# A Fuzzy Extension to the BLUE Active Queue Management Algorithm*

**M. H. Yaghmaee**[1] **M. B. Menhaj**[2] **H. Amintoosi**[1]

1-Computer Department, Ferdowsi University of Mashhad,
Institute for Studies in Theoretical Physics and Mathematics (I.P.M)
2- Dept. of Electrical Engineering, Amirkabir University of Technology, Tehran, Iran

**Abstract**:To support new Internet applications such as voice over IP and video on demand, it is necessary to design effective congestion control and queue management algorithms. However, such a design is known to be difficult because there are a variety of services supported in the Internet and their various demands for Quality of Service (QoS). This led to a new trend in using alternative techniques, such as Fuzzy Logic Controllers (FLC) which have the ability to cope with the aforementioned problems and provide more flexibility in modeling congestion controllers as well.
In this paper, we introduce Fuzzy BLUE queue management algorithm, a fuzzy improvement to the well-known BLUE active queue management algorithm. The proposed fuzzy controller, uses packet loss and normalized queue length as input linguistic variables. The output of fuzzy controller is the drop probability of BLUE mechanism. To evaluate the performance of the proposed model, a simulation software were developed in C++ environment. Based on simulation results obtained from different trials, it can be seen that the proposed Fuzzy BLUE algorithm outperforms the traditional BLUE mechanism.

**Keywords:** Active Queue Management (AQM), BLUE Mechanism, Fuzzy Logic Control, Congestion Control, Quality of Services

## 1. Introduction

During past few years, computer networks have experienced an explosive growth in number of users and amount of traffic. The TCP (Transmission Control Protocol) congestion control was introduced into the Internet in the late 1980's by Van Jacobson. The current Internet can only provide the best effort service. In the best effort services, however traffic is processed as quickly as possible, there is no guarantee of timeliness or actual delivery. By developing new Internet services, quality of service has become much needed. The Internet Engineering Task Force (IETF) has proposed two important models for providing quality of service in the Internet: Integrated Services (Intserv) [1-3] and Differentiated Services (Diffserv) [4-6]. In the Intserv model, for each individual flow, the network resources are reserved. This model uses the RSVP (Resource Reservation Protocol)[7] as the signaling protocol. The essence of the Diffserv is to divide traffic into different classes and give them differentiated treatments.

Each router in the Internet uses two classes of algorithms related to the congestion control, i.e., queue management and scheduling. The queue management algorithms manage the length of packet queues by dropping packets when necessary or appropriate, while the scheduling algorithms determine the packets to be sent in sequel and are used primarily to manage the allocation of bandwidth among flows. The FIFO (First-In First-Out) is the most common scheduling algorithm. Packet dropping can be considered as a form of implicit notification of congestion.

Internet routers use queues to smooth spikes in incoming packet rates. When the incoming packet rate is higher than that of the router's outgoing, the queue size will increase; the amount of increase will eventually exceed an available buffer space. When the buffer is full, some packets will have to be dropped. In the Drop Tail (DT)

* This research was in part supported by a grant from Institute for Studies in Theoretical Physics and Mathematics (I.P.M) (NO.CS1382-4-09)



Archive of SID

or tail-drop policy, which is the most widespread dropping policy, if a packet arrives and finds the queue full, it will then be dropped. It was shown that the DT mechanism interacts badly with TCP's congestion control mechanisms leading to poor a performance. The AQM is the pro-active approach of informing the sender about incipient congestion before a buffer overflow happens. By using AQM mechanisms, the senders are informed in advance about congestion so that to react accordingly.

Random Early Detection (RED)[8-11] and BLUE [12] are two important AQM mechanisms, proposed in the literature in order to resolve the afore-mentioned problems caused by the DT mechanism. The RED mechanism employs randomization to overcome efficiently both the lockout and full queue problems without requiring any changes at the end hosts. The purpose of RED is to avoid queue overflow and tail-drop at each router. By dropping packets randomly, packets of different TCP connections are likely to be dropped at different times. Therefore, the TCP flow control mechanism for these connections will reduce their sending rate at different times.

In [13], the AQM that uses both load and queuing delay to determine its packet drop/mark probabilities, has been examined and then the Load/Delay Controllers (LDC) which has the advantage of controlling the queuing delay as well as accurately anticipating incipient congestion, has been proposed.

In [14], a Closed-Loop Congestion Control (CLCC) scheme on top of an existing AQM scheme to achieve fair bandwidth distribution among concurrent responsive and non-responsive traffic, has been proposed. The new scheme has the advantage that it does not need to estimate the level of responsiveness of traffic. The simulations results show that this extension significantly improves fair bandwidth allocation for responsive and non-responsive traffic.

There has been a notable lack of standardized performance evaluation of AQM schemes. A rigorous study of the influence of parameterization on specific schemes and the establishment of common comparison criteria is essential for objective evaluation of the different approaches. In [15], a evaluation framework for AQM schemes has been proposed.

The effect of queue averaging on REDes ability to avoid a bias against bursty traffic, to minimize queue size oscillation, and to optimize the distribution of packet drops over time in order to avoid global synchronization, has been investigated in[16]. It has been shown that a small degree of averaging is best to optimize the performance metrics.

The approach given in [17], predicts the impact of its own actions on the behavior of reactive protocols and thus on the short-term future traffic without keeping per flow state. It allows much faster convergence of the main AQM parameters, at least towards a local optimum, thereby smoothing and minimizing both congestion feedback and queue occupancy.

It is well known that active queue management is a powerful tool to improve network performance, especially with respect to TCP's flow and congestion control mechanisms. In addition, it is also suitable to realize prioritization between distinct traffic aggregates. In [18] a model for traffic prioritization and differentiation with AQM has been proposed.

In [19] by using a previously developed nonlinear dynamic model of TCP, a active queue management control systems using RED, has been designed. The RED's parameters have been related to the network parameters such as propagation delay and load level.

In [20] a congestion control system that scales gracefully with network capacity, providing high utilization, low queuing delay, dynamic stability, and fairness among users, has been designed.

In [21] a stochastic model of an ECN(Explicit Congestion notification) /RED gateway with competing TCP sources sharing the capacity, has been considered. As the number of competing flows becomes large, the queue behavior at the gateway can be described by a two-dimensional recursion and the throughput behavior of individual TCP flows becomes asymptotically independent. Furthermore, a Central Limit Theorem has been presented, yielding insight into the relationship between the queue fluctuation and the marking probability function.

