Controller Design for Congestion Control: Some Comparative Studies

Teresa Alvarez, Hector de las Heras, Javier Reguera

***Abstract*— The congestion control problem in data networks is analyzed here from the point of view of control systems, and some solutions are compared. By analyzing in detail their linearized transfer functions, it is discussed why TCP Westwood and TCP NewReno present different behavior when dealing with the same network conditions, showing why TCPWestwood gives better performance. Moreover it is shown that adequate stability and performance can be obtained, even in the presence of the wide variations in the parameters of the networks, as long as the controller is properly designed.**

***Index Terms*— congestion control, control systems, robust performance, TCP Westwood, TCP NewReno, transfer function.**

I. Introduction

Routers are central components in modern data networks, as they simplify the connection of several networks, directing data to the final destination. When the router receives more data than the maximum limit it can process then congestion happens, causing excessive transmission delays, blocking of new connections, and losses of packets (see, for example, [1], [3], and references therein). If this congestion is not properly treated congestive collapse occurs, with the router providing very low throughput.

To correct this problem modern routers implement protocols that include some congestion control techniques, which provide feedback to the previous nodes in the network, in order for them to adapt the traffic to the maximum capacity of the congested router (see, for example, [4], [5]). This feedback is frequently provided by dropping packets: when the loss of some packets in a data link is detected, the protocols implemented in previous nodes reduce the traffic. Many congestion control algorithms have been proposed: this paper concentrates on those implemented in two of the most popular protocols: We assume that the protocol implemented at the transport layer is the Transmission Control Protocol (TCP), with an Active Queue Management (AQM) algorithm implemented in the network scheduler within the routers. These AQM algorithms implement the congestion control algorithm: a queue is implemented at the interface with each link, that holds the packets that have being scheduled (by TCP) to go out in that link; before the queue is full, in order for the sender to detect the congestion problem as soon as possible some packets are dropped (or marked, depending on the specific algorithm) [5,6]. Compared with other congestion control algorithms (such as the drop-tail implemented in many routers), AQM marks the packets probabilistically to improve sharing the bandwidth between different links, the improving the processing of bursty flows and avoiding synchronized oscillations between the links.

Many AQM algorithms have been proposed and tested (see, for example, [5]). Moreover, even if the same AQM technique is implemented results are different depending on the TCP protocol used. This paper concentrates on discussing the advantages of using algorithms derived from techniques that are standard in control systems and have being extensively tested in application areas other than data networks (see [7,8] for some general reference of control systems in general) considering two different TCP variants: Westwood and New Reno. These control system techniques normally require an approximate model of the system to be controlled, expressed as linear dynamic transfer function; some dynamical models of data congestion were already proposed in [10], and its linearization was discussed in [10].

From the many controller design methodologies proposed in control systems, PID control is probably the most widely used to develop AQM algorithms. We can cite [12], [13], [14], [15] and references therein.

In order to make these studies the methodologies described will be applied to a specific communication problem: using simulation would make possible to make a fair comparison, by reproducing the same traffic conditions when testing each algorithm.

This paper will present a comparison between Westwood TCP and NewReno TCP. The non linear fluid models are linearized and the transfer function models are then obtained. These transfer functions will be compared using the step response, the frequency response, the location of poles and zeros under different network configurations. These studies will help to understand why TCP Westwood behaves better when there are changes in the traffic conditions, number of users or delay in the network, regardless of the type of communication (wired or wireless).

The organization of the paper is the following: Sections 2 and 3 describe the TCP Westwood and TCP NewReno protocol and their fluid models. Section 4 presents a comparison between TCP Westwood and TCP NewReno. Section 5 briefly describes the controller to be implemented

Manuscript received March 2014. Work funded by MiCInn DPI2010-21589-C05-05.

T. Alvarez, H. de las Heras and J. Reguera are with the University of Valladolid (Telf: +34 983423162; email: tere@autom.uva.es).

and finally Section 6 discusses some experiments. A discussion concludes the paper.

II. TCP WESTWOOD

TCP Westwood (TCPW) [16] is a popular congestion control algorithm that has shown to provide robust selections for the congestion control parameters (in fact, it is implemented with minor modifications as part of the Linux kernel). It is particularly relevant when packet losses are not only generated by congestion control algorithms (for example in wireless networks).

TCPW relies on studying the stream of returning acknowledgment packets (ACKs) to adapt some parameters of the congestion control algorithm [16], in particular the size of the so-called *congestion window* (cwnd) that fixes the maximum number of bytes that can be accepted at each link: this window is slowly increased until some acknowledgements are not received or a timeout expires.

