2005 American Control Conference
June 8-10, 2005. Portland, OR, USA

Experimental verification of stabilizing congestion controllers using the network testbed

Takehito Azuma, Hiyoyuki Naito and Masayuki Fujita

***Abstract*— In this paper, we consider dynamical models of computer networks and derive a synthesis method for stabilizing congestion controllers. Moreover we verify the efficacy of the designed stabilizing congestion controllers using our developed network testbed. First we show dynamical models of TCP/AQM (Transmission Control Protocol/Active Queue Management) networks. The dynamical models of TCP/AQM networks consists of models of TCP window size, queue length and AQM mechanisms. Second we propose to describe the dynamical models of TCP/AQM networks as linear systems with self-scheduling parameters, which also depend on information delay. Here we focus on the constraints on the maximum queue length and TCP window-size, which are the network resources in TCP/AQM networks. Linear system with a self-scheduling parameter and an information delay are derived for dynamical models of TCP/AQM networks. We show a design method of memoryless state feedback controllers for derived linear system with a self-scheduling parameter and an information delay. Finally the effectiveness of the proposed method is evaluated by using `ns-2` (Network Simulator Ver.2) simulator and our developed network testbed.**

I. INTRODUCTION

The high reliable exchange of data using the Internet has been important for its explosive growth and utilization. The Transmission Control Protocol, which is called TCP, is well known as this exchange. Under TCP, a window flow-control mechanism is used to set its transmission rate(Figure 1). In this mechanism, TCP increases the window size during successful data transmission. Conversely TCP cuts the window size in half whenever a data does not reach the receiver. Such data losses called "packet losses" can affect network performance. One of causes of this is that TCP has no information of network mechanisms contributing to packet loss [16]. Considering this, recently TCP/AQM (Transmission Control Protocol/Active Queue Management) networks having queue information of the congested node are considered in some papers [9] [10] [17] [21].

Some kinds of AQM schemes are proposed, e.g. Random Early Detection(RED) [7], Virtual Queue [8], Random Early Marking(REM) [1], Adaptive Virtual Queue(AVQ) [14] and Proportional Integral Controller [12]. Based on the control theory, it seems possible to design that congestion controllers(AQM schemes) achieve better performances than those AQM schemes do. AQM design problems are important and become useful in future researches because AQM is embedded in the router having much information about circumstances of current networks.

Dr. Takehito Azuma, Mr. Hiyoyuki Naito and Prof. Masayuki Fujitais are with Department of Electrical and Electronics Engineering, Kanazawa University, 2-40-20 Kodatsuno, Kanazawa 920-8667, Japan `azuma@t.kanazawa-u.ac.jp`

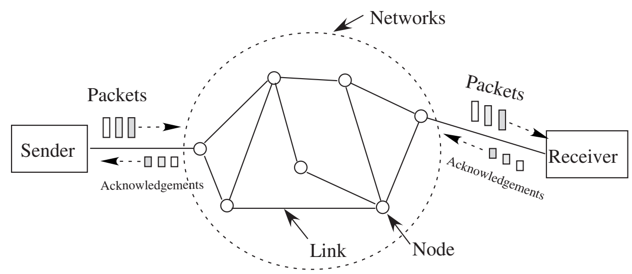

Fig. 1. Networks with TCP

In this article, we consider TCP/AQM networks and design a congestion controller using gain scheduling techniques [23] [18]. First we derive dynamical models of TCP/AQM networks as linear systems with self-scheduling parameters [4] and information delays. Next we propose a design method of congestion controllers for TCP/ AQM networks using the derived linear systems with self-scheduling parameters and information delays. Finally the effectiveness of the proposed method is evaluated by using `ns-2` (Network Simulator Ver.2) simulator by comparing RED and the proposed method.

II. DYNAMICAL MODELS OF TCP/AQM NETWORKS

In this section, dynamical models of TCP/AQM networks considered in this paper are shown. The dynamical models of TCP/AQM networks consist of models of the TCP window sizes, queue lengths and AQM mechanisms. First we show dynamical models of TCP window sizes and queue lengths. AQM mechanisms are introduced. Next the dynamical models of TCP/AQM networks are derived. Finally we propose to describe the dynamical models of TCP/AQM networks as linear systems with self-scheduling parameters, which also depend on time delay (we call this time delay as information delay).

