

# Fuzzy based PID Controller for Active Queue Management in TCP Networks

Mr. K. Vijayakumar,  
ME-II year(Applied Electronics)  
C.K.College of engineering and technology  
Cuddalore,india

Mrs. D. Sengeni M. E.,( Ph. D.),  
Assistant professor/dept of ECE,  
C.K.College of engineering and technology  
Cuddalore,india

**Abstract:** -A robust Active Queue Management (AQM) scheme based on fuzzy PID controller is introduced. In the TCP network ,AQM is important to regulate the queue length by passing or dropping the packets at the intermediate routers under dynamically changing network situations. The hybrid controller are used to reduce packet loss and improve network utilization in TCP/IP networks. Simulation and real time implementations will be carried out to demonstrate the effectiveness of the proposed method than the existing methods.

**Keywords:** AQM, Fuzzy controller, Fuzzy PID controllers, PID controllers, adaptive hybrid controllers

## 1. 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].

As PID is regarded as the standard control structures of the classical control theory, and fuzzy controllers have positioned themselves as a counterpart of classical PID controllers on the same dominant role at the knowledge rich spectrum [5].PID controllers are designed for linear systems and they provide a preferable cost/benefit ratio. However, the presences of nonlinear effects limit their performances. Fuzzy controllers are successful applied to non-linear system because of their knowledge based nonlinear structural characteristics. Hybridization of these two controller structures comes to ones mind immediately to exploit the beneficial sides of both categories. Naturally various hybrid controller structures have been arisen in literature [6, 7, 8, 9, 10, 11]. In some applications, these two control structures are combined by a switch [12, 13, 14, 15]. In [16] a fuzzy switching method between fuzzy controller and conventional PID controllers is used to achieve smooth control during switching.

Based on linearized fluid TCP/AQM model, a proportional-integral (PI) controller was developed [17] to regulate the queue length, round trip time and packet loss.

The virtual rate control (VRC) algorithm for AQM in TCP networks has been proposed in [18]. Additionally, a saturated nonlinearity of the control input usually exists in such a control problem due to the property of packet-dropping probability. Therefore, the effect of a saturated actuator should also be considered; otherwise it may cause serious degradation and instability of the network especially in large-scale, complex ones.

In this paper, a PID controller for a time-delayed TCP/IP network with input saturation is developed to achieve a stable and desired queue length, low packet loss and high link utilization. Numerical simulations show that the proposed scheme is of good stability and is robust with respect to variations in number of TCP sessions.

## 2. 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 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 [19]; the correct tuning of RED parameters implies a "global" parameterization that is very difficult, if not impossible to achieve as it is shown in [21]; some researchers have advocated against using RED, in part because of this tuning difficulty [20]; linearity of the dropping function has been questioned by a number of researchers (see for example [4, 22]).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 [20]. 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 [23]. 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, ARED 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 [24].

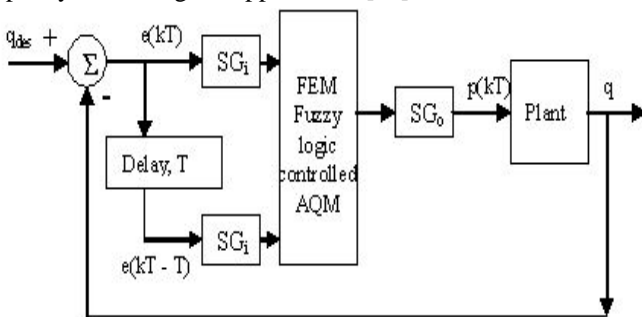


Figure: 1 Fuzzy logic controlled AQM system model

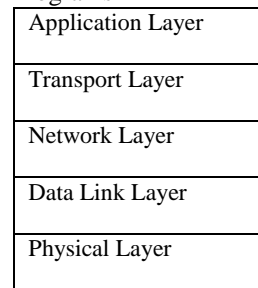
However, these AQM mechanisms still require a careful configuration of non-intuitive control parameters. They are often non-robust to dynamic network changes, and as a result, they exhibit greater delays than the target mean queuing delay with a large delay variation, and large buffer fluctuations, and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

