IJCCCE Vol.13, No.1, 2013

Design and Implementation of Adaptive Wavelet Network PID Controller for AQM in the TCP Network

Dr. Farooq Al- doraiee[1], **Dr. Salih Al-Qaraawi**[2], and **M.Sc. Hassan J. Hassan**[2]

[1] Department of Computer & Automation, College of Eng., University of Damascus, Syria
[2] Control & Systems Eng., University of Technology, Baghdad, Iraq

dr.salih.al_qaraawi@uotechnology.edu.iq, hassan61@yahoo.com

Received: 14/3/2011

Accepted: 14/9/2012

Abstract – Internet represents a shared resource wherein users contend for the finite network bandwidth. Contention among independent user demands can result in congestion, which, in turn, leads to long queuing delays, packet losses or both. Congestion control regulates the rate at which traffic sources inject packets into a network to ensure high bandwidth utilization while avoiding network congestion. In the current Internet, there are two mechanisms which deal with congestion; the end-to-end mechanism which is achieved by the Transmission Control Protocol (TCP) and the intermediate nodes algorithms such as Active Queue Management (AQM) in routers.

In this paper, a combined model of TCP and AQM (TCP/AQM) is formulated and first simulated without a controller. The results show that it is unable to track the desired queue size. So, to get better tracking performance, an adaptive PID controller based on wavelet network (AWNPID) is used as AQM in the router queue. The non-adaptive PID controller is also demonstrated, and its weakness to the network dynamic changes is compared to the robustness of adaptive controller (AWNPID). The analytical results for linearized TCP/AQM model are presented in MATLAB version 7.0.

Key Words

Active Queue Management, Wavelet Network, PID Controller, Adaptive Learning.

1. Introduction

Congestion control has become a critical problem in the development of the Internet. This is because congestion in the Internet can cause high packet loss rates, increased delay, and can even break the whole system by causing congestion collapse. TCP congestion control has the basis of the operation of the Internet. It adopts the end-to-end window-based flow control to avoid congestion [1]. Congestion in a computer network is a state in which performance degrades due to the saturation of network resources such as communication links, processor cycles, and memory buffers. Network congestion has been well recognized as a resource-sharing problem. When too many packets are contending for the same link, the queue overflows and packets have to be dropped. When such drops become common events, the network is said to be congested. Most networks provide a congestion-control mechanism to deal with just such a situation [2]. TCP congestion control mechanism, while necessary and powerful, is not sufficient to provide good service in all circumstances, especially with the rapid growth in size and the strong requirements to quality of service support, because there is a limit to how much control can be accomplished at end system. It is needed to implement some measures in the intermediate nodes to complement the end system congestion avoidance mechanisms. AQM, as one class of packet dropping/marking mechanism in the router queue, has been proposed to support the end-to-end congestion control in the Internet [3]. The simplest and earliest policy adopted by routers to manage their queues, is a Drop Tail strategy which is implemented by means of a First in First out (FIFO) queue management. This policy is easy to implement and works effectively in lightly loaded traffic conditions, but there are some drawbacks with this conventional scheme when the traffic load is heavy such as low network throughput and utilization, high packet loss rate and deteriorated network conditions. Regarding these problems, the Internet Engineering Task Force (IETF) recommends that Internet routers should implement some queue management mechanism which has the capability of controlling queue lengths and informing end hosts of incipient congestion. This mechanism is called (AQM) [4]. Because, RED (Random Early Detection) [4] algorithm can satisfy the objective of AQM, it is widely used in the network queue management to optimize router performance. But it still has some flaws, for example, RED is too sensitive to parameters configuration, readily leads to TCP global synchronization and induces queue oscillation under some special network environments, and then reduces link utilization and deteriorates delay jitter. Hence, RED absorbed so much attention in the network research. Except for theoretic analysis of RED stability [5] and validation to its effectiveness through experiments [6], most of works aimed to modify RED algorithm to overcome its shortcomings. Therefore, many RED variants were proposed, such as Stabilized-RED [7], Self-configuration RED [8] and Adaptive RED [9].

Some researchers [10], [11] analyze the performance of the networks based on the control theory, such as PI controller, PID controller. These control theories have lead into the research of TCP/AQM congestion control. RED, PI, and PID algorithms have been used for AQM, but, these algorithm show weakness in the detection and control of congestion under dynamically changing network situations. The AQM controller for the simplified and inaccurate linear TCP model is not optimal, because the real TCP network is rapidly changed, that is, the network parameters, the number of TCP sessions, and the capacity of the link are hardly kept at constant values for a long time. Therefore it is necessary to design an adaptive controller for nonlinear control system, so do the researchers [12] and [13].

