Engineering and Technology Journal
Journal homepage: engtechjournal.org

Congestion Avoidance and Control in Internet Router Based on Fuzzy AQM

Hanan M. Kadhim [a]*, **Ahmed A. Oglah** [b]

[a] Department of Control and Systems Engineering, University of Technology, Baghdad, Iraq
61276@student.uotechnology.edu.iq

[b] Department of Control and Systems Engineering, University of Technology, Baghdad, Iraq
60008@uotechnology.edu.iq

*Corresponding author.

Submitted: 16/08/2020 Accepted: 30/10/2020 Published: 25/02/2021

KEYWORDS

TCP/AQM model, fuzzy PID, congestion control, computer network.

ABSTRACT

The internet has made the world a little community, linking millions of people, organizations, and equipment for different purposes. The great impact of these networks in our lives makes their efficiency a vital matter to take care of, and this needs handling some problems including congestion. In this paper, the fuzzy-PID controller is used to control the nonlinear TCP / AQM model. This controller adjusts congestion of the computer network and commits controlled pressurized signaling features. Many experiments were carried out using different network parameter values, various queue sizes, and additional disturbances to verify the robustness and efficiency of the proposed controller. From all experiments carried out in the NS-2 simulator version (2.35), the results show the superiority of the FPID controller under different network traffic conditions.

How to cite this article: H. M. Kadhim and A. A., Oglah, "Congestion Avoidance and Control in Internet Router Based on Fuzzy AQM," Engineering and Technology Journal, Vol. 39, Part A, No. 02, pp. 233-247, 2021.
DOI: https://doi.org/10.30684/etj.v39i2A.1799

1. INTRODUCTION

Through many bandwidth-limited links, senders and receivers are communicating across the network. The network is congested when the flow rate reaches the connection's capacity [1]. There're two types to control the congestion: first is Transmission Control Protocol (TCP) which is based on end hosts, and the second is AQM which is based on intermediate routers. End hosts introduced multiple congestion management strategies, such as Slow Start, AIMD, and Fast Retransmit. Such techniques are inadequate to prevent congestion, as they only control the transmit rate after packet-drop detection. Active queue management, which is based on the closed-loop control of TCP, is implemented to compensate for the TCP deficiency. The fundamental idea behind AQM is: firstly, to sample a possible congestion signal in real-time, then to apply a queue control approach to control the drop probabilities of packets, whilst dropping packets at a random manner before buffer overflow happens and lastly, sources adjust their sending rate to reduce network congestion or enhance the link utilization by the feedback signal (packet-drop/ ECN) [2].

There are many proposed AQM schemes. The first AQM method, known as Random Early Detection (RED) [3], has been implemented for routers to solve the synchronization problem and keep the average queue length small but it sentient to traffic loads and configuration of the parameters. Intelligent computing technologies such as fuzzy inference and neural networks have provided intensive research in the control and networking community over the last decades as a powerful technique to deal with nonlinearity and uncertainty in the systems. Such approaches can be applied to the AQM problem to achieve better reliability and robustness in a complex network environment. Several (AQMs) based on the fuzzy logic were brought forward. H. Ashtiani et al. [4], 2010 proposed an AQM theme based on fuzzy logic techniques as a (hybrid-fuzzy-PID) controller to offer efficient crowding control with less delay time, high utilization, and low packets loss. Results demonstrated that the Hybrid Fuzzy Control approach can significantly improve congestion control over TCP / IP networks. It allows a very fast response in comparison with the traditional adaptive controllers, RED and PID.

In 2011 [5], the Fuzzy Proportional-Integral (FPI) controller was developed as an (AQM) in the Internet router to improve the efficiency of PI controllers to avoid congestion in computer networks. The genetic algorithm (GA) was applied as an optimization tool to adjust the Fuzzy-PI parameters. J. V. Chen et al. [6] in 2012, proposed a new AQM system called FPID for improving network congestion, that combines the random early detection (RED) system and fuzzy proportional integral derivative methodology (Fuzzy-PID). The Fuzzy logic applied together with PID helped to keep the buffer queue size near the target value. A simulation test in this paper was performed and evaluated in comparison to the popular methods of active queue management. The results demonstrated the superiority of the proposed system. J. Liu et al. [7] in 2013, et al. introduced a distributed traffic management system in which routers are configured with intelligent data rate controllers for handling traffic mass using fuzzy logic theory to develop an explicit rate-based traffic control scheme (called the Intel Rate controller) for high-speed IP networks. In 2016, three robust control methods were proposed for the design of AQM strategies: conventional H∞ controller, robust (PSO)-based PID (PSOPID) controller, and robust (ACO)-based PID (ACOPID) controller. The results showed that the suggested robust ACOPID provided more acceptable performance than the PSOPID controller and some controllers presented in previous works [8].

In this work, a conventional PID controller and a PID-like fuzzy logic controller are implemented to control a nonlinear TCP / AQM model. Those controllers enable control of congestion within computer networks to provide the optimal signal features.

The article is arranged as: Section 2, introduces a dynamic model of (TCP / AQM). In Section 3, the controllers: conventional PID controller and PID-like- FLC are designed while section 4 presents the performance evaluation for two controllers. In section 5, the comparisons and discussion are introduced while section 6 demonstrates the conclusions from the article.

2. DYNAMIC MODEL OF TCP/AQM

A dynamic TCP model conduct was formed with the use a fluid-flow and stochastic differential equation analysis [9]. This model links the average value of the main network variables and the following coupled nonlinear differential equations are defined as follows [10]:

$$\dot{W}(t) = \frac{1}{R(t)} - \frac{W(t)W\big(t - R(t)\big)}{2R\big(t - R(t)\big)} P\big(t - R(t)\big) \tag{1}$$

$$\dot{q}(t) = \begin{cases} \dfrac{W(t)}{R(t)} N(t) - C(t) \ ; q(t) > 0 \\ 0 \ ; \text{else} \end{cases} \tag{2}$$

Such that (W) is predicted size of the TCP window (packets), (q) refers to the size of queue in (packet), (t) is a time in (sec), R is a round trip time in (sec), N represents TCP sessions, C is the capacity of the link (packet/sec) , P is a packet marking/dropping probability. The marking probability should be only between [0, 1]. A round trip time is [10]:

$$R(t) = \frac{q(t)}{C(t)} + T_p \tag{3}$$

Where T_p is propagation time (sec). These differential equations were illustrated in the block diagram shown in Figure 1, which highlights the TCP window-control and queue dynamics.

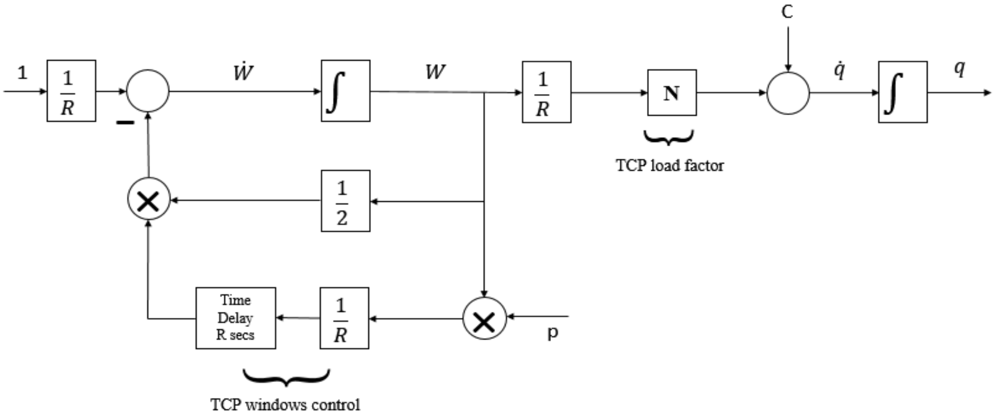


Figure 1: Block-diagram of TCP's congestion-avoidance flow-control mode [10]

The TCP / AQM models ' dynamics are often analyzed with the assistance of linearization around the equilibrium points of the nonlinear system established for investigating the stability of TCP / AQM around the equilibrium. Linearization, miss to capture system trajectories through various regions determined by nonlinear equations derived [11]. Besides, even when a linearized system is stable at equilibrium, there is no assurance that the nonlinear system will stay stable [12], particularly if the deviations from the equilibrium are at times high.

