Design of PID Controllers for TCP/AQM Wireless Networks

Teresa Alvarez

***Abstract*—Internet users rely on the good capabilities of TCP/IP networks for dealing with congestion. The network delay and the number of users change constantly, which might lead to transmission problems. Usually, they are handled following approaches such as drop tail or random early detection (RED) algorithms. During the last few years, automatic control techniques are providing practical solutions. Moreover, if there are wireless links, congestion control is more difficult to detect. This paper presents a methodology to design a PID controller with a linear gain scheduling that allows a network with wireless links to deal with control congestion under a variety of configurations. The proposed technique is demonstrated through non-linear simulations and compared with the classical PID**

***Index Terms*— PID, Westwood TCP/AQM, congestion control, robustness, gain scheduling.**

I. Introduction

THE importance of Internet cannot be denied. However, there are situations in which problems arise such as long delays in delivery, lost and dropped packets, oscillations or synchronization problems ([1], [2]). Congestion is responsible for many of these problems. It is harder to detect if there are wireless links in the network. Thus, techniques to reduce congestion are of great interest. There are two basic approaches ([3], [4]): c*ongestion control,* which is used after the network is overloaded, and c*ongestion avoidance,* which takes action before the problem appears.

Feedback control can help to solve congestion control. The end-to-end transmission control protocol (TCP), and the active queue management (AQM) scheme define the two parts implemented at the routers' transport layer where congestion control is carried out. The most common AQM objectives ([5]) are: *efficient queue utilization* (to minimize the occurrences of queue overflow and underflow, thus reducing packet loss and maximizing link utilization), *queuing delay* (to minimize the time required for a data packet to be serviced by the routing queue) and *robustness* (to maintain closed-loop performance in spite of changing conditions).

Manuscript received April 2012. This work was supported in part by AECI under project AP/039213/11 and by MiCInn project DPI2010-21589-C05-05.

T. Alvarez is with the University of Valladolid (Tlef: +34 983423162; e-mail: tere@autom.uva.es).

In general, AQM schemes enhance the performance of TCP, although they have some disadvantages as they do not work perfectly in every traffic situation. Numerous algorithms have been proposed (a good survey can be found in [4]). The most widely used AQM mathematical models were published in [5]. Since then, several control approaches have been published: fuzzy, predictive control, robust control, etc. The automatic control technique that has received more attention is the PI controller, followed by the PID ([6]).

AQM techniques should be robust and give good results when the network is not operating in nominal situation, when the delays are changing, or the number of users or the intensity of the packets' flow vary, i.e. the network should deliver the users' data on time and without errors regardless of the changes in the working parameters.

So, the AQM congestion control algorithm should be robust, stable and perform in a changing environment. If PID is chosen as the AQM algorithm, there are many useful references in the literature: [7] and [8] cover quite a lot of the situations that can arise. Reference [9] gives a graphical characterization of the stable region of a PID controller applied to the AQM router, but they do not consider changing traffic conditions. These works are of special interest because they explicitly consider delays in the design of the PIDs and this is the situation that arises when working with networks: a system with delay.

Motivated by these issues, this paper presents how to derive PIDs with guaranteed stability and robustness that performs well under a network with wireless links and changing traffic conditions. The work presented in this paper applies the generic method by [7] for finding stable PID controllers for time delay systems in the dynamic TCP Westwood/AQM router model. A linear gain scheduling is included in the design (reducing the gain variation) to enhance the robustness and ensure adequate behaviour in extreme working scenarios. The metrics applied to study the controller performance are: the router queue size (real value and standard deviation), the link utilization and the probability of packet losses.

Non-linear simulations will show the fine performance of the technique by applying it to a problem of two routers connected in a Dumbbell topology, which represents a single bottleneck scenario. The length of their queues is controlled with a PID whose properties are guaranteed following the results presented in the paper.

This paper is organized as follows. Section 2 briefly describes TCP Westwood. A fluid flow model is presented in Section 3 and, in Section 4, the methodology is described. Simulation results are shown in section 5. Finally, some

ISBN: 978-988-19252-1-3
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

conclusions and discussion on future work are presented.

II. TCP WESTWOOD

TCP Westwood (TCPW) ([10]-[14]) is a modification of TCP NewReno at the source side, intended for networks where losses are not only due to congestion such as the wireless networks studied in this paper. The protocol from the receiver point of view is the same in both TCPs, but there are some changes in the way the source calculates the available bandwidth. These modifications affect the dynamics of the system, which justifies the necessity of specific controllers for TCPW.

In TCPW, the congestion window increases during the slow start phase, but is the same as in NewReno during the congestion avoidance phase. A packet loss is indicated by a coarse timeout ending or the reception of three Duplicate ACKnowledgeS (DUPACKS). The basic idea of TCPW is to use the flow of returning ACKs to estimate the available bandwidth (BWE); then this value is used for setting the size of the slow start threshold size (*sstresh*) and the congestion window (*cwnd*) ([10]):

- IF (3 DUPACKS are received) THEN
  sstresh = (BWE*RTTmin)/MSS
  IF (sstresh<2) THEN sstresh=2
  cwnd = sstresh;
- IF (coarse timeout expires) THEN
  sstresh = (BWE*RTTmin)/MSS
  IF (sstresh<2) THEN sstresh=2
  cwnd = 1;
- IF ACKs are successfully received THEN
  cwnd is increased following the RFC2581 document for congestion control

This TCPW captures the basic NewReno behavior, but Westwood improves the stability of the TCP multiplicative decrease.

III. FLUID FLOW MODEL FOR TCP WESTWOOD

This section presents a fluid flow model derived in [10]. As in the original approach ([5]), there is a single bottleneck. Now all the TCP connections are assumed to follow the Westwood formulation.