In [22] the stability conditions for congestion control of the present Internet characterized by TCP-controlled sources and buffer-based active queue management schemes, has been developed.

During the past few years, fuzzy logic has been found many applications in telecommunication networks [23]. In [24-29], the authors proposed many different fuzzy logic controllers for traffic management in ATM (Asynchronous Transfer Mode) networks including Call Admission Control (CAC) and Usage Parameter Control (UPC). The concept of fuzzy threshold and adaptive buffer management was proposed in [30]. As shown in this paper, the fuzzy threshold buffer management has a better performance than the traditional binary thresholds. In [31] the Fuzzy Explicit Marking (FEM) was proposed, which can support Explicit Congestion Notification (ECN), to provide congestion control in TCP/IP best-effort networks using a fuzzy


Archive of SID

logic control approach. In [32], a methodology was proposed to choose optimized fuzzy controller parameters by using the Wang-Mendel and genetic algorithms and applied to voice over IP problems in DiffServ domains. In [33], the results of a fuzzy logic control approach for RED implementation was developed for differentiated services.

In this paper, we present a fuzzy extension of the BLUE algorithm. The proposed model employs a fuzzy logic controller to determine the drop probability. Simulation results, which are shown in section 4, clearly indicate that the proposed model has a better performance than the traditional BLUE mechanism does. . The initial results of this research effort has been reported in [34].

The reminder of this paper is organized as follows. In section 2, some of the most popular AQM mechanisms are discussed. The proposed fuzzy model is introduced in section 3. Section 4 presents simulation results and compares the performance of the proposed fuzzy BLUE with that of traditional BLUE mechanism. Finally, section 5 concludes the paper.

## 2. Evolution of AQM Algorithms

The Internet is increasingly facing packet loss and queue delays due to its rapid growth. This may lead to a congestion collapse [35] that will reduce the quality of Internet applications. Active queue management techniques try to detect and react to the congestion before its consequences such as packet drops or queuing delays. In reaction to a suspected congestion, AQM algorithms proposed in [36] either drop packets early or mark them as ECN to inform the congestion to the traffic sources. The most important differences among AQM schemes are the guess congestion time determination and the marked/dropped packets selection. In this section, we describe the evolution of AQM algorithms and list their shortcomings.

### 2.1. The DT Algorithm

In this algorithm, only when a packet loss occurs due to the queue overflow, sources are informed to reduce their transmission rates. The time between detecting a packet loss in the router and informing the source to reduce its rate causes a large number of packets to be dropped since the router cannot provide space for incoming packets which have been sent with the previous rate. This algorithm keeps the queues always full. It is unfair with bursty traffics and can affect in lockouts [37].

### 2.2. The RED Algorithm

This algorithm is one of the first proposed solutions to the active queue management. RED simply sets minimum and maximum dropping thresholds, which are presented by minth and maxth , respectively. If the average queue size (avg) exceeds the minimum threshold, RED starts dropping packets based on a probability depending on the average queue size. If average queue size exceeds the maximum threshold, then every packet is dropped. A pseudocode of the RED algorithm is given below:

```
For every packet arrival {
Calculate avg
If( )
{
Drop the packet
}
else if (avg > minth) {
Calculate the dropping probability pa
Drop the packet with probability pa, otherwise forward it
}
else {
Forward the packet
}
}
```

As expressed completely in [11,12], RED contains severe problems. The fundamental one is that it uses queue length as a congestion indicator. This indicator cannot completely show the severity of congestion. On the other hand, average queue length varies with the level of congestion as well as with the parameter settings. As a result, the queuing delay of RED is too sensitive to a traffic load and parameter settings [12]. Different variants of RED such as Stabilized RED (SRED) [9] and Adaptive RED (ARED) [10] have been proposed to fix some of its shortcomings.

### 2.3. The BLUE Algorithm

The main idea of this algorithm is to use different parameters such as packet loss and link utilization to detect a congestion and adjust the rate of packet dropping/marking [12]. BLUE uses single probability, Pm, which is used to mark or drop packets when they are enqueued. If the queue is continually dropping packets due to overflow, Pm is increased. If the link is underutilized, the probability is decreased. The amount of increase and decrease are d1 and d2, respectively.

The rate of updating Pm is 1/freeze_time. This allows Pm to take effect before the next update. The probability is also updated when the queue length exceeds a threshold L. In other words, it does not let the queue become full. This allows space to be reserved in the queue for occasional bursts. The BLUE algorithm is


Archive of SID

given below. Notice that "now" corresponds to the current time while "last_update" corresponds to the last time Pm was tuned.

```
Upon packet loss (or queue length > L) event
        If ( (now-last_update) > freeze_time ) then
                Pm = Pm + d1
                last_update = now
Upon link idle event
        If ( (now-last_update) > freeze_time ) then
                Pm = Pm - d2
             last_update = now
```

The important features of BLUE are the minimal amount of buffer space, the reduction of end to end delay, the high link utilization and a very low rate of packet loss [12].

### 3. The Proposed Fuzzy BLUE

In this section, we propose a novel approach for the TCP congestion control, Fuzzy BLUE, which is an extension to the traditional BLUE mechanism. The main part of designing the Fuzzy BLUE is to design the Fuzzy BLUE Controller (FBC) in a way to act as a congestion controller in the routers.

There exist two different methodologies for design of fuzzy logic controller: trial-and-error approach and the theoretical approach. To design Fuzzy BLUE, we used the trial-and-error approach. In this approach, a set of IF-THEN rules are collected from an introspective verbalization of experience-based knowledge; then fuzzy controllers are constructed from these fuzzy IF-THEN rules; finally, the fuzzy controllers are tested and if the performance is not satisfactory, the rules are fine-tuned or designed in a number of trial-and-error cycles until the performance is satisfactory.

The steps of the Fuzzy BLUE design are taken as follows:

**- Selection of input and output linguistic variables:** The input variables should characterize the key features of the BLUE mechanism and the output variable should be able to influence the state of BLUE mechanism. By analyzing the BLUE active queue mechanism, we have chosen packet loss and the normalized queue length as two input linguistic variables. The output linguistic variable is the drop probability (Pm).

- **Selection of proper membership functions for input and output variables:** Because triangular and trapezoidal shaped membership functions offer more computational simplicity, we have used them for all three variables: packet loss, normalized queue and the Pm.

