Analysis of the virtual rate control algorithm in TCP networks

Eun-Chan Park, Hyuk Lim, Kyung-Joon Park, and Chong-Ho Choi
School of Electrical Engineering and Computer Science
Seoul National University, Seoul, Korea
{ecpark, hyuklim, kjpark, chchoi}@csl.snu.ac.kr

Abstract— **The virtual rate control (VRC) algorithm has been proposed for active queue management (AQM) in TCP networks. This algorithm uses an adaptive rate control instead of queue length control in order to respond quickly to traffic change with high utilization and small loss. By introducing the notion of *virtual target rate*, the VRC algorithm can maintain an input rate around the target rate, while attempting to regulate the queue length. In this paper, we analyze the stability of the VRC algorithm in a linearized model. From the results of our analysis, we provide a design guideline for the system to remain stable. We show the validity of our analysis and the effectiveness of the VRC algorithm compared to RED, PI, REM and AVQ algorithms through *ns-2* simulations.**

I. Introduction

Considerable research has been undertaken on active queue management (AQM) for congestion control in TCP networks. The Random Early Detection (RED) gateways and many other algorithms have been proposed for AQM [1-9].

One of the most prevalent AQM algorithms is RED [1], which detects congestion using an exponentially weighted moving average of the queue length, $\bar{q}$, and drops or marks packets proportional to $\bar{q}$ at a router buffer before the buffer overflows. RED can prevent global synchronization, reduce packet loss rates, and minimize bias against bursty sources. It has been shown that the system's equilibrium point is stable for proportional marking schemes like RED [10]. However, getting appropriate values of RED parameters that show good performance under different congestion scenarios remains an inexact science [11]. It was also pointed out that the average queueing delay and throughput are sensitive to traffic load and control parameters.

To overcome these shortcomings of RED and to design a robust controller for AQM, the proportional-integral (PI) control algorithm [4] and Random Exponential Marking (REM) [3] has been recently proposed. The PI algorithm is proposed based on classical control system techniques and REM is proposed based on the approach from an optimization standpoint. The PI algorithm marks packets with probabilities proportional to the difference between the current instantaneous queue length and the target length and to its accumulation (time integral). While RED uses average queue length for congestion control, the PI algorithm uses instantaneous queue length. REM aims to stabilize both the input rate and queue length (" match rate, clear buffer") regardless of the number of users sharing the link. In order to achieve this, REM introduces a variable, *price*, as a congestion measure: REM decouples congestion measure from performance measure such as loss, queue length. The price is updated using the rate mismatch (i.e., the difference between the input rate and the output link capacity) and the queue mismatch (i.e., the difference between the queue length and its target value). It has been proven that the input rate converges to the link capacity and the queue length approaches its target value (zero) [12], [13]. However, when we consider the update rule of the price, REM is similar to PI algorithm, i.e., the control law of REM depends on the queue length mismatch and its accumulation (i.e., discrete-time integral) rather than queue length mismatch and rate mismatch, on which the update rule depends. This integral action of REM and PI algorithm can result in slow response to abrupt traffic changes.

Another AQM algorithm, the Adaptive Virtual Queue (AVQ) algorithm [5] utilizes a virtual queue to make the input rate achieve the desired utilization. When the virtual queue overflows, packets in the real queue are dropped or marked. This algorithm basically uses the rate mismatch and regulates utilization instead of the queue length as done in the RED, REM and the PI algorithms.

We have proposed the virtual rate control (VRC) algorithm [14] to regulate the queue length with small variation and to achieve high utilization with small packet loss in different network conditions and dynamic traffic changes. Since the algorithm is based on rate control, i.e., the marking probability is primarily proportional to the queue occupancy rate, the VRC algorithm can achieve a rapid response to traffic fluctuations. The virtual target rate is adopted to maintain the equilibrium input rate around the link capacity and to regulate the queue length. This paper makes the following contributions: (1) we analyze the stability of the VRC algorithm with TCP dynamics from a control-theoretic standpoint, (2) using the result of analysis, we derive the parametric ranges that can make the system remain stable and present simulation results to validate the analysis.

