Analytical Design of First-Order Controllers for the TCP/AQM Systems with Time Delay

Sana Testouri\(^1\) and Karim Saadaoui\(^1\) and Mohamed Benrejeb\(^1\)

\(^1\) Department of Electric Engineering, National School of Engineering of Tunis  
testouri_sana@yahoo.fr

Abstract

In this paper, the AQM controller of a first-order controller’s type is proposed. The model of TCP/AQM is described by a second-order system with time delay. An analytical approach to analyze the stability of TCP/AQM Networks is used, based on the D-decomposition method and lemma Kharitonov for quasi-polynomial. The proposed method for design first-order controller is verified and compared with other existing AQM schemes, using NS-2 simulator.

Keywords

Stability, Time delay, First-Order Controller, Active Queue Management

1. Introduction

The congestion-control mechanism becomes indispensable in an over-charged network. TCP (Transmission Control Protocol) has been the basis of congestion control. It adopts the end-to-end window-based flow control to avoid congestion [1]. Recently, a growing interest in designing AQM (Active Queue Management) has been proved an efficient approach to enhance congestion control, indeed, a significant research has been devoted to the use of control theory to develop more efficient AQM [6,7,8]. The goal of AQM is to maintain shorter queuing delay and higher throughput by dropping packets at intermediate nodes. It has therefore attracted attention in the research for transmission control protocol (TCP) of end-to-end congestion control. Random early detection (RED) [2] is the first well known AQM algorithm, which aims to drop packets with a certain probability as a function of the average queue size. Furthermore, it is difficult to obtain adequate values of RED parameters which provide satisfactory performance in terms of overall quality of service (QoS). Central to our approach is utilizing the reformulated AQM schemes as a controller for congestion control. Consequently, feedback control principles appear to be an appropriate tool in the analysis and design of AQM strategies. Using dynamical model TCP/AQM, some P (Proportional), PI (Proportional Integral) have been designed [3], this nonlinear model is linearized at an operating point to address the feedback control nature. Proportional (P) and proportional-integral (PI) AQM controllers were introduced by comparison to RED. A structural nonlinear component of RED was considered, and the describing function approach was applied to obtain a stability criterion for RED and PI controller was proposed, which adaptively adjusted the controller parameters based on network parameters estimation [4], [5]. In order to stabilize the network traffic system and achieve the desired QoS, [3],[6],[7] paid much attention to the setting of controller parameters. Our objective is to obtain the stabilizing...
regions of a first-order controller in order to choose parameters which stabilize the system modeled by a second-order plant with time delay. In earlier works, with a P type AQM controller, in the case of delay-free marking, the system’s equilibrium point is stable for all proportional gains. In a more realistic case of delayed feedback, there exists a boundary for proportional gain to guarantee the closed-loop AQM system stability [8]. When the AQM controller is of PI type, the stabilizing boundary of proportional gain has been given by using a parameter space approach [7]. Our approach consists of determining the stabilizing regions in the parameter space of the first-order controller, then choosing controller’s parameters within these regions. Finally, we illustrate the proposed methodology with an example and simulations experiments implemented by both Matlab and Network Simulator NS-2 for validate our analysis. In this paper, the AQM controller is of first-order controller type is presented. A parametric approach for stabilizing closed-loop for AQM system is proposed based on the D-decomposition method and lemma Kharitonov for quasi-polynomial. Section 2 introduces the TCP/AQM dynamic model. Next, the stabilizing regions in the parameter space of the controller are determined in section 3. Finally, in section 4, simulation results both with Matlab and NS-2 are given.

2. Dynamic Model of an AQM Router

2.1. Fluid-Flow Model of TCP Behaviour

The dynamic model of TCP flows is developed by using a fluid flow model without considering slow start and timeout mechanisms [4] .Based on this system, a type of AQM is constructed, which takes into account delays into the network.