A. Fluid-flow Model

The nonlinear dynamical model of TCPW ([15], [16], [17], [18]) is now presented, as it will be the base of the models used for controller design and comparison. If *W(t)* denotes the size of the TCPW window at a given link in packets, the number of packets at the corresponding queue (*q*(*t*):*queue length*) can be derived from the following equations as follows:

$$\frac{dq}{dt} = -C(t) + \frac{N(t)}{\frac{q(t)}{C(t)} + T_p} W(t) \tag{1}$$

$$\frac{dW}{dt} = \frac{1}{\frac{q(t)}{C(t)} + T_p} - \frac{W(t)W\left(t - R(t)\right)}{\frac{q\left(t - R(t)\right)}{C(t - R(t))} + T_p} p\left(t - R(t)\right) + T_p \left(\frac{W\left(t - R(t)\right)}{\frac{q\left(t - R(t)\right)}{C(t - R(t))} + T_p} \right)^2 p\left(t - R(t)\right) \tag{2}$$

Where T_p is the propagation delay (in seconds), *C(t)* is the *link capacity* (in packets/s), *N(t)* is the number of TCPW sessions (*load factor*) and *R(t)* is the round-trip time (equal to *q(t)/C(t)+Tp*) and *p(t)* is the variable that is modified by the AQM algorithm: the probability of discarding a packet (or marking it in some AQM implementations).

B. Model for controller design

As it has been mentioned, a linear model with constant parameters is frequently used to design control systems, as it simplifies the analysis and design of controllers. This kind of models can be easily derived by linearization: in our case, temporarily assuming that the number of active TCPW sessions, the link capacity and the round-trip time do not change, and discarding high-frequency components, a Taylor approximation of (1) and (2) at a given set of operating parameters gives the following transfer-function description of our system (see [11]or [15] for details):

$$q(s) = \frac{\frac{-C^2}{N}\left(1 - \frac{T_p}{R}\right)e^{-Rs}}{s^2 + \frac{CR + 2N}{CR^2}s + \left(\frac{N}{CR^3}\frac{2R - T_p}{R - T_p}\right)} p(s) \tag{3}$$

Or equivalently:

$$q(s) = \frac{\frac{-CQ}{NR}e^{-Rs}}{s^2 + \left(\frac{1}{R} + \frac{2N}{CR^2}\right)s + \frac{N}{CR^3}\left(\frac{CR}{Q} + 1\right)} p(s) \tag{4}$$

Where C, N, R and Q are the values of *C*(*t*), *N*(*t*), *R*(*t*) and *q*(*t*) at the current working point.

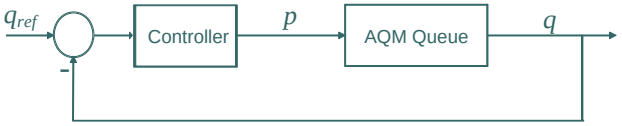

Figure 1: Block diagram of the AQM feedback control system

The static gain of this linearized system would be:

$$\frac{-C^3R^2}{N^2}\frac{\left(R - T_p\right)^2}{\left(2R - T_p\right)} = \frac{-Q^2C^2R^2}{(CR + Q)N^2} \tag{5}$$

With the system responding as a stable second-order system with delay R, underdamped if Tp>2NR.

III. TCP NewReno

TCP NewReno (TCPNR) is still probably the most widely used congestion control technique (with several variants), although TCPW provides better performance when the link includes large paths or might generate unexpected packet losses (like wireless networks). This simpler algorithm can be modelled as follows:

$$\frac{dq}{dt} = -C(t) + \frac{N(t)}{\frac{q(t)}{C(t)} + T_p} W(t) \tag{6}$$

$$\frac{dW}{dt} = \frac{1}{\frac{q(t)}{C(t)} + T_p} - \frac{W(t)W\left(t - R\right)}{\frac{q\left(t - R\right)}{C(t - R)} + T_p} p\left(t - R\right) \tag{7}$$

The corresponding transfer function model is then

$$q(s) = \frac{\frac{-C^2}{2N}e^{-Rs}}{\left(s + \frac{2N}{CR^2}\right)\cdot\left(s + \frac{1}{R}\right)} p(s) \tag{8}$$

Or equivalently:

$$q(s) = \frac{\frac{-C^2}{2N}e^{-Rs}}{s^2 + \left(\frac{1}{R} + \frac{2N}{CR^2}\right)s + \frac{2N}{CR^3}} p(s) \tag{9}$$

ISBN: 978-988-19253-5-0
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Thus, it can be seen that the TCPNewReno static gain is

$$\frac{-C^3R^3}{4N^2} \qquad (10)$$

and the dynamic response corresponds to a second order overdamped system with delay R, and real poles placed at

$$-\frac{2N}{C\cdot R^2}, -\frac{1}{R} \qquad (11)$$

Where the dominant pole is given by $-2N/(CR^2)$. Please, take into account that only if 2N/C is bigger than 1/R the dominant pole would be -1/R.

IV. Comparison between TCPW and TCPNR

Research has shown ([16]) that TCPW is a good approach when there are wireless links in the network. This section presents a comparison using the transfer function models of TCPW and TCPNR in order to achieve some conclusions regarding the static gain, the open loops and the frequency response of each system. This analysis will allow designing adequate controllers. The number of users (N) and the RTT (R) should have coherent values.

In [20] and [21] some guidelines on how to choose these values were given. The best scenario is to have many users and a small delay. The worst situation is to have few users and a big delay. This conclusion is true for both TCP approaches under study.

