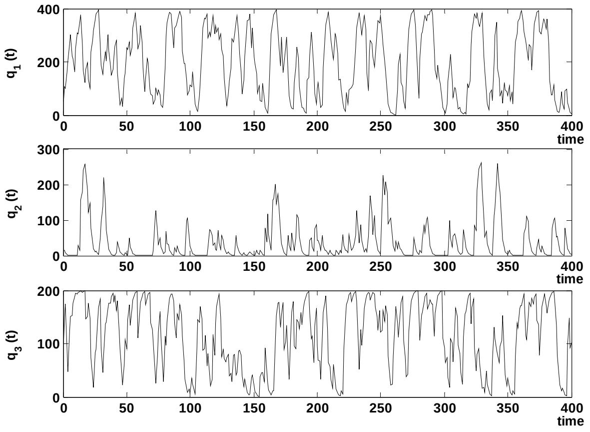
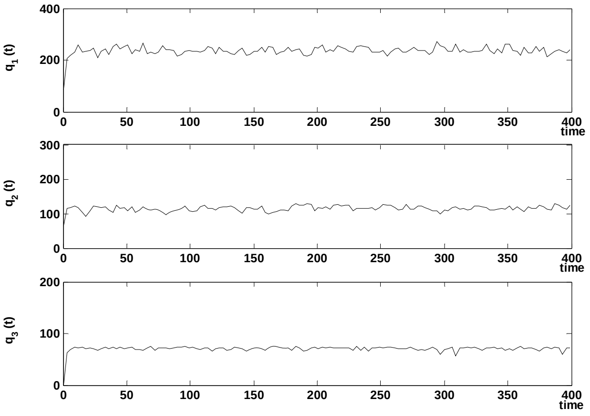


Fig. 6. Membership functions for Example II.



(a) WRR scheduling.



(b) Fuzzy scheduling.

Fig. 7. Queue dynamics for Example II.

Table 3. Simulation results for Example II [packets].

Scheduling Method	Queue Type	Max	Mean	Variance
WRR	q_1	397.8	212.0	1.7×10^4
	q_2	281.9	43.6	3.1×10^3
	q_3	199.9	112.7	4.0×10^3
Fuzzy	q_1	276.2	165.0	2.7×10^3
	q_2	215.4	134.2	1.8×10^3
	q_3	151.6	99.6	9.7×10^2

scheduling methods for this experiment. The table shows that fuzzy scheduling results in the smaller peak queue size and in smaller oscillations around the mean queue size. Thus our fuzzy scheduler again outperforms WRR. In particular, the reduced maximum queue levels for fuzzy scheduling are likely to reduce the probability of packet loss.

4. CONCLUSIONS

We developed a fuzzy dynamic queue scheduler for class-based queuing. A fuzzy rule base is derived through fuzzy Lyapunov stability theory to obtain a stable queue scheduler. We simulated a queuing mechanism which includes three queue types with different priorities. Compared to WRR, a well known conventional queue scheduling policy, the proposed fuzzy scheduler yields superior performance in our simulation experiments. The queue levels in the router with our fuzzy scheduler have lower steady-state responses and much lower average queue lengths than WRR. Consequently, the fuzzy based queue scheduler is an efficient dynamic queuing mechanism. We will apply the fuzzy inference scheduling scheme to multi-path routing in wireless networks in future work. For this implementation, we will develop a more elaborate decision algorithm in order to efficiently utilize network resources and to increase network throughput. In addition, more complex network scenarios and fuzzy scheduling models will be explored, and stability analysis for time-delay scheduling mechanism will be accomplished.

REFERENCES

[1] S. Floyd, “A report on recent developments in TCP congestion control,” *IEEE Communications Magazine*, vol. 39, no. 4, pp. 84-90, 2001.

[2] Z. Wang, *Internet QoS: Architectures and Mechanisms for Quality of Service*, Morgan Kaufmann Publishers, 2001.

[3] J. Nagle, “On packet switches with infinite storage,” *IEEE Trans. on Communications*, vol. 35, no. 4, pp. 435-438, 1987.

[4] A. Demers, S. Keshav and S. Shenker, “Analysis and simulation of a Fair-Queuing algorithm,” *Proc. ACM SIGCOMM*, pp. 1-12, 1989.

[5] F. Risso and P. Gevros, “Operational and performance issue of a CBQ router,” *Computer Communication Review* (*ACM SIGCOMM*), vol. 29, no. 5, pp. 47-58, 1999.

