Fuzzy Logic Congestion Control in TCP/IP Best-Effort Networks

C. Chrysostomou[+], A. Pitsillides[+], G. Hadjipollas[+], A. Sekercioglu[++] and M. Polycarpou[+++]

+ Department of Computer Science,*
University of Cyprus,
75 Kallipoleos Street,
P.O. Box 20537,
1678 Nicosia, Cyprus.
e-mail: {cchrys, andreas.pitsillides, hpollas}@ucy.ac.cy

++ Centre for Telecommunications and Information Engineering,
Monash University,
Melbourne, Australia.
e-mail:
ahmet@titania.ctie.monash.edu.au

+++ Department of Electrical and Computer Engineering,
University of Cyprus,
75 Kallipoleos Street,
P.O. Box 20537,
1678 Nicosia, Cyprus.
e-mail: mpolycar@ucy.ac.cy

Abstract - This paper presents a new active queue management (AQM) scheme -Fuzzy Explicit Marking (FEM)- supporting explicit congestion notification (ECN), to provide congestion control in TCP/IP best-effort networks using a fuzzy logic control approach. While many AQM mechanisms have recently been proposed, these require careful configuration of non-intuitive control parameters, and show weaknesses to detect and control congestion under dynamic traffic changes, and a slow response to regulate queues. The proposed fuzzy logic approach for congestion control allows the use of linguistic knowledge to capture the dynamics of nonlinear probability marking functions, uses multiple inputs to capture the (dynamic) state of the network more accurately, and can offer effective implementation. A simulation study over a wide range of traffic conditions shows that the FEM controller outperforms a number of representative AQM schemes in terms of queue fluctuations and delays, packet losses, and link utilization.

I. INTRODUCTION

The increased demand to use the Internet necessitates the design and utilization of effective congestion control algorithms. Recently, many active queue management (AQM) schemes have been proposed to provide high network utilization with low loss and delay by regulating queues at the bottleneck links in TCP/IP best-effort networks, including random early detection (RED) [1], adaptive RED (A-RED) [2], proportional-integral (PI) controller [3], and random exponential marking (REM) [4].

The AQM approach can be contrasted with the “Tail Drop” (TD) queue management approach, employed by common Internet routers, where the discard policy of arriving packets is based on the overflow of the output port buffer. Contrary to TD, AQM mechanisms [5] start dropping packets earlier in order to be able to notify traffic sources about the incipient stages of congestion. AQM allows the router to separate policies of dropping packets from the policies for indicating congestion. The use of Explicit Congestion Notification (ECN) [6] was proposed in order to provide TCP an alternative to packet drops as a mechanism for detecting incipient congestion in the network. The ECN scheme requires both end-to-end and network support. An AQM-enabled gateway can *mark* a packet either by dropping it or by setting a bit in the packet’s header if the transport protocol is capable of reacting to ECN. The use of ECN for notification of congestion to the end-nodes generally prevents unnecessary packet drops.

In this paper, we use fuzzy logic techniques to develop a new AQM scheme, Fuzzy Explicit Marking (FEM), to provide congestion control in TCP/IP best-effort networks. The application of fuzzy control techniques to the problem of congestion control in networks is suitable due to the difficulties in obtaining a precise mathematical model using conventional analytical methods, while some intuitive understanding of congestion control is available. The proposed fuzzy control system is designed to regulate the queues of IP routers in a predefined level, by achieving a specified target queue length (TQL), in order to maintain both high utilization and low mean delay. A fuzzy inference engine (FIE) is designed to operate on router buffer queues, and uses linguistic rules to *mark* packets in TCP/IP networks. The proposed fuzzy logic strategy is shown via simulations to be robust with respect to traffic modeling uncertainties and system nonlinearities, yet provide tight control. As a result, it can effectively regulate the queues of the bottleneck links, while achieving high utilization, low loss and delay.

The paper is organized as follows. Section II discusses important aspects of AQM. In Section III we briefly review some of the properties of Fuzzy Logic Control and present our proposed FEM implementation. Then Section IV presents simulative examples and discusses the performance of FEM. Finally in Section V we present our conclusions.