In this paper, and in order to provide queue space in the intermediate nodes (routers) to absorb bursts of packet arrivals, our proposed algorithm will be based on fixed learning rate (FLR)with adaptive PID according to the tuning of wavelet network parameters.

2. Proposed Control Strategy

2.1. Wavelet Network Structure and Algorithm

Before beginning tracking using wavelet network based PID controller, the unknown nonlinear dynamic model of TCP/AQM must be identified according to a certain model. In this particular identification process, the model consists of a neural network topology with the wavelet transform embedded in the hidden units. In cascades with the network is a local infinite impulse response (IIR) block structure as shown in figure (1). The algorithm of wavelet network is similar to those in [14] where any desired signal$y(t)$ can be modeled by generalizing a linear combination of mother function (daughter function) $h_{a,b}(t)$, where $h_{a,b}(t)$ are generated by dilation, a, and translation, b, from a mother wavelet.

The approximated signal of network $\hat{y}(t)$ can be modeled by:

$$\hat{y}(t)=\sum_{i=0}^{M} c_i z(t-i)u(t)+\sum_{j=1}^{N} d_j\,\hat{y}(t-j)v(t) \qquad (1)$$

Where:

$$z(t)=\sum_{k=1}^{K} w_k h_{ak,bk}(t) \qquad (2)$$

k is the number of wavelets, w_k is the k^{th} weight coefficient. M and c_j are the number of feed forward delays and coefficient of the IIR filter, respectively, N and $d_j$are the number of feedback and recursive filter coefficients, respectively. The signal $u(t)$ and $v(t)$are the input and co-input to the system at time$t$, respectively. Input $v(t)$ is usually kept small for the feedback stability purpose.

The wavelet network parametersa_k, b_k, w_k, c_i, and $d_j$can be optimized in the LMS (Least mean square) sense by minimizing a cost function or the energy function, E, over all time$T$. Thus,

$$e(t)=y(t)-\hat{y}(t) \qquad (3)$$

is a time varying error function at time t, where $y(t)$ is the desired (target) response. The energy function is defined by:

$$E=\frac{1}{2}\sum_{t=1}^{T} e^2(t) \qquad (4)$$

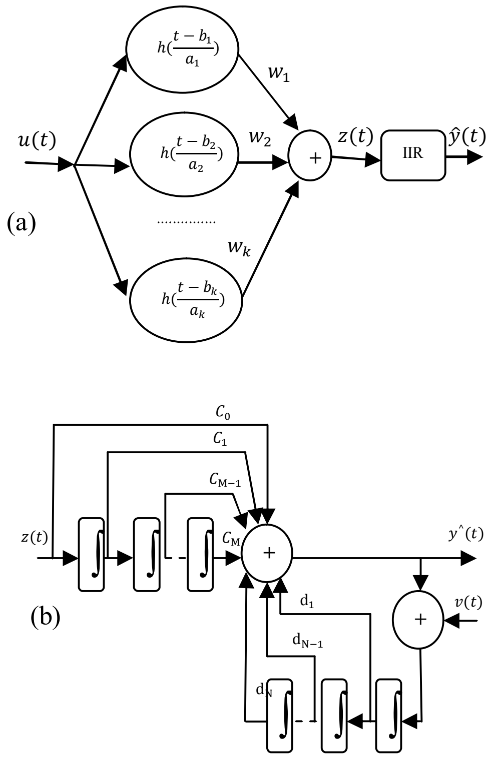


Figure (1): IIR adaptive wavelet network structure (a) Wavelet network architecture. (b) IIR model.

2.2. TCP/AQM Modeling

The non linear dynamic model for TCP/AQM was developed by [15] and linearized in [16]. Figure (2) depicts the linearized model of AQM control system:

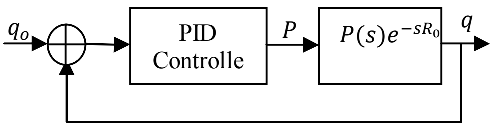


Figure (2): Block diagram of AQM control system

$P(s)e^{-sR_0}$ is the plant of AQM control system. The transfer function $p(s)$ is the product of $P_{tcp}(s)$ and $P_{queue}(s)$, where:

$$P_{tcp}(s) = \frac{\frac{R_0\, C^2}{2N^2}}{S+\frac{2N}{R_0^2 C}} \quad \text{and,}$$

$$P_{queue}(s) = \frac{\frac{N}{R_0}}{S+\frac{1}{R_0}} \qquad \textbf{(5)}$$

Where:

C: Link capacity (packet/s).

q_0: Queue reference value.