### 3. THE TCP/IP PROTOCOL SUIT

The protocol stack used on the Internet is the Internet Protocol Suite. It is usually called TCP/IP after two of its most prominent protocols, but there are other protocols as well. The TCP/IP model is based on a five-layer model for networking. From bottom (the link) to top (the user application), these are the physical, data link, network, transport, and application layers. Not all layers are completely defined by the model, so these layers are “filled in” by external standards and protocols. The layers have names but no numbers, and although sometimes people speak of “Layer 2” or “Layer 3,” these are not TCP/IP terms. Terms like these are actually from the OSI Reference Model. The TCP/IP stack is open, which means that there are no “secrets” as to how it works. (There are “open systems” too, but with TCP/IP, the systems do not have to be “open” and often are not.) Two compatible end-system applications can communicate regardless of their underlying architectures, although the connections between layers are not defined. The TCP/IP Layers the TCP/IP

protocol stack models a series of protocol layers for networks and systems that allows communications between any types of devices. The model consists of five separate but related layers, as shown in Figure2. The Internet protocol suite is based on these five layers. TCP/IP says most about the network and transport layers, and a lot about the application layer. TCP/IP also defines how to interface the network layer with the data link and physical layers, but is not directly concerned with these two layers themselves. The Internet protocol suite assumes that a layer is there and available, so TCP/IP does not define the layers themselves. The stack consist of protocols, not implementations, so describing a layer or protocols says almost nothing about how these things should actually be built. Not all systems on a network need to implement all five layers of TCP/IP. Devices using the TCP/IP protocol stack fall into two general categories: a host or end system (ES) and an intermediate node (often a router) or an intermediate system (IS).

User Application Programs



Network Link(s)

Figure: 2 the five layers of TCP/IP.

The intermediate nodes usually only involve the first three layers of TCP/IP (although many of them still have all five layers for other reasons, as we have seen). In TCP/IP, as with most layered protocols, the most fundamental elements of the process of sending and receiving data are collected into the groups that become the layers. Each layer’s major functions are distinct from all the others, but layers can be combined for performance reasons. Each implemented layer has an interface with the layers above and below it (except for the application and physical layers, of course) and provides its defined service to the layer above and obtains services from the layer below. In other words, there is a service interface between each layer, but these are not standardized and vary widely by operating system. TCP/IP is designed to be comprehensive and flexible. It can be extended to meet new requirements, and has been. Individual layers can be combined for implementation purposes, as long as the service interfaces to the layers remain intact. Layers can even be split when necessary, and new service interfaces defined. Services are provided to the layer above after the higher layer provides the lower layer with the command, data, and necessary parameters for the lower layer to carry out the task. Layers on the same system provide and obtain services to and from adjacent layers. However, a peer-to-peer protocol process allows the same layers on different systems to communicate. The term peer means every implementation of some layer is essentially equal to all others. There is no “master” system at the protocol level.

Communications between peer layers on different systems use the defined protocols appropriate to the given layer.

#### 4. PID CONTROLLER

A proportional-integral-derivative controller (PID controller) is a common feedback loop component in industrial control systems. The controller takes a measured value from a process or other apparatus and compares it with a reference setpoint value. The difference is then used to adjust some input to the process in order to bring the process' measured value back to its desired setpoint. Unlike simpler controllers, the PID can adjust process outputs based on the history and rate of change of the error signal, which gives more accurate and stable control. PID controllers do not require advanced mathematics to design and can be easily adjusted (or "tuned") to the desired application, unlike more complicated control algorithms based on optimal control theory.