3. CONTROLLERS DESIGN

I. Conventional PID-controller

The equation for continuous-time PID controller was explained as following [13]:

$$u(t) = K_p e(t) + K_i \int e(t)dt + K_d \frac{de(t)}{dt} \tag{4}$$

Where, K_p . K_i and K_d represented a proportional, integral, and derivative gain, respectively. Figure 2 shows a block diagram of a PID controller.

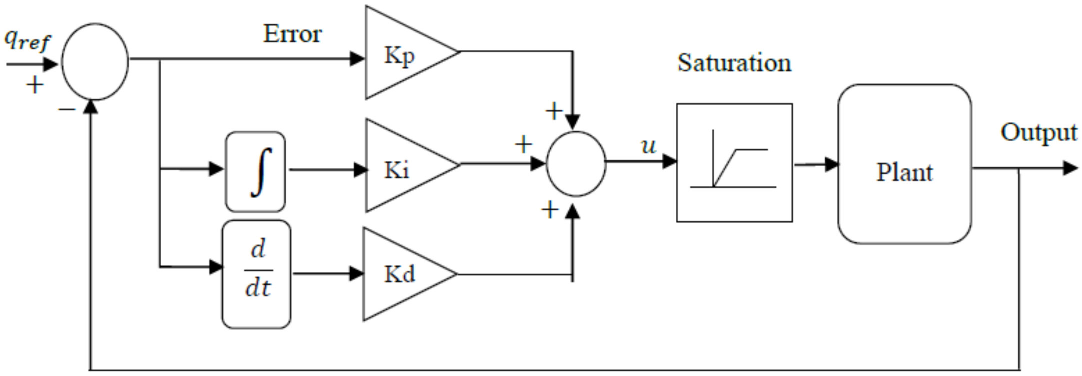


Figure 2: The block diagram for a PID controller [14]

II. PID-like fuzzy logic controller

As for IC (Intelligence Control), fuzzy logic control was considered. It is a procedure used to construct robust systems that can deal with popular adverse factors like nonlinearity in the system, uncertainty of parameters, inaccuracy of measurement, and modeling. Therefore, the fuzzy logic principle offers a simple design methodology for controllers based on expert experience that is similar to human decisions and helps engineers to easily model complex, non-linear systems. Fuzzy logic control has been commonly applied in industrial process control and showed exceptional and mature precision, transient response, robustness, and stability control performance [15].

E	**EC**	**NB**	**NM**	**NS**	**ZE**	**PS**	**PM**	**PB**

The FLC is consisting of many elements which are: (fuzzifier, defuzzifier, inference engine, and rule base). Inputs and output(s) are crisp numbers, while the block of fuzzification generates fuzzy sets from the crisp inputs. The inference process generates fuzzy conclusions from fuzzy rules that are in the rule base. The block of defuzzification transforms those fuzzy conclusions to the crisp output [13]. Figure 3 shows a block diagram for PID-FLC as AQM.

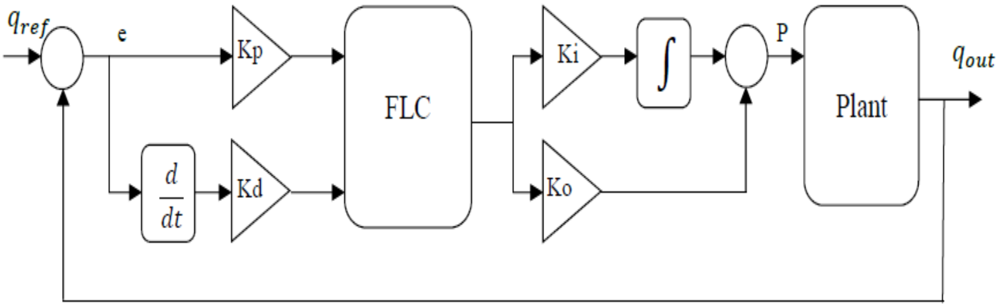


Figure 3: Block diagram of PID-FLC for AQM

A two inputs error (E) and change of error (EC) membership functions are formed by seven triangles shaped with range is equal to [-1, 1] and seven singletons shaped for output (u) with range is equal to [-1, 1]. Figure 4 shows the input and output MFs. Table 1 and Table 2 represent the linguistic variable of FLC membership functions and the fuzzy rule base, respectively [16]. The flowchart of the FPID algorithm was shown in Figure 5, where qmax represented the highest length of the queue for the AQM router which equal 800 packets

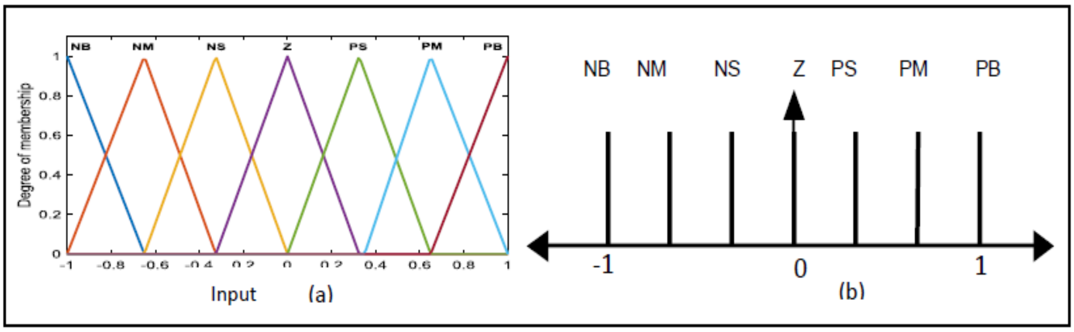


Figure 4: Membership function, (a) input (E and EC), (b) output

TABLE I: Linguistics description of MFs

Item	linguistics description	Linguistics abbreviation
1	NB	Negative Big
2	NM	Negative Medium
3	NS	Negative Small
4	Z	Zero
5	PS	Positive Small
6	PM	Positive Medium
7	PB	Positive Big

TABLE II: Fuzzy rule base

NB	NB	NB	NB	NB	NB	NM	ZE
NM	NB	NB	NB	NB	NM	ZE	PM
NS	NB	NB	NB	NM	ZE	PM	PB
ZE	NB	NB	NM	ZE	PM	PB	PB
PS	NB	NM	ZE	PM	PB	PB	PB
PM	NM	ZE	PM	PB	PB	PB	PB
PB	ZE	PM	PB	PB	PB	PB	PB