Figure1 shows the membership functions of the input and output linguistic variables. Based on this figure, the term sets of linguistic variables packet loss and normalized queue length are given below:

T(packet loss) ={ small, med (medium), big}

T(normalized queue length) ={low, mid (middle), high }

The output term sets of fuzzy logic controller are also defined as:

T(Pm)={zero, low, moderate, high}

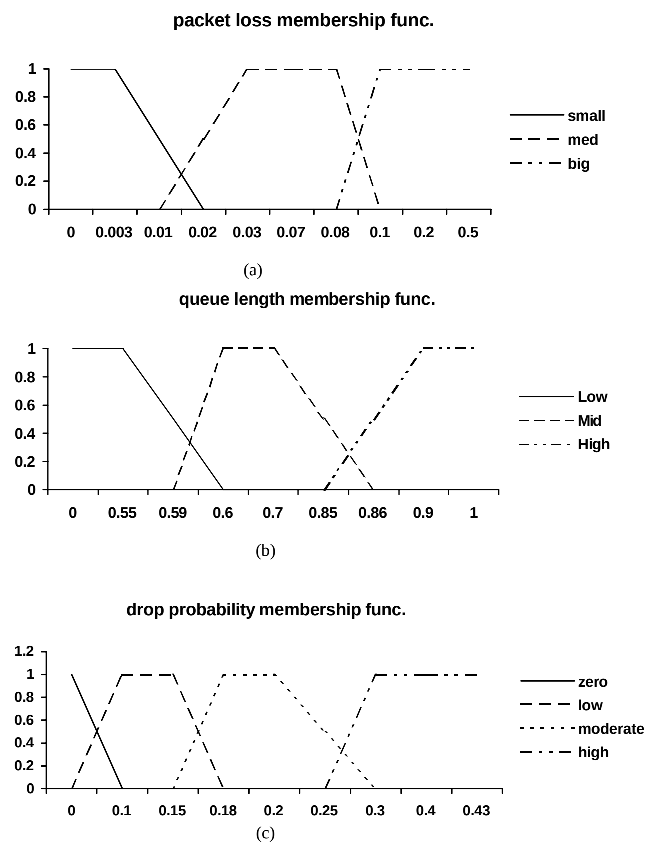



**Fig. 1:**Membership functions of the input and output variables:

(a) packet loss (b) normalized queue length (c) drop probability

**- Design of the fuzzy IF-THEN rule base:** The design of a proper fuzzy IF-THEN rule base is based on our experience and beliefs on how the system should work [38]. Design of a fuzzy IF-THEN rule base consists of two steps: first, linguistic rules are set and then membership functions of linguistic values are determined. In Figure 2, we present the fuzzy linguistic rules used in





Archive of SID

the simulation. This fuzzy IF-THEN rule base is achieved by means of human expertise and the knowledge about system There are two ways for designing the fuzzy IF-THEN rule base: trial and error approach and theory approach [39]. In the trial and error approach, based on the knowledge obtained from the experiment, a set of fuzzy IF-THEN rules are constructed and then the system is tested. If the desired behavior is not observed, rules are again tuned. This process is continued till the functionality of the controller is satisfied. In the theory approach, rules are designed in such a way that a specific functionality of a parameter (such as throughput) is guaranteed.

where $\mu^{conse)}(Pm)$ is the fuzzy set of control action.

**- The Main process:** As we mentioned earlier, the FBC utilizes two input variables, packet loss and normalized queue length. In certain periods of time, the values of these two variables are monitored and fed to the FBC. These values are then fuzzified using linguistic variables and the fuzzifier procedure. Referring to the rule base, rules which are fired according to these fuzzified values (i.e., the rules with the correct 'if' parts) are determined. To select between fired rules, we used the max-min combination method which defines the zone which the output value belongs to. The final step is to defuzzify the output value using the defuzzifier procedure.

```
/* Linguistic rules of FBC */
Rule 1: if packet loss is small and normalized queue length is low   then pm is zero;
Rule 2: if packet loss is small and normalized queue length is mid  then pm is zero;
Rule 3: if packet loss is small and normalized queue length is high then pm is zero;
Rule 4: if packet loss is med  and normalized queue length is low   then pm is zero;
Rule 5: if packet loss is med  and normalized queue length is mid  then pm is zero;
Rule 6: if packet loss is med  and normalized queue length is high then pm is moderate;
Rule 7: if packet loss is big   and normalized queue length is low   then pm is zero;
Rule 8: if packet loss is big   and normalized queue length is mid  then pm is low;
Rule 9: if packet loss is big   and normalized queue length is high  then pm is high;
```

**Fig. 2**: Linguistic rules for FBC

For example, consider the first rule of Fuzzy BLUE rule base as below:

Rule 1: if packet loss is small and normalized queue length is low then Pm is zero.

It is clear that when both packet loss and queue length are low, so the congestion density is low too. In this case, to increase the channel utilization, the drop probability of BLUE mechanism should be decreased.

**- The fuzzifier and defuzzifier procedures:** The fuzzifier procedure defines the zone which each input value belongs to according to its membership function. The deffuzzifier procedure, on the other hand, computes a crisp value of an output variable. In the proposed Fuzzy BLUE mechanism, we used the single-tone fuzzifier and the Center Of Gravity (COG) defuzzification method. The COG method chosses the control action that corresponds to the center of the area with membership greater than zero. The area is weighted with the value of the membership function. For the proposed Fuzzy BLUE, the final crisp output is calculated as:

$$Pm^{COG} = \frac{\int Pm.\mu^{conseq}(Pm)\,dPm}{\int \mu^{conseq}(Pm)\,dPm}$$

To understand the functionality of proposed Fuzzy BLUE, we provide an example. In our Fuzzy BLUE controller, we assume that the current packet loss and normalized queue length are 0.2 and 0.595, respectively. Thus we get that packet loss is "big" with degree 1. A similar definition of the linguistic variables in the normalized queue length case yields "Low" with degree 0.1 and "Mid" with degree 0.5. Based on fuzzy rule base given in figure2, we see that the following two rules have a degree of match greater than zero:

Rule 7: if packet loss is big and normalized queue length is low then Pm is zero;
Rule 8: if packet loss is big and normalized queue length is mid then Pm is low;