A. Dynamics of TCP networks

Dynamical models of TCP behaviors were investigated by [19]. Simulation results demonstrated that the models investigated by [19] accurately captured the dynamics of TCP networks. The dynamical models of TCP networks are given as the following nonlinear differential equations(Figure 2, Figure 3).

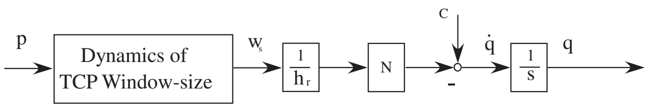

Fig. 2. Models of TCP networks

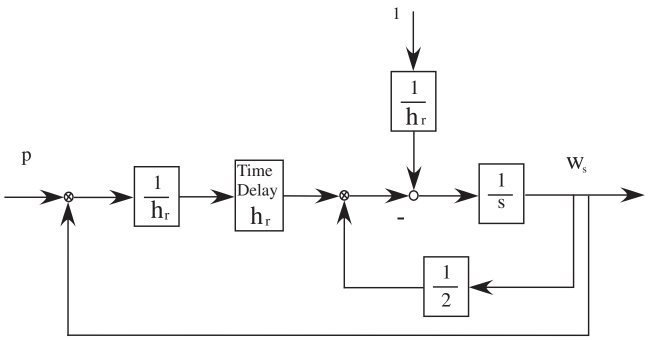

Fig. 3. Dynamics of TCP window-size

$$\dot{w}_s(t) = \frac{1}{h_r(t)} - \frac{w_s(t)}{2}\frac{w_s(t-h_r(t))}{h_r(t-h_r(t))}p(t-h_r(t)), \quad (1)$$

$$\dot{q}(t) = N(t)\frac{w_s(t)}{h_r(t)} - C, \quad (2)$$

where w_s is the average TCP window size, q is the average queue length, C is the queue capacity, N is the number of TCP sessions and p is probability of packets dropped at AQM. h_r is round-trip time $RTT = \frac{q}{C} + T_p$ (T_p : propagation delay). This parameter h_r is information delay in TCP networks. We also assume that w_s and q are constrained as follows,

$$q \in [0,\ q_{max}],, \quad (3)$$

$$w_s \in [0,\ w_{smax}]. \quad (4)$$

In the equation (1), the parameter p is very important. In case of $p = 0$, (1) is equal to

$$\dot{w}_s(t) = \frac{1}{h_r(t)}.$$

This dynamics denotes that TCP window size increases linearly. On the other hand, in case of $p = 1$, (1) can be described as

$$\dot{w}_s(t) = \frac{1}{h_r(t)} - \frac{w_s(t)}{2}\frac{w_s(t-h_r(t))}{h_r(t-h_r(t))}.$$

These fact means that the TCP window size decreases largely. Thus it is important to change the TCP window size by setting p adequately to avoid congestions in networks but traditional AQM designs are impossible to change the TCP window size by setting p adequately because parameter tunings of AQM schemes, which are explained in the next subsection, are manual procedures.

B. AQM scheme

Active queue management is a core process where packets are dropped depending on the queue length. Dropped

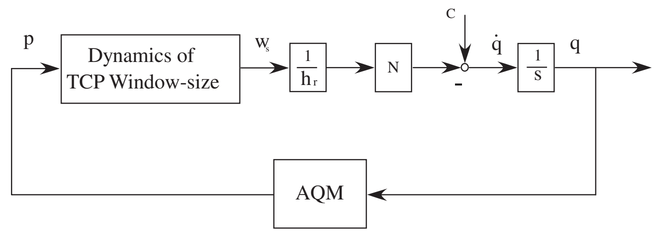

Fig. 4. TCP/AQM networks

packets in this way amounts to reducing TCP source rate as the queue length grows. The objective of AQM is to manage the buffer size as a mean value (the average queue length).

In (1) and (2), we can control the average queue length by adjusting the TCP window size. Thus it is proper to avoid congestions that we make the TCP window size small if the average queue length is large. It is also proper to achieve effective data transmission that we make the TCP window size large if the average queue length is small. In this paper, our objective is to design AQM scheme which has the above properties(Figure 4).