II. AQM MECHANISMS

AQM mechanisms aim to provide high link utilization with low loss rate and queuing delay, while responding quickly to load changes. Several schemes have been proposed to provide congestion control in TCP/IP networks. RED [1], which was the first AQM algorithm proposed, simply sets some minimum and maximum *marking* thresholds in the router queues. In case the

---

* Work partially supported by IST program SEACORN

average queue size exceeds the minimum threshold, RED starts randomly *marking* packets based on a probability depending on the average queue length, whereas if it exceeds the maximum threshold every packet is dropped.

The properties of RED have been extensively studied in the past few years. Issues of concern include: problems with performance of RED under different scenarios of operation and loading conditions [7]; the correct tuning of RED parameters implies a “global” parameterization that is very difficult, if not impossible to achieve as it is shown in [9]; some researchers have advocated against using RED, in part because of this tuning difficulty [8]; linearity of the dropping function has been questioned by a number of researchers (see for example [4, 10]).

Recently, new proposed AQM mechanisms have appeared to give alternative solutions, and approached the problem of congestion control differently than RED, due to the difficulties of appropriately setting RED parameters based on dynamic network conditions [8]. Specifically, REM [4] algorithm uses the instantaneous queue size and its difference from a target value to calculate the *mark* probability based on an exponential law. Also, a PI controller [3] uses classical control theory techniques to design a feedback control law for the router AQM. It introduces a TQL, in order to stabilize the router queue length around this value. Moreover, A-RED [2], proposed by the same author of RED [1], attempts to solve the problem for the need of tuning RED parameters by modifying a similar proposal [11]. In particular, A-RED adjusts the value of the maximum *mark* probability to keep the average queue size within a target range half way between the minimum and maximum thresholds. Thus, A-RED maintains a desired average TQL twice the minimum threshold (if the maximum threshold is kept three times the minimum threshold). Furthermore, A-RED also specifies a procedure for automatically setting the RED parameter of queue weight as a function of the link capacity, following the approach in [12].

However, these AQM mechanisms still require a careful configuration of non-intuitive control parameters. As indicated in Section IV, they are often non-robust to dynamic network changes, and as a result, they exhibit greater delays than the target mean queuing delay with a large delay variation, and large buffer fluctuations, and consequently cannot effectively control the router queue.

III. FUZZY LOGIC: IMPLEMENTATION OF FEM

A. *Fuzzy logic*

Fuzzy logic is one of the tools of what is commonly known as Computational Intelligence (CI). CI [13, 14] is an area of fundamental and applied research involving numerical information processing. While these techniques are not a panacea (and it’s important to view them as supplementing proven traditional techniques), we are beginning to see a lot of interest not only from the academic research community [15], but also from industry [16]. Fuzzy Logic Control (FLC) may be viewed as a way of designing feedback controllers in situations where rigorous control theoretic approaches cannot be used due to difficulties in obtaining a formal analytical model, while at the same time some intuitive understanding of the process is available. The control algorithm is encapsulated as a set of linguistic rules. FLC has been applied successfully [17, 18] for controlling systems in which analytical models are not easily obtainable or the model itself, if available, is too complex and possibly highly nonlinear.

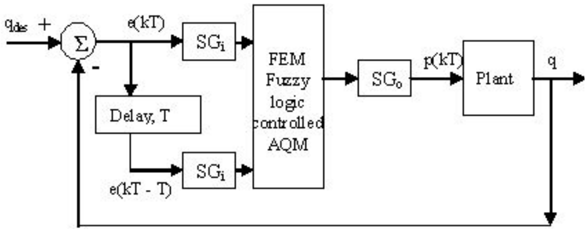


Figure 1. Fuzzy logic controlled AQM system model

In recent years, a number of research papers using fuzzy logic investigating solutions to congestion control issues, especially to ATM networks, have been published (e.g. [19]). A survey is given in [15].

B. *FEM implementation*

Our design of a fuzzy control system is based on a fuzzy logic controlled AQM scheme to provide congestion control in TCP/IP best-effort networks. The system model of FEM is shown in Figure 1, where all quantities are considered at the discrete instant kT, with T the sampling period, $e(kT) = q_{des} – q$ is the error on the controlled variable queue length, q, at each sampling period, $e(kT – T)$ is the error of queue length with a delay T (at the previous sampling period), $p(kT)$ is the *mark* probability, and SG_i and SG_o are scaling gains.

