On using Lyapunov function to design optimal controller for AQM routers supporting TCP flows

ZHANG Peng¹, YE Chengqing¹, MA Xueying¹², LI Xin¹, CHENG Yanhua¹

¹Dept. of Computer Science, Zhejiang University, Hangzhou 310027, P. R. China
pzhang1974@163.com

²Dept. of Information Management, Zhejiang Institute of Finance & Economics, Hangzhou, 310012, P. R. China
sbkmxy1@sina.com

Abstract: Recently it has shown that the active queue management schemes implemented in the routers of communication networks supporting transmission control protocol (TCP) flows can be modeled as a feedback control system. In this paper, based on Lyapunov function we developed an optimal controller to improve active queue management (AQM) router’s stability and response time, which are often in conflict with each other in system performance. Using ns-2 simulations, it is shown that optimal controller outperform PI controller significantly.

Keywords: TCP flows; AQM; Lyapunov function; optimal control.

1. INTRODUCTION

Internet congestion occurs when the aggregated demand for a resource (e.g., link bandwidth) exceeds the available capacity of the resource. Resulting effects from such congestion include long delays in data delivery, wasted resources due to lost or dropped packets, and even possible congestion collapse [1]. Therefore it is very necessary to avoid or control network congestion. The active queue management (AQM) scheme implemented in routers is designed for this purpose [2]. AQM can maintain stable queuing and higher throughput by purposefully dropping packets at intermediate nodes.

Several AQM Schemes have been studied in recent literatures to provide early congestion notification to users. Among these AQM mechanisms, Random Early Detection (RED), which was originally proposed to achieve fairness among sources with different burst-ness and to control queue length, is probably the best known [3]. The performance of RED has been evaluated through simulations and experiments in real networks [4]. And it is well known that RED is quite sensitive to parameter settings. S.Floyd, the designer of RED, and other researchers have given several guidelines in parameter settings, such as “gentle_RED”, ARED, SRED [13,14,15] etc. Unfortunately as its performance is improved, RED becomes more and more complex to implement.

Another approach for AQM design is through control theory. There are many kinds of controllers have been proposed, such as PI, PD, PID controller and so on [5,6,7]. Among
these schemes, proportional-integral (PI) controller designed by C.V.Hollot is the most representative one, and others can be looked upon as variations of it. This controller is expected better responsiveness by calculating packet drop probability based on the current queue length instead of the average queue length [5,8]. The results of various simulations show that PI outperforms RED in regulating steady state queue length to a desired reference value with changing levels of congestion. Although this controller improves the performance of AQM, there are many experiments and simulations and calculations dedicate that PI is sluggish system and the regulating time is too long [7]. In fact stability and response time are often in conflict with each other in system performance. It is an intractable problem to find tradeoff between them.

To solve this problem, we apply time optimal control theory to design a novel controller that can decrease the responsive time and improve the stability of TCP/AQM congestion control in this paper. Different from PI and variations controller of it, time optimal controller focus on finding an admissible control policy which transfers control variable from current value to desired reference value in minimum amount of time. Moreover, we design this time optimal controller through Lyapunov stability theory, so we can expect this algorithm has better performance of response time and stability than PI controller.

The remainder of the paper is organized as follows. In Section 2, we present a novel algorithm called time optimal controller (TOC) and give a theoretic law of choosing the parameters to achieve the system stability. The simulation results demonstrate that the better network performance of TOC can be achieved compared with PI scheme in different network conditions by ns-2 in Section 3. Finally, we conclude our work in section 4.

2. DESIGNING TIME OPTIMAL CONTROLLER

The simplified dynamic model of TCP behavior [9] has been shown in the following equations:

\[
\begin{align*}
\frac{dW}{dt} &= \frac{1}{R(q)} \left( W W(t-\tau) - W R(q) \right) p(x(t-\tau)) \\
\frac{dq(t)}{dt} &= \frac{W(t)}{R(t)} N(t) - C \\
\end{align*}
\]

(1)

where

- \(W\) □ expected TCP window size (packets);
- \(q\) □ expected queue length (packets);
- \(R\) □ round-trip time = \(\frac{q}{C} + T_p\) (secs);
- \(C\) □ link capacity (packets/sec);
- \(T_p\) □ propagation delay (secs);
load factor (number of TCP sessions);

probability of packet mark/drop.