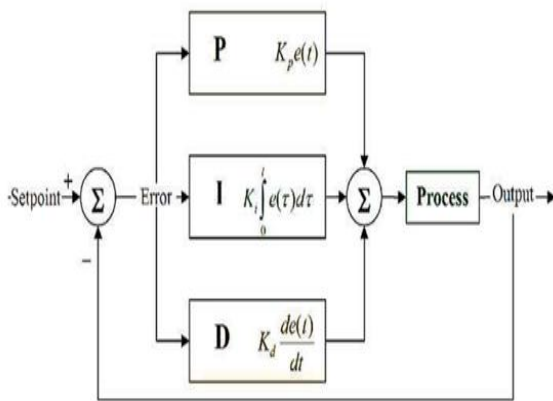


Figure: 3 A traditional PID controller

#### CONTROLLER DESCRIPTION

This section presents the approach that has been followed to deal with the AQM congestion control problem in a network with wireless links under varying traffic. Some parameters, such as the link capacity  $C$ , do not usually change, but the round trip delay  $R_0$  and the number of users  $NTCP$  vary frequently. The proposed technique is simple, but works well in networks with a wide range of users and varying delays. As network traffic changes constantly, this affects the performance of the system. We tune the PID for a certain set of conditions, but they change. The method proposed here will improve the router's performance no matter how many users or how much delay there is in the network. The PID is tuned following a model-based approach, taking the linearized dynamic model of TCPW described in previous section as the starting point. The delay  $e^{sR_0}$  is explicitly considered in the design. There are two elements in the controller  $C(s)$ :  
 1. CPID(s) is a PID controller and

2.  $K_1$  is a variable gain; a simple gain scheduling that allows the controller to work satisfactorily in a broad range of situations.

The steps in the design are:

1. Choose the worst network scenario in terms of  $NTCP$  and  $R_0$ : i.e., the biggest delay and the smallest number of users.
2. Choose the best network scenario in terms of  $NTCP$  and  $R_0$ : i.e., the smallest delay and the biggest number of users.
3. Design a controller CPID(s) that is stable in these two situations and the scenarios in between.
4. Add a simple gain scheduling  $K_1$  to improve the system's response, based on the expected variations of the system's gain. The PID should be tuned to be stable in the above mentioned situations. It is also required to have good responses in terms of settling time and overshoot. Thus, the tuning of the controller is done following the method presented in [25]: the system can be of any order and the roots can be real or complex. Using the provided Matlab toolbox, a three dimensional region is obtained.  $K_p$ ,  $K_i$  and  $K_d$  chosen inside this region give a stable closed-loop response.

#### 5. DESIGN OF FUZZY LOGIC CONTROLLER

PID controller [26] is a standard control structure for classical control theory. But the performance is greatly distorted and the efficiency is reduced due to nonlinearity in the process plant. The fuzzy PID controllers are the natural extension of their conventional version, which preserve their linear structure of PID controller. The fuzzy PID controllers are designed using fuzzy logic control principle in order to obtain a new controller that possesses analytical formulas very similar to digital PID controllers. Fuzzy PID controllers have variable control gains in their linear structure. These variable gains are nonlinear function of the errors [27] and changing rates of error signals. The main contribution of these variable gains in improving the control performance is that they are self-tuned and they can adapt to rapid changes of the errors and rate of change of error caused by time delay effects, nonlinearities and uncertainties of the underlying process.

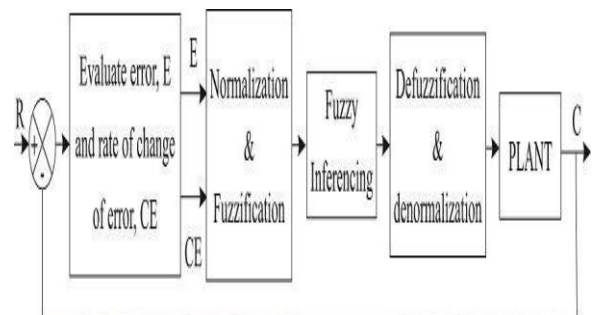


Figure: 4 shows the basic architecture of fuzzy logic controller

6. HYBRID TYPE FUZZY PID CONTROLLER.

The proposed hybrid controller which is shown in Figure 5 consists of two main parts: the classical PID controller and fuzzy PID controller.