The proposed fuzzy control system is designed to regulate the queues of IP routers by achieving a specified desired TQL, q_{des}, in order to maintain both high utilization and low mean delay. A fuzzy inference engine (FIE) is designed to operate on router buffer queues, and uses linguistic rules to *mark* packets in TCP/IP networks. As shown in Figure 1, the FIE dynamically calculates the *mark* probability behavior based on two network-queue state inputs: the error on the queue length (i.e., the difference between the desired (TQL) and the current instantaneous queue length) for *two* consecutive sample periods (which can be interpreted as a prediction horizon). We have implemented FEM with *marking* capabilities, so that FEM routers have the option of either dropping a packet or setting its ECN bit in the packet header, instead of relying solely on packet drops (for the rest of the paper, by *marking* a packet it is meant setting its ECN bit). The decision of *marking* a packet is based on the *mark* probability, which is dynamically calculated by the FIE.

The scaling gains, SG_i and SG_o, shown in Figure 1, are defined as the maximum values of the universe of discourse of the FIE input and output variables, respectively. In order to achieve a normalized range of the FIE input variables from -1 to 1, the input scaling gain SG_i is set to be equal to $-1/(q_{des}–QueueBufferSize)$, if the instantaneous queue length is greater than the TQL; otherwise SG_i is equal to $1/q_{des}$. The output scaling gain SG_o is determined so that the range of outputs that are possible is the maximum, but also ensuring that the input

p(kT)		Q_{error} (kT - T)						
		NVB	NB	NS	Z	PS	PB	PVB
Q_{error} (kT)	NVB	H	H	H	H	H	H	H
	NB	B	B	B	VB	VB	H	H
	NS	T	VS	S	S	B	VB	VB
	Z	Z	Z	Z	T	VS	S	B
	PS	Z	Z	Z	Z	T	T	VS
	PB	Z	Z	Z	Z	Z	Z	T
	PVB	Z	Z	Z	Z	Z	Z	Z

Table 1. Linguistic rules – Rule base

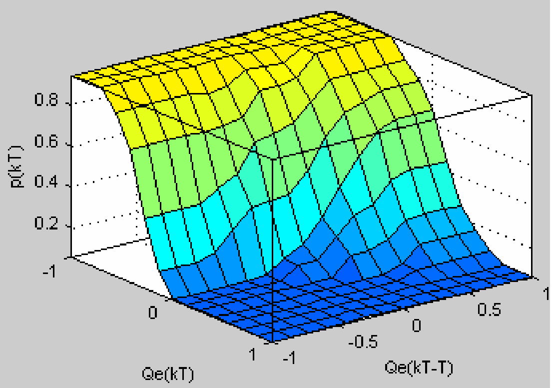
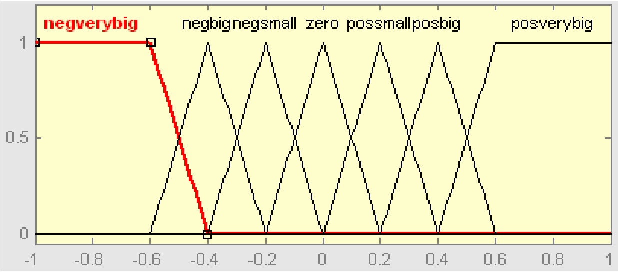


Figure 2. Decision surface of the fuzzy inference engine The control surface is shaped by the rule base and the linguistic values of the linguistic variables.

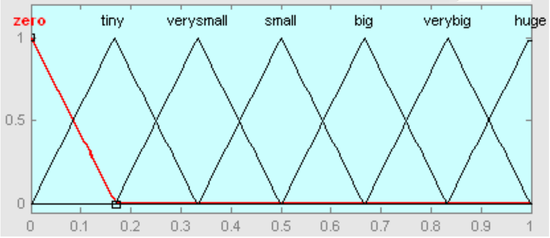
to the plant will not saturate around the maximum. Following the approach in [2], SG_o is set to a value indicating the maximum *mark* probability (e.g. 10%) that can also be adjusted in response of changes of the queue length.