First, let's compare the static gain of both systems:

- TCPW static gain (5): $\frac{-Q^2C^2R^2}{(CR+Q)N^2}$
- TCPNR static gain (10): $\frac{-C^3R^3}{4N^2}$

Which one is bigger? It depends on the parameters' values, but (usually) (5) will be bigger than (10), i.e., the input will be amplified:

$$\left|\frac{-Q^2C^2R^2}{(CR+Q)N^2}\right| \le \left|\frac{-C^3R^3}{4N^2}\right|$$

Practical examples will show this. This is a very important inference for the designing controllers: when the number of users or the delay change in the network, TCPW will be more robust than TCPNR.

Both systems have no zeros and both have two poles. However, the TCPNR transfer function always has two real poles and the TCPW system's poles can be real or complex conjugate. The denominators of both transfer functions are:

- TCPW: $s^2+\left(\frac{1}{R}+\frac{2N}{CR^2}\right)s+\frac{N}{CR^3}\left(\frac{CR}{Q}+1\right)$
- TCPNR: $s^2+\left(\frac{1}{R}+\frac{2N}{CR^2}\right)s+\frac{2N}{CR^3}$

The second and first order terms are equal, the difference is on the independent, term and they have very similar:

$$\frac{N}{CR^3}\left(\frac{CR}{Q}+1\right) \text{ versus } \frac{2N}{CR^3} \text{ is the same than}$$

$$\left(\frac{CR}{Q}+1\right) \text{ versus } 2$$

When is $\left(\frac{CR}{Q}+1\right)$ greater than 2? It depends on C, R and Q, but in real networks the initial queue length will be smaller than the R times C. As a result, for a given N, C, R and Q, the two poles of TCPW are always placed between the two poles of TCPNR. Another conclusion is that TCPW will be a bit faster than TCPNR. This also can be justified looking at the analytical solution of the poles:

If we look at the solution of the second order equation of TCPNR:

$$poles_{1,2} = \frac{-\left(\frac{1}{R}+\frac{2N}{CR^2}\right)\pm\sqrt{\left(\frac{1}{R}+\frac{2N}{CR^2}\right)^2-4\frac{2N}{CR^3}}}{2} = \frac{-\left(\frac{1}{R}+\frac{2N}{CR^2}\right)\pm\sqrt{\frac{1}{R^2}+\frac{4N^2}{C^2R^4}-\frac{4N}{CR^3}}}{2}$$

On the other hand, the poles of TCPW will be given by:

$$poles_{1,2} = \frac{-\left(\frac{1}{R}+\frac{2N}{CR^2}\right)\pm\sqrt{\left(\frac{1}{R}+\frac{2N}{CR^2}\right)^2-4\frac{N}{CR^3}\left(\frac{CR}{Q}+1\right)}}{2} = \frac{-\left(\frac{1}{R}+\frac{2N}{CR^2}\right)\pm\sqrt{\frac{1}{R^2}+\frac{4N^2}{C^2R^4}-\frac{4N}{QR^2}}}{2}$$

When $\frac{4N}{QR^2} > \frac{1}{R^2}+\frac{4N^2}{C^2R^4}$, then the poles will be complex conjugate.

A second order system is characterized by this equation:

$q(s) = \frac{Kw_n^2}{s^2+2\xi w_n s+w_n^2}\,p(s)$, where ξ is the damping ratio and w_n the natural frequency.

Considering TCPW:

$$Kw_n^2 = \frac{-CQ}{NR}$$

$$s^2+2\xi w_n s+w_n^2 = s^2+\left(\frac{1}{R}+\frac{2N}{CR^2}\right)s+\left(\frac{CR}{Q}+1\right)\frac{N}{CR^3}$$

$$w_n^2 = \left(\frac{CR}{Q}+1\right)\frac{N}{CR^3}$$

ISBN: 978-988-19253-5-0
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

$$\xi = \frac{\left(\frac{1}{R} + \frac{2N}{CR^2}\right)}{2\sqrt{\left(\frac{CR}{Q}+1\right)\frac{N}{CR^3}}} = \frac{(CR+2N)\sqrt{Q}}{2\sqrt{CRN(CR+Q)}}$$

When the poles are complex conjugate:

$$poles_{1,2} = -\xi w_n \pm j w_n \sqrt{1-\xi^2}$$

V. Controller design

This section briefly describes how a controller can be implemented into an AQM approach. Figure 1 shows the basic structure. As it has been mentioned the control action in AQM algorithm is probability of dropping (or marking) packets $p(t)$, with is adapted by comparing the current queue length $q(t)$ with a desired value q_{ref} (a percentage of the real capacity of the queue): see Figure 1.

We have chosen the PID controller [2] as the AQM congestion control algorithm, as this controller is widely used and gives good results. The structure of the controller is:

$$\begin{aligned} u(t) &= K_p\left(e(t) + \frac{1}{T_i}\int_0^t e(\tau)d\tau + T_d\frac{de(t)}{dt}\right) \\ &= K_p e(t) + K_i\int_0^t e(\tau)d\tau + K_d\frac{de(t)}{dt} \end{aligned} \quad (12)$$

It must be pointed out that this is the ideal equation: When the controller is implemented, the derivative term should be treated as a first order filter.