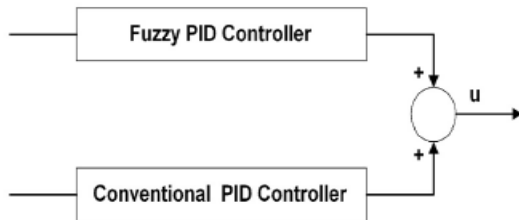


Figure: 5 Block diagram of hybrid type fuzzy PID

A standard PID controller is also known as the “three-term” controller, The “three-term” functionalities are highlighted by the following:

- i. The proportional term is providing an overall control action proportional to the error signal through the all-pass gain factor.
- ii. The integral term is reducing steady-state errors through low-frequency compensation by an integrator.
- iii. The derivative term is improving transient response through high-frequency compensation by a differentiator.

The structure of the fuzzy PID controller, which has two inputs and one rule base, is shown in Fig 6. The inputs are the classical error (e) and the rate of the change of error(ė).

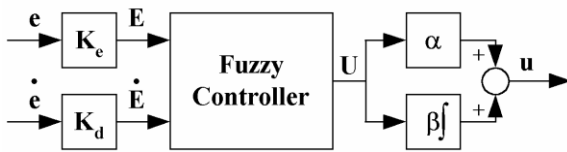


Fig6: fuzzy PID controller structure

Triangular membership functions are used for input variables as it is shown in Fig. 7. For the output variable u, singleton membership functions are defined as in Fig. 8. The fuzzy PID controller rule base composed of 49 (7x7) rules as shown in Table 1.