A. Nonlinear Model

The nonlinear model used for the simulations is described by the following two coupled nonlinear differential equations:

$$\dot{W} = \frac{1}{\frac{q(t)}{C}+T_p} - \frac{W(t)W(t-R_0)}{\frac{q(t-R_0)}{C}+T_p}\,p(t-R_0) + T_p\left(\frac{W(t-R_0)}{\frac{q(t-R_0)}{C}+T_p}\right)^2 p(t-R_0) \tag{1}$$

$$\dot{q} = \begin{cases} -C + \dfrac{N}{\frac{q(t)}{C}+T_p}W(t) & \text{if} \quad q(t)>0 \\ \max\left\{0, \dfrac{N}{\frac{q(t)}{C}+T_p}W(t) - C\right\} & \text{if} \quad q(t)=0 \end{cases} \tag{2}$$

Where

W: average TCP window size (packets),

$\dot{q}$: average queue length (packets),

R: round-trip time = $q/C+Tp$ (secs),

C: link capacity (packets/sec),

T_p: propagation delay (secs),

$N_{TCP}=N$: load factor (number of Westwood TCP sessions) and

p: probability of packet mark.

Equation 1 describes the TCP window control dynamic and Equation 2 models the bottleneck queue length as an accumulated difference between the packet arrival rate and the link capacity. The queue length and window size are positive, bounded quantities, i.e., $q \in [0,\bar{q}]$ and $W \in [0,\bar{W}]$, where $\bar{q}$ and $\bar{W}$ denote buffer capacity and maximum window size, respectively.

B. Linearized Model

Although an AQM router is a non-linear system, a linearized model is used to analyze certain required properties and design controllers. To linearize (1) and (2), it is assumed that the number of active TCP sessions and the link capacity are constant, i.e., $N_{TCP}(t)=N_{TCP}=N$ and $C(t)=C$.

As described in [5] and [10], the dependence of the time delay argument $t-R$ on the queue length q is ignored and it is assumed to be fixed to $t-R_0$. This is a big assumption and there could be situations where it would not be acceptable. In [15], it was deduced that this is a good approximation when the round-trip time is dominated by the propagation delay. When the model is linearized, the same supposition is made. It cannot be denied that the calculations are significantly simplified, but further research in the matter would be advisable.

Then, local linearization of (1) and (2) around the operating point (q_0, p_0, W_0) results in the following equations:

$$\left.\begin{aligned} \partial\dot{W}(t) &= -\frac{N}{R_0 T_{q_0} C}\left(\partial W(t) + \partial W(t-R_0)\left(1-2\frac{T_p}{R_0}\right)\right) \\ &\quad -\frac{1}{R_0^2 C}\left(\partial q(t) - \partial q(t-R_0)\right)\frac{R_0-2T_p}{T_{q_0}} \\ &\quad -\frac{R_0^2 C}{N^2}\left(1-\frac{T_p}{R_0}\right)\partial p(t-R_0) \\ \partial\dot{q}(t) &= \frac{N}{R_0}\partial W(t) - \frac{1}{R_0}\partial q(t) \end{aligned}\right\} \tag{3}$$

where $T_{q_0} = R_0 - T_p$ and $\partial\dot{W}(t) = W - W_0$ and $\partial p = p - p_0$, represent the perturbed variables. Taking

ISBN: 978-988-19252-1-3
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

(W,q) as the state and *p* as input, the operating point (q_0, p_0, W_0) is defined by $\dot{W}=0$ and $\dot{q}=0$, that is,

$$\dot{W}=0 \Rightarrow W_0 = -\frac{R_0\left(R_0-T_p\right)^2}{N^3\left(2R_0-T_p\right)}p_0,$$

$$\dot{q}=0 \Rightarrow q_0 = N\cdot W_0 . \quad (4)$$

Equation (3) can be further simplified by separating the low frequency ('nominal') behavior P(s) of the window dynamic from the high frequency behavior Δ(s), which is considered parasitic. Taking (4) as the starting point and following steps similar to those in [5], we can obtain the feedback control system (Fig. 1) for TCPW/AQM ([16]).

The action implemented by an AQM control law is to mark packets with a discard probability *p(t)* as a function of the measured queue length *q(t)*. Thus, the larger the queue, the greater the discard probability becomes.

$$P(s)=\frac{\frac{-C^2}{N}\left(1-\frac{T_p}{R_0}\right)}{s^2+\frac{C\cdot R_0+2N}{C\cdot R_0^2}s+\left(\frac{N}{C\cdot R_0^3}\frac{2R_0-T_p}{R_0-T_p}\right)} \quad (5)$$

$$\Delta(s)=\frac{1}{\beta}\frac{1}{R_0C}\left(1-e^{-sR_0}\right) \quad (6)$$

$$\beta=\frac{C^2R_0}{N^2}\left(1-\frac{T_p}{R_0}\right) \quad (7)$$

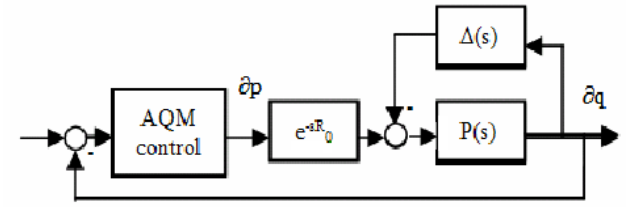

Fig. 1. Block diagram of AQM as a feedback control system

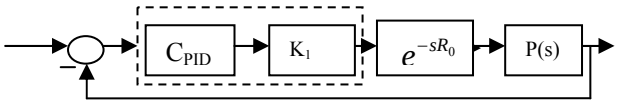

Fig. 2. Block diagram of the AQM feedback control system

IV. 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 N_{TCP} 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. Fig. 2 depicts the block diagram corresponding to the feedback control approach followed in the paper: P(s) is the transfer function obtained in Section 3. The delay e^{-sR_0} is explicitly considered in the design. There are two elements in the controller C(s):

1. $C_{PID}(s)$ 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 N_{TCP} and R_0: i.e., the biggest delay and the smallest number of users.
2. Choose the best network scenario in terms of N_{TCP} and R_0: i.e., the smallest delay and the biggest number of users.
3. Design a controller $C_{PID}(s)$ 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.