The degree of membership is :

$$\alpha_7 = \min\{1, 0.1\} = 0.1$$

$$\alpha_8 = \min\{1, 0.5\} = 0.5$$

Accordingly, the consequences of the rules are:

$$\mu_7^{conseq}(Pm) = \min\{0.1, \mu^{zero}(Pm)\}$$

$$\mu_8^{conseq}(Pm) = \min\{0.5, \mu^{low}(Pm)\}$$



Archive of SID

Figure 3, depicts the resulting fuzzy set of control action as below:

$$\mu^{conseq}(Pm) = \max\{\mu_7^{conseq}(Pm), \mu_8^{conseq}(Pm)\}$$

By using the COG defuzzification technique, the final crisp output of the Fuzzy BLUE is computed as below:

$$Pm^{COG} = \frac{\int_0^{0.01} 0.1Pm dPm + \int_{0.01}^{0.05} 10P^2 m dPm + \int_{0.05}^{0.165} 0.5Pm dPm + \int_{0.165}^{0.18} \frac{100(0.18-Pm)Pm}{3} dPm}{\int_0^{0.01} 0.1 dPm + \int_{0.01}^{0.05} 10Pm dPm + \int_{0.05}^{0.165} 0.5 dPm + \int_{0.165}^{0.18} \frac{100(0.18-Pm)}{3} dPm} = 0.10499$$

So, the packet are dropped with probability of 0.10499.

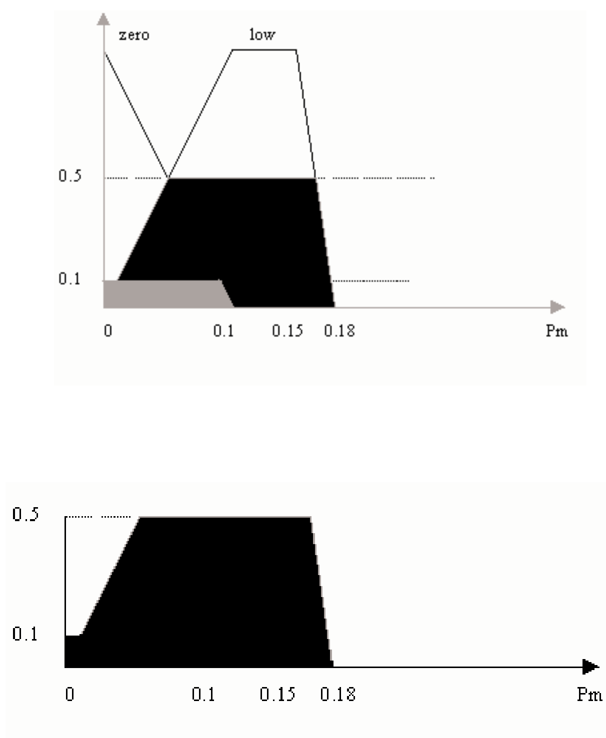



**Fig.3:** a) Rule consequence in the Fuzzy BLUE b) the output of max-min combination method

**4. Performance Evaluation**

In this section, by using computer simulation, we evaluated the performance of the proposed Fuzzy BLUE and compared it with the traditional BLUE mechanism. For this purpose, a software for simulation was developed in c++ environment. We perform the following different trials to test the performance of our algorithm.

**Trial 1**

In order to test the performance of the Fuzzy BLUE controller, we used the network topology shown in figure 4. As shown in this figure, four traffic sources (each have different link delay varying from 1ms to 2ms) send packets to the network. The traffic source #1,#2,#3 and #4 start to send traffic at time 0,0.2s,0.4s and 0.6s , respectively. All links between the traffic sources and the router have the capacity equal to 100Mbps. Packets are passed through a bottleneck link in order to reach the destination. The output link capacity of the router is equal to 45Mbps. We also implemented the TCP New Reno congestion control mechanism with TCP window size of 100 packets. The buffer size of the router was set to 100 packets. All packets are assumed to be of 1000 bytes length. For both Fuzzy BLUE and BLUE algorithms, d1 was set to 0.01 and d2 to 0.002. Also, the freeze_time was set to 10ms. The queue threshold, L, was selected equal to 60% of the buffer size.

When the simulation starts, sources begin to send packets one after another. Packets are then enqueued within the router's buffer. Whenever the queue length exceeds L, the drop probability, Pm, is computed by executing a fuzzy process. Arriving packets are then dropped according to this newly computed probability in order to prevent congestion. In addition, when the link is idle (i.e. the queue is empty), the drop probability is recomputed in order to reflect the new conditions of the network. Like BLUE, there is a delay (i.e. freeze_time) between two successive computations of Pm in order to let it take necessary effects.

In order to enhance the validity of results, we run the simulation ten times. In figure 5, for both the proposed Fuzzy BLUE and the traditional BLUE mechanisms, the queue length is plotted versus simulation time. As shown in this figure, it is clear that the Fuzzy BLUE manages to keep the queue size significantly better than the BLUE does.

In figure 6, the number of lost packets is plotted versus simulation time. It can be seen that for the proposed Fuzzy BLUE, the number of lost packets is less that that of the traditional BLUE mechanism.

The throughput of both mechanisms is shown in figure 7. According to this figure, the Fuzzy BLUE has a better throughput than the traditional BLUE mechanism. This figure shows that the Fuzzy BLUE matches a near 100% throughput (99.99%).

The queuing delay of of both mechanisms is plotted in figure 8. This figure confirms that both Fuzzy BLUE



Archive of SID

and traditional BLUE mechanisms have nearly the same queuing delay.