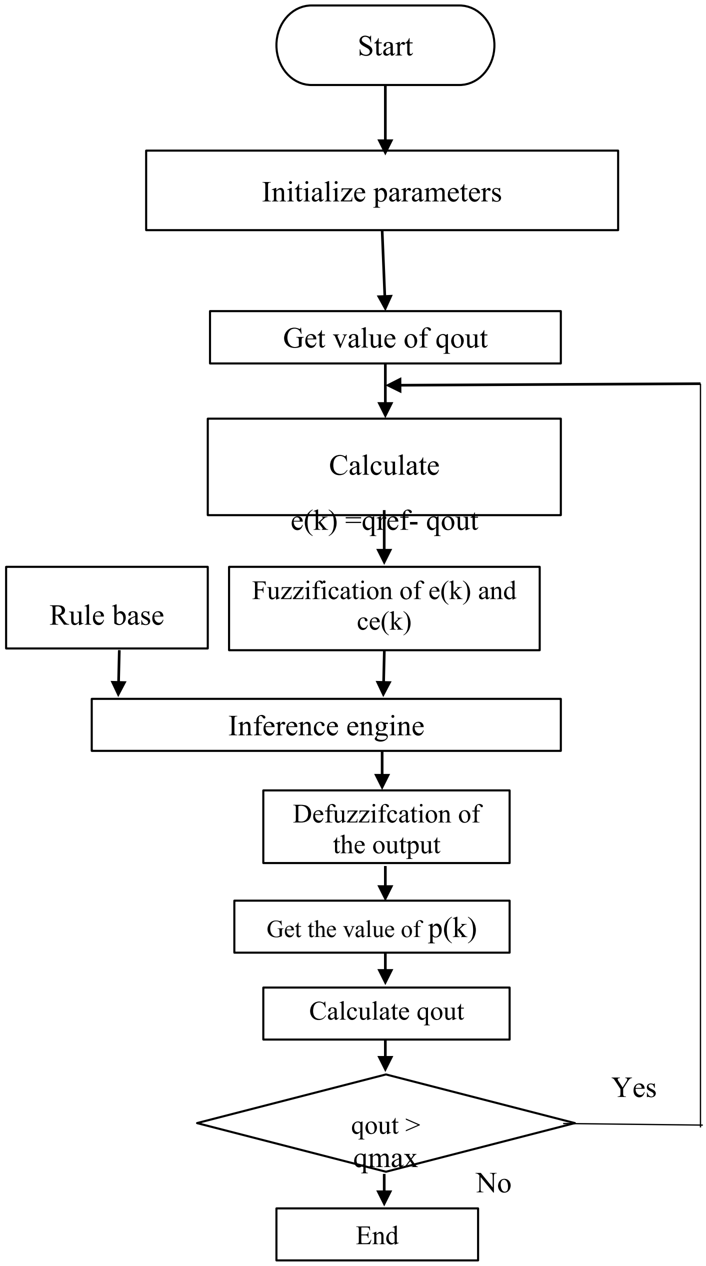


Figure 5: flowchart of the FPID algorithm

4. PERFORMANCE EVALUATION

The non-linear simulation is performed by using Network Simulator-2 (NS-2) package version (NS-2.35) to evaluate the performance of the designed FPID controller and compare it with the PID controller under a more dynamic traffic situation using the topology depicted in Figure 6.

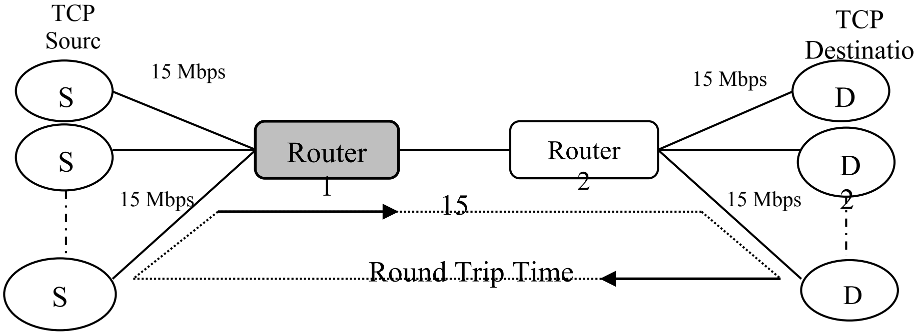


Figure 6: Simulation Network Topology

The link capacity is set to **C=3750** packets/sec which equivalents to a **15**Mbps where the packet is **500** bytes, the reference queue length and the propagation delay (Tp) is set to **200** packets and **5 msec** respectively. The highest length of queue for the AQM router which is Router 1 (qmax) is **800** packets. The number of TCP sessions (**N**) is **60** source and destination. An AQM scheme (FPID) is used at Router 1, and the Drop-tail is used at rest gateways. Table 3 shows a parameter settings of AQM controllers (PID, and FPID).

TABLE III: Parameters of AQM Controller

Title	PID	FPID
Kp	6e-6	0.00119
Ki	9e-6	0.0222
Kd	3e-5	0.01
Ko		0.001

For evaluating the efficiency and robustness of the FPID controller, many experiments have been conducted by changing various network parameters values, several values of queue sizes, and additional disturbances.

I. Changing the number of TCP sessions (N)

The FPID controller stability and robustness have been analyzed under FTP flows. The propagation time (Tp) and the link capacity (C) were set to 5 ms and 5 Mbps respectively, while the TCP sessions (N) are chosen to be 60, 80, 100, and 200. The system response using the PID controller and FPID controller is shown in Figures 7 and 8. These figures showed that both PID and FPID controllers were succeeded in making the queue size tracking its desired level even with changes in the number of TCP sessions. Table 4 displayed results of (mean) and (standard deviation) in the length of queue of a PID and FPID controller. We can be deduced that a more desirable response could be achieved by the FPID controller.

TABLE IV: Mean and standard deviation values for a length of queue in various number of TCP sessions (N)

TCP source	Mean		Standard deviation	
	PID	*FPID*	*PID*	*FPID*
60	215.43	196.92	42.25	38.29
80	232.78	211.43	34.73	30.62
100	244.24	220.39	27.24	22.0
200	286.49	241.12	25.89	20.5

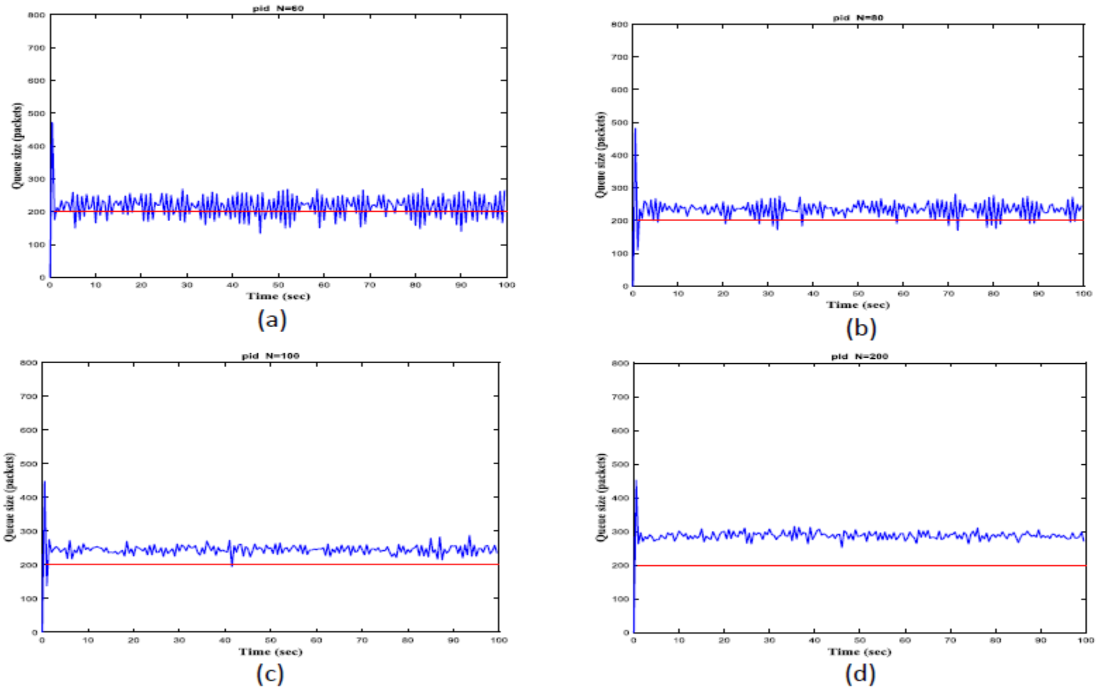


Figure 7: System response using the PID controller for a different numbers of TCP sessions (a) *N*=60 (b) *N*=80 (c) *N*=100 and (d) *N*=200