The rest of the paper is organized as follows. Section II introduces the VRC algorithm briefly. We analyze the stability of the VRC algorithm using the linearized model of TCP dynamics in Section III. In Section IV, we present simulation results to verify the validity of the stability analysis and compare the performances of the VRC algorithm with RED, PI, REM, and AVQ. Our conclusions follow in Section V.

II. Virtual Rate Control Algorithm

To maximize link utilization and regulate the queue length effectively, it is desired that the aggregate input rate is kept equal to the output link capacity. An excess of the input rate over the output link capacity should be quickly compensated for before it affects the queue length and leads to buffer overflow, which can result in low utilization. Furthermore, the difference between input rate and output link capacity results in the variation

0-7803-7632-3/02/$17.00 ©2002 IEEE

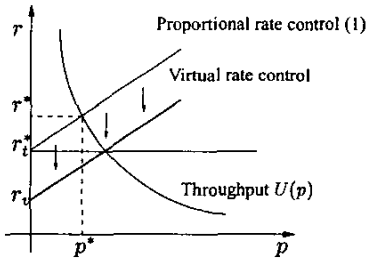

Fig. 1. Equilibrium points for proportional control and virtual rate control

of the queue length and this variation is related to delay jitter. The rate control can regulate the queue length and reduce delay jitter. Instead of queue control, consider a proportional rate control which can keep the input rate around a given target rate as follows:

$$p(t) = [\alpha(r(t) - r_t(t))]^+, \alpha > 0. \quad (1)$$

Here $p(t)$ is the marking probability, $r(t)$ is the aggregate input rate of the queue, $r_t(t)$ is the target rate, and $[\cdot]^+ = \max(\min(\cdot, 1), 0)$. One can simply take the target rate $r_t(t)$ as the link capacity C to maximize the throughput. However, instead of a constant target rate a modified target rate is adopted in order to keep the queue length $q(t)$ around the target queue length q_t. The target rate is set as the sum of the link capacity and the difference between q_t and $q(t)$.

$$r_t(t) = C + \gamma(q_t - q(t)), \gamma > 0. \quad (2)$$

If the queue length becomes smaller than its target, there is more room to accommodate packets and target rate increases. Otherwise, if the queue length grows and becomes larger than its target, target rate decreases. By rewriting (1) with (2), the marking probability becomes

$$p(t) = \left[\alpha((r(t) - C) + \gamma(q(t) - q_t))\right]^+. \quad (3)$$

The marking probability (3) is a weighted sum of the rate difference $(r(t) - C)$ and the queue length difference $(q(t) - q_t)$, similar to the price of REM; REM updates the price to minimize the rate mismatch and queue mismatch. However, REM responds slowly to network congestion compared to (3) because the update rule of REM depends on the accumulations of these mismatches, while (3) depends on current mismatches.

However, the rate control (1) and (2) cannot ensure that the input rate converges to the link capacity in equilibrium.

To explain this discrepancy, the overall steady-state TCP behavior is considered using a graphical method. Fig. 1 illustrates the equilibrium of the marking probability and the input rate when the rate control (1) is applied for congestion control. In Fig. 1, the throughput $U(p)$ is assumed to be a strictly decreasing function of p, i.e., the throughput decreases as the marking probability increases. The equilibrium point (p^*, r^*) is at the intersection of (1) and $U(p)$. In Fig. 1, the equilibrium input rate r^* is greater than the equilibrium target rate r_t^*, and consequently there is always a rate error at equilibrium.