In literature [8], linearized equation of (1) is proposed as the following:

\[
\begin{align*}
\delta W(t) &= -\frac{2N}{R_0^2C} \delta W(t) - \frac{R_0 C^2}{2N^2} \delta p(t - R) \\
\delta q(t) &= \frac{N}{R_0} \delta W(t) - \frac{1}{R_0} \delta q(t)
\end{align*}
\] (2)

Where \( \dot{x} \) denotes the time-derivative of \( x \)

\( \delta W \) denoting \( W - W_o \)

\( \delta q \) denoting \( q - q_o \)

\( \delta p \) denoting \( p - p_o \)

Here, operating point \( (W_o, q_o, p_o) \) is defined by \( \dot{W} = 0 \) and \( \dot{q} = 0 \) [8]

\[
W_o = \frac{R_o C}{N}
\]

\[
R_o = q_o + T_p
\]

let \( \alpha = \frac{1}{R_0} \), \( \beta = \frac{2N}{R_0^2C} \), \( \gamma = \frac{N}{R_0} \), \( \eta = -\frac{R_0 C^2}{2N^2} \), \( x_1 = \delta q \), \( x_2 = \delta W \), \( u = \delta p(t - R_0) \), we get following matrix equation:

\[
\begin{pmatrix}
\dot{x}_1 \\
\dot{x}_2
\end{pmatrix} = 
\begin{pmatrix}
-\alpha & \gamma \\
0 & -\beta
\end{pmatrix}
\begin{pmatrix}
x_1 \\
x_2
\end{pmatrix} + \begin{pmatrix}
0 \\
\eta
\end{pmatrix} u
\] (3)

Let us consider the following Lyapunov function:

\[
V(X) = \begin{pmatrix} x_1 & x_2 \end{pmatrix} \begin{pmatrix} 1 & b \\ b & a \end{pmatrix} \begin{pmatrix} x_1 \\
x_2 \end{pmatrix} \quad (a-b^2 > 0)
\]

It is easy to verify that \( V(X) \) satisfies the necessary conditions to enable it to be used as a Lyapunov function. According to the Lyapunov stability theory, if there exists appropriate parameters to make \( \dot{V}(X) < 0 \), the equilibrium state of dynamic system is asymptotically
stable.

\[
\dot{V}(X) = - \left( x_1 \quad x_2 \right) \begin{pmatrix} 2\alpha & b(\alpha + \beta) - \gamma \\ b(\alpha + \beta) - \gamma & 2(\alpha \beta - \beta \gamma) \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + 2(bx_1 + ax_2)\eta u
\]  \hspace{1em} (4)

If the following inequation is hold

\[
a > \frac{(b(\alpha + \beta) - \gamma)^2 + b\gamma}{4\alpha \beta}
\]  \hspace{1em} (5)

the first item of right hand of (4) is negative-definite, that means:

\[
- \left( x_1 \quad x_2 \right) \begin{pmatrix} 2\alpha & b(\alpha + \beta) - \gamma \\ b(\alpha + \beta) - \gamma & 2(\alpha \beta - \beta \gamma) \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} < 0
\]

Now considering the second item of right hand of (4),

\[
2(bx_1(t) + ax_2(t))\eta \delta p(t - R_0)
\]  \hspace{1em} (6)

The \( x_1 \) and \( x_2 \) can be rewritten as following:

\[
x_1(t) = x_1(t - R_0) + \int_{t-R_0}^{t} \dot{x}_1 \, d\tau
\]

\[
x_2(t) = x_2(t - R_0) + \int_{t-R_0}^{t} \dot{x}_2 \, d\tau.
\]

According to TCP window-based Additive Increase and Multiplicative Decrease (AIMD) control scheme, TCP sending window’s size does not change its value until next RTT (\( R_0 \)).