The FIE uses linguistic rules to calculate the *mark* probability based on the input from the queues (see Table 1[1]). Usually multi-input FIEs can offer better ability to linguistically describe the system dynamics. We expect that we can tune the system better, and improve the behavior of the queue, by achieving high utilization, low loss and delay. The dynamic way of calculating the *mark* probability by the FIE comes from the fact that according to the error of queue length for *two* consecutive sample periods, a different set of fuzzy rules, and so inference apply. Based on these rules and inferences, the *mark* probability is calculated more dynamically than other AQM approaches [1, 2, 3, 4].

This point can be illustrated by observing the visualization of the decision surface of the FIE used in the FEM scheme (see Figure 2). An inspection of this decision surface and the linguistic rules shown in Table 1 provides hints on the operation of FEM. The *mark* probability behaviour under the region of equilibrium (i.e., where the error on the queue length is close to zero) is smoothly calculated. On the other hand, the rules are aggressive about increasing the probability of packet *marking* sharply in the region beyond the equilibrium point. These rules reflect the particular views and experiences of the designer, and are easy to relate to human reasoning processes and gathered experiences.



linguistic input variables



linguistic output variable

Figure 3. Membership functions of the linguistic values representing the input variables "normalized error on queue length for two consecutive sample periods", and the output variable "mark probability"

Usually, to define the linguistic rules of a fuzzy variable, Gaussian, triangular or trapezoidal shaped membership functions are used. Since triangular and trapezoidal shaped functions offer more computational simplicity, we have selected them for our rule base (see Figure 3). Then, the rule base is fine tuned by observing the progress of simulation, such as packet *marking* and delay occurrences, and throughput curves. The tuning can be done with different objectives in mind. For example, any gain in throughput must be traded off by a possible increase in the delay experienced at the terminal queues. Alternatively, an adaptive fuzzy logic control method [20] can be used, which is based on tuning the parameters of the fuzzy logic controller on line, using measurements from the system. The tuning objective can be based on a desired optimization criterion, for example, a trade-off between maximization of throughput with minimization of end-to-end delay experienced by the users. This is part of our future work.

The design of FEM aims to generally provide better congestion control and better utilization of the network, with lower losses and delays than other AQM schemes [1, 2, 3, 4], especially by introducing additional input variables and on-line (dynamic) adaptivity of the rule base (self-tuned).

IV. SIMULATION RESULTS

In this section we evaluate the performance and robustness of the proposed FEM AQM in a wide range of environments, and compare with other published results by taking some representative AQM schemes, namely A-RED [2], PI controller [3] and REM [4], using a recent version of NS-2 [21] simulator (Version 2.1b9a). The network topology used is shown in Figure 4. We use TCP/Newreno with an advertised window of 240 packets.

---

[1] Table content notations: negative/positive very big (NVB/PVB), negative/positive big (NB/PB), negative/positive small (NS/PS), zero (Z), huge (H), very big (VB), big (B), small (S), very small (VS), tiny (T).

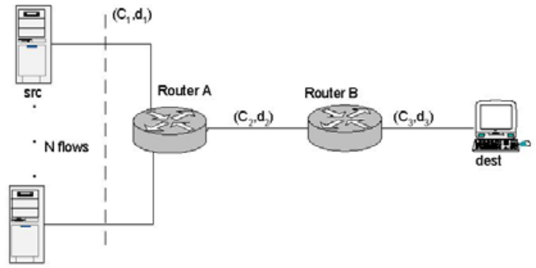


Figure 4. Network topology

The size of each packet is 1000 bytes. The buffer size of all queues is 500 packets. We use AQM in the queues of the bottleneck link between router-A and router-B. All other links have a simple TD queue. All sources (*N* flows) are greedy sustained FTP applications, except for Scenario II, where we also introduce web-like traffic. The links between all sources and router-A have the same capacity and propagation delay pair (C_1, d_1), whereas the pairs (C_2, d_2) and (C_3, d_3) define the parameters of the bottleneck link between router-A and router-B, and the link between router-B and the destination, respectively. The sampling period for FEM AQM is fixed to 0.006 sec (close to the one used in [3]). The TQL of all AQM schemes, except otherwise defined, is set to 200 packets, as this is used in [3] (for A-RED, we set the minimum threshold to 100 packets, and the maximum to 300, giving an average TQL of 200 packets). The simulation time is *100 sec*.