To compensate for this rate error, the concept of a *virtual target rate* is introduced. Instead of using $r_t(t)$ in (1), the virtual target rate $r_v(t)$ is adopted to match r^* with r_t^*.

$$p(t) = [\alpha(r(t) - r_v(t))]^+ \quad (4)$$

The virtual target rate $r_v(t)$ is updated to minimize the difference between $r(t)$ and $r_t(t)$ as follows:

$$\begin{aligned} r_v(t) &= r_t(t) - \Delta r_v(t), \ t = nT_s \\ \Delta r_v(t + T_s) &= \Delta r_v(t) + \beta T_s(r(t) - r_t(t)), \ \beta > 0 \end{aligned} \quad (5)$$

where T_s is the sampling interval and $r(t)$ can be estimated using the algorithm in [15]. The complete algorithm can be implemented as the following pseudo code.

```
At every sampling instant
1. /* calculate the target rate r_t */
2. r_t ← C + γ * ( q_t − q )
3. /* calculate the virtual target rate r_v */
4. Δr_v ← Δr_v + β * T_s * ( r − r_t )
5. r_v ← r_t − Δr_v
6. /* calculate the marking probability p */
7. p ← α * ( r − r_v )
```

To show that the input rate converges to the target rate with (4) and (5), consider the rate error $e_r(t) = r(t) - r_t^*$ in equilibrium. For the discrete-time, $e[n] = e(nT_s)$ decreases with time, i.e.,

$$\frac{e[n+1]}{e[n]} = \frac{1 - \alpha\Delta U(p)}{1 - \alpha(1+\beta)\Delta U(p)} < 1 \quad (6)$$

where

$$\Delta U(p) = \frac{U(p[n+1]) - U(p[n])}{p[n+1] - p[n]} < 0. \quad (7)$$

Here, $\Delta U(p)$ is a linear approximation of the slope of $U(p)$. Since $U(p)$ is assumed to be a strictly decreasing function, (7) holds and consequently the ratio $e[n+1]/e[n]$ becomes smaller than one, i.e., $e[n]$ goes to zero as $t \to \infty$ and $r(t)$ converges to r_t^*.

III. Stability Analysis

In this section, we will analyze the stability of the VRC algorithm combined with TCP dynamics. We adopt the fluid-based TCP dynamic model [16]. For the sake of simplicity, we ignore the slow-start and time-out mechanism of TCP. The simplified model is as follows:

$$\begin{aligned} \dot{W}(t) &= \frac{1}{R(t)} - \frac{W(t)W(t - R(t))}{aR(t - R(t))} p(t - R(t)), \\ \dot{q}(t) &= N(t)\frac{W(t)}{R(t)} - C. \end{aligned} \quad (8)$$

Here, $W(t)$ [packets] and $q(t)$ [packets] are the window size and the queue length, respectively. We assume that the round-trip time $R(t)$ [sec] and the number of TCP connections $N(t)$ are constant, i.e., $R(t) = R$ and $N(t) = N$, and consider the aggregate input rate $r(t)$ [packet/sec] instead of window size $W(t)$ as $r(t) = NW(t)/R$. Now, we can rewrite the differential equations (8) with respect to $r(t)$ and $q(t)$.

$$\dot{r}(t) = \frac{N}{R^2} - \frac{1}{aN} r(t)r(t - R)p(t - R), \quad (9)$$

$$\dot{q}(t) = r(t) - C. \quad (10)$$

The input rate r^* at equilibrium can be obtained by setting $\dot{r}(t) = 0$ in (9). In order to make r^* agree with the steady-state throughput of TCP [17] ,i.e., $\sqrt{(3/2)/p^*}/R$ for a single

source, the constant a is set to 3/2.
On the other hand, the marking probability of VRC can be represented as

$$p(t) = \alpha(r(t) - C) + \alpha(\beta + \gamma)(q(t) - q_t) + \alpha\beta\gamma z(t), \quad (11)$$

$$\dot{z}(t) = q(t) - q_t. \quad (12)$$

from (4), (5) and (10). Defining the error of queue length as $e(t) = q(t) - q_t$, (11) can be written as