[6] J. Joutsensalo, G. Gomzikov, and T. Hamalainen, “Enhancing revenue maximization with adaptive WRR,” *Proc. of the 8th IEEE Int. Symposium on Computers and Communication*, vol. 1, pp. 175-180, 2003.

[7] K. M. Passino and S. Yurkovich, *Fuzzy Control*, Addison-Wesley, 1998.

[8] M. Margaliot and G. Langholz, *New Approaches to Fuzzy Modeling and Control: Design and Analysis*, World Scientific, 2000.

[9] J. R. Perkins and P. R. Kumark, “Stable, distributed, real-time scheduling of flexible manufacturing /assembly /disassembly systems,” *IEEE Trans. on Automatic Control*, vol. 34, no. 2, pp. 139-148, 1989.

[10] J.-J. E. Slotine and W. Li, *Applied Nonlinear Control*, Prentice Hall, 1991.

[11] C. V. Hollot, V. Misra, D. Towsley, and W. Gong, “Analysis and design of controllers for AQM routers supporting TCP flows,” *IEEE Trans. on Automatic Control*, vol. 47, no. 6, pp. 945-959, 2002.

[12] J.-S. R. Jang, C.-T. Sun, and E. Mizutani, *Neuro-fuzzy and Soft Computing*, Prentice Hall, 1997.

[13] D. Bertsekas and R. Gallager, *Data Networks*, Prentice Hall, 1992.

Hyun Cheol Cho received the B.S. degree from the Pukyong National University in 1997, the M.S. degree from the Dong-A University, Korea in 1999, and the Ph.D. degree from University of Nevada-Reno, USA in 2006. He is currently a Post-doc Researcher in the Dept. of Electrical Engineering, Dong-A University. His research interests are in the areas of control systems, stochastic systems, and neural networks.

M. Sami Fadali earned a B.S. in Electrical Engineering from Cairo University in 1974, an M.S. from the Control Systems Center, UMIST, England, in 1977 and a Ph.D. from the University of Wyoming in 1980. He was an Assistant Professor of Electrical Engineering at the University of King Abdul Aziz in Jeddah, Saudi Arabia 1981-1983. From 1983-85, he was a Post Doctoral Fellow at Colorado State University. In 1985, he joined the Electrical Engineering Dept. at the University of Nevada, Reno, where he is currently Professor of Electrical Engineering. And he is a senior member of IEEE and ISCA. In 1994 he was a visiting professor at Oakland University and GM Research and Development Labs. He spent the summer of 2000 as a Senior Engineer at TRW, San Bernardino. His research interests are in the areas of robust control, robust stability, fault detection, and fuzzy logic control.

Jin Woo Lee received the B.S. degree from the Kyungnam University in 1998, and the M.S. and Ph.D. degrees from the Dong-A University, Korea in 2000 and 2006 respectively. He is currently a Post-doc Researcher in the Dept. of Electrical Engineering, Dong-A University. His research interests are image processing, adaptive control, intelligent control, and automatic port systems.

Young Jin Lee earned the B.S., M.S., and Ph.D. degrees from the Dong-A University, Korea in 1992, 1994, and 2000, respectively. He is currently a Professor in the Dept. of Avionics Electrical Eng., Korea Aviation Polytechnic College. His research interests are control systems, immune algorithm, and automatic port systems.

Kwon Soon Lee received the B.S. degree from Chungnam National University, Daejeon, and M.S. degree in Electrical Engineering from Seoul National University, Seoul, Korea, in 1977 and 1981, respectively, and the Ph.D. degree in Electrical and Computer Engineering from Oregon State University, USA, in 1990. He joined the Department of Electrical Engineering, Dong-A University, Busan, Korea, as an Assistant Professor from 1982 to 1994. Since October 1, 1994, he has been with Division of Electrical, Electronic, and Computer in Dong-A University, Busan, Korea, where he is currently a Professor. He has authored or coauthored over 130 articles in archival journals and conference proceedings. His research interests include all aspects of port automation systems, intelligent control theory, and application of immune algorithm, etc. Prof. Lee is a responsible person of National Research Laboratory nominated by the Korean Ministry of Science & Technology, the Team Leader of New University for Regional Innovation (NURI) in Dong-A University, and the Director of international cooperation department of Regional Research Center (RRC) in Korea.