A. Scenario I

1. Scenario I-1

In this scenario, we examine whether FEM AQM can regulate the queue to stabilize at arbitrary TQLs. Given that N = 60, (C_1, d_1) = (15Mbps, 40ms), (C_2, d_2) = (15Mbps, 5ms), and (C_3, d_3) = (30Mbps, 5ms), we choose the TQL to be equal to 50, 100, 200, and 300 packets. The results, shown in Figure 5, show the ability of FEM AQM to adequately regulate the queue length at the target values, and, consequently, controlling the queuing delay.

2. Scenario I-2

Based on scenario *I-1*, we have conducted a comparison of FEM with the other AQM schemes, for a TQL of 200 packets (see Figure 6). FEM quickly regulates the queue to the reference value, while PI controller spends considerably long time. A-RED and REM shows good control performance, however, after a significant transient period. Furthermore, PI and REM have larger queue fluctuations than the other two schemes.

3. Scenario I-3

In order to explore the transient performance of the AQM schemes, we increase the number of flows from 60 to 100. The performance of the AQM schemes under dynamic traffic changes is also examined. We provide some time-varying dynamics by stopping half of the flows at time *t = 40 sec*, and resuming transmission at time *t = 70 sec*. The results (see Figure 7) show that FEM is very robust against the dynamic traffic changes and keeps very good response by successfully maintaining the queue length at the target value. PI and REM are not as robust, as they are slower to settle down to the reference value, resulting in large queue fluctuation. A-RED responds well, except for some larger overshoots at the time of the traffic changes.

B. Scenario II

1. Scenario II-1

In this scenario, we investigate the performance of AQM schemes under higher link capacities and propagation delays, that is, we set (C_1, d_1) = (100Mbps, 5ms), (C_2, d_2) = (15Mbps, 120ms), and (C_3, d_3) = (200Mbps, 5ms), while N = *100*. We also keep the time-varying dynamics on the network, as used in Scenario I-3. We specifically examine the effect of the round-trip time (RTT) by increasing the propagation delay of the bottleneck link (i.e., *120 ms*). In general, an increase of RTT degrades the performance of an AQM scheme. The results (see Figure 8) show the superior steady performance of FEM with stable queue length dynamics, while PI, A-RED, and REM exhibit large queue fluctuations that result in degraded utilization and high variance of queuing delay.

2. Scenario II-2

We also investigate the effect of the traffic load factor (N) in the last experiment, by increasing N from 100 to 200, 300, 400, and 500. The expected queuing delay experienced at router-A is 106.7 ms (15Mbps link capacity corresponds to 1875 packets/sec; for a TQL of 200 packets the expected mean delay is 200/1875 = 0.1067. Note that the parameters of bottleneck link capacity and TQL are the same as in [3]). Figure 9 shows the packet losses as traffic load increases, where it can be seen that FEM has the lowest drops. FEM shows stable and low packet loss over large traffic load. A-RED has the largest drops with a large increase of packet loss with respect to higher loads. Figure 10 shows the utilization of the bottleneck link with respect to the mean queuing delay. FEM outperforms other AQM schemes on both high utilization and low mean delay, thus it exhibits a more stable, and robust behavior. The other AQM schemes show a poor performance as the number of traffic load increases, achieving much lower link utilization, and large queuing delays, far beyond the expected value. Table 2 lists the statistical results of the mean queuing delay and its standard deviation. It is clear that FEM has the lowest variance in queuing delay, resulting in a stable and robust behavior. On the other hand, the other AQM schemes exhibit very large queue fluctuations with large amplitude that inevitably deteriorates delay jitter.