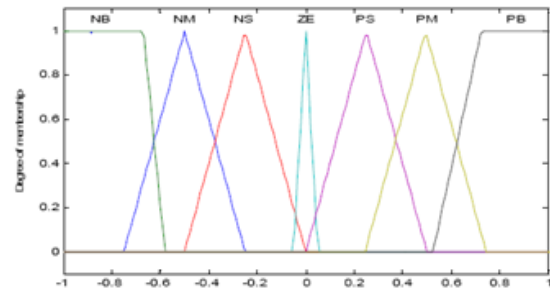


Fig7: The membership functions of e and ė

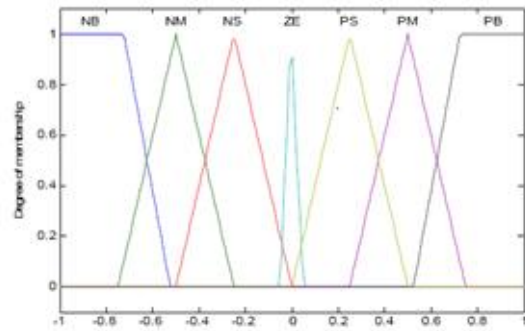


Fig8: The membership functions of u

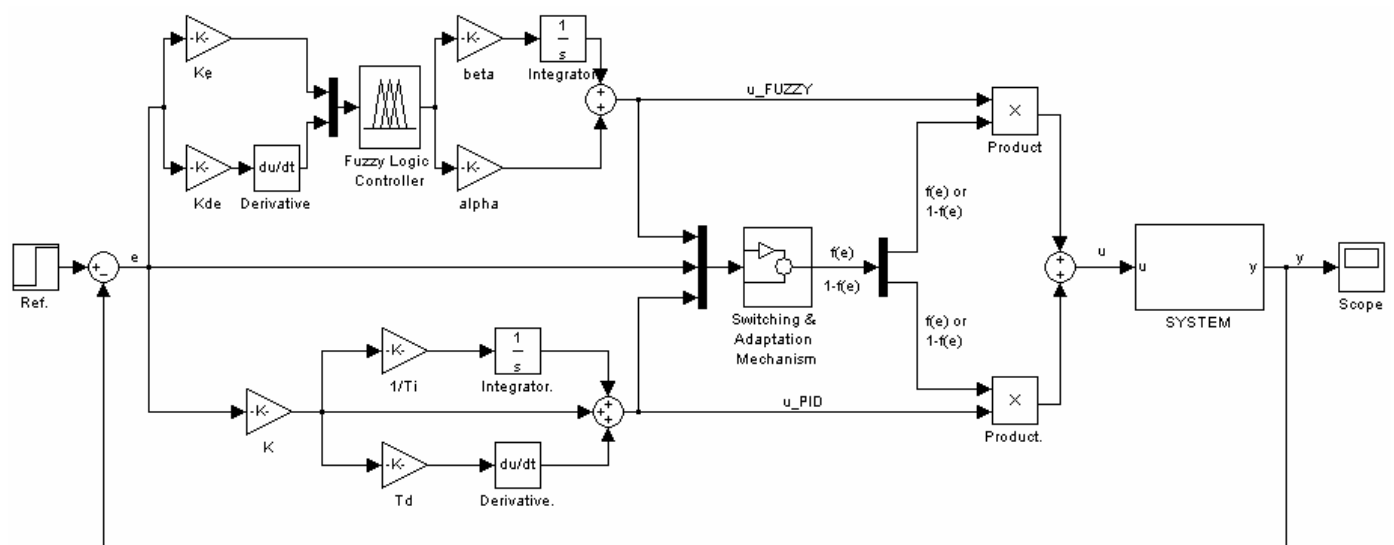


Figure: 9 Adaptive Hybrid Fuzzy PID Controller Structure

In this paper, the classical PID and fuzzy PID controller are combined by a blending mechanism which depends on a certain function of actuating error [28]. Moreover, an intelligent switching scheme is induced on the blending mechanism that makes a decision based on the priority of the two controller parts. The Matlab/Simulink simulation model of the proposed intelligent hybrid PID controller is shown in Figure 9.

The parameters of the PID controller are denoted by K, TI, and TD. These stand for proportional gain, integral and derivative time constants respectively. The parameters of the fuzzy controller are defined as  $K_e$ ,  $K_d$ ,  $\alpha$ , and  $\beta$

TABLE I PID TYPE FUZZY CONTROLLER RULE BASE

CE \ E	NB	NM	NS	Z	PS	PM	PB
NB	NB	NB	NB	NM	NS	NVS	Z
NM	NB	NB	NM	NS	NVS	Z	PVS
NS	NB	NM	NS	NVS	Z	PVS	PS
Z	NM	NS	NVS	Z	PVS	PS	PM
PS	NS	NVS	Z	PVS	PS	PM	PB
PM	NVS	Z	PVS	PS	PM	PB	PB
PB	Z	PVS	PS	PM	PB	PB	PB

A switching & blending mechanism firstly decides the dominant one of the two controller structures; namely, classical and fuzzy controllers. The outputs of the fuzzy PID controller and the classical PID controller are then multiplied by either one of the functions  $1-f(e)$  and  $f(e)$ .  $1-f(e)$  and  $f(e)$  are the weighing factors of the blending part of the mechanism. They quantify the level of the activity of the contributing controller and help us to achieve a reasonable tradeoff between the actions generated by the individual controllers. Since the function  $f(e)$  has to be positive valued, it has been selected as  $f(e)=e^2$ . Consequently the hybrid controller's output becomes either

$$U_{HYBRID} = f(e).U_{PID} + (1-f(e)).U_{FUZZY}$$

or

$$U_{HYBRID} = (1-f(e)).U_{PID} + f(e).U_{FUZZY}$$

It is obvious that when the error is large the controller output multiplied by  $f(e)$  is activated more than the other controller part. For this reason, at the early stages of the control action, the controller output which gives the faster response must be multiplied by  $f(e)$ . The switching part of the mechanism tries to catch the bigger one of the control efforts of the two main controller parts. The idea behind this is that higher control effort should produce faster system response.