$$p(t) = K_D\dot{e}(t) + K_P e(t) + K_I \int_0^t e(\tau)d\tau. \quad (13)$$

Note that this is a Proportional-Integral-Derivative (PID) type [1] control with respect to queue length error with $K_D = \alpha$, $K_P = \alpha(\beta + \gamma)$ and $K_I = \alpha\beta\gamma$. From (11) and (13), it can be observed that β and γ can be extended to have complex conjugate values, keeping K_P and K_I real. Compared to PI and REM algorithm, the proposed algorithm can achieve more rapid response due to the introduction of differentiation action. Hence the proposed algorithm is a generalized form of PI and REM algorithm.
At the equilibrium points of the model (9),(10) and (12),

$$\begin{aligned} r^* &= C, \\ q^* &= q_t, \\ z^* &= \frac{p^*}{K_I} = \frac{a}{K_I}\left(\frac{N}{RC}\right)^2. \end{aligned} \quad (14)$$

From (14), it is apparent that the VRC algorithm makes the input rate and the queue length converge to the link capacity and the target length, respectively.
In order to linearize the non-linear model (9),(10) and (12) at the equilibrium point; we let $r(t) = r^* + \delta r(t)$, $q(t) = q^* + \delta q(t)$, $z(t) = z^* + \delta z(t)$. The linearized model around (r^*, q^*, z^*) can then be written as follows:

$$\begin{aligned} \delta\dot{r}(t) &= -A_1\delta r(t) - A_2\delta r(t - R) \\ &\quad -A_3\delta q(t - R) - A_4\delta z(t - R), \\ \delta\dot{q}(t) &= \delta r(t), \\ \delta\dot{z}(t) &= \delta q(t), \end{aligned} \quad (15)$$

where $A_1 = \tau_1$, $A_2 = \tau_1 + K_D\tau_2$, $A_3 = K_P\tau_2$, $A_4 = K_I\tau_2$, and $\tau_1 = N/R^2C$, $\tau_2 = C^2/aN$.

Theorem: *Assuming that delay-bandwidth product is greater than the number of TCP connections, i.e., $RC > N$, the approximated system of (15) is stable if the positive constants K_D, K_P and K_I satisfy*

$$K_D > \frac{2(2R - \tau_1 - {\tau_1}^2 R)}{(1 + \tau_1 R)\tau_2}, \quad (T1)$$

$$4/\tau_2 < K_P < K_1/\tau_2 R, \quad (T2)$$

$$K_I < K_P\frac{K_2}{1 + \tau_1 R}, \quad (T3)$$

$$K_3 K_P < K_I < K_4 K_P, \quad (T4)$$

where,

$$\begin{aligned} K_1 &= (1 + \tau_1 R)(2\tau_1 + K_D\tau_2), \\ K_2 &= (2\tau_1 + K_D\tau_2) - \frac{K_P\tau_2 R}{1 + \tau_1 R}, \\ K_3, K_4 &= \frac{K_2\left(1 \mp \sqrt{1 - 4/K_P\tau_2}\right)}{2}. \end{aligned}$$

[1] After this paper has been accepted for presentation, we have noticed that the PID control structure has been proposed using a different approach in [9].

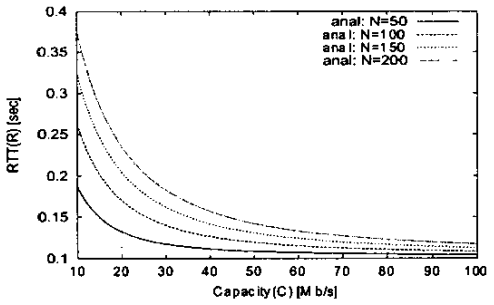

Fig. 2. Stability region of R and C at fixed parameters α, β, and γ: for each N, the region below the curve is stable.