So we approximately have \( \dot{x}_2(t-R_0) = \frac{\Delta x_2(t-R_0)}{R_0} \approx \frac{x_2(t)-x_2(t-R_0)}{R_0} \) in this model, \( i.e. \)

\[
x_2(t) = x_2(t - R_0) + \dot{x}_2(t - R_0)R_0.
\]

According to equation (1), we know that \( \dot{x}_1 \) keeps a constant value within RTT interval as long as \( \delta W(t) \) does not change its value. So the following equation

\[
x_1(t) = x_1(t - R_0) + \dot{x}_1(t - R_0)R_0
\]

is hold.

Furthermore, according to equation (1), we have

\[
\frac{R_0 \dot{x}_1(t-R_0)}{N} = x_2(t-R_0);
\]

And according to equations (2), we have

\[
\dot{x}_2(t-R_0) = -\frac{2N}{RC} x_2(t-R_0) - \frac{RC^2}{2N^2} \delta p(t-2R_0).
\]

Rewriting (6) as following:
If the following inequation is hold

\[ 2(bx_1(t-R_0)+x_2(t-R_0)(bN+a\frac{R_0C-2N}{R_0C})-aR_0^2C^2 \frac{\delta p(t-R_0)}{2N^2})\eta \delta p(t-R_0) \]

we can see that \( (6) < 0 \). According to Lyapunov stability theory, this means the system is asymptotically stable. So according to inequation (8) the control policy can be choose as following:

\[
\delta p(t) = \begin{cases} 
> 0, & \eta(bx_1(t)+x_2(t)(bN+a\frac{R_0C-2N}{R_0C})-aR_0^2C^2 \frac{\delta p(t-R_0)}{2N^2}) < 0 \\
\leq 0, & \eta(bx_1(t)+x_2(t)(bN+a\frac{R_0C-2N}{R_0C})-aR_0^2C^2 \frac{\delta p(t-R_0)}{2N^2}) > 0
\end{cases}
\]

Note here \( x_2 = \delta W = W - \frac{RC}{N}, \frac{NW}{R} = \text{flow_rate} \), that means \( x_2 = \delta W = \frac{R}{N} (\text{flow_rate} - C) \).

For \( \delta p(t) = p(t) - p_0 \) (\( 0 < p_0, \ p(t) < 1 \)), we have

\[
\begin{cases} 
\delta p(t) < 0, & p(t) = 0 \\
\delta p(t) > 0, & p(t) = 1
\end{cases}
\]

This leads to the following results:

\[
p(t) = \begin{cases} 
1, & bx_1(t)+x_2(t)(bN+a\frac{R_0C-2N}{R_0C})-aR_0^2C^2 \frac{\delta p(t-R_0)}{2N^2} > 0 \\
0, & bx_1(t)+x_2(t)(bN+a\frac{R_0C-2N}{R_0C})-aR_0^2C^2 \frac{\delta p(t-R_0)}{2N^2} < 0
\end{cases}
\]

or to say

\[
p(t) = \begin{cases} 
1, & b \delta q(t) + \frac{R_0}{N} (bN+a\frac{R_0C-2N}{R_0C})(\text{flow_rate} - C) - aR_0^2C^2 \frac{\delta p(t-R_0)}{2N^2} > 0 \\
0, & b \delta q(t) + \frac{R_0}{N} (bN+a\frac{R_0C-2N}{R_0C})(\text{flow_rate} - C) - aR_0^2C^2 \frac{\delta p(t-R_0)}{2N^2} < 0
\end{cases}
\]
Letting \( a_0 = \frac{R_0}{N} (bN + a(R_0C - 2N)) \), \( a_i = \frac{aR_0^2C^2}{2N^2} \), and using algorithm in literature \([10]\) to compute \( p_0 \), we can describe our control algorithm as the following pseudo-code in Fig.1.

To obtain values of \( \beta \), \( a_0, a_i \), we set \( b \) at 30, according to (5), value of \( a \) can be computed as following:

\[
\frac{(b(\alpha + \beta) - \gamma)^2}{4\alpha} + b\gamma
\]

Then we can compute out \( a_0, a_i \). Note here \( R_0 = \frac{q_0}{C} + \text{average}_\text{propagation}_\text{delay} \).