Simulation Model

In order to evaluate the performance and robustness of our proposal, we performed some simulations using the topology depicted in figure 10.

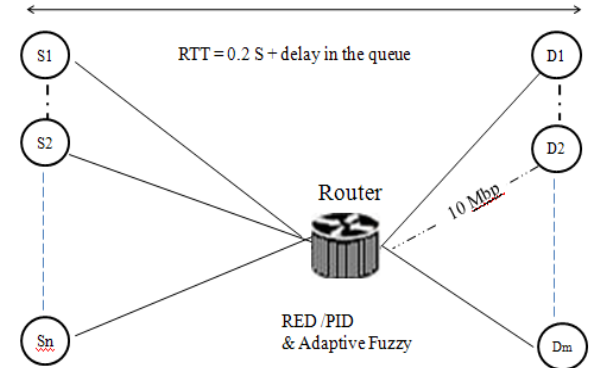


Figure:10. Simulation Network Topology

The bottleneck is located at the central router level which implements one of the three AQM algorithms: RED, PID or Adaptive Hybrid Fuzzy (our proposal). Its capacity is 10 Mbps (2621 packets/s, default packet size is 500 bytes), and delay is 20 ms. Each queue may contain up to 400 packets .

However, the most performing queue length is of 80 packets.

The controller parameters of the classical PID controller are set to  $K=2$ ,  $TI = 0.25$ ,  $TD = 0.025$  in order to have a small rise time. On the other hand, the fuzzy PID controller has the following parameters:  $\alpha = 0.05$ ,  $\beta = 4.5$ ,  $K_e = 1$ .

Experimental Results

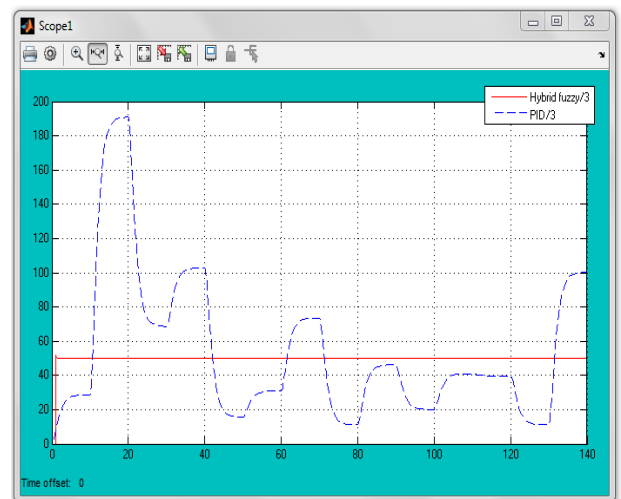


Figure: 11. Instantaneous queue length

Figure 11 presents the queue evolution, which shows that the Hybrid Fuzzy controller regulates the queue to the reference value very quickly in spite of the workload variations.

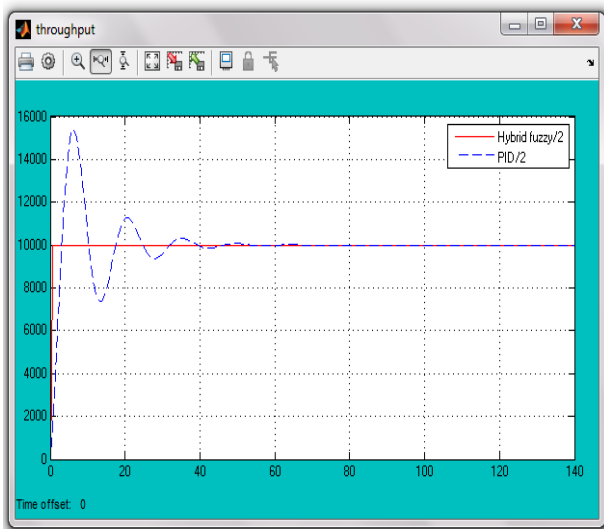


Figure: 12. Throughput

As It can be seen from Figure 12, all two controllers permit good performance. However, Hybrid Fuzzy gives a constant and maximal throughput value by stabilizing of the queue size as is shown in Figure 11.