We have chosen the PID controller (Equation 8, [6]) as the AQM congestion control algorithm, as it is the most common form of feedback control technique and relevant results can be found in the literature regarding the stability properties and the tuning. The structure of the controller is:

$$\begin{aligned} u(t) &= K_p\left(e(t)+\frac{1}{T_i}\int_0^t e(\tau)d\tau+T_d\frac{de(t)}{dt}\right) \\ &= K_p e(t)+K_i\int_0^t e(\tau)d\tau+K_d\frac{de(t)}{dt} \end{aligned} \quad (8)$$

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 [7]: 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.

The simple gain scheduling approach proposed in the paper takes into consideration the big gain variation that the AQM dynamic model has:

$$\frac{-C^2}{N}\left(1-\frac{T_p}{R_0}\right) \quad (9)$$

Moreover, if a fair comparison among all the network configurations were done, the open-loop gain of the systems should be the same, so the independent term of the denominator of the transfer function should be included:

$$\frac{N}{C\cdot R_0^3}\frac{2R_0-T_p}{R_0-T_p} \quad (10)$$

This term greatly affects the size of the stable region and the magnitude of the PID's tuning parameters (as will be shown in next section). So we take out this term from P(s) and calculate the stable region for:

$$P_{normalized}(s)=P_n(s)=\frac{1}{s^2+a\cdot s+1} \quad (11)$$

ISBN: 978-988-19252-1-3
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

And:

$$P(s) = \frac{-\frac{C^2}{N}\left(1-\frac{T_p}{R_0}\right)}{\frac{N}{C\cdot R_0^3}\frac{2R_0-T_p}{R_0-T_p}} P_n(s)$$
$$= \frac{-C^3R_0^2}{N^2}\frac{(R_0-T_p)^2}{(2R_0-T_p)}\cdot P_n(s) \qquad (12)$$
$$= P_{cte}\cdot P_n(s)$$

So, the gain scheduling term is given by (13).

$$K_1 = -\frac{2R_0-T_p}{(R_0-T_p)^2}\frac{N^2}{C^3R_0^2} \qquad (13)$$

It is important to choose the worst and best scenarios properly. A good knowledge of the network is required.

In the next section the network parameters are defined. The stable regions are depicted and the PID is tuned. Then, linear and non-linear simulations will show the goodness of the method.

V. SIMULATIONS

A. Tuning the controller

The basic network topology used as an example to test the controller is depicted in Fig. 3. It is a typical single bottleneck topology. The link capacity (C) is kept constant in all experiments. Although this parameter could change, its adjustments are more related with network updates than with everyday traffic conditions. The different scenarios come from changing the number of users (N) and the round trip time (R_0) (see Table 1).

TABLE I
NETWORK CONFIGURATIONS

	N	R_0
Best case	80	0.1
Extreme Worst case	40	0.9
Working case	60	0.25
Worst case	45	0.45

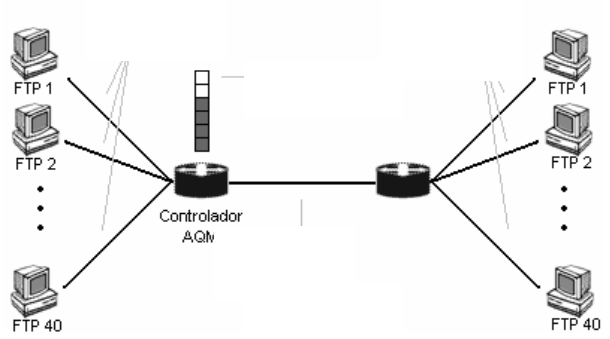

Fig. 3. Dumbbell topology

The situations deal with a broad range of conditions: few users and big delay, many users and small delay and in-between situations. As can be seen, the worst situation (extreme worst case) is when the number of users is 40 and the round trip delay is 0.9 seconds. The best situation has a delay of only 0.1 seconds and 80 users.

Fig. 4 shows the open loop poles of the system for the best and extreme working scenarios, without considering the delay. The lower graphs depict the open loop poles and zeros if a Padé approximation of first order is used. The other scenarios have their poles and zeros between these two situations.

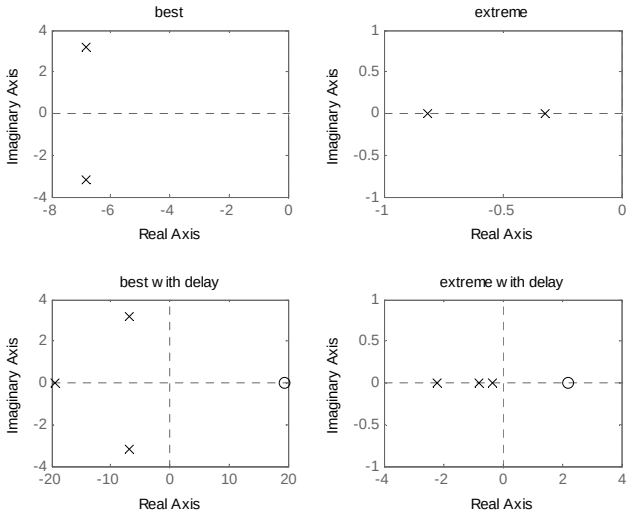

Fig. 4. Open loop poles and zeros

The open loop step response for each of the studied scenarios of the original linear system ($P(s)$) and the normalized one ($P_n(s)$) are shown in Fig. 5 (upper and lower, respectively). The step has amplitude of 0.01 because the input of the system is the probability of marking a packet and it always takes values between 0 and 1. In fact, the smaller the changes, the better for the system.