N: Load factor (number of active TCP sessions).

q: Instantaneous queue.

R: Round-Trip Time (RTT), where:

$$R = (\tfrac{q}{c} + T_p) \qquad \textbf{(6)}$$

T_p is the fixed propagation delay.

P: Dropping/marking probability.

It is a recognized fact that there are many different methods for approximating a time delay inherent in a system by a rational function by a Maclaurin series [17]:

$$e^{-R_0 s} \cong \frac{1-\frac{R_0}{2}s}{1+\frac{R_0}{2}s} \qquad (7)$$

In this paper, equation (7) will be used as an approximation of TCP round-trip delay model.

2.3. System Model and Adaptive Wavelet Network PID Controller

Consider a general SISO dynamical system to be represented by the state equations:

$$x(k+1) = f(x(k), u(k), (k)) \qquad \textbf{(8)}$$

$$y(k) = g(x(k), k) \qquad \textbf{(9)}$$

Where $x(k) \in R^n$ and $u(k), y(k) \in R$. Further, let the unknown functions$f, g \in C^1$. The only accessible data are the input u and output y. Itwas stated in [14] that if the linearized system around the equilibrium state is observable, an input-output representation exists which has the form:

$$y(k+1) = \phi\big(y(k)\big) + \Gamma\big(y(k)\big).u(k) \qquad \textbf{(10)}$$

Where $y(k)$ and $u(k)$ denote the input and the output at the K^{th} instant of time, respectively.

If $\phi(.)$ and $\Gamma(.)$ are unknown, the idea is to use the neural network adaptive wavelets model to approximate the system dynamics i.e.

$$\hat{y}(k+1) = \hat{\phi}\big(y(k), \Theta_\phi\big) + \hat{\Gamma}(y(k), \Theta_\Gamma)u(k) \qquad \textbf{(11)}$$

Comparing the model of Eq. (11) with the one of Eq. (1), we can conclude that:

$$\hat{\phi}\big(y(k), \Theta_\phi\big) = \sum_{j=1}^{N} d_j \hat{y}(k-j)v(k) \qquad \textbf{(12)}$$

$$\hat{\Gamma}(y(k), \Theta_\Gamma) = \sum_{i=0}^{M} C_i z(k-i) \qquad \textbf{(13)}$$

Where:

$$z(k) = \sum_{l=1}^{K} w_l h\left(\frac{k-b_l}{a_l}\right) \qquad \textbf{(14)}$$

After the nonlinearities $\phi(.)$ and $\Gamma(.)$ are approximated by the two distinct neural network functions $\hat{\phi}(.)$ and $\hat{\Gamma}(.)$ with adjustable parameters (including weight w_k, dilation a_k, translation b_k, IIR feed-forward coefficients c_k, and IIR feedback coefficients d_k), represented by Θ_ϕ and Θ_Γ respectively, the PID control $u(k)$ for tracking a desired output $r(k+1)$ can be obtained from:

$$u(k) = u(k-1) + P[\varepsilon(k) - \varepsilon(k-1)]$$
$$+I\varepsilon(k) + D[\varepsilon(k) - 2\varepsilon(k-1) + \varepsilon(k-2)] \qquad \textbf{(15)}$$

Where$P$,I, and D are proportional, integral, and differential gain, $u(k)$ is a plant input at KT, where T is a sampling interval, and:

$$\varepsilon(k) = r(k) - y(k) \qquad \textbf{(16)}$$

P, I, D parameters are considered as part of the function of E and can be optimized and update according to the cost function E of Eq. (4),

$$P(k) = P(k-1) + \mu_P e(k)\Gamma(k)(\varepsilon(k) - \varepsilon(k-1)) \qquad \textbf{(17)}$$

$$I(k) = I(k-1) + \mu_I e(k)\Gamma(k)\varepsilon(k) \qquad \textbf{(18)}$$

$$D(k) = D(k-1) + \mu_D e(k)\Gamma(k)(\varepsilon(k) - 2\varepsilon(k-1) + \varepsilon(k-2) \qquad \textbf{(19)}$$

Where:

$e(k)$ comes from Eq.(3)

$\hat{\Gamma}$ comes from Eq. (13), and μ is the fixed learning rate of each adaptive PID parameter. Figure (3) depicts the model considered in this paper.