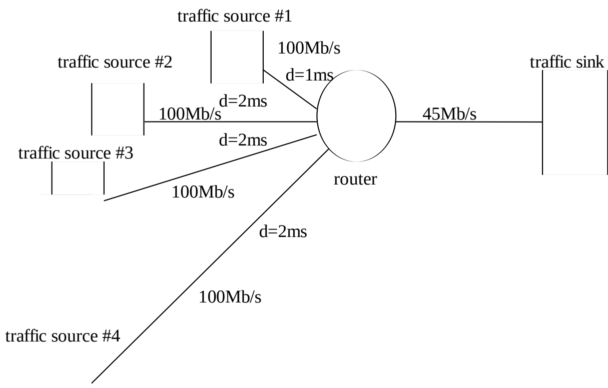



**Fig. 4:** Network topology used for simulation

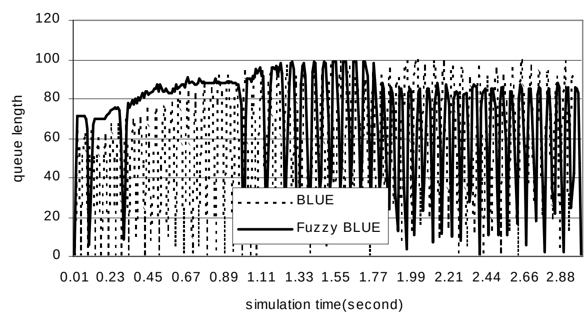



**Fig. 5:** Queue length versus simulation time for both Fuzzy BLUE and traditional BLUE

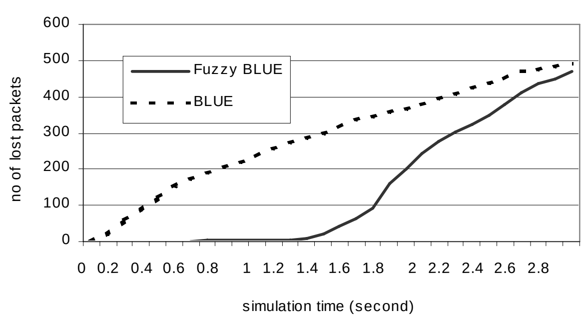



**Fig. 6:** Number of lost packets versus simulation time for both Fuzzy BLUE and traditional BLUE

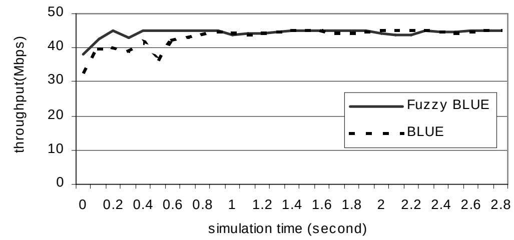



**Fig. 7:** Throughput versus simulation time for both Fuzzy BLUE and traditional BLUE

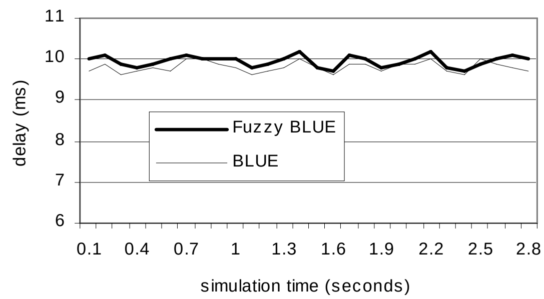



**Fig. 8:** Queuing delay versus simulation time for both Fuzzy BLUE and traditional BLUE

### Trial 2

In order to show the dynamic behavior of the proposed Fuzzy BLUE, we performed a new trial with the same network topology considered in the previous trail (see figure 3). The purpose of this trial is to evaluate the performance of the Fuzzy BLUE under different network traffic loads. At first, by increasing the TCP window to 120 packets we create conditions for more traffic. As it is expected, the number of lost packets is increased but the throughput matches nearly 100%. From figures 9(a) and 10(a), we easily observe that the Fuzzy BLUE completely outperforms the traditional BLUE in both queue behavior and packet loss.

In order to see how the algorithm behaves in lower traffic conditions, we reduced the TCP window size from 100 packets to 80 packets. As shown in figure 9(b), it is clear that the queue size of the proposed Fuzzy BLUE remains at a better level than that of the BLUE without leaving the buffer empty. Furthermore, from figure 10(b), it is clear that the Fuzzy BLUE has a lower packet loss than the traditional BLUE mechanism does.

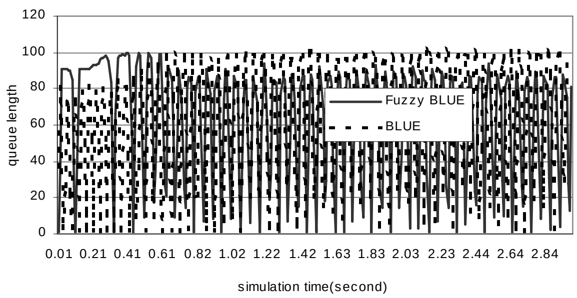
These figures approve that in compared with the BLUE, the Fuzzy BLUE achieves a better queue length while keeping a higher throughput with a lower packet loss rate.



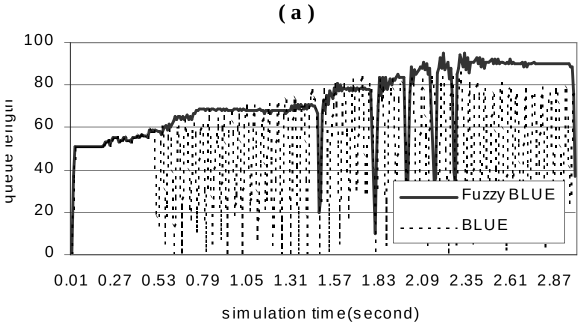
Archive of SID

**Trial 3**

In order to show how sensitive the variance of the packet loss probability of the FBC is to the TCP window sizes variations, we changed the window size from 5 to 200 packets. Figure 11 shows the results for both Fuzzy BLUE and BLUE mechanisms. As it was expected, the Fuzzy BLUE has a lower packet loss rate than the traditional BLUE mechanism does. As shown in this figure, although for both mechanisms the packet loss rate increases as the TCP window size increases , it is clear that for each TCP window size, the proposed Fuzzy BLUE mechanism in compared with the traditional BLUE has a lower packet loss.