C. Representation as linear systems with self-scheduling parameters and information delays

Here we derive dynamical models of TCP/AQM networks as linear systems with self-scheduling parameters and information delays. To simplify the discussion, we assume that the number of TCP sessions and round-trip time(information delay) are time invariant, e.g. $N(t) = N$, $h_r(t) = h_r$. (1) and (2) can be described as follows,

$$\dot{w}_s(t) = \frac{1}{h_r} - \frac{w_s(t)}{2}\frac{w_s(t-h_r)}{h_r}p(t-h_r), \quad (5)$$

$$\dot{q}(t) = N\frac{w_s(t)}{h_r} - C. \quad (6)$$

Additionally the right-hand side of (5) can be approximated by

$$\frac{1}{h_r} - \frac{w_s(t)}{2}\frac{w_s(t)}{h_r}p(t-h_r),$$

if $w_s >> 1$ [12]. Thus the dynamical models of TCP/AQM networks are given as

$$\dot{w}_s(t) = \frac{1}{h_r} - \frac{w_s(t)^2}{2h_r}p(t-h_r), \quad (7)$$

$$\dot{q}(t) = N\frac{w_s(t)}{h_r} - C. \quad (8)$$

Introducing the equilibrium points ($w_{s0},\ q_0,\ p_0$) which are given by

$$p_0 = \frac{2}{w_{s0}^2}, \quad w_{s0} = \frac{h_r C}{N}, \quad (9)$$

(5) and (6) become

$$\delta\dot{w}_s(t) = -p_0\frac{\delta w_s(t)+2w_{s0}}{2h_r}\delta w_s(t) - \frac{(\delta w_s(t)+w_{s0})^2}{2h_r}\delta p(t-h_r), \quad (10)$$

$$\delta\dot{q}(t) = N\frac{\delta w_s(t)}{h_r}, \quad (11)$$

where

$$w_s(t) = w_{s0} + \delta w_s(t),$$
$$q(t) = q_0 + \delta q(t),$$
$$p(t) = p_0 + \delta p(t).$$

Now the state variable and the input variable are introduced as follows,

$$x(t) = \begin{bmatrix} \delta w_s(t) \\ \delta q(t) \end{bmatrix}, \quad (12)$$

$$u(t) = \delta p(t). \quad (13)$$

Then TCP/AQM networks (5) and (6)((1) and (2)) can be described as the following linear system with a self-scheduling parameter and an information delay.

$$\dot{x} = A(\theta_x(x))x + B(\theta_x(x))u(t-h_r), \quad (14)$$

$$\theta_x(x(t)) = x_1(t) = \delta w_s(t), \quad (15)$$

where

$$A(\theta_x(x)) = \begin{bmatrix} -p_0\frac{1}{2h_r} & 0 \\ 0 & 0 \end{bmatrix}\theta_x(x) + \begin{bmatrix} -p_0\frac{w_{s0}}{h_r} & 0 \\ \frac{N}{h_r} & 0 \end{bmatrix}, \quad (16)$$

$$B(\theta_x(x)) = \begin{bmatrix} -\frac{1}{2h_r} \\ 0 \end{bmatrix}\theta_x^2(x) + \begin{bmatrix} -\frac{w_{s0}}{h_r} \\ 0 \end{bmatrix}\theta_x(x) + \begin{bmatrix} -\frac{w_{s0}^2}{2h_r} \\ 0 \end{bmatrix}, \quad (17)$$

and $\theta_x(x)$ is called as a self-scheduling parameter. From the above discussion, TCP/AQM networks are described as linear system with a self-scheduling parameter and an information delay (14)-(17) having constraints (3) and (3).

III. STABILIZING CONGESTION CONTROLLER SYNTHESIS

Now consider the following linear time-delay system with a self-scheduling parameter,

$$\dot{x} = A(\theta_x(x))x + B(\theta_x(x))u(t-h_r), \quad (18)$$

$$\theta_x(x(t)) = x_1(t), \quad (19)$$

where $x \in R^n$ is the state, $u \in R^m$ is the input and h_r is a constant time delay. $\theta_x(x(t))$ is a self-scheduling parameter and is equal to $x_1(t)$ which denotes the first element of the state x. In this paper, we consider a simple case such as (19). Of course it is possible to extend more general case, but to simplify the discussion we consider only a simple case. Additionally we consider the constraint of the state variable x as follows,

$$x \in X_e = \{x | x'Q_x^{-1}x \le 1\}. \quad (20)$$

The matrix Q_x is a given matrix and denotes a constraint of the state variable x. Since the linear time-delay system with a self-scheduling parameter (18) is nonlinear, the constraint of the state variable x (20) is usual and meaningful. Thus the global problem of the linear time-delay system with a self-scheduling parameter (18) is not discussed in this paper and only a semi-global problem is considered.