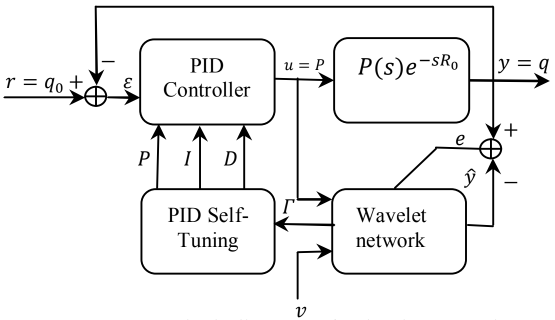


Figure (3): Block diagram of Adaptive Wavelet Network PID Controller for AQM

3. Network Topology

Figure (4) shows the network case study taken where the simulation is conducted for a single link (Bottleneck link) that has a bandwidth capacity **C=3750** packets/sec (corresponds to a **15** Mbps with packet size **500** bytes), and the same bandwidth capacity is used at other links, the Round Trip Time (**R0**) is**0.253** second which is calculated in Eq. (6), where the desired queue size is**200** packets and the propagation delay is **0.2** second. The number of TCP sessions (**N**) is **100**. The maximum queue length in the AQM router A is **800** packets. The AQM mechanism (PID controller) is configured at router A, and drop Tail is used at other gateways. The simulation is carried out by using MATLAB version 7.0.

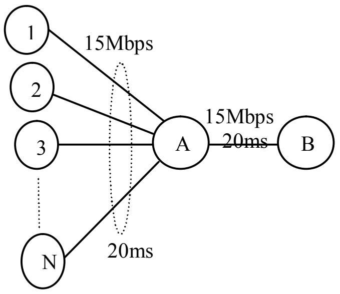


Figure (4): Simulation network topology

3.1. Simulation Results

Consider the TCP/AQM model with network parameters as set in the previous section and the reference input (queue size) which has rectangular form changes every 50 seconds as shown in equation (22). First, the simulation is done for the system **without controller** as shown in figure (5).

$$q_{ref} = \begin{cases} 300 \text{ packet} & 0< t < 50; \\ 200 \text{ packet} & 50< t < 100; \\ 500 \text{ packet} & 100< t < 150; \\ 200 \text{ packet} & 150< t < 200; \end{cases} \qquad \textbf{(20)}$$

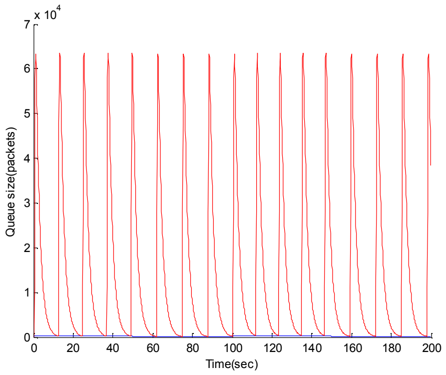


Figure (5): System Response without Controller.

From figure (5) it is shown that the system without controller is unable to track the queue length around the queue length to the desired level, where the system goes into a sustained oscillation with high congestion exceeding the maximum buffer size.

In order to eliminate this sustained oscillation and get better tracking performance, an adaptive wavelet network PID controller (AWNPID) and non-adaptive PID controller are applied at the same time. As shown in figure (6) the system response with PID controller (Adaptive and non-adaptive) has the ability to get the desired performance, where the desired queue length is achieved within the buffer size of the router.