3. Scenario II-3

We further investigate the performance of AQM schemes by introducing additional web-like traffic that can be seen as noise-disturbance to the network. In particular, we keep the same parameters as in scenario *II-1*, without the time-varying dynamics. The number of flows is kept to 100 for FTP applications, with an additional 100 web-like traffic flows. We have conducted experiments for two specific values of the TQL (i.e., 100 and 200 packets) to examine the robustness of the AQM schemes. For both cases the results are shown in Table 3 where we obtain the mean queuing delay and its standard deviation, link utilization and packet losses. It is clear that, for both cases, FEM keeps a queuing delay close to the expected one with the lowest variance, while it exhibits the highest link utilization with the lowest drops. This is in contrast with the other AQM schemes that exhibit very large variations of the queue; consequently, this has the effect of having degraded link utilization with large number of drops.

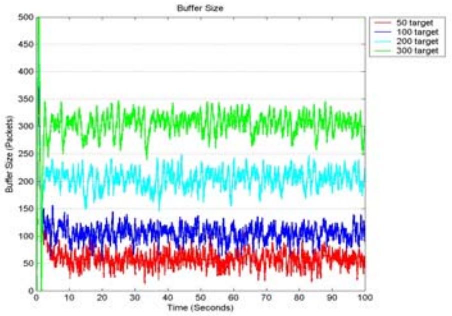
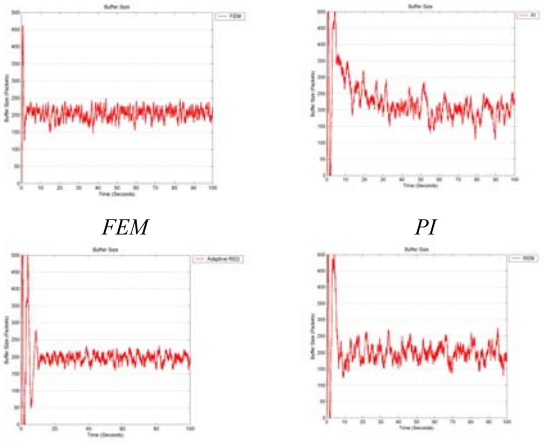


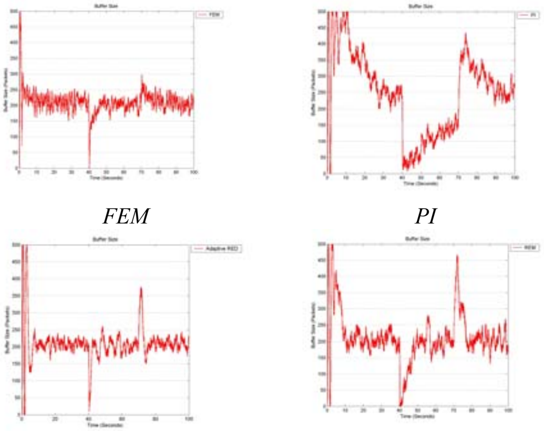
Figure 5. FEM queue length under various target values (queue ranges from 0-500 packets with a time evolution of 100sec; similarly for Figures 6-8)