Both systems have the same settling time, but as will be shown later in the section, the behavior once the loop is closed will be very different. The worst scenario gives the slowest open loop response (settling time: 14.7 seconds) and the best scenario gives the fastest one (settling time: 0.74 seconds).

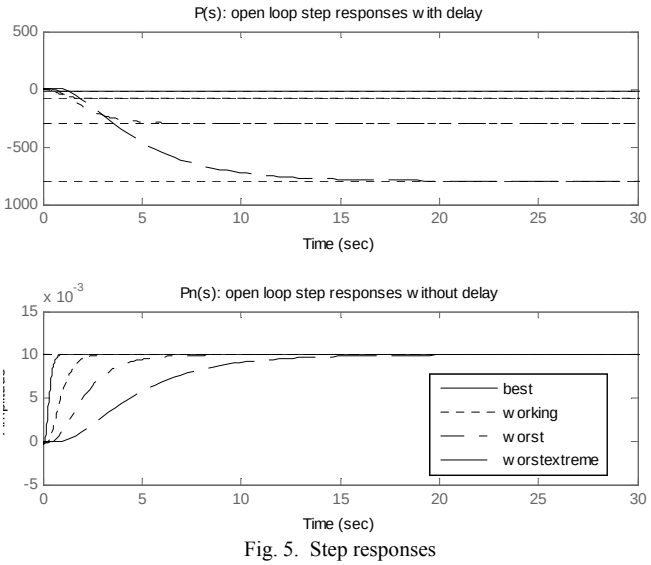

Fig. 5. Step responses

Let us consider the normalized transfer function approach. Using Hobenbichler's approach ([7]), the 3-dimensional stable region for the PID is depicted in Fig. 6. The color code is the same as in Fig. 5. Note that if K_p, K_i and K_d are chosen inside these regions, the closed loop response will be stable. We are looking for a controller that

ISBN: 978-988-19252-1-3
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

is stable in all the working scenarios: the best, the worst and those in between. Maybe the controller is not the best for each of the cases, but it would work well in all of the scenarios.

Moreover, due to the normalization in the transfer function, the scales of the graph are in the same range or magnitude, making the tuning of the controller easier.

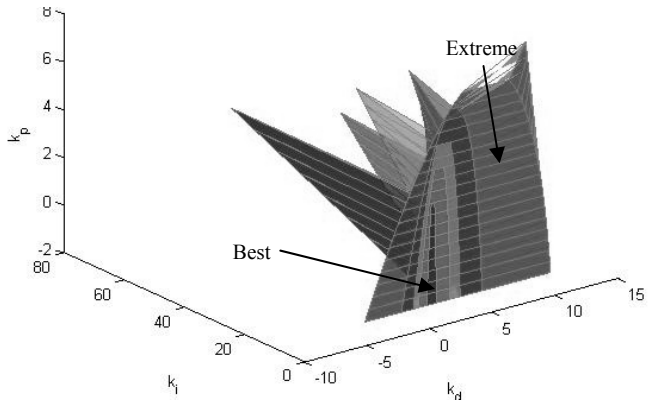

Fig. 6. Stable region with proposed method

This is clearer if we see the same 3-dimensional representation for the system without the methodology proposed in the paper. In Fig. 7, the stable regions are shown, but the scales are in the range of 10^{-3} to 10^{-4}. Moreover, the PID controller will give a worse performance, as will be shown in the simulations. It is more difficult to see where all the regions overlap.

In [17] and [18], the effect of N and R_0 on the size of the controller's stability region was studied. They concluded (and our experiments also agree with their result) that the smallest stability region is obtained for the smallest number of users (N) and the minimum round trip delay (R_0) (Fig. 7 shows the region).

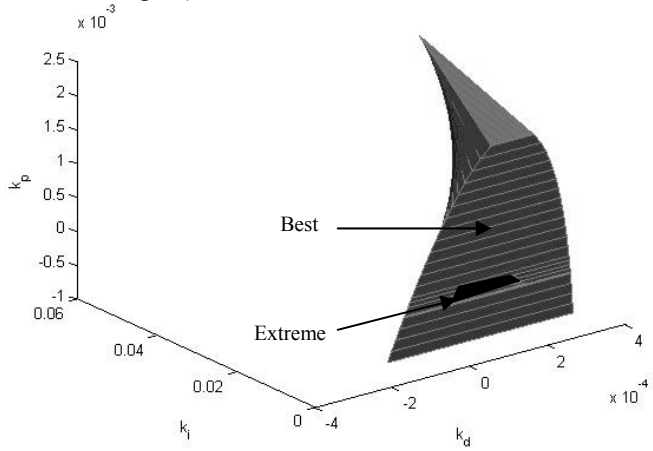

Fig. 7. Stable region with classical approach

B. Linear Simulations

The controller's parameters, when the method proposed in the paper is applied, are chosen as K_p=0.5, k_i=0.3 k_d=0.01. Fig. 8 shows the closed loop step response of the linear systems for the four scenarios under study. The settling time is small and all the systems have a settling time smaller than 15 seconds.