Proof: The details of the proof are given in Appendix. ■

We analyze the effect of feedback delay (i.e. round-trip time), link capacity and the number of flows on the stability of the system. The conditions $(T2)$ and $(T3)$ can be written in terms of R, C, and N as

$$K_P < \frac{1}{R}\left(1 + \frac{N}{RC}\right)\left(K_D + \frac{2a}{C}\left(\frac{N}{RC}\right)^2\right), \quad (T2')$$

$$K_I < K_P\left(K_D\frac{C^2}{aN} + \frac{2aN - C^2R}{aR(RC + N)} - K_P\frac{R^3C^4}{aN(RC + N)^2}\right). \quad (T3')$$

From $(T2')$ and $(T3')$, it is shown that the upper bounds of K_P and K_I decrease as R or C increases and consequently the upper bound of $(T4)$ also decreases.

We calculate and plot the stability region for $K_D = 0.0001$ and $K_P = K_I = 0.001$ in Fig. 2: the region below the curve is stable. From Fig. 2, we can see that the region of system stability decreases as the delay or capacity increases and the number of flows decreases. This agrees well with the results in [5], [18].

IV. Simulation

To verify the validity of our analysis and compare the performance of the VRC algorithm with other AQM algorithms, we conducted simulations using the *ns*-2 network simulator. A simple bottleneck network configuration was implemented consisting of two routers and a number of TCP connections. The routers are connected through a link of capacity 10 Mb/s. The target queue length at the bottleneck router is set to 50 packets, and the allocated buffer size is set to 100 packets. The average packet length is set to 1000 bytes. We use TCP-Reno as the default transport protocol and ECN marking [19] instead of dropping packets.

A. Simulation 1: For the validity of analysis

In this simulation, we verify the validity of our analysis. For fixed K_D=0.0003, N=100, C=10 Mb/s and several R=0.2s, 0.3s, 0.4s, we plot the boundaries of K_P and K_I that satisfy $(T1) - (T4)$ in Fig. 3. Also, via simulation, we measure several sets of K_P and K_I for $R = 0.2s(\square)$, $0.3s(\bigcirc)$, $0.4s(\triangle)$ at which the system remains stable. Here, the set of K_P and K_I is obtained by increasing K_I at fixed K_P until the queue length consistently oscillates from zero to the maximum buffer size.

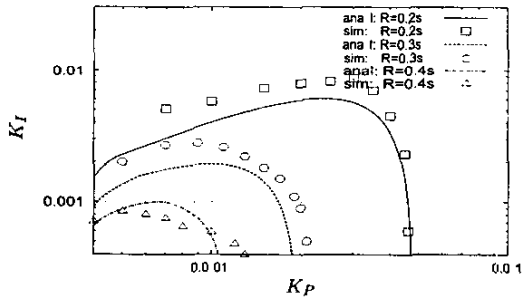

Fig. 3. K_P and K_I regions required for system stability at fixed k_D and several round-trip times: for each R, the region above the curve is unstable and the one below is stable.

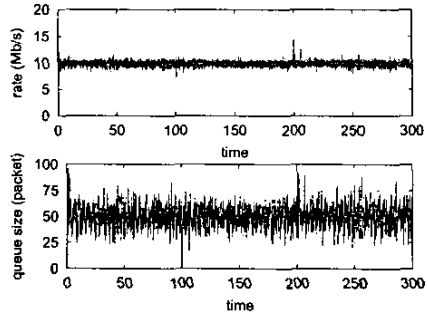

Fig. 4. Input rate and queue length of VRC algorithm in dynamic traffic

As shown in Fig. 3 (log-scale), the region of system stability rapidly decreases as the delay increases. Some quantitative differences are shown between the results of analysis and simulation, which may be attributed to the linearization and Padé approximation. However, the trend of change of both K_P and K_I is similar to the analytic results at fixed R. Also, K_P and K_I reduce significantly as R increase, which agrees with our analysis. Thus, the conditions $(T1) - (T4)$ can be used as an effective guideline for designing parameters.