FEM *PI*

A-RED *REM*

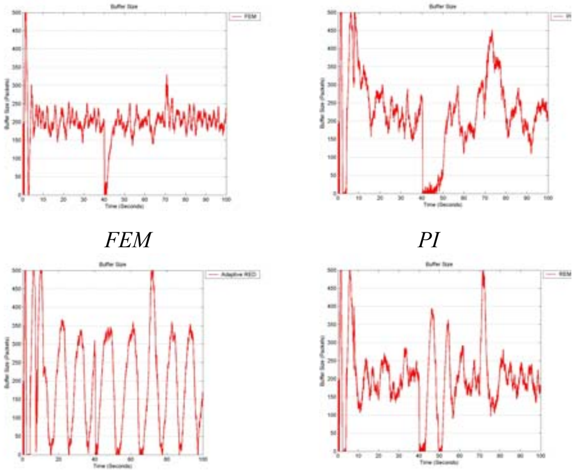
Figure 6. Scenario I-2: Queue lengths



FEM *PI*

A-RED *REM*

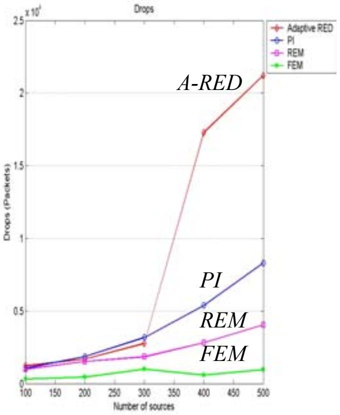
Figure 7. Scenario I-3: Queue lengths



FEM *PI*

A-RED *REM*

Figure 8. Scenario II-1: Queue lengths



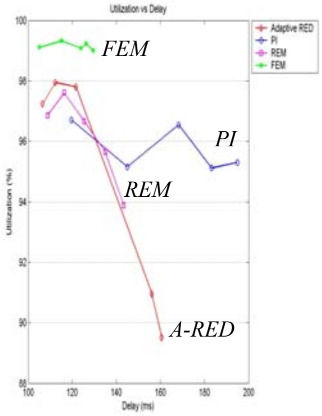


Figure 9. Scenario II-2: Packet Losses vs Traffic Load (for 100, 200, 300, 400, 500 flows)

Figure 10. Scenario II-2: Utilization vs Mean Delay (for 100, 200, 300, 400, 500 flows)

		Mean-Delay (ms)	Std-Deviation (ms)
100 Sources	**FEM**	105.343	25.2407
	PI	119.508	54.8057
	ARED	106.531	72.8443
	REM	108.769	47.7956
200 Sources	**FEM**	115.246	30.0087
	PI	144.998	85.6514
	ARED	112.356	52.2939
	REM	116.298	50.1747
300 Sources	**FEM**	123.945	36.8476
	PI	168.225	96.2637
	ARED	121.653	51.1104
	REM	125.403	63.0991
400 Sources	**FEM**	126.329	35.3921
	PI	183.278	99.527
	ARED	156.144	58.4707
	REM	134.916	75.5712
500 Sources	**FEM**	129.619	38.0314
	PI	194.903	94.0823
	ARED	160.633	58.7155
	REM	143.333	82.2324

Table 2. Scenario II-2: Summary of mean delay and standard deviation

		Mean-Delay (ms)	**Std-Deviation (ms)**	**Utilization (%)**	**Drops (packets)**
TQL 100 (expected mean delay: 53.3 ms)	**FEM**	57.7093	23.9208	99.44	167
	PI	69.6015	44.9733	97.9	1029
	ARED	57.2572	42.6883	97.6	1155
	REM	57.5126	32.8804	97.9	919
TQL 200 (expected mean delay: 106.7 ms)	**FEM**	107.227	21.393	99.28	360
	PI	136.754	37.9652	97.92	1015
	ARED	108.91	69.9759	96.85	2708
	REM	108.629	32.6228	97.89	981

Table 3. Scenario II-3: Summary of statistical results

V. CONCLUSIONS

We have presented a new AQM scheme, which we refer to as Fuzzy Explicit Marking, implemented in TCP/IP networks, using fuzzy logic techniques, to provide effective congestion control by achieving high utilization, low losses and delays. The proposed scheme is contrasted with a number of well-known AQM schemes through a wide range of scenarios.

The proposed fuzzy logic approach for congestion control is implemented with *marking* capabilities (either dropping a packet or setting its ECN bit). In this paper the design of the fuzzy knowledge base is kept simple, using a linguistic interpretation of the system behavior. We have successfully used the reported strength of fuzzy logic (a CI technique) and have addressed limitations of existing AQM algorithms implemented in TCP/IP networks. This is clearly shown from the simulative evaluation. FEM controller is shown to exhibit many desirable properties, like robustness and fast system response, and behaves better than other AQM schemes in terms of queue fluctuations and delays, packet losses, and link utilization, with capabilities of adapting to highly variability and uncertainty in network.

We believe that future work can include the design of a fuzzy model reference learning controller, which can tune the parameters of the fuzzy logic controller on line, using measurements from the system, to obtain even better behavior. Furthermore, it is worth investigating the implementation of FEM in a differentiated services environment in TCP/IP networks, using separate linguistic rules for each predefined class of service. Of course, many open issues still require investigation, such as a more formal analysis of stability issues and fairness.

From the results presented we are optimistic that the Fuzzy Control methodology will offer significant improvements on controlling congestion in TCP/IP networks.