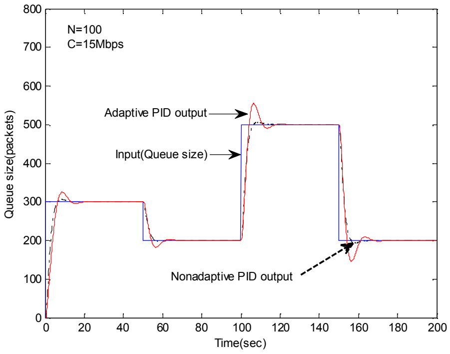


Figure (6): System Response for adaptive and non-adaptive PID Controller (N=100, C=15Mbps)

The PID controller parameters for both adaptive and non-adaptive are selected by trial and error method as follows:

$$K_{P=}3\times10^{-6}, K_I = 1\times10^{-6} \mathit{and}$$
$$K_D = 3\times10^{-7}.$$

The activation function for wavelet network was POLYWOG1 which can be represented in the following equation.

$$h(\tau) = \tau\, e^{-0.5\tau^2} \qquad \dots \textbf{(21)}$$

3.2 Robustness of Adaptive and Non-Adaptive PID Controller

To check the robustness of adaptive and non-adaptive controllers and their ability to manage the congestion avoidance in different circumstances, the values of system parameters [N=100, C=15Mbps] are changed to study the effect of changing them on system response.

a) Link Capacity (C)

For the first parameter of the case study on the link capacity (C) if it is perturbed by decreasing it from 15 Mbps down to 10 Mbps from the value that it took in the first case study, the system response is shown in figure (7) below:

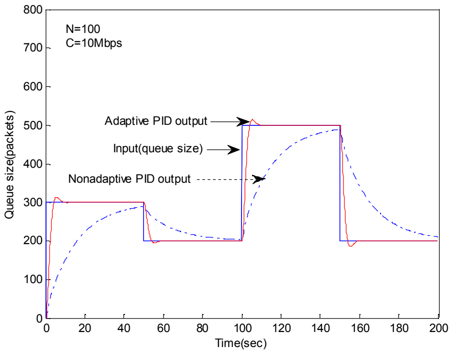


Figure (7): System Response for adaptive and non-adaptive PID Controller (N=100, C=10Mbps)

In Figure (7) it is seen when decreasing link capacity (C) to 10 Mbps, the non-adaptive controller parameters must be tuned to keep the desired queue length within the buffer size of the router. From the same figure, we can see that the adaptive wavelet network controller has the ability to self- tune parameters and then keep the desired queue length within the buffer size of router.

b) Number of TCP Flows (N)

The number of the TCP flows (N) is increased to 160 nodes (C=10 and N=160), it gets the system response as shown in Figure (8).

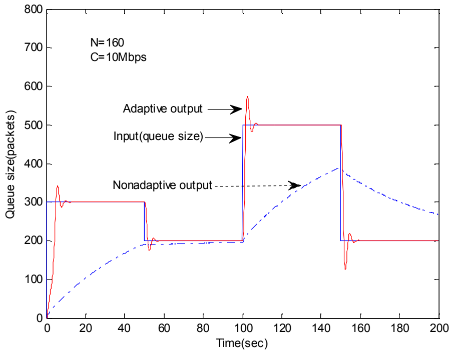
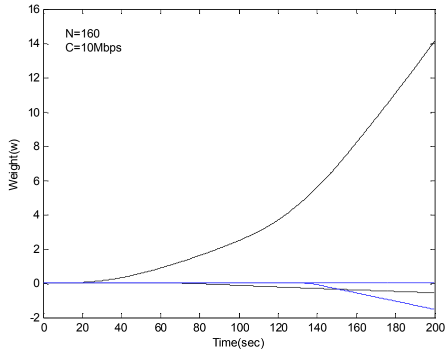


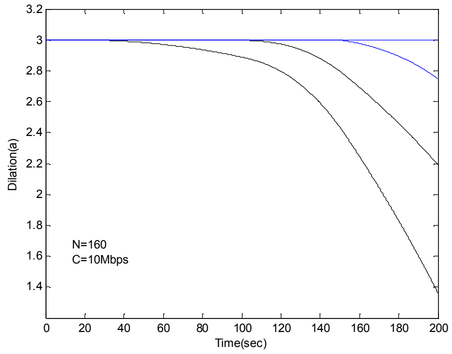
Figure (8): System Response for adaptive and non-adaptive PID Controller (N=160, C=10Mbps)

From figure (8), we can see the potential of adaptive wavelet PID controller to absorb the dynamic changes that may occur in the network by self-tuning the system parameters (weight, dilation , translation, feed forward IIR, feedback IIR, and the PID gain parameters), while the non-adaptive controller failed to track changes in the network under some conditions.