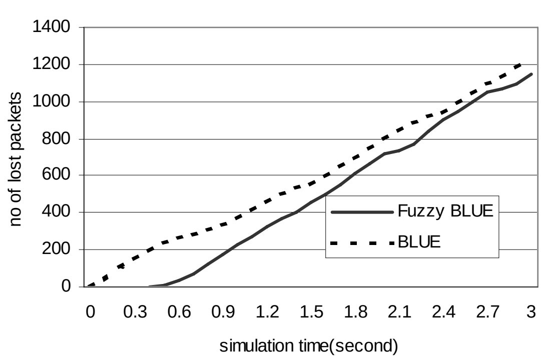



**( a )**

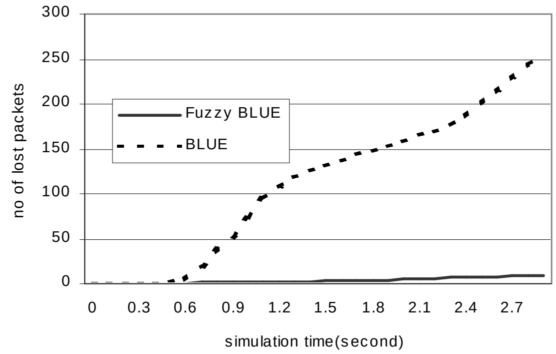



**( b )**

**Fig. 9:** Queue length versus simulation time for both Fuzzy BLUE and traditional BLUE
a) TCP window = 120 packets
b) TCP window = 80 packets




**( a )**




**( b )**

**Fig. 10:** Number of lost packets versus simulation time for both Fuzzy BLUE and traditional BLUE
a)TCP window = 120 packets
b) TCP window =80 packets

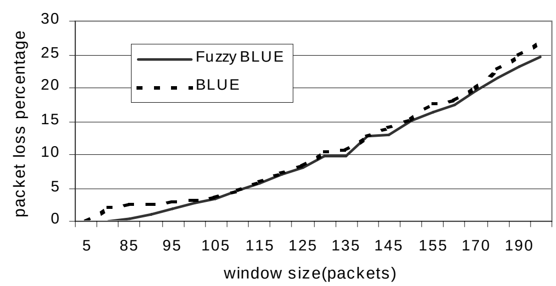



**Fig. 11:** Packet loss percentage versus TCP window size for both Fuzzy BLUE and traditional BLUE

**Trial 4**

To evaluate the performance of both mechanisms in a bursty traffic condition, we change the simulation conditions in a way that the traffic source #0 starts sending packets at the beginning of the simulation and the other three traffic sources starts to send packets, 0.2 second after the beginning of the simulation. This new scenario causes sudden increases in the traffic load at the router.

We executed trials 1 to 3 for this new case. Figures12 to 16 present the results. As shown in figure 12, the proposed Fuzzy BLUE has a better control over the queue size with lower fluctuations even in a sudden traffic increase situation. In addition, the packet loss rate of the Fuzzy BLUE is lower than that of the traditional BLUE (see figure 13).



Archive of SID

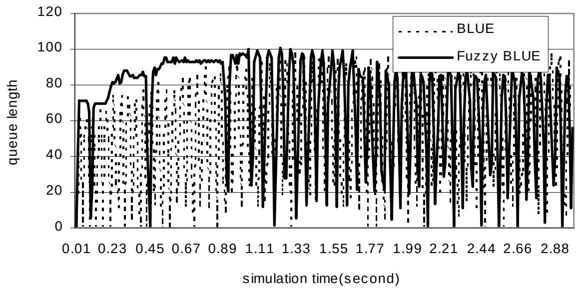



**Fig. 12:** Queue length versus simulation time for both Fuzzy BLUE and traditional BLUE (new case)

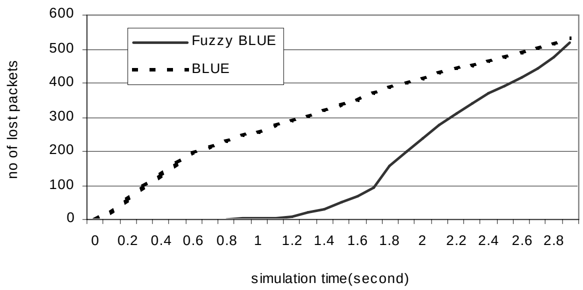



**Fig. 13:** Number of lost packets versus simulation time for both Fuzzy BLUE and traditional BLUE (new case)

In figure 14, for different TCP window sizes (120 and 80 packets), the queue length is plotted versus simulation time. This figure shows that the proposed Fuzzy BLUE can control the queue length better than the traditional BLUE mechanism. The packet loss percentage at different values of TCP window sizes, are shown in figure 15. Based on the results shown in figure 14(a), by increasing the window size, more fluctuations is caused in the queue length diagram of both algorithms. Furthermore, from figure 15(a) it is clear that the packet loss rate of the proposed Fuzzy BLUE is less than the traditional BLUE mechanism. Based on the results shown in figure 14(b), by reducing the window size to 80 packets, the queue size of both algorithms varies slowly; however the fuzzy BLUE has a better queue length behavior than the traditional BLUE mechanism does. In this case the packet loss rate of Fuzzy BLUE is further lower than the traditional BLUE, see figure 15(b).

Figure 16 shows the result of trial 3 for this new case. In this figure, for both Fuzzy BLUE and traditional BLUE mechanisms packet loss percentage is plotted versus TCP window size. As it can be observed from Figure 16, at different window sizes, in comparison with the traditional BLUE mechanism, the proposed Fuzzy BLUE has a better performance.

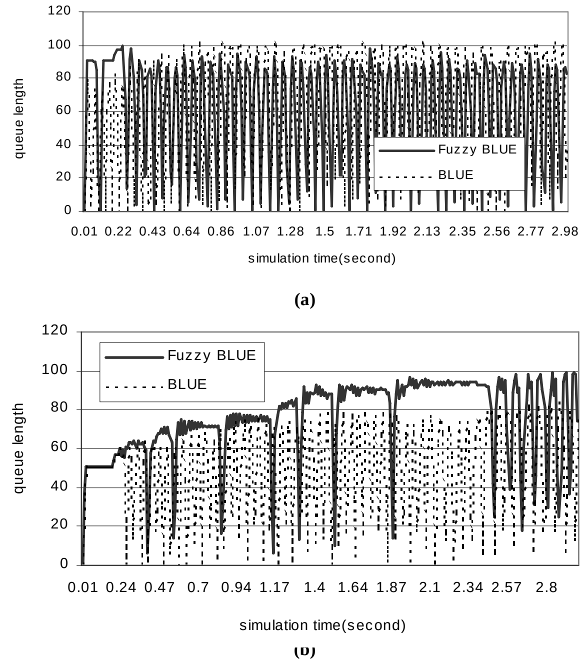



**Fig. 14:** Queue length versus simulation time for both Fuzzy BLUE and traditional BLUE (new case)
a)TCP window = 120 packets
b) TCP window =80 packets