Now considering the state constraint (20), the linear time-delay system with a self-scheduling parameter can be treated as a linear parameter varying(LPV) system with a time delay as follows,

$$\dot{x} = A(\theta(t))x + B(\theta(t))u(t-h_r), \quad (21)$$

$$\theta(t) \in [\theta_{min}, \theta_{max}], \quad (22)$$

where $\theta(t)$ is the scheduling parameter of this LPV system with a time delay and θ_{min}, θ_{max} are given from the constraint (20). We call this LPV system with a time delay as a linear time-delay system with a scheduling parameter.

Note that the system (18)-(20) includes the linear time-delay systems with a scheduling parameter (21) with (22). Here we define the design problem of stabilizing controllers for the linear time-delay system with a self-scheduling parameter (18).

Problem : Design the memoryless state feedback controller

$$u(t) = Kx(t), \quad (23)$$

which stabilizes the linear time-delay system with a self-scheduling parameter (18) with (19) and (20) and assures that the state of the closed loop system satisfies the constraint (20).

The following theorem can solve the problem defined in the previous subsection[5].

Theorem 1: Let θ_{min} and θ_{max} be given parameters. If there exist matrices $X > 0$, $P_1 > 0$, $P_2 > 0$ and Y such that the following two LMIs are feasible for all $\theta \in \Theta = [\theta_{min}, \theta_{max}]$,

$$\begin{bmatrix} M_c(\theta, X, Y) & XA'(\theta) & Y'B'(\theta) & B(\theta)Y \\ A(\theta)X & -P_1 & 0 & 0 \\ B(\theta)Y & 0 & -P_2 & 0 \\ Y'B'(\theta) & 0 & 0 & -X \end{bmatrix} < 0, \quad (24)$$

$$X > P_1 + P_2, \quad (25)$$

where

$$M_c(\theta, X, Y) = h_r^{-1}\{XA'(\theta) + A(\theta)X + Y'B(\theta)' + B(\theta)Y\},$$

and moreover defining K, A_c and B_c as

$$K = YX^{-1},$$
$$A_c(\theta) = A(\theta) + B(\theta)K,$$
$$B_c(\theta) = B(\theta)K,$$

be using the matrices of (24) and (25), if there exists $\alpha > 0$ which satisfies

$$\begin{bmatrix} A_c(\theta)Q_x + Q_xA_c'(\theta) + \alpha Q_x & B_c(\theta) \\ B_c'(\theta) & -2\alpha Q_x \end{bmatrix} < 0, \quad (26)$$

for all $\theta \in \Theta$.

Then the controller

$$u(t) = Kx(t), \ K = YX^{-1} \tag{27}$$

is a stabilizing controller for the system (18)-(20) and the state of the closed loop system satisfies the constraint (20).

Here a simple explanation of Theorem 1 is given. The first LMIs (24) and (25) assure that the closed-loop system (18)-(19) is asymptotically stable without considering the constraint (20). The next LMI (26) assures that the state of the closed-loop system satisfies the constraint (20).

Remark 1: Parameters θ_{min} and θ_{max} are design parameters and should be chosen adequately because conservativeness of the result from Theorem 1 is decided by these parameters. Usually θ_{min} and θ_{max} are obtained by considering the constraint (20) in case of a given matrix Q_{x0} because the self-scheduling parameter $\theta_x(x(t))$ is equal to $x_1(t)$. The given matrix Q_{x0} is an estimated matrix from a reachable set of the controlled system (18)-(19).

Remark 2: Since conditions in Theorem 1 depend on the parameter $\theta \in \Theta = [-w_{s0}, \ w_{smax} - w_{s0}]$ in this paper(See the next section), it is generally needed to solve the infinite number of conditions which are obtained by fixing θ on Θ. But using the technique [2], [3], the conditions in Theorem 1 can be reduced to the finite (feasible) number of conditions which do not depend on θ as follows.