If the classical approach is followed, the controller parameters are K_p= $-4\cdot10^{-5}$, k_i= -10^{-5} and k_d= $-2\cdot10^{-5}$. As shown in Fig. 9, the settling times are very different. The best case scenario has a settling time around 250 seconds and the worst extreme is much faster (10 seconds). These settling times are very different: there are two orders of magnitude between them. This way, it is not feasible to have the same PID controller giving good closed-loop responses.

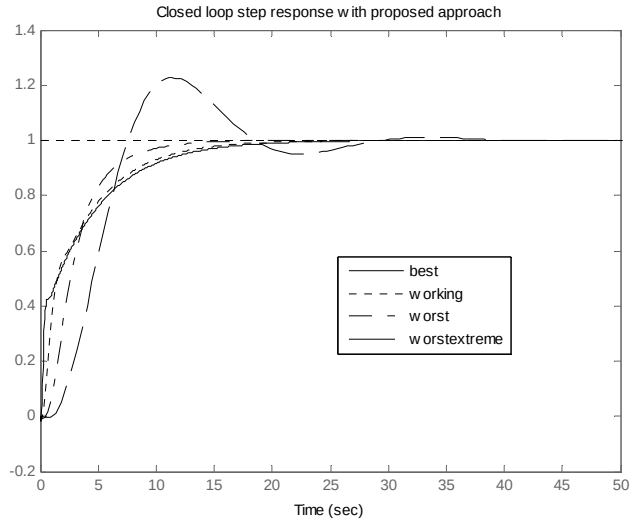

Fig. 8. Closed-loop step response with proposed method

As can be seen, using the approach presented in the paper (Fig. 8), the settling times have the same order of magnitude and the same PID can work well for all the scenarios.

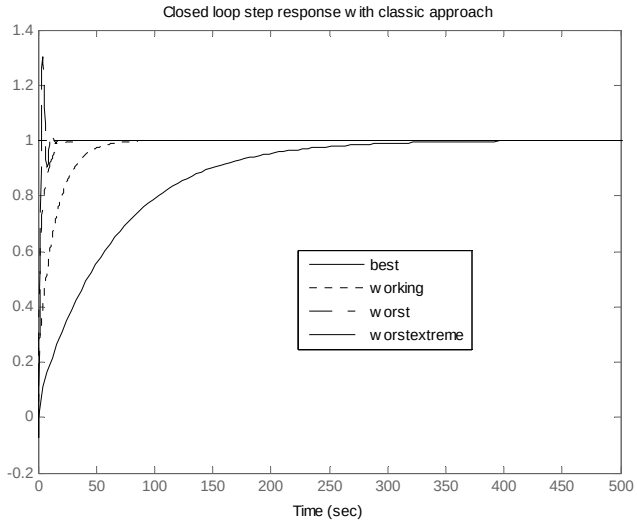

Fig. 9. Closed-loop step response with classical approach

C. Nonlinear Simulations

Taking the model described in Section III and the controllers described in Section IV, the following experiments were carried out.

In the first experiment, the reference is set to 230 packets (30 above the working point). Looking at the probability of marking a packet, there are not many differences between the two approaches (Figs. 10 and 11). However, the graphs with the queue evolution (Figs. 12 and 13) show significant differences. It should be noted that the settling time when the proposed method (Fig. 12) is applied is smaller than when the classic approach is followed (Fig. 13). As was expected, the slowest response corresponds to the system with the biggest number of users.