B. Simulation 2: For dynamic traffic

We focus on the responsiveness of the VRC algorithm to dynamic traffic and on the variations of queue length, and investigate both the input rate and the queue length when some of the TCP connections are off and then on for some period. Initially at $t = 0$ s, 100 TCP connections are established, and then 50 TCP connections are dropped at $t = 100$ s. At $t = 200$ s, another 100 TCP connections are established. The propagation delay between the two routers is set to be uniformly distributed between 25 ms and 75 ms. To satisfy the stability conditions $(T1) - (T4)$ with $N = 100$ and $R = 0.1$ s, we set the parameters of VRC as $\alpha = 0.0003, \beta = 3, \gamma = 5$. (i.e., $K_D = 0.0003, K_P = 0.0024, K_I = 0.0045$)
As shown in Fig. 4, both the input rate and the queue length remain at around the link capacity (10 Mb/s) and the target queue length (50 packets) respectively, even during an abrupt traffic change.

C. Simulation 3: For static traffic

In this simulation, we focus on the queueing delay, the delay jitter, and the utilization and packet loss rate for static traffic, and compare these performances of VRC with those of RED, PI, REM, and AVQ. Since the queueing delay and the delay jitter are closely related to the queue length and its variation, we measure the average queue length and its standard deviation. We also measure the throughput and packet loss rate. Furthermore, in order to test the robustness of the algorithms for traffic load we change the number of TCP connections and perform the simulation repeatedly. The parameters of VRC are the same as in Simulation 2. Others are set to $min_{th} = 20, max_{th} = 80$ for RED, $a = 0.000304, b = 0.00003, w = 100$ for PI, $\phi = 1.002$ for REM, $\gamma = 1.0$ for AVQ and others that are not addressed are set to their default values.
As shown in Fig. 5, the average queue length of RED and AVQ are sensitive to traffic load, while those of REM, PI and VRC do not change appreciably from the target length regardless of the number of connections. The average queue length of RED becomes larger as the number of connections increases. AVQ maintains a small average queue length, however, this depends on the number of connections, because the AVQ does not control the queue length directly.
Fig. 5 also shows that the standard deviation of the VRC is smaller than those of the other algorithms, in most cases. Since the VRC algorithm uses rate information as well as queue length information, it can compensate for the rate variation even before the variation affects the queue length, consequently queue length variations can be reduced.
Next, the throughput and loss rate are investigated. As shown in Fig. 6, the throughput of VRC is higher than other algorithms, and it is relatively constant, while those of the other algorithms drop when the number of connections is small. RED performs poorly compared with the other algorithms. Because AVQ queue length is small, its queue is sometimes empty, leading to lower utilization than PI, REM or VRC. The packet loss rates of PI, AVQ and VRC are nearly the same, and they show better performance than the other algorithms. When the traffic is heavy, the loss rate of RED is substantially higher than those of the other algorithms.
These simulation results show that VRC has a consistent queueing delay, high utilization and a low loss rate regardless of the traffic load.

V. Conclusion

The stability of the VRC algorithm, which is based on a rate control to improve response to changing traffic load with high utilization and small loss and delay, has been analyzed in terms of control parameters. From the results of our analysis, we have presented a design guideline of control parameters for system stability. The guideline has been shown to work well by simulation. The simulation results also have shown the effectiveness of the VRC algorithm regardless of the change of traffic load.

References

[1] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Trans. on Networking*, vol. 1, no. 4, pp. 397–413, 1993.
[2] F. P. Kelly, A. Maulloo, and D. Tan, "Rate control for communication networks: Shadow prices, proportional fairness and stability," *Journal of Operations Research Society*, vol. 49, no. 3, pp. 237–252, 1998.
[3] S. Athuraliya, S. H. Low, V. H. Li, and Q. Yin, "REM: Active queue management," *IEEE Network*, vol. 15, no. 3, pp. 48–53, 2001.