This model is described by the following non-linear differential equations

\[
\begin{align*}
\dot{W}(t) &= \frac{1}{R(t)} - \frac{W(t)}{2R(t)} \left(\frac{1}{R(t)} - \frac{R(t)}{R(t)}\right) \frac{1}{R(t)} - \frac{R(t)}{R(t)} \frac{1}{R(t)} - \frac{R(t)}{R(t)}, \\
\dot{q}(t) &= \frac{w(t)}{R(t)} \left(\frac{1}{R(t)} - \frac{R(t)}{R(t)}\right) - \frac{c(t)}{R(t)} - \frac{T_p}{R(t)}, \\
\dot{R}(t) &= \frac{c(t)}{R(t)} + T_p.
\end{align*}
\]

where \( \dot{W}(t) \) and \( \dot{q}(t) \) denote the time-derivatives of \( W(t) \) and \( q(t) \), respectively. \( W(t) \) denotes the TCP window size, \( q(t) \) denotes the queue length in the router.

\( p(t) \) denotes the probability packet marking/dropping \( (p(t) \in [0,1]) \). \( R(t) \) denotes the round-trip time, \( c(t) \) denotes the link capacity. \( T_p \) denotes the propagation delay. \( N(t) \) denotes the load factor (number of TCP sessions).The first differential equation in (1) describes the TCP window control dynamic and the second equation models the bottleneck queue length. The queue length and window size are positive, bounded quantities, i.e., \( q \in [0,\bar{q}], W \in [0,\bar{W}] \) window size, respectively. Also, the marketing probability \( p \) takes value only in \( [0,1] \).

In this model, the congestion window \( W(t) \) increase linearly if no packet loss is detected; otherwise it halves.
2.2. Linearization

Although an AQM router is a non-linear system, in order to analyze certain types of properties and design controllers we need a linear model which is presented in this sub-section. To linearize (1), we first assume that the number of TCP sessions and link capacity are constant, i.e., \( N(t) = N \), \( C(t) = C \).

Taking \( [w, q] \) as the state and \( p \) as input, the operating point \( (w_0, q_0, p_0) \) is then defined by \( \dot{w} = 0 \) and \( \dot{q} = 0 \) so that

\[
\begin{align*}
\dot{w} &= 0 \Rightarrow w_c + p_0 = 2, \\
\dot{q} &= 0 \Rightarrow W_0 = \frac{RC}{N}, \\
R_c &= \frac{q_0}{C} + T_p.
\end{align*}
\]

(2)

We linearize (1) about the operating point to obtain

\[
\begin{align*}
\delta \dot{w}(t) &= -\frac{N}{RC} (\delta w(t) + \delta W(t - R_c)) - \frac{RC^2}{2N^2} \delta p(t - R_c), \\
\delta \dot{q}(t) &= \frac{N}{R_0} \delta W(t) - \frac{1}{R_0} \delta q(t).
\end{align*}
\]

(3)

where \( \delta w = w - w_0 \), \( \delta q = q - q_0 \), \( \delta p = p - p_0 \) represent the perturbed variables around the operating point.

For typical network conditions [4],

\[
\frac{N}{R_0^2} = \frac{1}{W_0 R_0} \ll \frac{1}{R_0}
\]

\[
\begin{align*}
\delta \dot{w}(1) &= -\frac{2N}{R_1 C} \delta w(1) - \frac{R_1 C}{2N} \delta p(1 - R_1), \\
\delta \dot{q}(1) &= \frac{N}{R_1} \delta w(1) - \frac{1}{R_1} \delta q(1).
\end{align*}
\]

(4)

We just consider the following dynamics. Performing Laplace transform on (4), we have:

\[
\begin{align*}
G_{TCP}(s) &= \frac{R_0 C^2}{s + \frac{2N}{R_0}} e^{-sR_0}, \\
G_{queue}(s) &= \frac{R_0}{s + \frac{1}{R_0}}
\end{align*}
\]

(5)

where \( G_{TCP}(s) \) is the TCP’S dynamic, \( G_{queue}(s) \) is the queue’s dynamic.
3. STABILIZING FIRST-ORDER CONTROLLERS

We consider the closed-loop AQM system with $C(s)$ the transfer function of the controller, and $G_0$ the transfer function of the plant dynamic. This model presents the dynamics of the queue and the congestion window as a time delay system.

Indeed, taking into account this characteristic, we expect to reflect the TCP/AQM behaviour in control congestion.