VI. REFERENCES

[1] S. Floyd, V. Jacobson, "Random Early Detection gateways for congestion avoidance", IEEE/ACM Trans. on Networking, Aug. 1993.

[2] S. Floyd, R. Gummadi, S. Shenker, "Adaptive RED: An Algorithm for Increasing the Robustness of RED's Active Queue Management", Technical report, ICSI, August 2001.

[3] C. V. Hollot, V. Misra, D. Towsley, W.-B. Gong, "Analysis and Design of Controllers for AQM Routers Supporting TCP Flows" IEEE Transactions on Automatic Control, vol. 47, no. 6, pp. 945-959, June 2002.

[4] S. Athuraliya, V. H. Li, S. H. Low, Q. Yin, "REM: Active Queue Management", IEEE Network Magazine, 15(3), pp. 48-53, May 2001.

[5] B. Braden et al, "Recommendations on Queue Management and Congestion Avoidance in the Internet", IETF RFC2309, April 1998.

[6] K. Ramakrishnan, and S. Floyd, "The Addition of Explicit Congestion Notification (ECN) to IP", IETF RFC 3168, September 2001.

[7] Iannaccon G, Brandauer C, Ziegler T, Diot C, Fdida S, May M (2001) "Tail Drop and Active Queue Management Performance for bulk-data and Web-like Internet Traffic", 6th IEEE Symposium on Computers and Communications, Hammamet, 2001.

[8] M. May, T. Bonald, and J.C. Bolot, "Analytic Evaluation of RED Performance", Tel Aviv, IEEE Infocom 2000.

[9] W. Feng, D. Kandlur, D. Saha, and K. Shin, "Blue: A New Class of Active Queue Management Algorithms" Technical Report, UM CSE-TR-387-99, 1999.

[10] E. Plasser, T. Ziegler and P. Reichl, "On the Non-linearity of the RED Drop Function", in Proc. of Inter. Conference on Computer Communication (ICCC), August 2002.

[11] W. Feng, D. Kandlur, D. Saha, and K. Shin, "A Self-Configuring RED Gateway", IEEE Infocom, March 1999.

[12] T. Ziegler, S. Fdida, and C. Brandauer, "Stability Criteria of RED with TCP Traffic", in Proc. of IFIP ATM&IP Working Conference, Budapest, Hungary, 2001.

[13] J. C. Bezdek, "What is Computational Intelligence: Imitating Life", edited by J.M. Zurada, et al, IEEE Press, pp. 1-12, 1994.

[14] W.Pedrycz, "Computational Intelligence: An Introduction", CRC Press, 1998.

[15] A. Sekercioglu, A. Pitsillides, A. Vasilakos, "Computational intelligence in management of ATM networks", Soft Computing Journal, 5(4), pp. 257-263, 2001.

[16] B. Azvine, and A. Vasilakos, "Application of soft computing techniques to the telecommunication domain", ERUDIT Roadmap, (G. Tselentis, Ed.), pp. 89-110, 2000, http://www.erudit.de/erudit/papers/Roadmap.pdf.

[17] S. Yasunobu, S. Miyamoto, "Automatic Train Operation by Predictive Fuzzy Control", in Industrial Applicatons of Fuzzy Control, M. Sugeno, Ed., pp. 1-18, Elsevier Science Publishers, 1985.

[18] E. Morales, M. Polycarpou, N. Hemasilpin, J. Bissler, "Hierarchical Adaptive and Supervisory Control of Continuous Venovenous Hemofiltration", IEEE Transactions on Control Systems Technology, Vol. 9, No. 3, pp. 445-457, May 2001.

[19] A. Pitsillides, A. Sekercioglu, "Congestion Control", Computational Intelligence in Telecommunications Networks, (Ed. W. Pedrycz, A. V. Vasilakos), CRC Press, ISBN: 0-8493-1075-X, September 2000, pp- 109-158.

[20] Sekercioglou A, Pitsillides A, Egan GK (1994) Study of an adaptive fuzzy controller based on the adaptation of relative rule weights. In Proceedings of ANZIIS'94, Brisbane, Queensland, Australia, pp 204-208.

[21] Network Simulator, NS-2, Homepage, http://www.isi.edu/nsnam/ns/.