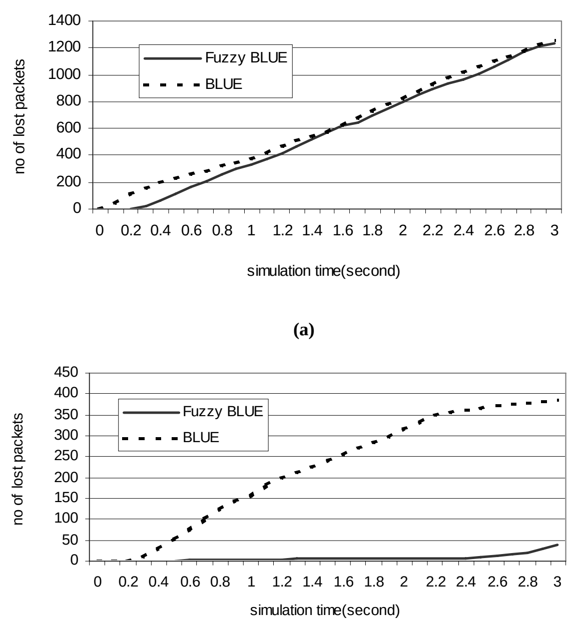



**Fig. 15:** Number of lost packets versus simulation time for both the Fuzzy BLUE and the traditional BLUE (new case)
a)TCP window = 120 packets
b)TCP window = 80 packets



Archive of SID

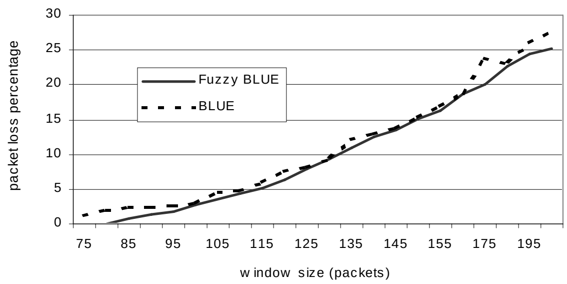



**Fig. 16:** Packet loss percentage versus TCP window size for both Fuzzy BLUE and traditional BLUE (new case)

**Trial5**

To simulate a more practical case, we added another router to the simulation topology; this lead to a two router network with two queues, one for each router. Furthermore, to have a more traffic load, we increased the window size to 120 packets. The link capacity between the new router and the destination was set to 43 Mbps. We then measured the average of the two queue sizes and also the average of the two packet loss rates for 10 different runs. Results are shown in Figures 17 -19. Obviously observed that the fuzzy BLUE has both a lower packet loss rate and a better management of queue size.

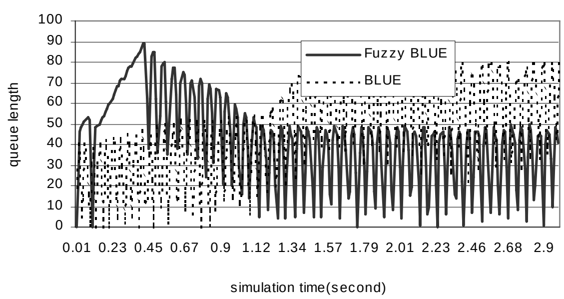



**Fig. 17:.** Average queue length versus simulation time for both Fuzzy BLUE and traditional BLUE (two routers)

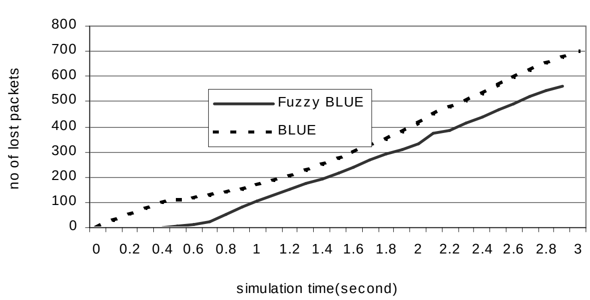



**Fig. 18:**Mean number of lost packets versus simulation time for both Fuzzy BLUE and traditional BLUE (two routers)

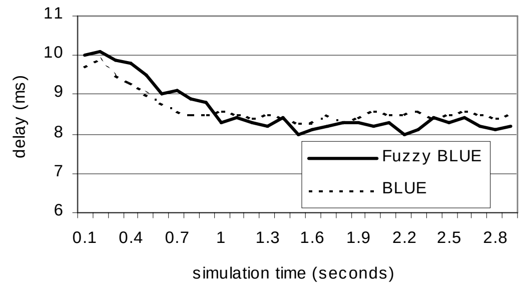



**Figure 19.** Queuing delay versus simulation time for both Fuzzy BLUE and traditional BLUE (two routers)
`

**5. Conclusions**

With new services emerging in the Internet every day, the current AQM and congestion control mechanisms seem not to have enough capability to cope with new needs. Fuzzy logic control, in turn, is a well-known technique for designing feedback control systems. The application of fuzzy control technique to the problem of congestion control is appropriate due to the problems we will face in obtaining a precise model using conventional methods. In this paper, we introduced the Fuzzy BLUE which acts more effective and robust in comparison with other approaches including the traditional BLUE. To evaluate the performance of the proposed Fuzzy BLUE, different trials were performed. Based on the simulation results, it was shown that the proposed Fuzzy BLUE achieve a high throughput. Furthermore, simulation results show that the proposed Fuzzy BLUE mechanism has a better loss performance as well as a queue length behavior than the traditional BLUE mechanism does.

**References:**

[1] R. Braden, D. Clark, S. Shenker, Integrated Services in the Internet Architecture: an Overview, IETF RFC 1633, June 1994.

[2] Integrated Services Charter, http://www.ietf.org/html.charters/intservcharter.html.

[3] "Integrated Services over Specific Link Layers (issl) Charter," http://www.ietf.org/html.charters/issl-charter.html.

[4] T. Li, Y. Rekhter , "A Provider Architecture for Differentiated Services and Traffic Engineering (PASTE)", IETF RFC 2430, October 1998.

[5] K. Nichols, V. Jacobson, L. Zhang, "A Two-bit Differentiated Services Architecture for the Internet", IETF RFC 2638, July 1999.



Archive of SID

[6] S. Blake, et al, "An Architecture for Differentiated Services", RFC 2475, 1998.