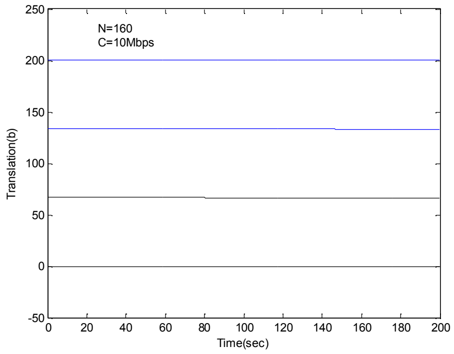
For the case depicted in figure (8), the wavelet network parameter updates can be shown in figure (9-a, b, c, d, and e).



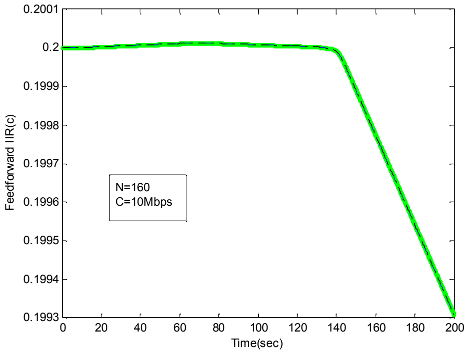
(a)



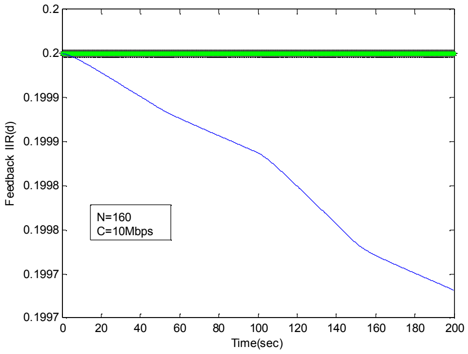
(b)



(c)



(d)



(e)

Figure (9): Wavelet network parameter updates .(a) Weigh. (b) Dilation. (c) Translation. (d) ff IIR. (e) fb IIR

Figure (10) provides P, I, and D parameter updates.

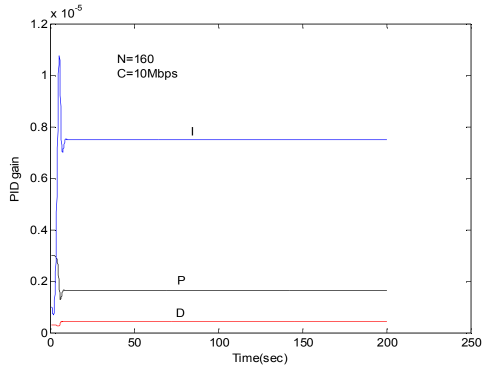


Figure (10):P, I, D parameter updates

The update of nonlinearity parameter Gama (Γ)is shown in figure (11).

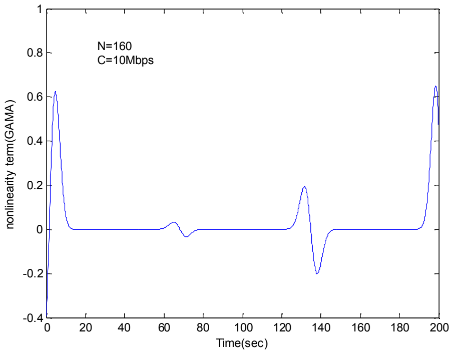
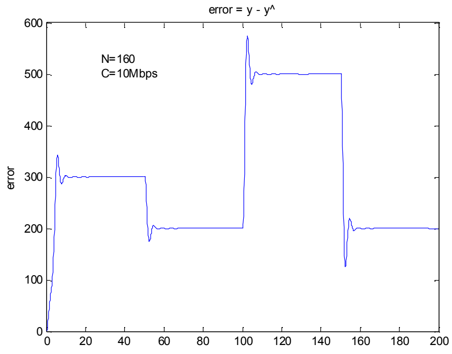
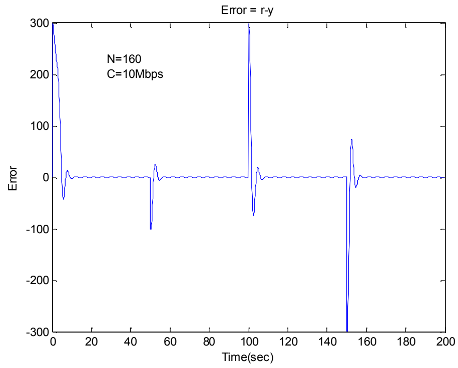


Figure (11): Gama (Γ) parameter update

Finally, the error parameter updates e, and εwhich were represented in Eq. (3) and Eq. (18) are shown in figure (12- a and b) respectively.