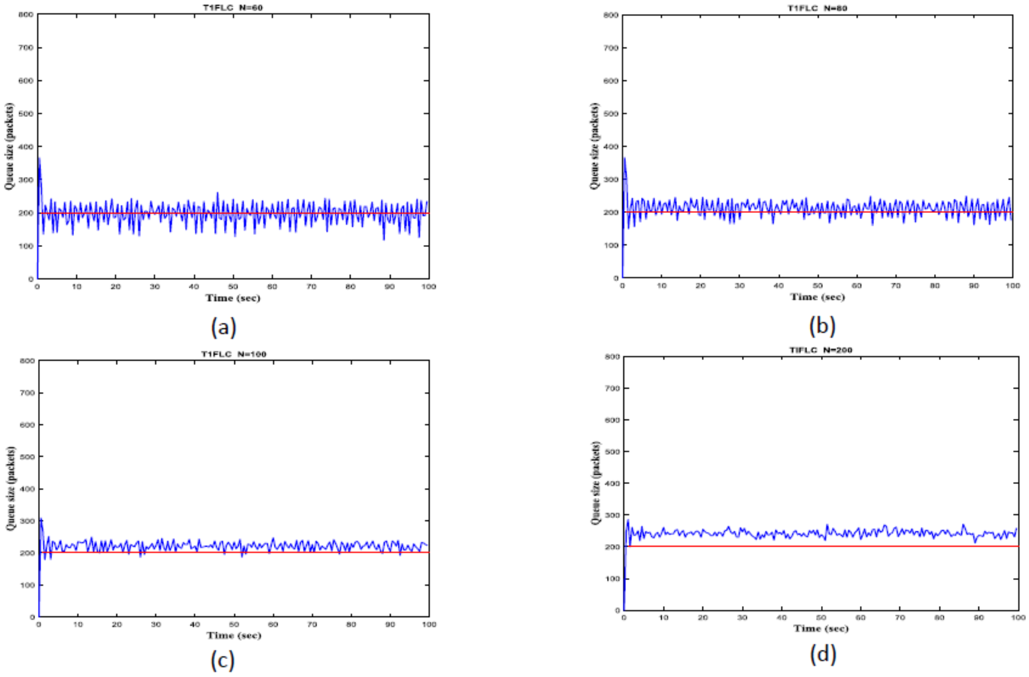
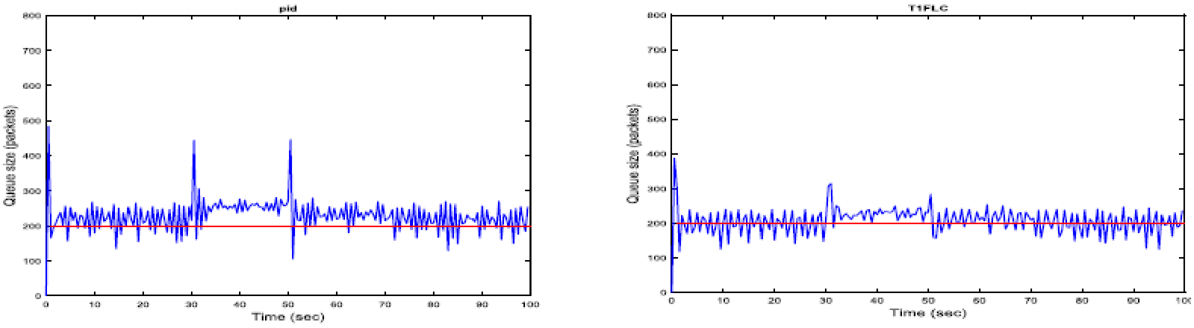


Figure 8: System response using the FPID controller for a different numbers of TCP sessions (a) *N*=60 (b) *N*=80 (c) *N*=100 and (d) *N*=200

II. Sudden variation in the number of TCP flows

The FPID controller robustness was evaluated in a more dynamic situation; where the number of long-lived TCP flows (N) has been changed suddenly. The number of TCP flows that are used is 60, 120, 80 and 60 at times 0, 30, 50 and 70, respectively as illustrated in Figure (9). The figure indicates that a queue for the PID controller is more fluctuating. The use of FPID controllers reduced the fluctuations and provided a much more stable queue. The (mean) and (standard deviation) for a length of queue for a (PID controller) was:- (mean) = 228.93; (standard deviation) = 48.34, while the FPID controller values are : mean = 207.28 and standard deviation = 38.91.



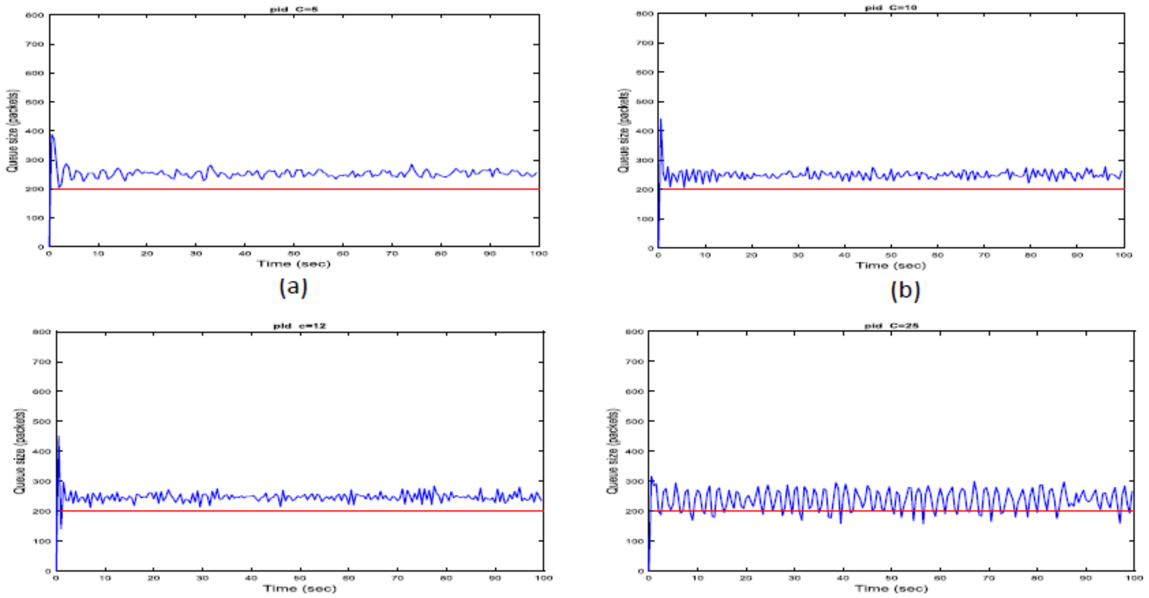
(a) (b)

Figure 9: System pesponse using the two controllers for a sudden variation in the TCP sessions (a) PID controller (b) FPID controller

III. Changing the link capacity value (C)

Some experiments have been carried out to study the response of the PID and FPID controllers for the change of the capacity of the link. The performances of the system using the PID and FPID controllers with the following capacity of the link values: (5, 10, 12, and 25) Mbps were illustrated within figures 10, 11.