VI. Examples

This section compares TCPW and TCPNR under different traffic conditions. The basic topology is depicted in Figure 2. Different scenarios will be considered.

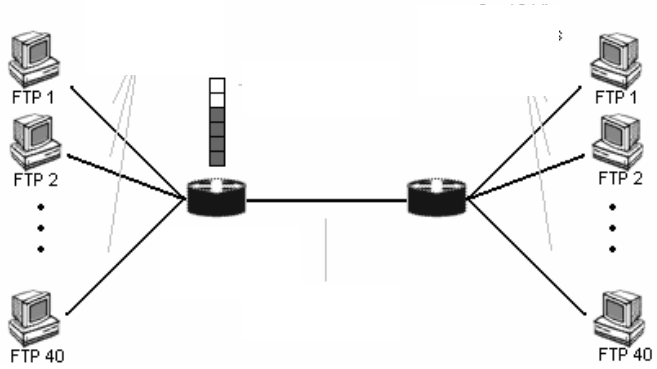

Figure 2: Dumbbell topology

A. First experiment: Fixed N, changing R

The first batch of experiments considers the parameters shown in Table I. We consider a fixed N and a changing R. The greater the delay, the smaller the initial marking probability.

TABLE I
Network Configurations

	N	R	C	P_0	Q
Case 1	50	0.15	3750	0.0158	200
Case 2	50	0.3	3750	0.0040	200
Case 3	50	0.6	3750	0.0010	200
Case 4	50	0.8	3750	0.0006	200
Case 5	50	1.1	3750	0.0003	200

TABLE II. Fixed N, changing R

	Caso 1	**Caso 2**	**Caso 3**	**Caso 4**	**Caso 5**
TCPW tf	$\frac{-100000}{s^2+7.852s+15.06}$	$\frac{-50000}{s^2+3.63s+3.272}$	$\frac{-25000}{s^2+1.741s+0.7562}$	$\frac{-18750}{s^2+1.292s+0.4167}$	$\frac{-13640}{s^2+0.9311s+0.2166}$
TCPNR tf	$\frac{-140625}{s^2+7.852s+7.901}$	$\frac{-140625}{s^2+3.63s+0.987}$	$\frac{-140625}{s^2+1.741s+0.1235}$	$\frac{-140625}{s^2+1.292s+0.05208}$	$\frac{-140625}{s^2+0.9311s+0.02004}$
TCPW s ga	-6639	-15283	-33061	-45000	-62948
TCPNR s ga	-17800	-1412400	-1139100	-27000000	-7018900
TCPW band.	2.4513	1.1563	0.5578	0.4143	0.2988
TCPNR band.	1.1475	0.2933	0.738	0.0415	0.0220
TCPW Poles	-4.5185 -3.3333	-1.9630 -1.6667	-0.9074 -0.8333	-0.6667 -0.6250	-0.4766 -0.4545
TCPNR Poles	-6.6667 -1.1852	-3.3333 -0.2963	-1.6667 -0.0741	-1.2500 -0.0417	-0.9091 -0.0220
TCPW Set tim	1.69 sec.	3.55 sec.	7.32 sec.	9.85 sec.	13.6 sec
TCPNR Se Ti	3.62 sec	13.8 sec	54 sec	95.5 sec	180 sec

ISBN: 978-988-19253-5-0
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

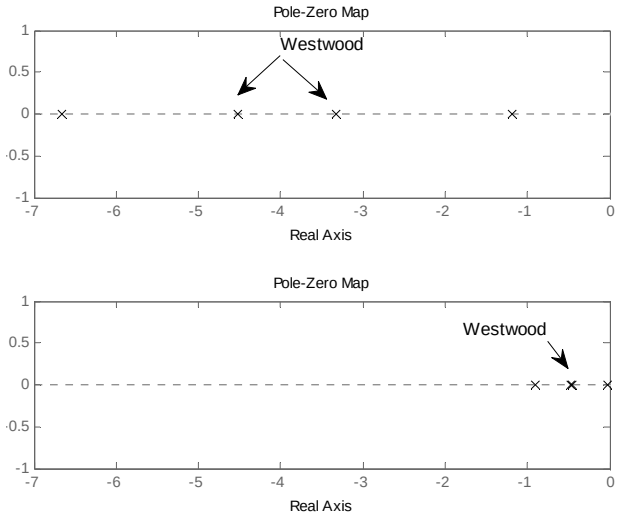

Figure 3: Case 1 and 5, open loop poles

Table II shows a comparison between TCPW and TCPNR in terms of open loop transfer function, static gain, bandwidth and poles and zeros.

As it was hinted in previous section, the span of variation of TCPW's static gain is smaller than the one of TCPNR. In fact, the standard deviation of TCPW static gain is $2.3 \cdot 10^4$ and TCPNR standard deviation is $2.9 \cdot 10^6$. If a controller is tuned for a certain network parameters and then there are changes, TCPW will perform much better.

Looking at the last two rows of Table II and Figure 3, we can see that the open loop poles of TCPW lie between the poles of TCPNR. Moreover, the greater the delay, the closer the poles to the origin, i.e., the slower the system.