Now we explain why this scheme can be looked upon as time optimal controller (TOC). According to control theory, \( |\dot{V}(X,u)| \), the absolute value of Lyapunov function’s time derivative along the trajectories of the system, approximately reflects how fast this system converges to equilibrium state. If we choose \( \dot{V}(X,u) = \min_{u} \dot{V}(X,u) \), system will achieve to the desired reference value with the shortest regulating time \([11,12]\).

3. PERFORMANCE EVALUATION

We evaluate the effectiveness and performance of the TOC algorithm by simulations using...
ns-2.1b9 simulator. The network topology used in the simulation is the same one used in [7]. It is a simple dumbbell topology based on a single common bottleneck link of 10 Mb/s capacity with many identical, long-lived and saturated TCP/Reno flows. The round-trip propagation delay is randomly chosen from (1ms, 20ms).

In the first experiment, bottleneck link is 10Mbps, target queue length \(q_0\) is 80 packets, and buffer size for AQM is 300 packets, average packet size is 1040 Bytes. There are 100 FTP connects all together.

The performance of TOC is compared with PI controller. The parameters used are as follows: for PI scheme, the parameters are set as same as literature [5], that is \(a=1.822e-5\) and \(b=1.816e-5\) respectively. For TOC, the parameters are set at \(b=30, a_0=1.147, a_1=670.4188\).

Figure 2 and Figure 3 demonstrate the dynamic change of the real queue length of the PI and TOC algorithm respectively. It can be seen that, TOC shows higher performance than PI.

TOC settle down reference point within 3 seconds, while PI controller takes more than 7 seconds to settle down the reference point. Note how the overshoot and oscillation essentially have been eliminated when we use optimal controller. All together, the system has become more stable than using PI controller.

Experiment 2: in this experiment the bottleneck link is 10Mbps. There are 150 FTP connects, and \(q_0\) is 80 packets, buffer size is 300 packets.
PI controller’s parameter setting is same as experiment 1, and parameters of TOC is set at $b = 30, a_0 = 0.7050, a_1 = 308.0166$.

Fig 4 and Fig 5 shows the experiment result. The queue length evolution of PI controller shows it varies acutely. On the other hand TOC controller regulate the queue length to the equilibrium point more quickly and more stability.

Experiment 3: to evaluate the performance of controller at different target queue length and $R_0$ settings, in this experiment we set $q_0$ to 150, 100, 50 each other with 100 FTP connections and 150 FTP connections respectively. Note here different target queue length will result in different $R_0$.

Fig.6 shows this experiment’s result. Left figure plots queue evolution with 100 FTP connections while right one plots the evolution with 150 FTP connections. It can be seen that all queues length are stabilized at target value with short response time. This proves that TOC can force system to achieve the desired reference value fast, and reduce the overshoot and oscillation evidently.

Experiment 4: in last experiment we test the robust and capable of resisting mismatches of model. At 0 second, there are 80 FTP connections to start, at 10 second, another 80 FTP connections start, at 20 second the other 80 FTP connections start, and at 30 second the first group stops, at 45 second the second group stops. The target queue length is 100 packets.

We compare performance of TOC with PI controller. Parameters of these two controllers are set as same as experiment 1.

The queue evolution of PI controller is plotted in Fig.7, while that of TOC is plotted in Fig.8. Evidently, the Fig.7 shows that queue oscillate with great magnitude, while TOC algorithm operates in a relatively stable state. We can draw a conclusion from this experiment that TOC is more robust than PI algorithm and it can resist the mismatches of model very well.
4. CONCLUSION

In this paper we focus on improving TCP/AQM performance of stability and response time in router. In fact stability and response time are often in conflict with each other in system performance. To find a tradeoff between them, we propose to regulate system with time optimal controller (TOC) that is designed through Lyapunov stability theory. We took a complete comparison with PI controller under various scenarios. The extensive simulation results by ns-2 demonstrate that the integrated performance of optimal controller is superior to that of PI controller. The TOC is responsive, stable and robust.

REFERENCES

IEEE International Symposium on, 30 June-3 July 2003 Pages:985 - 990 vol.2