The rest of the parameters were set as (N = 100) and (Tp = 5 ms). The figures proved the FPID controller was succeeded in making the queue size tracking its desired level even with changes in link capacity values. Table 5 displayed results of (mean) and (standard deviation) in the length of queue of a PID controller and FPID controller.



(a) (b) (c) (d)

Figure 10: System response using the PID controller for the different *C* values

(a) C=(5 Mbps) (b) C=(10 Mbps) (c) C=(12 Mbps) (d) C=(25 Mbps)

TABLE V: (Mean) and (standard deviation) values for a length of queue in various **link capacity values (C)**

C (Mbps)	Mean		Standard deviation	
	PID	*FPID*	*PID*	*FPID*
5	252.39	225.27	25.03	19.40
10	248.82	222.03	25.81	21.06
12	246.24	222.24	27.83	20.31
25	236.97	215.86	39.94	37.38

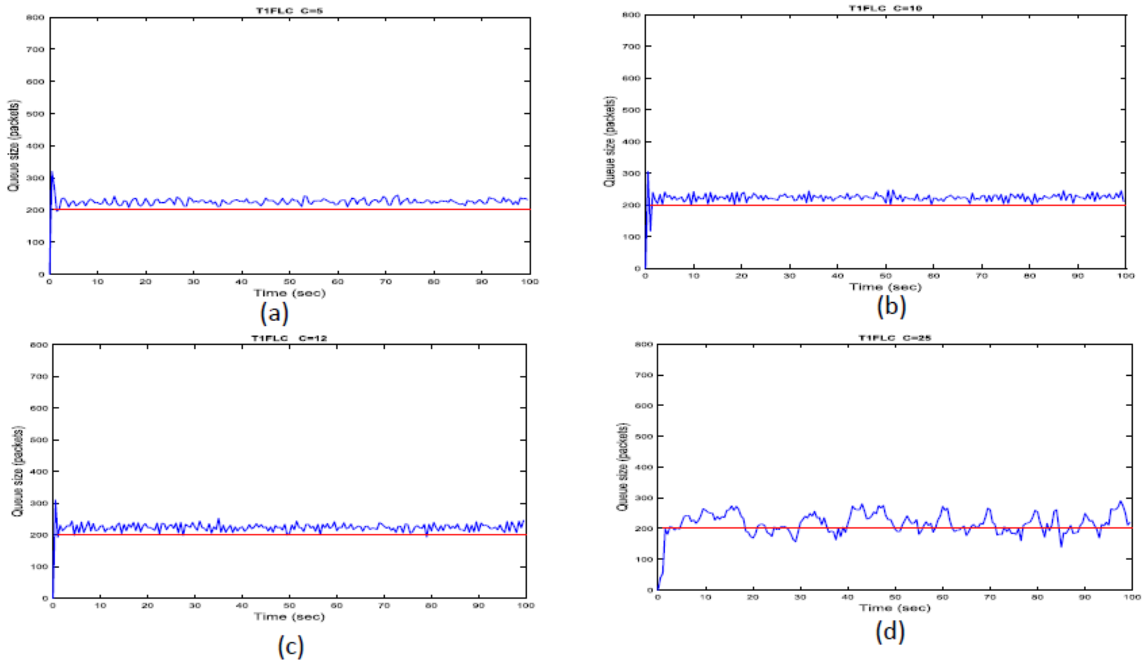


Figure 11: System response using the FPID controller for the different *C* values (a) C=(5 Mbps) (b) C=(10 Mbps) (c) C=(12 Mbps) (d) C=(25 Mbps)

IV. Disturbance with different FTP connections

File transfer protocol (FTP) is used to establish rules for file exchange among computers. It is responsible for transfer files from the server to the client computers and move files from the client to the server. The robustness of the proposed controllers was examined by applying an additional FTP connection as a disturbance. The TCP sessions were set to 100, and further FTP connections of (100, 200, 300, 500) were applied in an interval within (50 and 80) second. The system response using the PID and FPID controllers can be seen in Figures 12 and 13, respectively. Figures show that the system response with FPID controller appears to be more robust compared to the PID controller. This can be seen in Table 6 by looking at the lower means and standard deviation of FPID compared to PID controller.

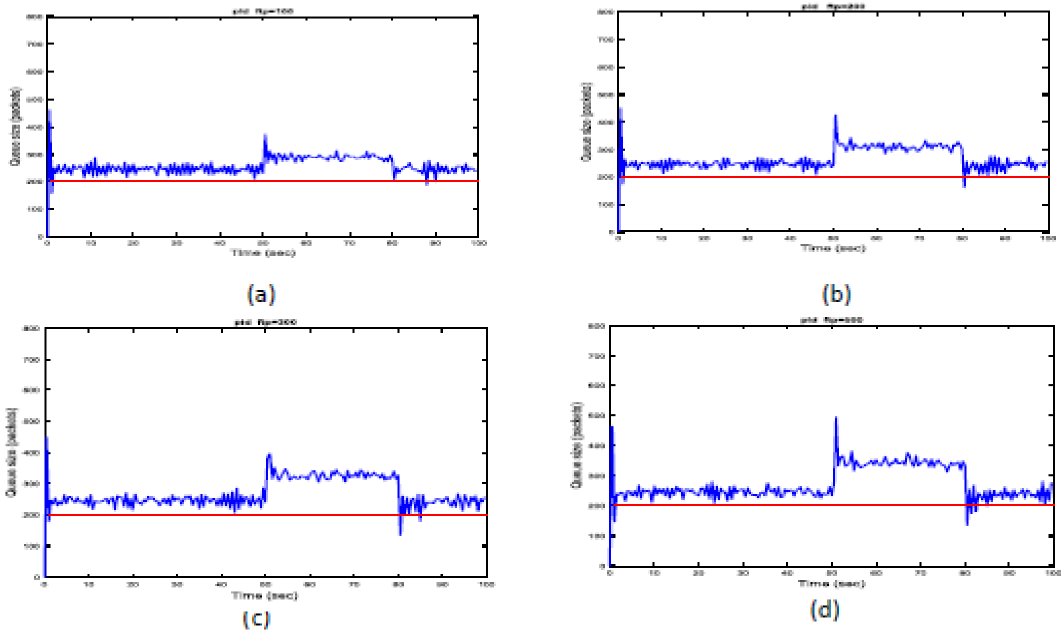


Figure 12: System response using the PID controller with the disturbance of further FTP connections (a) (FTP=100) (b) (FTP=200) (c) (FTP=300) (d) (FTP=500)