The conditions in Theorem 1 can be written in the form of the following parameter dependent LMI condition,

$$F_0(M) + f_1(\theta)F_1(M) + \cdots + f_r(\theta)F_r(M) < 0, \tag{28}$$

where $\theta \in \Theta = [-w_{s0}, \ w_{smax} - w_{s0}]$, f_i is a continuous function of θ, and a symmetric matrix function F_i depends affinely on the unknown matrix $M = [X, \ P_1, P_2, Y] or [\alpha]$. The parameter dependent LMI condition (28) can be reduced to the finite number of LMI conditions as follows.

Theorem 2: [2] Let $\{p_1, \ p_2, \ \cdots, \ p_q\}$ be vertices of a convex polyhedron which includes the curved surface T,

$$T = \{[f_1(\xi) \ f_2(\xi) \ \cdots \ f_r(\xi)]' \mid \xi \in \Xi\}.$$

Assume that there exists M which satisfies the following LMI condition for all $p_i (i = 1, 2, \cdots, q)$,

$$F_0(M) + p_{i1}F_1(M) + \cdots + p_{ir}F_r(M) < 0, \tag{29}$$

where p_{ij} is the jth element of p_i. Then M satisfies (28) for all $\theta \in \Theta$.

A general technique to construct a convex polyhedron which includes the curved surface T is proposed in [2]. If the parameter is scalar and f_i is given as a general polynomial function, a technique to construct a convex polyhedron which includes the curved surface T is proposed in the paper [3] and less conservative results can be obtained by using this technique.

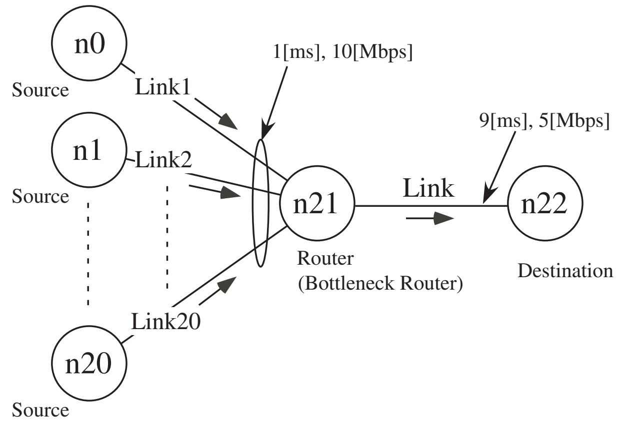

Fig. 5. The general network topology

IV. Experimental verification of stabilizing congestion controllers

A. Simulation results using ns-2

The general network topology on this simulation using `ns-2` is shown in Figure 5. In Figure 5, senders are from n0 to n20 and each senders send data to a router n21, which is the bottleneck router, through Link1 to Link20 respectively. The router n21 sends data from n0 to n20 to the receiver n22. The application command on this simulation is ftp and TCP is used as the protocol. On senders(n0 $\cdots$ n20), the size of one packet is 500 [Bytes] and the size of one acknowledgement is 52 [Bytes]. The maximum window size w_{smax} and the maximum queue length q_{max} are set as follows,

$$w_{smax} = 5, \quad q_{max} = 60.$$

The result using default RED parameters on `ns-2` is shown in Figure 6.

Next the proposed method is applied to Figure 5. We set the round trip time (information delay) h_r as 68 [ms] because the propagation delay is 20 [ms] and the maximum computation time on the router n21 is 48 [ms]. Using the proposed method (Theorem 1), the following memoryless feedback gain is obtained.

$$K = [4.68 \times 10^{-2} \quad 2.48 \times 10^{-5}]. \tag{30}$$

Embedding this memoryless feedback controller in `ns-2`, the result is shown in Figure 7.