[7] R. Braden, Ed., L. Zhang, S. Berson, S. Herzog, S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", IETF RFC 2205, September 1997

[8] S.Floyd, V.Jacobson, "Random Early Detection Gateways for Congestion Avoidance", IEEE/ACM Trans. On Networking, vol.1, Number 4, pp. 397-413, August 1993

[9] T.Ott, T.Lakshman, L.Wong, "SRED: Stabilized RED", in proceeding IEEE INFOCOM 1999

[10] S.Floyd, R.Gummadi, S.Shenker, "Adaptive RED: An Algorithm for Increasing the Robustness of RED", Technical Report, 2001

[11] M.May, J.Bolot, C.Diot, B.Lyles, "Reasons Not to Deploy RED", Proc. of 7th International Workshop on Quality of Service (IWQoS'99), pp. 260-262, June 1999

[12] W.Feng, D.Kandlur, D.Saha, K.Shin, "BLUE: A New Class of Active Queue Management Algorithms", U.Michigan CSE-TR-387-99, April 1999

[13] M. Kwon and S. Fahmy, "A comparison of load-based and queue-based active queue management algorithms", in Proc. SPIE ITCom, vol. 4866, Aug. 2002.

[14] Roman Pletka, Andreas Kind, Marcel Waldvogel, and Soenke Mannal "Active queue management for fair bandwidth allocation of mixed responsive and nonresponsive traffic using a closed-loop congestion control scheme", Research report, IBM, February 2003.

[15] A. Bitorika, M. Robin, and M. Huggard, "An evaluation framework for active queue management schemes", in Proc. MASCOTS'03. IEEE, Oct. 2003.

[16] T. Ziegler, "On Averaging for Active Queue Management Congestion Avoidance", IEEE ISCC 2002, Italy.

[17] Roman A. Pletka, Marcel Waldvogel, and Soenke Mannal. "PURPLE: predictive active queue management utilizing congestion information", in Proceedings of Local Computer Networks (LCN), pages 21--30, October 2003.

[18] Walter U. and Wehrle K., "Traffic Priorization and Differentiation with Active Queue Management", http://www.tm.uka.de/~walter/papers/walter_wehrle_ICTSM2002.pdf.

[19] C. Hollot, V. Misra, D. Towsley, and W.-B. Gong. "A Control Theoretic Analysis of RED", in Societies INFOCOM'2001.

[20] F. Paganini, Z. Wang, S. Low, and J. Doyle, "A new TCP/AQM for stable operation in fast networks", in Proc. IEEE IFOCOM, (San Francisco, CA), 2003.

[21] P.Tinnakornsrisuphap, A.Makowski, "Limit Behavior of ECN/RED Gateways Under a Large Number of TCP Flows", IEEE IFOCOM 2003.

[22] H. Han, C. V. Hollot, Y. Chait, and V. Misra, "TCP networks stabilized by buffer-based AQMs", in IEEE INFOCOM 2004.

[23] S. Ghosh, Q. Razouqi, H.J. Schumacher, A. Celmins, A survey of recent advances in fuzzy logic in telecommunications networks and new challenges, IEEE Trans. Fuzzy Systems 6 (3) (1998),pp. 443-447.

[24] M.H.Yaghmaee , M.Safavi, M.B.Menhaj., "A novel fuzzy traffic controller for ATM networks", in proceeding of IEEE International Workshop on Intelligent Signal Processing and Communication Systems (ISPACS98), Melbourn, Australia, 1998, pp. 199-203.

[25] M.H.Yaghmaee , M.Safavi, M.B.Menhaj, "A high performance fuzzy traffic controller for ATM networks", in proceeding of IEEE ATM workshop'99, Kochi , Japan, 1999, pp.379-384.

[26] M.H.Yaghmaee , M.Safavi, M.B.Menhaj, "A fuzzy connection admission controller for ATM networks", in proceeding of IEEE International Workshop On Intelligent Signal Processing and Communication System (ISPACS99), Pucket, Thailand, 1999.

[27] M.H.Yaghmaee , M.Safavi, M.B.Menhaj "An efficient fuzzy based traffic policer for ATM networks", IEICE Transaction on Communication, Vol. E83-B, No.1, pp.10-19, 2000.

[28] M.H.Yaghmaee , M.Safavi, M.B.Menhaj. ,"An intelligent usage parameter controller based on dynamic rate leaky bucket for ATM networks", Elsevier Science Journal on Computer Networks, Vol. 32, pp.17-34, 2000.

[29] M.H.Yaghmaee , M.Safavi, M.B.Menhaj , "A novel FLC based approach for ATM traffic control", Elsevier Science Journal on Computer Networks, Vol. 36/5-6, pp.643-658, 2001.

[30] A.R. Bonde, S. Ghosh, "A comparative study of fuzzy versus fixed threshold for robust queue management in cell-switching networks", IEEE/ACM Trans. Networking 2 (4) (1994) 337±344.

[31] C. Chrysostomou, A. Pitsillides, G. Hadjipollas, A. Sekercioglu and M. Polycarpou, "Fuzzy Logic Congestion Control in TCP/IP Best-Effort Networks"

[32] Marcial Porto Fernandez, Aloysio de Castro P. Pedroza and Jos´e Ferreira de Rezende, " Optimizing Fuzzy Controllers with Genetic Algorithms for QoS Improvement", International Telecommunications Symposium - ITS2002, Natal, Brazil,2002.

[33] C. Chrysostomou1, A. Pitsillides, L. Rossides, M. Polycarpou, A. Sekercioglu, "Congestion Control in Differentiated Services Networks using Fuzzy-RED" , IFAC Journal Control Engineering Practice.




Archive of SID

[34] M.H. Yaghmaee,Hale Amintoosi, "A Fuzzy Based Active Queue Management Algorithm", in the proceeding of the International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS2003), 20-24 July, Montreal, Canada, pp.458-365,2003.

[35] V.Jacobson, "Congestion Avoidance and Control", in Proceeding of ACM SIGCOMM, pp. 314-329, August 1988

[36] K.Ramakrishnan, S.Floyd, "A Proposal to Add Explicit Congestion Notification (ECN) to IP", RFC 2481, January 1999

[37] B.Barden et al, "Recommendation on Queue Management and Congestion Avoidance in the Internet", RFC2309, 1998.

[38] A.Pitsillides, Y.Ahmet, G.Ramamutthy, "Effective Control of Traffic Flow in ATM Networks Using Fuzzy Explicit Rate Marking (FERM)", IEEE journal on Selected Areas in Communications, vol.15, No.2, February 1997

[39] Wang, Lie-Xin, " A Course in Fuzzy Systems and Control",1999



Archive of SID