ISBN: 978-988-19252-1-3
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

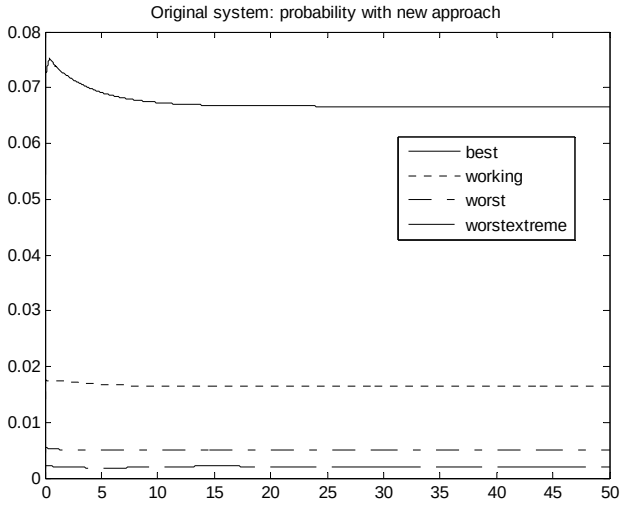

Fig. 10. Experiment 1, probability with new method

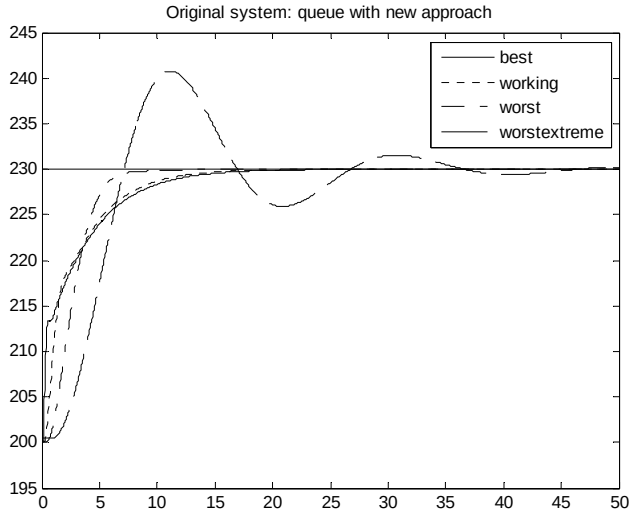

Fig. 12. Experiment 1, queue with new method

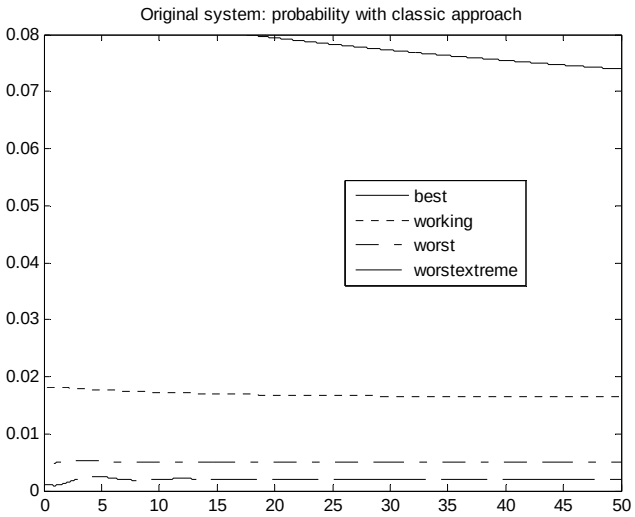

Fig. 11. Experiment 1, probability with classic approach

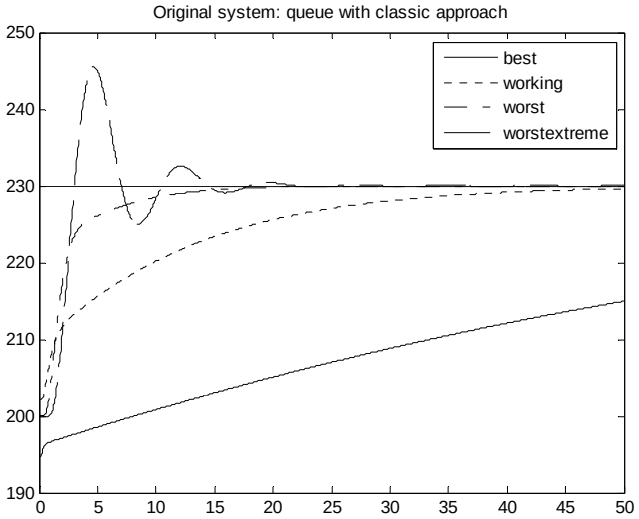

Fig. 13. Experiment 1, queue with classic approach

In experiment 2, a variable reference was applied. First, the queue set-point (in packets) was increased by 30, then by 60 and finally decreased by 80 packets. All the plants performed well (Fig. 14-17).

The performance of the system follows the same patterns as in experiment 1. Fig. 14 shows the evolution of the probability for the proposed method. The peaks are smaller than when using the classic approach (Fig. 15).

Fig. 16 shows the queue values with the modified PID. Clearly, the settling times are smaller than when the classic approach (Fig. 17) is followed (as happened in the first experiment).

Again, the slowest response corresponds to the system with the biggest number of users

Finally, let us consider an experiment where the number of users (N) changes dynamically.

The network default configuration is the working scenario described in Table I. The set point does not change and is equal to the queue nominal value (200 packets).

The changes in the number of users are depicted in Fig. 18 and the probability and queue size for the classic and proposed approaches in Fig. 19 and Fig. 20, respectively.

Looking at Fig. 19, we can conclude that the changes and the real value of the probability are smaller with the new method. Smaller probability values mean that the probability of discarding packets is smaller.

As is shown in Fig. 19 and 20, the classic approach cannot handle the sudden change in the number of users, whereas the new approach recovers very fast from the variations The queue has almost no changes with the proposed PID, but with the classic approach suffered greatly from the changes in the number of users.

ISBN: 978-988-19252-1-3
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

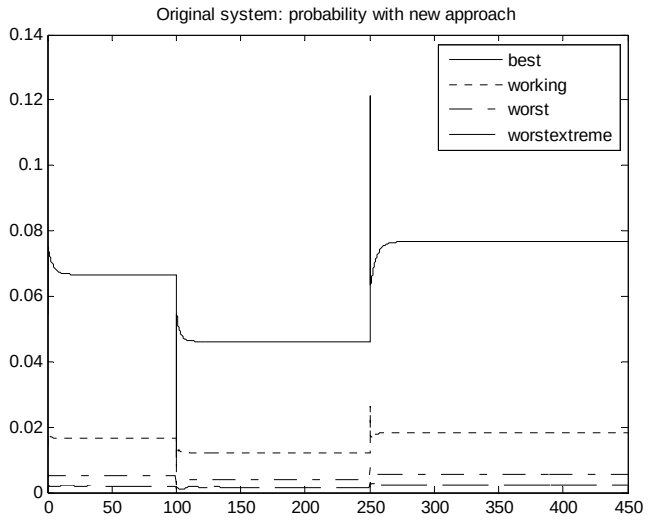

Fig. 14. Experiment 2, probability with new method

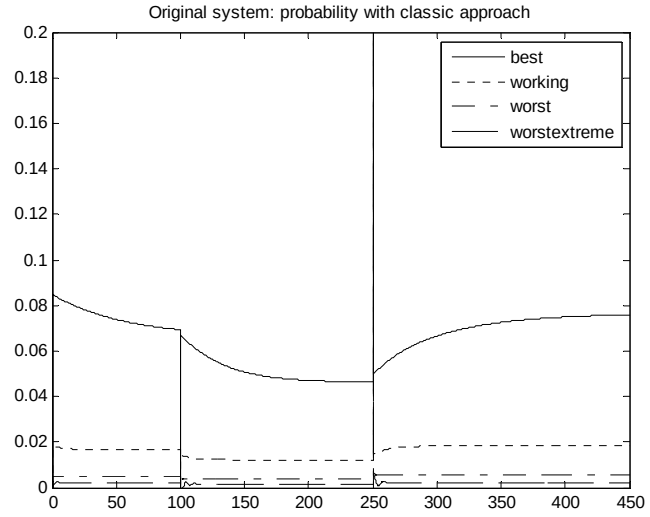

Fig. 15. Experiment 2, probability with classic method

The mean, standard deviation and variance for the queue size following the classic and the new approach are shown in Table II. The reference is set to 200 packets and the mean value for the proposed methodology is 201.4 packets. This is a value very close to the reference, especially if we compared it with the mean of 210 packets of the classic approach.