![Block diagram of the AQM system](image)

$$G(s) = G_T C_P(s) G_q e = \frac{B}{Q(s)} e^{-R_0 s}$$

(6)

where $B = \frac{C^2}{2 NR}, Q(s) = (s + \frac{2N}{R^2 C})(s + \frac{1}{R_0})$.

As the network parameters $\{N,C,R_0\}$ are positive, where $R_0 > 0$ is the time delay, and $C(s)$ is the first order controller having the form

$$C(s) = \frac{\alpha_2 s + \alpha_3}{s + \alpha_1}$$

(7)

The closed-loop AQM system is a second-order system with time delay, whose characteristic equation is

$$1 + C(s)G(s) = 0$$

(8)

which leads to the following characteristic quasi-polynomial

$$V^*(s) = (s + \alpha_1)Q(s) + B(\alpha_2 s + \alpha_3)e^{-sR_0}$$

(9)

Multiplying both sides of (9) by $e^{R_0 s}$ yields

$$V(s) = (s + \alpha_1)Q(s)e^{R_0 s} + B(\alpha_2 s + \alpha_3)$$

(10)

As $e^{-s}$ does not have any finite zeros [10], the zeros of $V^*(s)$ are identical to those of $V(s)$. The characteristic quasi-polynomial $V^*(s)$ of the closed-loop AQM system is stable if and only the zeros of $V(s)$, are in open left half plane (LHP). Then, $V(s)$ is defined as Hurwitz or stable. Determining stabilizing controller parameters $\{\alpha_1, \alpha_2, \alpha_3\}$ will be done in the next section.
3.1. Computing the admissible values of $\alpha_1$

First, the admissible values of $\alpha_1$ are calculated, the following Lemma, gives a condition for the stability of $V(s)$, where $V(s)$ denotes the derivative of $V(s)$.

**Lemma 1.** [12] Consider the quasi-polynomial

$$\Delta(s) = \sum_{i=0}^{n} \sum_{j=1}^{m} h_{ij} s^{i-j} e^{j\tau}$$

such that $\tau_1 < \tau_2 < \ldots < \tau_r$ with main term $h_0$ and $\tau_1 + r \tau > 0$. If $V(s)$ is stable then $V'(s)$ is also a stable quasi-polynomial.

Now, using Lemma 1, if $V(s)$ is stable then $V'(s)$ is also a stable quasi-polynomial, where

$$V(s) = \left[ (R(s + \alpha_1) + 1)Q(s) + (s + \alpha_1)Q'(s) \right] e^{R_s} + B\alpha_2$$

(12)

Note that only two parameters $(\alpha_1, \alpha_2)$ appear in the expression of $V'(s)$. Repeating the same reasoning once more: if $V'(s)$ is stable, then $V''(s)$ is also stable, $V''(s)$ given by

$$\Delta'(s, \alpha_1) = \left[ sQ'(s) + (2R_s^2 + 2)Q'(s) + (R_s^3 + 2R_s)Q(s) \right]$$

$$+ \alpha_1 \left[ Q'(s) + 2R_s Q(s) + R_s^2 Q(s) \right] e^{\alpha_r}$$

(13)

Note that only one controller parameter $\alpha_1$ appears in the expression of $V(s)$, moreover the term $e^{\alpha_r}$ has no finite roots, so stability of $V(s)$ is equivalent to stability in expression (13) without term $e^{\alpha_r}$. To sum up, using the condition of Lemma 1, we can get an admissible stabilizing range for the controller parameter $\alpha_1$ [11].

3.2. Stabilizing Regions in the Plane of $(\alpha_2, \alpha_3)$

Once the admissible value of $\alpha_1$ is fixed within the range determined by the above procedure, the set of the stabilizing regions in the plane of the parameters $(\alpha_2, \alpha_3)$ are determined by using the D-decomposition method [13], which is described in what follows

- Substitute $s$ by $j\omega$ and set the real and imaginary parts of $V(s)$ to zero.
- The $(\alpha_2, \alpha_3)$ plane can be partitioned into root-invariant regions.
- Stabilizing regions in the $(\alpha_2, \alpha_3)$ plane can be determined by choosing a point inside the root-invariant regions and applying classical methods.