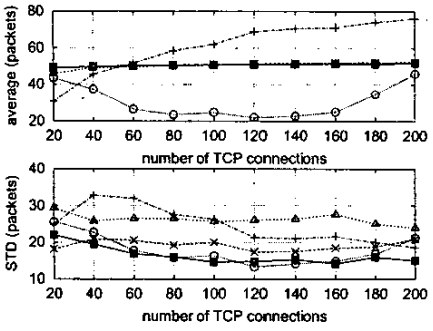

Fig. 5. The average and the standard deviation of the queue length of RED(+), PI(×), REM(△), AVQ(○) and VRC (■)

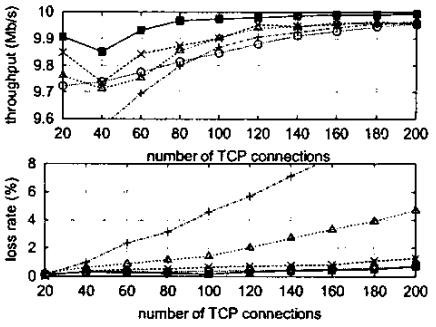

Fig. 6. The throughput and the packet loss rate of RED(+), PI(×), REM(△), AVQ(○) and VRC (■)

[4] C. V. Hollot, V. Misra, D. Towsley, and W. Gong, "On designing improved controllers for AQM routers supporting TCP flows," in *Proceedings of IEEE INFOCOM*, vol. 3, pp. 1726–1734, 2001.

[5] S. Kunniyur and R. Srikant, "Analysis and design of an adaptive virtual queue (AVQ) algorithm for active queue management," in *Proceedings of ACM SIGCOMM*, pp. 123–134, 2001.

[6] J. Aweya, M. Ouellette, and D. Y. Montuno, "A control theoretic approach to active queue management," *Computer Networks*, vol. 36, no. 2-3, pp. 203–235, 2001.

[7] V. Firoiu and M. Borden, "A study of active quene management for congestion control," in *Proceedings of IEEE INFOCOM*, vol. 3, pp. 1435–1444, 2000.

[8] H. Lim, K.-J. Park, E.-C. Park, and C.-H. Choi, "Active queue management algorithm with a rate regulator," in *Proceedings of 15th IFAC World Congress on Automatic Control*, Barcelona, Spain, July 21-26 2002.

[9] K. B. Kim and S. H. Low, "Analysis and design of AQM for stabilizing TCP," *Tech. Rep., caltechCSTR:2002.009*, March 2002.

[10] C. V. Hollot and Y. Chait, "Nonlinear stability analysis for a class of TCP/AQM networks," in *Proceedings of IEEE Confernce on Decision and Control*, vol. 3, pp. 2309–2314, 2001.

[11] M. Christiansen, K. Jeffay, D. Ott, and F. Smith, "Tuning RED for web traffic," in *Proceedings of ACM SIGCOMM*, pp. 139–150, 2000.

[12] Q. Yin and S. Low, "Convergence of REM flow control at a single link," *IEEE Comunication Letters*, vol. 5, no. 3, pp. 119–121, 2001.

[13] F. Paganini, "On the stability o optimization-based flow control," in *Proceedings of American Control Conference*, vol. 6, pp. 4689–4694, 2001.

[14] H. Lim, K.-J. Park, E.-C. Park, and C.-H. Choi, "Virtual rate control algorithm for active queue management in TCP networks," *accepted for publication in IEE Electronics Letters*.

[15] I. Stoica, S. Shenker, and H. Zhang, "Core-stateless fair queueing: Achieving approximately fair bandwidth allocations in high speed networks," in *Proceedings of ACM SIGCOMM*, pp. 118–130, 1998.

[16] V. Misra, W. Gong, and D. Towsley, "Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED," in *SIGCOMM*, pp. 151–160, 2000.

[17] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP throughput: A simple model and empirical validation," in *Proceedings of ACM SIGCOMM*, pp. 303–314, 1998.

[18] S. H. Low, F. Paganini, J. Wang, S. Adlakha, and J. C. Doyle, "Dynamics of TCP/RED and a scalable control," in *Proceedings of IEEE INFOCOM*, 2002.

[19] K. Ramakrishnan and S. Floyd, "A proposal to add explicit congestion notification (ECN) to IP," RFC 2481, 1999.