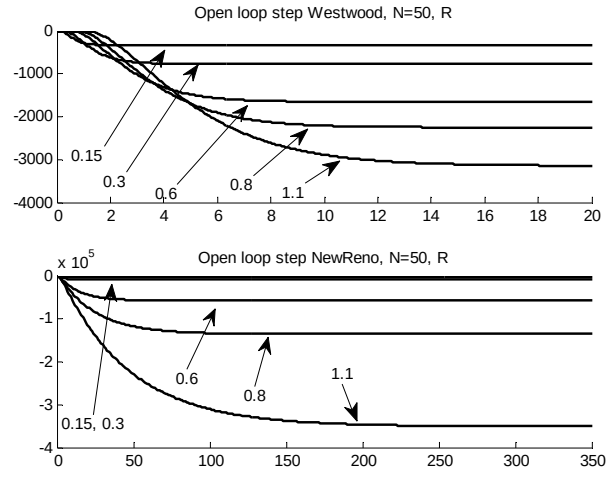

Figure 4: Open loop step responses and settling times

Figure 4 shows the open loop step response of both TCP variants considering the delay. Settling times are smaller for TCPW (as shown in Table II). The greater the delay is, the slower the system and greater the gain are. Clearly, TCPNR is slower than TCPW.

Figures 5 and 6 depict the open loop magnitude Bode diagram of each system. Again, TCPNR has a worst behavior than TCPW. TCPW's bandwidth is bigger than TCPNR's one: Westwood is more robust when there are perturbations or noise in the system.

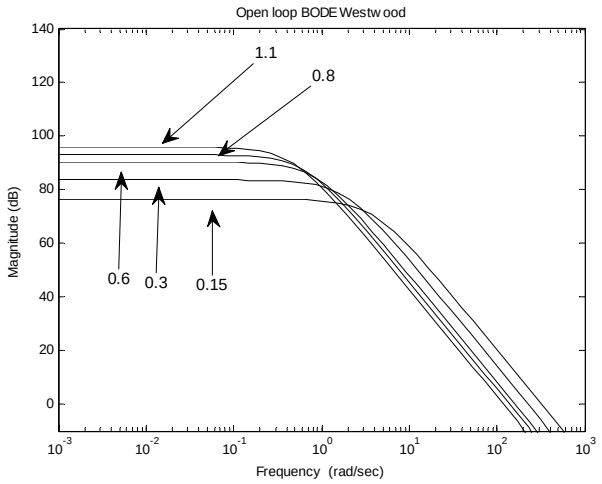

Figure 5: Open loop TCPW Magnitude Bode plot

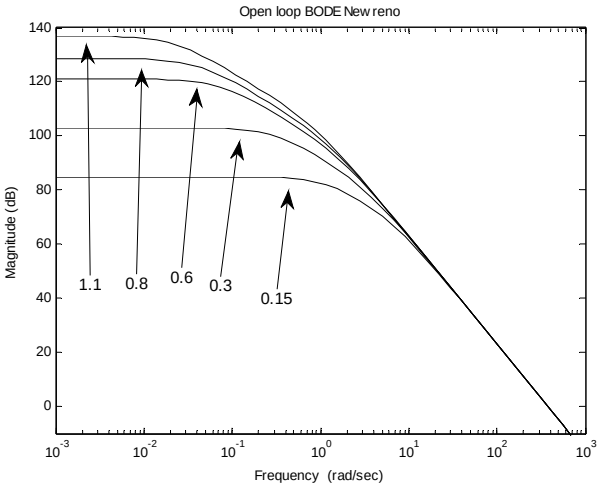

Figure 6: Open loop TCPNR Bode diagram

A. Second experiment: Changing N, fixed R

The second experiment considers a fixed R and a changing number of users:

TABLE III
NETWORK CONFIGURATIONS

	N	R	C	P_0	Q
Case 1	40	0.3	3750	0.0025	200
Case 2	50	0.3	3750	0.0040	200
Case 3	70	0.3	3750	0.0077	200
Case 4	100	0.3	3750	0.0158	200
Case 5	2000	0.3	3750	0.0632	200

Observing Table IV, it can be concluded that the bigger N give smaller the static gain. Again the standard deviation of the static gain is smaller for TCPW (9284) than for TCPNR (86497). Moreover, the bandwidth increases with the number of users.

As in experiment 1, all the poles are stable (Table IV and Figure 7). In fact, TCPW's poles are always placed between TCPNR's poles. As R is constant, the farthest pole is located at -1/R.