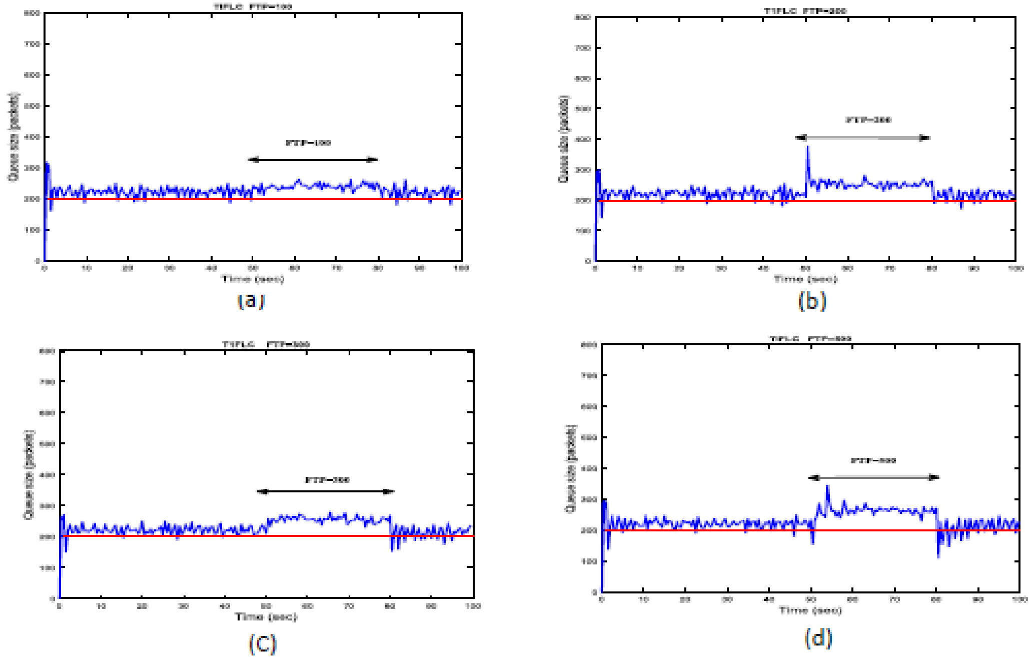


Figure 13: System response using the FPID controller with the disturbance of further FTP connections (a) (FTP=100) (b) (FTP=200) (c) (FTP=300) (d) (FTP=500)

TABLE VI: (Mean) and (standard deviation) for a length of queue with a disturbance of further FTP connections

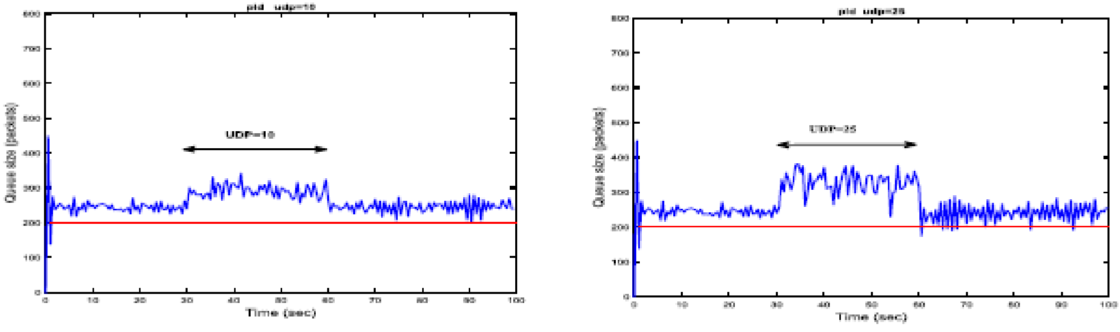
FTP Connections	Mean		Standard deviation	
	PID	*FPID*	*PID*	*FPID*
100	257.42	226.22	36.26	25.36
200	264.16	228.71	43.89	30.09
300	268.13	229.57	49.18	28.66
500	271.62	230.22	57.63	34.75

V. Disturbance with additional UDP flows

A different form of disturbance represented by non-responsive UDP flows was used to check the robustness of proposed controllers. The additional UDP flows (10 and 25) was applied in an interval within (30 and 60) second. System response using PID and FPID controllers under additional UDP flows can be seen in Figures 14 and 15. The simulation results were showed that the FPID controllers gave a good response when the additional UDP flows are 10 with lower mean and standard deviation. on the other hand, as increase the UDP flows to 25, FPID controller response was showed an oscillation from the desired length of the queue while the PID controller gave a better response with lower (mean) and (standard deviation). However, a response oscillation was within an acceptable value (It did not exceed the maximum buffer size of the router). Table 7 displays results (mean) and (standard deviation) in the length of queue of a PID and FPID controllers.

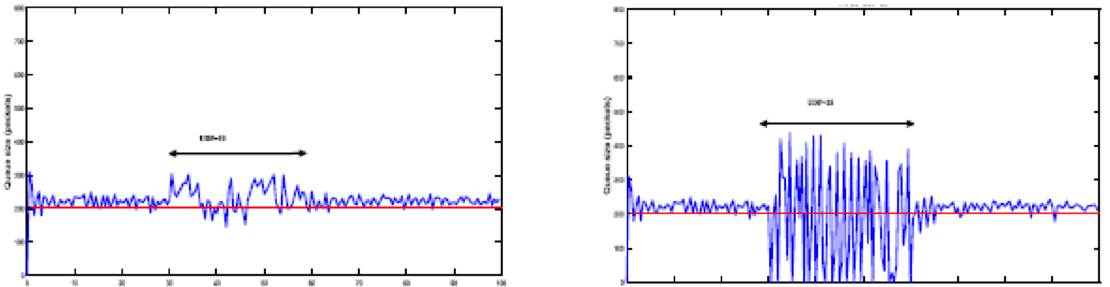
TABLE VII: (Mean) and (standard) deviation values for a length of queue within a disturbance of further UDP flows

UDP flows	Mean		Standard deviation	
	PID	*FPID*	*PID*	*FPID*
10	257.72	224.74	36.28	30.52
25	266.56	208.18	53.09	91.12



(a) (b)

Figure 14: System response of the PID controller with a disturbance as a further UDP flows a- (UDP=10) b- (UDP=25)

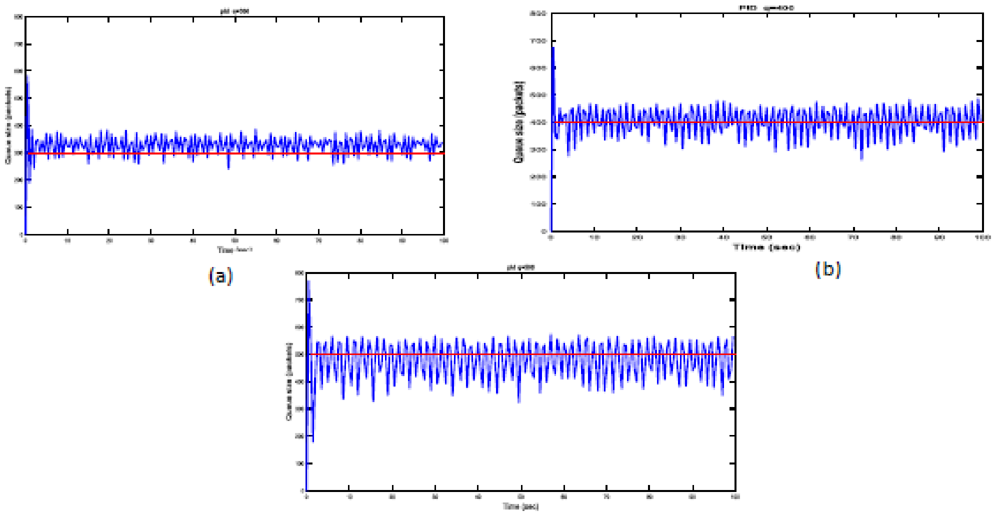


(a) (b)

Figure 15: System response of the FPID controller with a disturbance as a further UDP flows a- (UDP=10) b- (UDP=25)

VI. Changing size of reference queue size

The robustness of the proposed controllers was checked under a different reference queue size where the queue size was increased by 50% and 100% and 150% which equivalent to (300, 400, and 500 packets) respectively. The simulation results of the system using PID and FPID controller was displayed in figures 16 and 17. Both controllers were succeeded in making the queue size tracking its desired level and the FPID controller was superior with lower mean and standard deviation compared to the PID controller as shown in Table 8. Besides, the response of the system using a PID controller was showed an increase in the maximum overshoot in contrast to the FPID controller that has not exceeded 550 packets.