[20] G. F. Franklin, J. D. Powell, and A. Emami-Naeini, *Feedback control of dynamic systems*, Addsion-Wesley, 3rd ed., 1995.

APPENDIX

We analyze the stability of the linearized model (15) using its characteristic equation. By taking the Laplace Transform of (15), we obtain the characteristic equation

$$s^3 + A_1 s^2 + e^{-sR}(A_2 s^2 + A_3 s + A_4) = 0. \qquad \text{(A-1)}$$

Using the Padé approximant for the time delay, i.e., $e^{-Rs} \approx 1/(1+Rs)$ [20]. In the low frequency region, the Padé approximant is effective, and the resulting approximation of (A-1) is

$$s^4 + a_1 s^3 + a_2 s^2 + a_3 s + a_4 = 0. \qquad \text{(A-2)}$$

where, $a_1 = A_1 + 1/R$, $a_2 = (A_1 + A_2)/R$, $a_3 = A_3/R$, $a_4 = A_4/R$. Since (A-2) is the characteristic equation of a linear time-invariant system, we can apply the Routh-Hurwitz stability criterion [20]. The necessary and sufficient conditions for the approximated system to be stable are

$$(a_1 a_2 - a_3) > 0, \qquad (C1)$$

$$a_3(a_1 a_2 - a_3) - a_1^2 a_4 > 0, \qquad (C2)$$

$$a_1 a_4\big(a_3(a_1 a_2 - a_3) - a_1^2 a_4\big) - a_3(a_1 a_2 - a_3)^2 > 0. \qquad (C3)$$

Proof of Theorem: We will show that the set of conditions $(T1)$-$(T4)$ is derived from the set of $(C1)$,$(C2)$ and $(C3)$. Starting with any positive K_D satisfying $(T1)$, we take K_P satisfying $(T2)$. We can easily find that the right inequality of $(T2)$ is identical to $(C1)$. Here, $(T1)$ is introduced to make the upper bound of $(T2)$ greater than its lower bound. For fixed K_D and K_P, we can derive conditions from $(C2)$ and $(C3)$ in terms of K_I. The condition $(T3)$ can be directly derived from $(C2)$. Note that the term on the right-side of $(T3)$ is greater than zero, i.e., $K_2 > 0$, which is ensured by the right inequality of $(T2)$.

Next, $(C3)$ can be written in terms of K_P and K_I as

$$K_I^2 - K_2 K_P K_I + K_2^2 K_P/\tau_2 < 0. \qquad (C3')$$

Having determined K_D and K_P satisfying $(T1)$ and $(T2)$ respectively, we can consider $(C3')$ as a second-order inequality with respect to $X = K_I$. Defining $F(X)$ as $F(X) = X^2 - K_2 K_P X + K_2^2 K_P/\tau_2$, we will show that $(T4)$ is equivalent to $(C3')$, i.e., there exists an interval of X satisfying $F(X) < 0$.

Note that the axis of symmetry of $F(X)$ lies in the right-half plane, i.e., $K_2 K_P/2 > 0$, since K_2 is positive. The fact that $F(0)$ is positive, together with the left inequality $K_P > 4/\tau_2$ of $(T2)$, guarantees that $F(X) = 0$ has two positive real roots $K_3 K_P$ and $K_4 K_P$. Thus, $F(X) < 0$ for X between $K_3 K_P$ and $K_4 K_P$.

Finally, we have to show that there exists an intersection between $(T3)$ and $(T4)$. Making the assumption that $RC > N$, the axis of the symmetry is smaller than the upper bound in $(T3)$, i.e., $K_2 K_P/2 < K_2 K_P/(1+\tau_1 R)$, therefore, there always exists a K_I satisfying both $(T3)$ and $(T4)$. Consequently, the set of conditions $(T1)-(T4)$ satisfies $(C1)$, $(C2)$ and $(C3)$ and it is sufficient condition for the approximated system to be stable. ■