TABLE II
MEAN AND VARIANCE RESULTS

	Mean	St. dev	Var
Classic	210.5	37.3	1393.4
New	201.4	31.9	1019.7

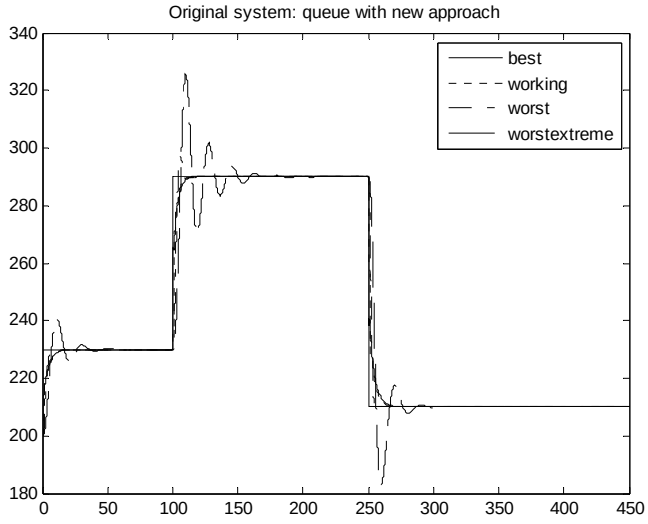

Fig. 16. Experiment 2, queue with new method

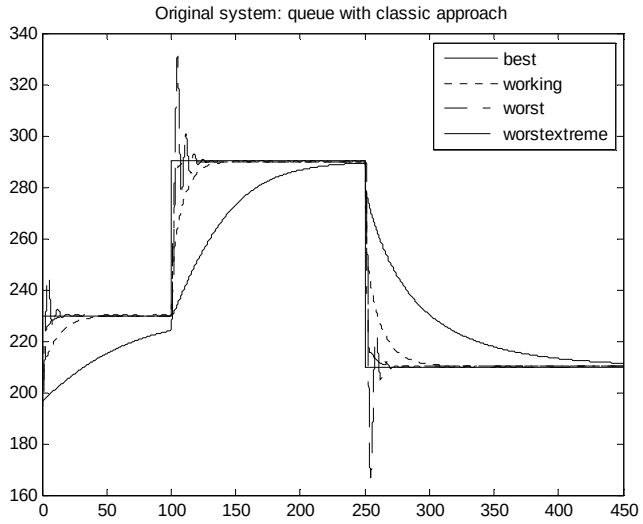

Fig. 17. Experiment 2, queue with new method

VI. CONCLUSION

The paper has presented a PID design with linear gain scheduling and normalized values that works very well under different network traffic conditions. The controller is tuned for the worst scenario and works properly in a wide range of situations.

As we are working with TCP Westwood, the analysis of the results is valid either in wired or wireless routers. Linear and nonlinear simulations confirm the technique's potential. This is especially convenient, as many design procedures are tested in restricted situations with no guarantee that the controller will behave similarly in other scenarios.

The technique presented in the paper gives more uniform results than the classical approach. It does not matter if the scenario is close to the nominal situation or a worst case setting, the controller reacts adequately and the settling times are all of the same order. The classic approach cannot deal well with extreme situations.

Results are encouraging and more work should be done.