Evaluating the characteristic function at the imaginary axis is equivalent to replacing $s$ by $j\omega$, $\omega \geq 0$ in (10), which gives

$$V(j\omega) = \left[ (j\omega + \alpha_1) [R(\omega) + jI(\omega)] \right] \left[ \cos(\omega \theta) + j \sin(\omega \theta) \right] + B(\alpha, j\omega + \alpha_1)$$

(14)
where \( R(\omega) \) and \( I(\omega) \) are the real and the imaginary part of \( \mathbf{Q}_j(\omega) \).

Three cases will be investigated:

Case 1. Setting \( \omega = 0 \), this leads to the following equation:

\[
\alpha_i = -\frac{1}{B} R(0) \alpha_i
\]  

(15)

Case 2. For \( \omega > 0 \), the following pair of \( (\alpha_2, \alpha_3) \) can be calculated for each fixed value of \( \alpha_1 \):

\[
\begin{align*}
\alpha_2 &= \frac{1}{B} \left[ I(\omega) - \alpha_1 \frac{R(\omega)}{\omega} \right] \sin(\text{Lo}) - \left[ R(\omega) + \alpha_1 \frac{I(\omega)}{\omega} \right] \cos(\text{Lo}) \\
\alpha_3 &= \frac{1}{B} \left[ \omega I(\omega) - \alpha_1 R(\omega) \right] \cos(\text{Lo}) + \left[ \alpha_1 R(\omega) + \omega I(\omega) \right] \sin(\text{Lo})
\end{align*}
\]  

(16)

By sweeping over all values \( \omega > 0 \), the \( (\alpha_2, \alpha_3) \) plane can be partitioned into root–invariant regions, therefore stabilizing regions can be determined. Repeating the same steps for the admissible values of \( \alpha_1 \) leads to the determination of all stabilizing \( (\alpha_1, \alpha_2, \alpha_3) \) values.

3.3. Control Action

AQM controller marks packets with a probability. The marking probability is calculated according to the first-order controller and it is a function of the difference between the instantaneous queue length and the desired queue length to which we want to regulate. \( \delta q \) is given by \( \delta q = q - q_0 \), and we assume \( \mu = 0 \), which makes \( \delta p = p \). In practice the sampled queue system needs the discrete form of controller, after substituting a series of sampling time \( kT \) for continues time \( t \) in equation (7), we obtain the increment expression of the form:

\[
p(kT) = a \delta_q (kT) - b \delta_q ((k-1)T) + p((k-1)T) * \alpha
\]  

(17)

For implement in ns2, the first order controller we have need convert the increment expression by pseudo code with a sampling frequency of 160 Hz [8].

\[
\begin{align*}
p &= a * (q - q_{-1}) - b * (q_{-1} - q_{-2}) + p_{-1} * \alpha \\
p_{-1} &= p \\
q_{-1} &= q \\
\alpha &= e^{-\epsilon \cdot t}
\end{align*}
\]  

(18)

4. SIMULATION

4.1. Simulation in Matlab

For determining stabilizing controllers for the system (6), we consider the network parameters \( N = 60, C = 3750 \text{packets/s} \) and \( \mathbf{R}_0 = 0.25s \), and we obtain (19)

\[
P(s) = \frac{117187.5}{(s+0.512)(s+0.4)} e^{-0.25s}
\]  

(19)
Applying the procedure given in the previous section, the admissible range of \( a_1 \) is found to be \( a_1 > -2 \).

Now, we choose the admissible values within the range \( a_1 = 0.5 \times 10^{-5}, 0.6 \times 10^{-5}, 0.7 \times 10^{-5}, 0.8 \times 10^{-5}, 0.9 \times 10^{-5}, 1 \times 10^{-5} \). The stabilizing region in the plane of the remaining of the two parameters \((\alpha_2,\alpha_3)\) derived from equations (15) and (16) is determined.