(a)



(b)

Figure (12): Error parameter updates. (a)e, (b)ε

4. Conclusion

In this paper, the Adaptive Wavelet Network PID controller (AWNPID) was proposed for AQM in the TCP network. AWNPID controller using adaptive parameters (weight, dilation, translation, feed forward IIR, and feedback IIR) is operated as a direct adaptive controller, where the output is dropping probability of packets at the router. The proposed controller maintains the actual queue size close to a reference queue value. Beside the adaptive controller, non-adaptive controller response has been demonstrated and compared to the adaptive controller response. The result of the simulation proved clearly the ability of AWNPID controller to track dynamic changes in the network in contrast to the non-adaptive controller which reflects its weakness in controlling the congestion under some different traffic conditions.

References

[1] Y. Hong, O.W.W. Yang, C. Huang, "Self-Tuning PI TCP Flow Controller for AQM Routers with Interval Gain and Phase Margin Assignment", IEEE Communications Society, Globcom, vol.3, no. 4, pp. 1324-1328, 2004.

[2] L. Peterson and B. Davie, "Computer Network a Systems Approach" 3rd edition, Elsevier Science (USA), 2003.

[3] S. Floyd and V. Jacobson, "Random Early Detection Gateway for Congestion Avoidance", IEEE/ACM Transactions on Networking Vol.1, No.4, August 1993.

[4] A. T. Al-Hammouri "Internet Congestion Control Complete Stability Region for PI AQM and Bandwidth Allocation in Networked Control". January, PhD Thesis, Case Western Reserve University, January 2008.

[5] V. Firoiu and M. Borden. "A study of Active Queue Management for Congestion Control". In Proceeding of IEEE INFACOM 2000.

[6] M. Christiansen, K. Jeffay, D. Ott, F.D. Smith, "Tuning RED for Web traffic". In Proceeding of the ACM SIGCOMM 2000 Conference, 2000, 139-150.

[7] Teunis J. Ott, T. V. Lakshman and Larry H. Wong. SRED: Stabilized RED. In Proceeding of INFOCOM 1999.

[8] W. Feng, D. Kandlur, D. Saha and K. Shin."A self-Configuring RED gateway". In Proceeding of INFOCOM 1999, 1320-1328.

[9] Sally Floyd, Ramakrishna Gumadi, and Scott Shenker."Adaptive RED: An Algorithm for Increasing the Robustness of RED's Active Queue Management".
http://www.cs.berkeley.edu/

[10]C. V. Hollot, V. Misra, D. Towsley, and W. B. Gong, "On Designing Improved Controllers for AQM Routers Supporting TCP Flows". In Proceedings of IEEE INFOCOM, pages 1726–1734, USA, 2001.

[11] F. Yanfei, R. Fengyuan, L. Chaung, "Design a PID Controller for Active Queue Management", proceeding of the Eighth International Symposium on Computers and Communication (ISCC), IEEE, 2003.

[12] p.p. Xiao, Y.T. Tiao, "Design of a Robust Active Queue Management Algorithm based on Adaptive Neuron PID", Proceeding of the Fifth International Conference on Machine Learning and Cybernetics, IEEE, 2006.

[13] J. M. Kim, J. B. Park, Y. H. Choi, "Wavelet Neural Network Controller for AQM in a TCP Network: Adaptive Learning Rates Approach", International Journal of Control, Automation, and systems, vol. 6, no. 4, pp. 526-533, August 2008.

[14] G. Lekutai, "Adaptive Self-Tuning Neuro Wavelet Network Controllers", Ph.D. Thesis. Virginia Polytechnic Institute and State University. College of Eng., Electrical Eng. The Bradley Dept., 1997.

[15] V. Misra, W.-B. Gong, and D. Towsley, "Fluid-Based Analysis of a Network of AQM Routers Supporting TCP Flows with an Application to RED", in Proc. ACM SIGCOMM, 2000. pp. 151-160.

[16] C. V. Hollot, V. Misra, D. Towsley, and W. B. Gong, "A Control Theoretic Analysis of RED", In Proceedings of IEEE INFOCOM, USA, 2001.

[17] K. Rahnami, P. Arabshahi, A. Gray, "Neural Network based Model Reference Controller for Active Queue Management of TCP Flows", IEEE, 2005.