ISBN: 978-988-19252-1-3
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

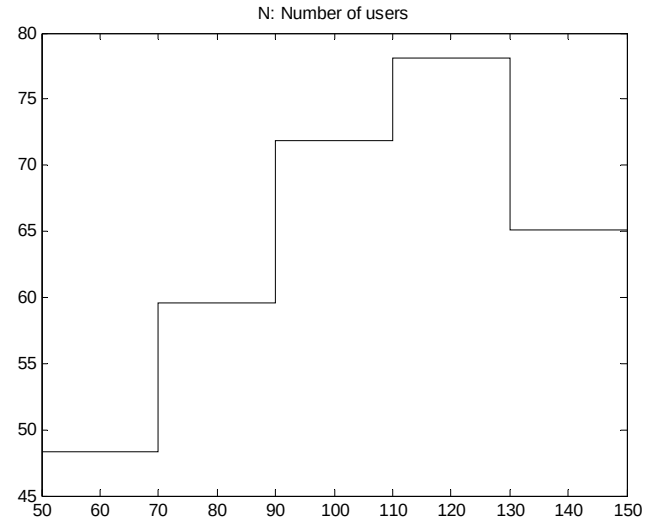

Fig. 18. Changing N

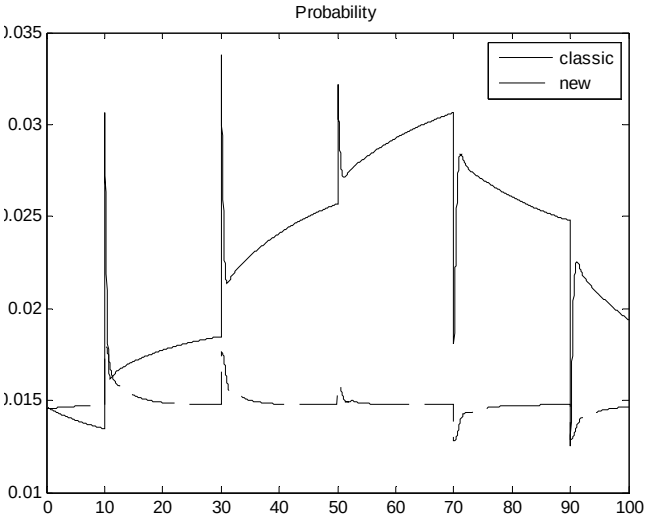

Fig. 19. Probability when the number of users changes

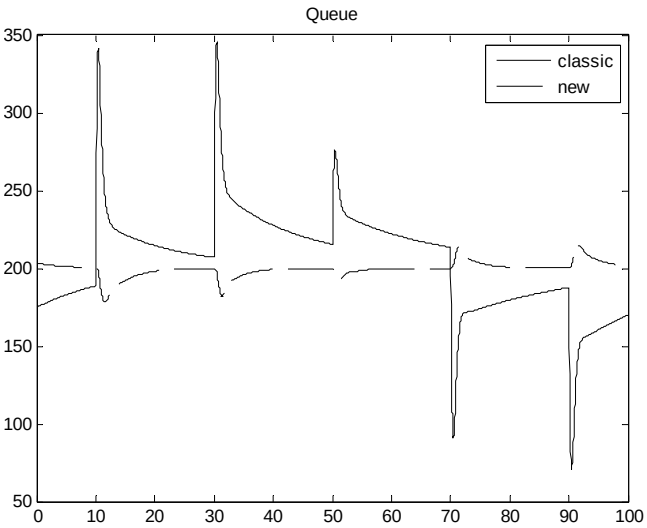

Fig. 20. Queue size when the number of users changes

ACKNOWLEDGMENT

The author would like to thank Prof. F. Tadeo for so many fruitful discussions.

REFERENCES

[1] Azuma, T., T. Fujita, M. Fujita (2006). Congestion control for TCP/AQM networks using State Predictive Control. Electrical Engineering in Japan, 156, 1491-1496.

[2] Deng, X., S. Yi, G. Kesidis. C.R. Das (2003). A control theoretic approach for designing adaptive AQM schemes. GLOBECOM'03, 5, 2947 – 2951.

[3] Jacobson, V. (1988). Congestion avoidance and control. ACM SIGCOMM'88.

[4] Ryu, S., C. Rump, C. Qiao (2004). Advances in Active Queue Management (AQM) based TCP congestion control. Telecommunication Systems, 25, 317-351.

[5] Hollot, C.V., V. Misra, D. Towsley, W. Gong (2002). Analysis and Design of Controllers for AQM Routers Supporting TCP flows. IEEE Transactions on Automatic Control, 47, 945-959.

[6] Aström K.J. and T. Hägglund (2006), Advanced PID control. ISA. NC.

[7] Hohenbichler, N. (2009). All stabilizing PID controllers for time delay systems. Automatica, 45, 2678-2684.

[8] Silva, G., A. Datta and S. Bhattacharyya. PID controllers for time-delay systems. Birkhäuser, 2005.

[9] Long, G.E., B. Fang, J.S. Sun and Z.Q. Wang (2010). Novel Graphical Approach to Analyze the Stability of TCP/AQM Networks. Acta Automatica Sinica, 36, 314-321.

[10] M. di Bernardo, L. A. Grieco, S. Manfredi, and S. Mascolo," Design of robust AQM controllers for improved TCP Westwood congestion control", Proceedings of the 16th International Symposium on Mathematical Theory of Networks and Systems (MTNS 2004), Katholieke Universiteit Leuven, Belgium, July, 2004.

[11] Chen, J., F. Paganini, M.Y. Sanadidi, R. Wang and M. Gerla (2006). Fluid-flow analysis of TCP Westwood with RED. Computer Networks, 50, 132-1326.

[12] Mascolo, S., C. Casetti, M. Gerla, M.Y. Sanadidi, R. Wang. TCP Westwood: bandwidth estimation for enhanced transport over wireless links, in: Proceedings of Mobicom, 2001.

[13] Wang, R., M. Valla, M.Y. Sanadidi, B. Ng, M. Gerla, Efficiency/friendliness tradeoffs in TCP Westwood, in:IEEE Symposium on Computers and Communications, Taormina, Italy, July 2002.

[14] Wang, R., M. Valla, M.Y. Sanadidi, M. Gerla, Adaptive bandwidth share estimation in TCP Westwood, in: Proceedings of IEE Globecom, Taipei, 2002.

[15] Hollot, C.V. and Y. Chait (2001). Nonlinear stability analysis for a class of TCP/AQM networks". In Proceedings of the 40th IEEE Conference on Decision and Control, Orlando, USA.

[16] Alvarez, T. (2012). Designing and Analysing Controllers for AQM routers working with TCP Westwood protocol. (Internal Report, University of Valladolid. To be submitted).

[17] Al-Hammouri, A.T. , V. Liberatore, M.S. Branicky, and S.M. Phillips (2006b). Parameterizing PI congestion controllers. Proc. First International Workshop on Feedback Control Implementation and Design in Computing Systems and Networks, Vancouver, CANADA.

[18] Al-Hammouri, A.T., V. Liberatore, M.S. Branicky, and S.M. Phillips (2006a). Complete stability region characterization for PI-AQM. SIGBED Review, 3(2).

ISBN: 978-988-19252-1-3
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)