ISBN: 978-988-19253-5-0
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

TABLE IV. CHANGING N, VARYING R

	Caso 1	Caso 2	Caso 3	Caso 4	Caso 5
TCPW tf	$\frac{-62500}{s^2+3.57s+2.617}$	$\frac{-50000}{s^2+3.63s+3.272}$	$\frac{-35710}{s^2+3.748s+4.58}$	$\frac{-25000}{s^2+3.926s+6.543}$	$\frac{-12500}{s^2+4.519s+13.09}$
TCPNR tf	$\frac{-175800}{s^2+3.57s+0.7901}$	$\frac{-140625}{s^2+3.63s+0.9877}$	$\frac{-140625}{s^2+1.741s+0.1235}$	$\frac{-70310}{s^2+3.926s+1.975}$	$\frac{-315160}{s^2+4.519s+3.951}$
TCPW s ga	-23880	-15283	-7797	-3821	-955
TCPNR s ga	222470	-1423800	-726400	-35600	-8900
TCPW band.	0.9044	1.1563	1.6546	2.3382	4.0307
TCPNR band.	0.2353	0.2933	0.4076	0.5738	1.0666
TCPW Poles	-2.5399 -1.0305	-1.9630 -1.6667	-1.8741 ± 1.0335i	-1.9630 ±1.6401i	-2.2593 ± 2.8253i
TCPNR Poles	-3.3333 -0.2370	-3.3333 -0.2963	-3.3333 -0.4148	-3.3333 -0.5926	-3.3333 -1.1852
TCPW Set tim	4.6 sec	3.55 sec	2.38 sec	2.46 sec	1.95 sec
TCPNR Set tim	17.1 sec	13.8 sec	10.1 sec	7.23 sec	3.97 sec

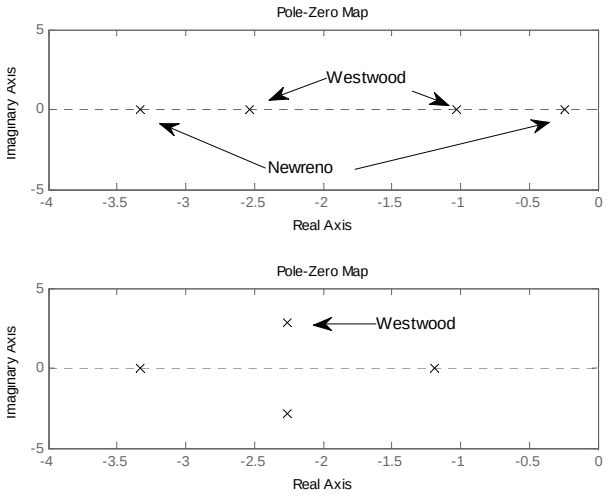

Figure 7: Experiment 2. Case 1 and 5, open loop poles

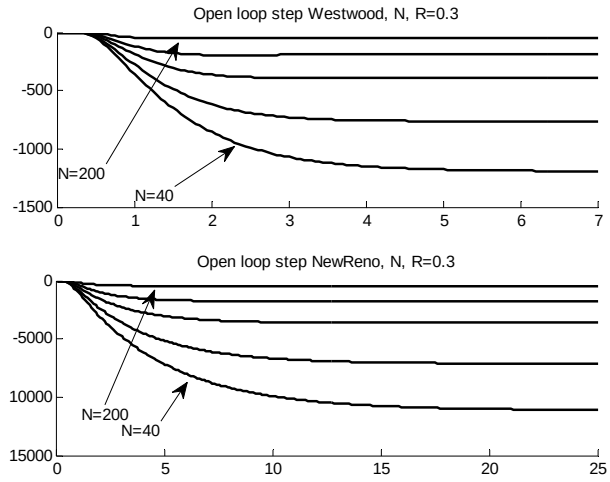

Figure 8: Experiment 2: Open loop step responses and settling times

The settling times (Figure 8, Table IV) are not so different as in experiment 1. Again TCPW is faster than TCPNR and the settling times are more similar. The magnitude Bode diagrams (Figures 9 and 10) are more similar than in experiment 1.

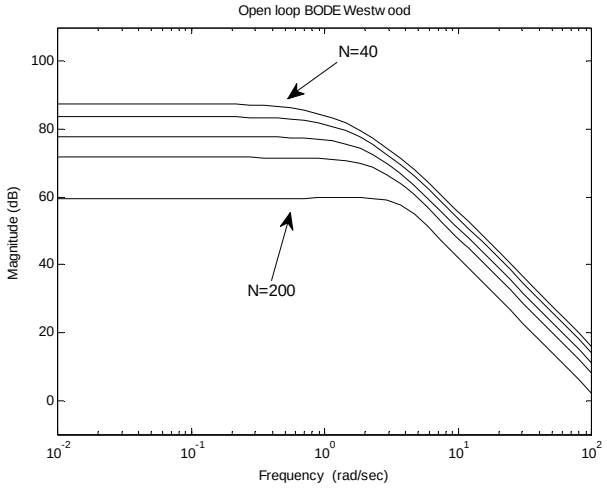

Figure 9: TCPW magnitude Bode plot, exp2

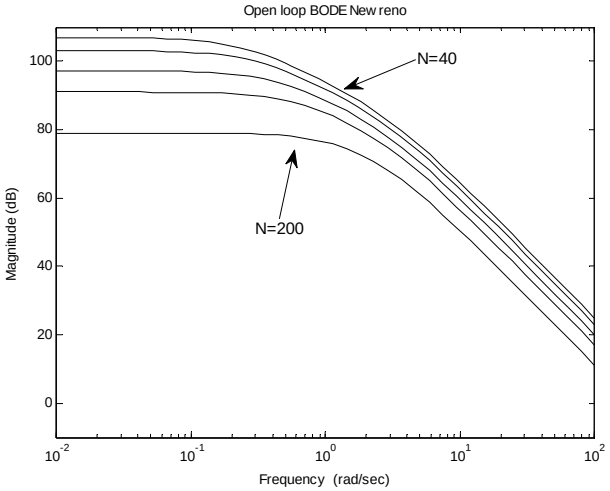

Figure 10: TCPNR magnitude Bode plot, exp2

A. Third experiment: Mixed situations

The final experiment will consider mixed situations (Table V). The objective is to design a PID controller for the scenario considered as a normal situation and to test the behavior of this controller in other scenarios.