(a) (b) (c)

Figure 16: System response using the PID controller with various queue sizes a- ($q_{reference}$ = 300) b- ($q_{reference}$ = 400) c- ($q_{reference}$= 500)

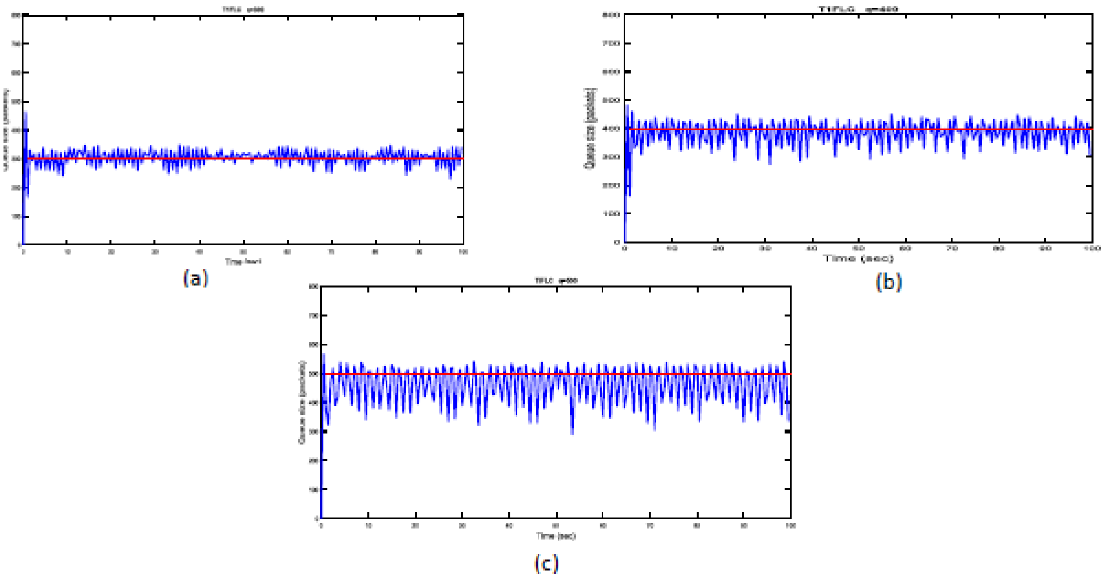


Figure 17: System response using the FPID controller with various queue sizesa- ($q_{reference}$ = 300) b- ($q_{reference}$ = 400) c- ($q_{reference}$= 500)

TABLE VIII: (Mean) & (standard deviation) values for the queue size with various queue reference size

Queue reference	Mean		Standard deviation	
	PID	***FPID***	***PID***	***FPID***
300	325.46	303.47	46.78	38.55
400	401.08	382.15	64.68	55.54
500	476.46	456.73	81.72	74.02

VII. Changing the propagation delays value (Tp)

The following experiments are presented to examine the capability and efficiency of PID and FPID as AQM techniques when the propagation delay is taking different values. Figures 18 and 19 display the simulation results of PID and FPID controllers for different Tp (10, 15 and 20 ms). The rest of the system parameters were set (N =100) and (C =15Mbps). The FPID controller as AQM was outperformed over PID controller by stabling the queue size for all values of Tp near the reference value, which is (200) packets. (Mean) and (standard deviation) values for the length of queue within a PID and FPID controllers were displayed in Table 9.

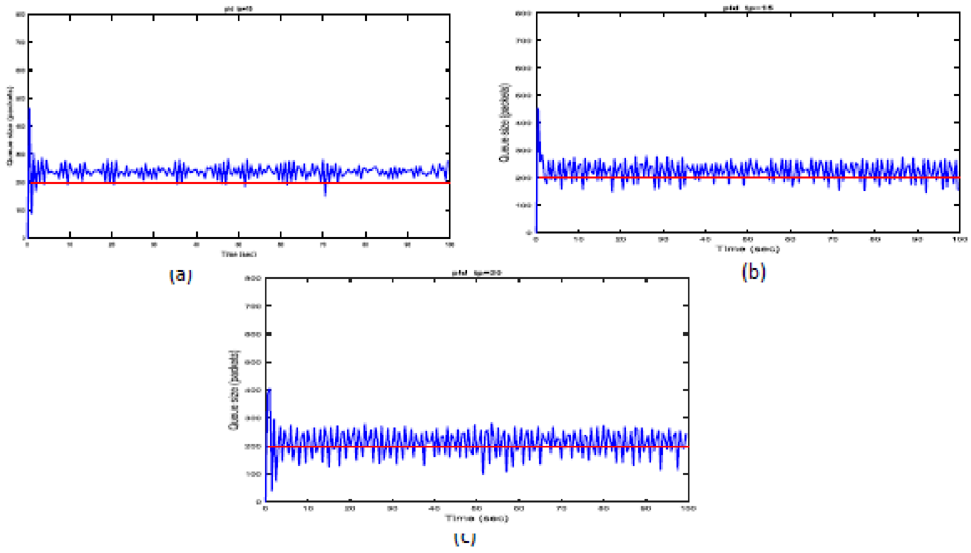
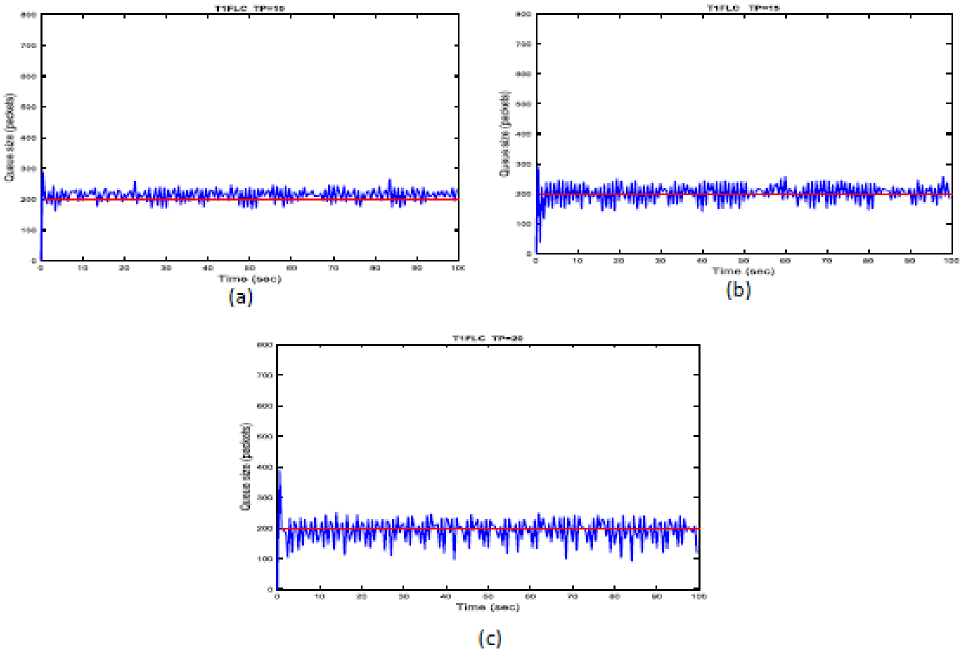