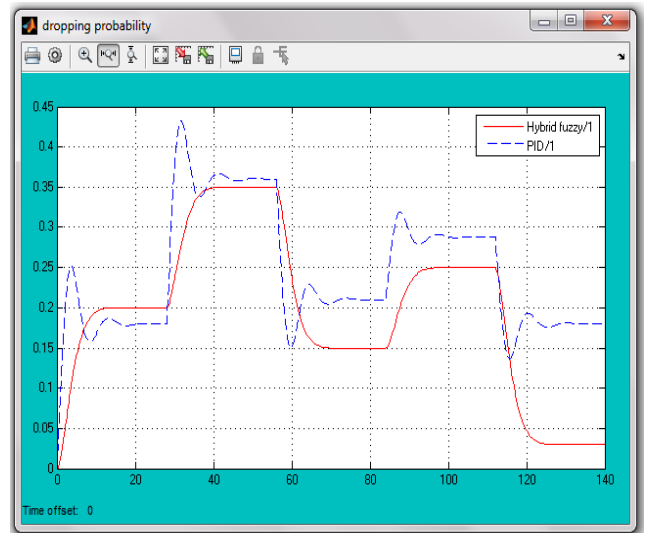


Figure 14. Dropping probability

By looking at figure 14, it can be observed that the dropping probability in Hybrid fuzzy is not regular. Moreover, it is clearly seen that the dropping probability value in Hybrid Fuzzy is better than PID.

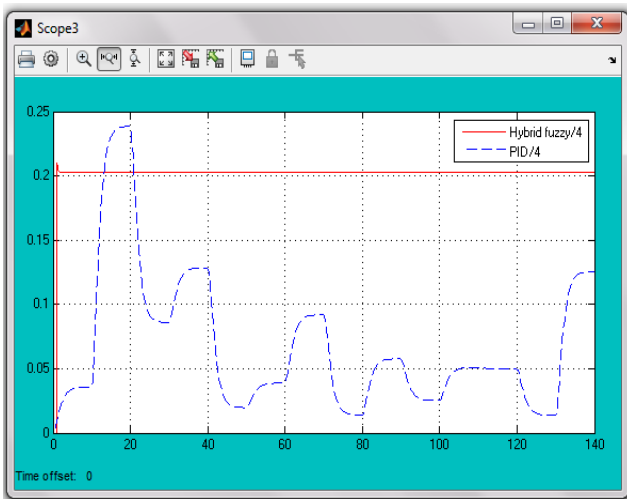


Figure :13. Queuing delay

From figure 12 and 13, it can clearly seen that queuing delays in Hybrid Fuzzy are constant. RED and PID controllers are sensitive to the number of session's variations.

### 7. CONCLUSION

A novel design methodology that blends the classical PID and the fuzzy controllers in an intelligent way is introduced in this paper, thus a new intelligent hybrid controller has been achieved. It can be implemented in TCP/IP networks to provide effective congestion control for high utilization, low losses and delays which are very important for multimedia applications. Hybrid Fuzzy controllers behave better than other AQM schemes in terms of queue fluctuations and delay, loss and link utilization of packets in TCP/IP networks. Hybrid Fuzzy Control methodology will offer significant improvements on controlling congestion in TCP/IP networks. It permits a very fast response in compared with the classical adaptive controllers, RED and PID.

### 6. 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] Åström, K.J. and Hagglund, T., 1995. PID Controllers: Theory, design and tuning. ISA Pres, Research Triangle Park, North Carolina.
- [6] Brehm, T., Rattan, K.S., 1993. "Hybrid fuzzy logic PID controller." Proceeding of the IEEE National Aerospace and Electronics Conference, Vol:2, 807-813.