ISBN: 978-988-19253-5-0
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

TABLE V
NETWORK CONFIGURATIONS

	N	R	C	P_0	Q
Case 1	80	0.1	3750	0.0910	200
Case 2	40	0.9	3750	0.0003	200
Case 3	60	0.25	3750	0.0082	200
Case 4	50	0.3	3750	0.0040	200

The open loop step responses are shown in Figure 11. The trends and comments regarding static gain, bandwidth and poles follow those given for experiments 1 and 2. It can be seen that the bigger the delay, the bigger the gain is. In fact, when the delay is big, TCPNR is clearly slower than TCPW (Figure 11).

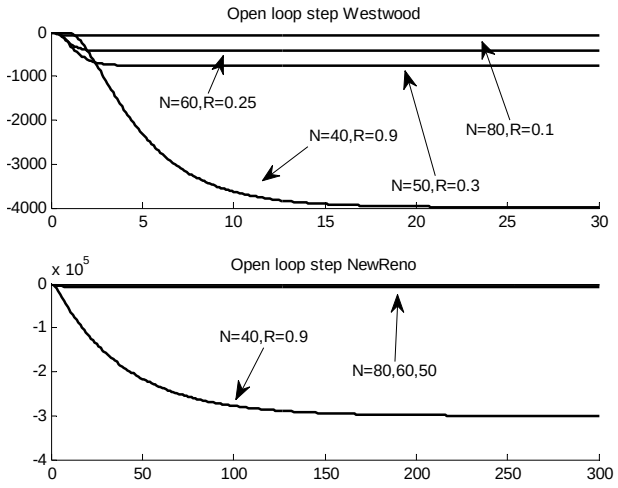

Figure 11: TCPW and TCPNR open loop step response

Case 4 is the working scenario: 50 users and a delay of 0.3 seconds. Using the classical method of Ziegler-Nichols [8] followed by some fine tuning, two controllers were tuned, one for TCPW and the other for TCPNR.:

- TCPW: Kp= -0.000015, Ti=5, Td=0.
- TCPNR: Kp= -0.00002, Ti=3, Td=0.05

The TCPW controller is a PI, the derivative term has been set to zero because there was no improvement when including this term.

Figure 12 shows the closed loop response of the TCPW system for the 4 scenarios under study. It represents the evolution of the router's queue (controlled variable) when a step is applied. Figure 13 shows the control variable (probability): how it changes to drive the queue to the desired value.

Figures 13 and 14 depict the closed loop behavior of the TCPNR system for the four scenarios under study. The controlled variable (queue size at the router) corresponds to Figure 12 (best settling time 18.2 sec., case 2 and the worst time is 867 sec. in Case1) and the probability of marking a packet is showed in Figure 13.

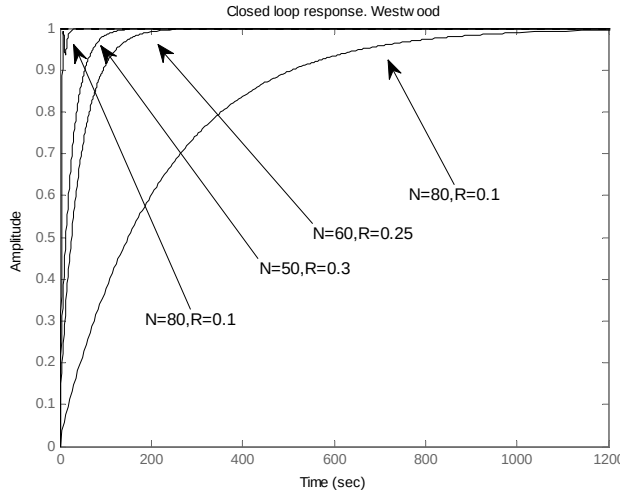

Figure 12: TCPW closed loop response, queue evolution.

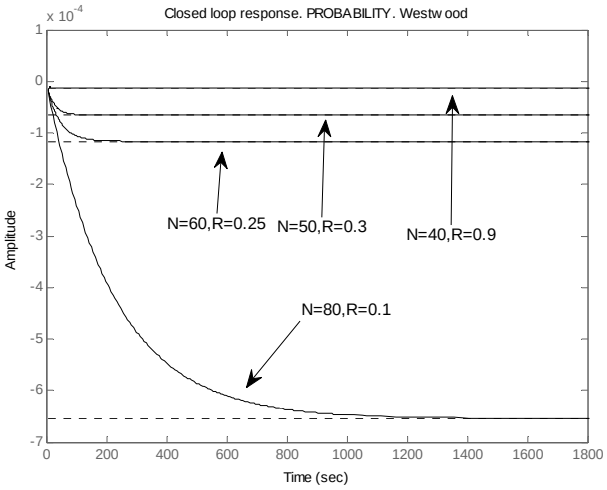

Figure 13: TCPW closed loop response, probability

Observing Figures 12 and 13, we can conclude that the step response is stable in all cases. When N and R are closer to the nominal scenario, the results are better. But what it is really important is that the system is robust to changes in the parameters. Looking at Figures 14 and 15 some conclusions are similar: the most similar the scenario to the nominal one (fourth graph) the better the results.