Figure 2 gives the set stabilizing first-order controller. Setting \( \alpha_1 = 0.6 \times 10^{-5} \), we have two groups of controller parameters. \( (a_{10}, a_{20}, a_{30}) \) locates inside the stabilizing region, \( (\alpha_{10}, \alpha_{21}, \alpha_{31}) \) locates outside the stabilizing region.

Figure 2. Stabilizing regions of controller parameters.

Figure 3 represents the step response of the closed loop system with \( (a_{10}, a_{20}, a_{30}) = (0.6 \times 10^{-5}, 4, 4) \) and \( (a_{10}, a_{21}, a_{31}) = (0.6 \times 10^{-5}, 4, 4) \).

Figure 3. Step responses of the closed-loop system.

We find that \( (a_{10}, a_{20}, a_{30}) \) leads the output convergence, and \( (\alpha_{10}, \alpha_{21}, \alpha_{31}) \) leads the output divergence.

Now, we turn our attention to analyzing the relations between stabilizing boundary of the gain in first-order controller and the network parameters \( [6, 7] \). For \( N \in [50, 300] \), according to the stability criterion given in the previous section, we check the stabilizing boundary of \( a_1 \) gain by (13) and we plot the curve \( \gamma_1 \) shown in Figure 4.(a)
Figure 4. Stabilizing regions for different network parameters. (The stabilizing region is above the curve).

Also, we plot of the curve \( \alpha_{1}(C) \) for \( N = 60, R_0 = 0.25s \) and ranges \( C \in [1500;5000] \) shown the stabilizing region of \( \alpha_1 \) in Figure 4. (b) and for \( N = 60, C = 3750 \) packets/s and \( R_0 \in [1500;5000] \).

From these two curves, we find that stabilizing boundary of increases with \( N \) and decreases with \( C \) and \( R_0 \), respectively.

4.2. Simulation in NS-2

To verify the stabilizing parameters in network environment, we conducted non linear simulation by NS, using the network topology depicted in Figure 5.
We introduced 60 TCP flows and the simulation time is 80 s. \( S_i \) \((i=1,\ldots,n)\) are TCP senders with average packet size 1000 Bytes. \( S_d \) is a FTP sender which has 10 Mbps capacity and 20 ms propagation delay, the traffic scenario. The only bottleneck link lies between Router \( R_A \) and \( R_B \), which has 15Mbps capacity and 5ms propagation delay. Router \( R_B \) uses the First-order controller, others use the Drop Tail. The sampling is \( 160 \text{ Hz} \). The buffer size is 800 packets and the desired queue length is 200 packets. As one of the controller parameters inside the stabilizing region, \( (\alpha_1, \alpha_2, \alpha_3) \) regulates the queue length to the desired 200 packets. We will compare the dynamic response of the first–order controller with that Hollot’s PI controller in [5]. First, we plot the queue lengths for different first-order controller parameters, inside and outside the stabilizing region.

Figure 6 shows the instantaneous queue length, \( (\alpha_1, \alpha_2, \alpha_3) \) regulates the queue length to the desired 200 packets pictured in Figure 6. (a).

Figure (b) shows significant oscillation in the case \( (\alpha_{10}, \alpha_{21}, \alpha_{31}) \) outside the region, when it is unstable.

The performance of first-order controller is compared by PI controller presented in Figure 6. (c).

It clear that the queue length with the first-order controller is fast, more stable and overshoot is negligible than Hollot’s PI controller, the PI controller takes long time to settle the queue length around target.
Figure 6. Instantaneous queue lengths for first-order controller for different parameters and PI controller.

For validate our analysis, the curves of the dropping probability are plotted in Figure 7.

The regulating time of PI controller is almost eight times larger than that of first order controller.

Figure 7. Instantaneous drop probabilities for first-order controller and PI controller.

5. CONCLUSION

This paper discusses the stability characteristic of TCP/AQM networks using control theory. The stabilizing regions of a first-order controller are given. Simulation experiments conducted by Matlab and NS have validated our criterion. In this paper, we have just determining stabilizing a first-order controller for TCP/AQM system with time delay. therefore, in order to achieve better performance criterion (no over-shoot, minimal rise time, Steady state error = 0), we will propose in our futur work, a method for determining the optimal parameters of first-order controller.
REFERENCES