Figure 18: System response using the PID controller with various *Tp* values

(a) ***Tp*** =10 ms (b) ***Tp*** =15 ms (c) ***Tp*** =20 ms

TABLE IX: (Mean) and (standard deviation) values for a length of queue within various *T*p

Tp (ms)	Mean		Standard deviation	
	PID	***FPID***	***PID***	***FPID***
10	236.21	212.63	35.38	26.76
15	224.42	201.12	43.17	37.86
20	210.91	190.42	54.97	44.76



(a)

(b)

(c)

Figure 19: System response using the FPID controller with various *Tp* values (a) *Tp* =10 ms (b) *Tp* =15 ms (c) *Tp* =20 ms

5. COMPARISONS AND DISCUSSION

The outcomes achieved from the preceding section demonstrated that the AQM based FPID controller has obtained more acceptable system performance based on the mean and standard deviation values got from the network simulations in the NS2 package. The FPID controller was succeeded in making the queue size tracking its desired level even with changes in several network parameters compared to the AQM based PID controller. besides, the FPID has robustly maintained the queue size near the target in the period that an additional disturbance was applied to the system. The comparison of proposed AQM in this paper with AQM techniques in the previous works was carried out based on the mean and standard deviation as shown in Table 10, which demonstrates the superiority of the proposed FPID controller in congestion control in the TCP networks. It is worth mentioning that the network parameters were set to N = 60, link capacity (C=15Mbps), propagation time (Tp=5ms), desired queue size (qref=200 packets) in all AQM techniques.

TABLE X: Performance comparison among the proposed PID, FPID, and the previous introduced controllers

Controllers	Mean	Standard Deviation
PI [17]	220.68	45.59
PID	215.43	42.25
Adaptive wavelet [18]	208.26	41.01
Robust PSOPID [8]	198	39.7
FPID	196.92	38.29

6. CONCLUSION

In this paper, we used fuzzy logic techniques to present an AQM scheme that called the Fuzzy PID controller. In TCP/IP networks, which can be introduced for obtaining an efficient control of congestion with high utilization, few packet losses, and delays that are so necessary to multimedia-applications. From the results of the simulation, we conclude as follows:

1) Compared to the classical PID controller, the configured FPID controller can perform the congestion avoidance with a good performance by tracking the reference queue size needed to be in a router buffer space.
2) The performances of the PID and FPID was compared with several designed controllers in previous works based on mean, standard deviation
3) Comparison of the performances of the PID and FPID with several designed controllers in other works showed the superiority of the FPID controller over other these controllers with lower standard deviation and the mean is close to the desired value.
4) The controller's robustness was checked through several experiments by changing many of the network parameters.
5) Good FPID controller capability under dynamic network changes, expressed by TCP session and link capacity.
6) The FPID controller was outperformed over PID controller by stabling the queue size for different values of Tp near the reference value.
7) Good FPID controller tracking performance under different traffic conditions, represented by additive responsive flows, such as FTP, and unresponsive flows, such as UDP.
8) In all previous experiments, the FPID response overshoot never exceeded 400 packets, in contrast to the classical PID controller.

References

[1] I. D. Barrera, S. Bohacek, G. R. Arce, Statistical detection of congestion in routers, IEEE Trans. Signal Process., 58 (2009) 957-968. 10.1109/TSP.2009.2034917

[2] W. Chen, W. Wang, Y. Jiang, G. Li, Q. Chen, PFL: Proactive Fuzzy-logic-based AQM Algorithm for Best-effort Networks, 2010 Int. Conf. Comput. Aspec. Soci. Netw., 2010, 315-318. 10.1109/CASoN.2010.77

[3] S. Misra, B. J. Oommen, S. Yanamandra, M. S. Obaidat, Random early detection for congestion avoidance in wired networks: a discretized pursuit learning-automata-like solution, IEEE Trans. Syst. Man Cybern., 40 (2009) 66-76. 10.1109/TSMCB.2009.2032363

[4] H. Ashtiani, H. M. Pour, M. Nikpour, Active Queue Management in TCP Networks Based on Fuzzy-Pid Controller, Int. J. Appl. Math. Comput., 6 (2012) .

[5] M. Z. Al-Faiz, A. L. A. M. Mahmood, Fuzzy-Genetic controller for congestion avoidance in computer networks, Int. J. Comput. Commun. Control., 11(2011) 20-27.

[6] J. V. Chen, F.-C. Chen, J. M. Tarn, D. C. Yen, Improving network congestion: A RED-based FuzzyPID approach, Comput. Stand. Interfaces, 34 (2012) 426-438. doi.org/10.1016/j.csi.2012.02.002

[7] J. Liu, O. W. Yang, Using fuzzy logic control to provide intelligent traffic management service for high-speed networks, IEEE Trans. Netw. Serv. Manag., 10 (2013) 148-161. 10.1109/TNSM.2013.043013.120264

[8] H. I. Ali, K. S. Khalid, Swarm intelligence based robust active queue management design for congestion control in TCP network, IEEJ Trans. Electr. Electron. Eng., 11 (2016) 308-324. 10.1002/tee.22220

[9] V. Misra, W.-B. Gong, D. Towsley, Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED, in Proceedings of the conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, (2000) 151-160. doi.org/10.1145/347059.347421

[10] C. V. Hollot, V. Misra, D. Towsley, W.B. Gong, A control theoretic analysis of RED, in Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society (Cat. No. 01CH37213), 3 (2001) 1510-1519. 10.1109/INFCOM.2001.916647

[11] C. Chrysostomou, "Fuzzy logic based AQM congestion control in TCP/IP networks," Ph.D. Thesis, Computer Science Dep., Univ. of Cyprus, 2006.

[12] E. Plasser , T. Ziegler, A RED function design targeting link utilization and stable queue size behavior, Comput. Net., 44 (2004) 383-410. doi.org/10.1016/j.comnet.2003.10.006

[13] R. Leonid, "Fuzzy controllers," Newnes*,* vol. 1, pp. 1-9, 1997.

[14] M. Malekabadi, M. Haghparast, F. Nasiri, Air Condition's PID Controller Fine-Tuning Using Artificial Neural Networks and Genetic Algorithms, Computers, 7(2018) 32. doi.org/10.3390/computers7020032

[15] B. Sangmitra, K. Narayana, and K. Kumar, Congestion Control in High Speed Networks Using Fuzzy Logic Control, Int. J. Soft Comput., 5 (2015) 45-55. 10.26708/IJMSC.2015.1.5.06

[16] H. M. Kadhim and A. A. Oglah, Interval type-2 and type-1 Fuzzy Logic Controllers for congestion avoidance in internet routers, IOP Conf. Ser: Mater. Sci. Eng., 881 , 2020, 012135. 10.1088/1757-899X/881/1/012135

[17] C. V. Hollot, V. Misra, D. Towsley, W.B. Gong, On designing improved controllers for AQM routers supporting TCP flows, in Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society (Cat. No. 01CH37213), 3 (2001) 1726-1734. 10.1109/INFCOM.2001.916670

[18] H. J. Hassan, Development of Adaptive Wavelet Network PID (AWNPID) controller for improving AQM in the TCP network , Ph.D. Thesis, University of Technology, Iraq, 2013.