- [7] Li, W. 1998. "Design of hybrid fuzzy logic proportional plus conventional integral-derivative controller." *IEEE Trans. Fuzzy Systems* 6(4), 449-463.
- [8] Li, W., Chang, X.G., Wahl, F.M., Tso, S.K., 1999. "Hybrid fuzzy P+ID control of manipulators under uncertainty." *Mechatronics* (9), 301-315.
- [9] Reznik, L., Ghanayem, O., Bourmistrov, A., 2000. "PID plus fuzzy controller structure as a design base for industrial applications." *Engineering Applications of Artificial Intelligence*, 13, 419-430.
- [10] Xiaoyin, L., Belmin, Q., 1993. "Fuzzy-PID controller." *IEE TENCON'93*, Beijing, 296-299.
- [11] Kwok, D.P., Tam, P., Li, C.K., Wang, P., 1990, "Linguistic PID controllers." *Proceedings of the 11th World Congress*, Tallin, Estonia, 7, 192-197.
- [12] Ketata, R., Geest, D.D., Titli, A., 1995. "Fuzzy controller: design, evaluation, parallel and hierarchical combination with a PID controller." *Fuzzy Sets and Systems* (71), 113-129.
- [13] Matsunaga, N., Kawaji, S., 1991. "Fuzzy hybrid control for DC servomotor." *Trans. Inst. Electrical Eng. Japan D* (111), 195-200.
- [14] Otsubo, A., Hayashi, K., Murakami, S., Maeda, M., 1998. "Fuzzy hybrid control using simplified indirect inference method." *Fuzzy Sets and Systems* (99), 265-272.
- [15] Parnichkun, M., Ngaecharoenkul, C., 2001. "Kinematics control of pneumatic systems by hybrid fuzzy PID." *Mechatronics* (11), 1001-1023.
- [16] Er, M. J., Sun, Y.L., 2001. "Hybrid fuzzy proportionalintegral plus conventional derivative control of linear and nonlinear systems." *IEEE Trans. On Industrial Electronics* 48(6), 1109-1117.
- [17] Hollot C V, Misra V, Towsley D and Gong W B 2001 On Designing Improved Controllers for AQM Routers Supporting TCP Flows in: Proc. of IEEE INFOCOM (Alaska, USA) pp.1726.
- [18.] H. Lim, K.J. Park, E.C. Park, and C.H. Choi, "Virtual rate control algorithm for active queue management in TCP networks," *IEEE Electron. Lett.*, vol.38, no.16, pp.873-874, Aug. 2002
- [19] 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.
- [20] M. May, T. Bonald, and J.C. Bolot, "Analytic Evaluation of RED Performance", Tel Aviv, IEEE Infocom 2000.
- [21] 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.
- [22] 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.
- [23] W. Feng, D. Kandlur, D. Saha, and K. Shin, "A Self-Configuring RED Gateway", IEEE Infocom, March 1999.
- [24] 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.
- [25] Hohenbichler, N. (2009). All stabilizing PID controllers for time delay systems. *Automatica*, 45, 2678-2684.
- [26] Atef Saleh Othman Al-Mashakbeh, "PID Control of DC Motor," *European Journal of Scientific Research*, Vol. 35, No. 2, 2009, pp. 198-203.
- [27] B Allaoua, B Gasbaoui, B Mebarki, "Setting up PID DC motor speed control alteration parameters using PSO strategy," *Leonardo Electronic Journal of Practices and technologies*, 2009, pp. 19-32.
- [28] [13.] I. Erenoglu, I. Eksin, E. Yesil, M. Guzelkaya, "AN INTELLIGENT HYBRID FUZZY PID CONTROLLER", European Conference on Modelling and Simulation, Wolfgang Borutzky, Alessandra Orsoni, Richard Zobel ©ECMS, 2006