But, now there is a stability problem: when N=40 and R=0.9, the closed loop system (plant with the controller) is unstable. It is true that the controller has not been tuned for this situation. Nevertheless as TCPNR has shown to be less robust (see previous sections and experiments 1 and 2) to changes in the parameters, this is a situation that can arise in reality.

ISBN: 978-988-19253-5-0
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

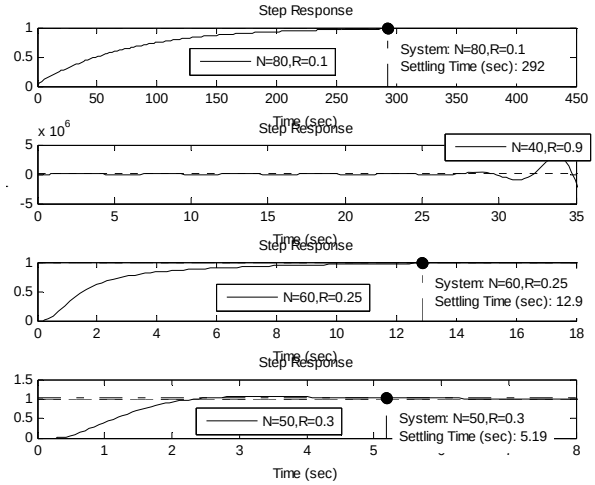

Figure 14: TCPNR closed loop response, queue evolution.

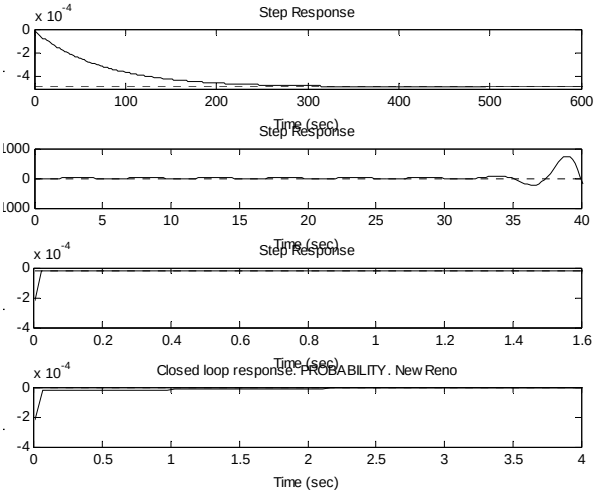

Figure 15: TCPNR closed loop response, probability

VI. Conclusion

The paper has presented a detailed analysis of TCP Westwood and TCP NewReno in terms of poles, static gain and frequency response of its linearized transfer functions, parametrized in terms of the network parameters. This has made possible to compared their expected responses for the same traffic conditions: The main conclusion is that as variations in the static gain and bandwidth are smaller for TCP Westwood and its bandwidth is bigger, the controlled system will be more robust in the inherent presence of changes in the network parameters: it has been shown that it can work properly in a wide range of situations as it has been shown through examples. Moreover, the TCP Westwood transfer function model has damping greater than 0.8.

Results are encouraging and more work is being done to use this information, such as using techniques for tuning the controller that takes into account the expected parameter variations.

Acknowledgment

The authors 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] Aström K.J. and T. Hägglund (2006), *Advanced PID control*. ISA. NC.

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

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

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

[6] Floyd, S., & Jacobson, V. (1993). Random early detection gateways for congestion avoidance. Networking, IEEE/ACM Transactions on, 1(4), 397-413.

[7] Mathis, M., Semke, J., Mahdavi, J., & Ott, T. (1997). The macroscopic behavior of the TCP congestion avoidance algorithm. ACM SIGCOMM Computer Communication Review, 27(3), 67-82.

[8] Dorf, R. C. (1995). Modern control systems. Addison-Wesley Longman Publishing Co., Inc.

[9] Franklin, G. F., Powell, J. D., & Emami-Naeini, A. (1994). Feedback control of dynamics systems. Addison-Wesley, Reading, MA.

[10] 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.

[11] Bolajraf, M., Tadeo, F., Alvarez, T., & Rami, M. A. (2010). State-feedback with memory for controlled positivity with application to congestion control. IET control theory & applications, 4(10), 2041-2048.

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

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

[14] 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.

[15] Teresa Alvarez, Diego Martínez (2013), Handling the Congestion Control Problem of TCP/AQM Wireless Networks with PID Controllers, in IAENG Transactions on Engineering Technologies, pages 365-379.

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

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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: 978-988-19253-5-0
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)