We can see that both queue lengths of the routers have oscillation behavior in Figure 6 and Figure 7. But the queue length in Figure 6 is over the maximum queue length $q_{max} = 60$ at the initial time within about 2 [sec] and also the amplitude of oscillation in Figure 6 is larger than that in Figure 7. The number of loss packets is 6476 [packets] in case of RED but the number of loss packets is 2 [packets] in case of the proposed method (Theorem1). Moreover throughputs of RED and the proposed method (Theorem1) are 1142 [packets/sec] and 1157 [packets/sec]

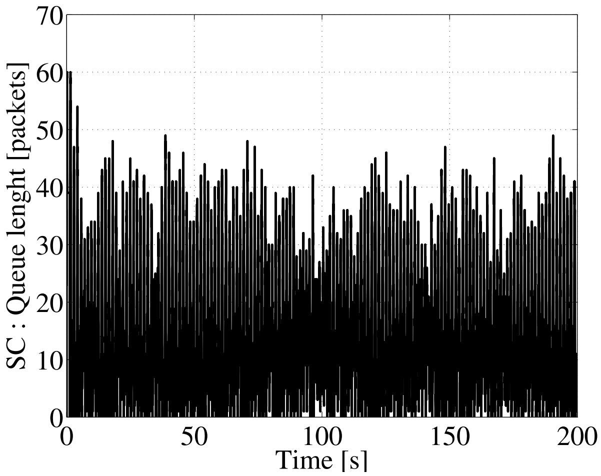

Fig. 6. The queue length of the router n21 using RED

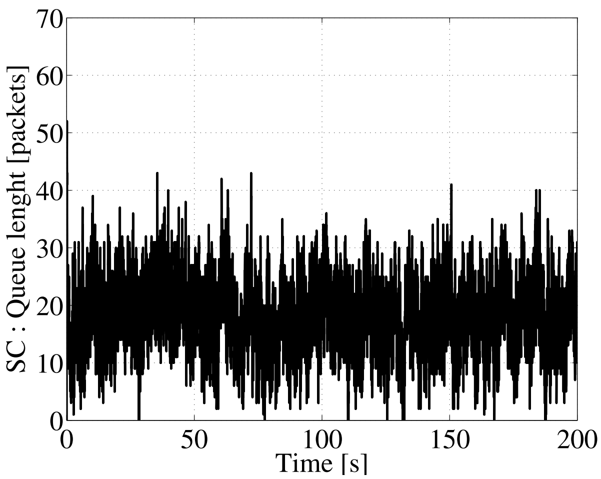

Fig. 7. The queue length of the router using the proposed method(Theorem 1)

respectively. Thus we can conclude that AQM using the proposed method(Theorem1) achieves better performances than AQM using RED does.

B. Experimental results using the network testbed

The network testbed is developed by using 4 computers with Linux 2.4.18. Fig. 9 shows the network topology of the testbed and the hardware configuration of the testbed. Fig. 8 is the photograph of the developed testbed for TCP/AQM networks. Fig. 10 shows the software(Linux kernel) configuration of the testbed and some packets dropping according to probability designed by the designed stabilizing congestion controller(AQM mechanism). Table I and Table I are parameters of links and computers of the network testbet and all parameters are same as simulation using *ns-2*.

Note that the setting of the testbed Fig. 9 $\sim$ 10 and Table I, II is very easy to occur the congestion in the testbed computer network. The designed stabilizing congestion controller(AQM mechanisms) is embedded in the bottleneck router as linux kernel programs at the testbed Fig. 9 $\sim$ 10.

Finally the designed stabilizing congestion controller is implemented in the bottleneck router in Fig. 9 $\sim$ 10. Fig. 11 shows the queue length in the bottleneck router implemented the designed stabilizing congestion controller(AQM mechanism). Considering Fig. 11 and Fig. 7, we can conclude that the designed stabilizing congestion controller is effective and TCP/AQM networks in case of using the designed stabilizing congestion controller achieves better performances than TCP networks in case of using RED. Moreover figures of the queue length in the bottle neck router suggest that our proposed approach, that is congestion controller synthesis problems based on the systems and control theory, is available and possible to achieve good performances.

V. CONCLUSIONS

In this paper, we proposed a design method of stabilizing controllers(memoryless state feedback controllers) for linear time-delay systems with self-scheduling parameters. Our proposed approach is based on a combination technique of gain scheduling and stabilization for linear time-delay systems. Next we applied the proposed approach to design of stabilizing controllers for TCP/AQM networks. We derived TCP/AQM networks as linear time-delay systems with self-scheduling parameters. In a numerical example, the proposed approach was applied to design of stabilizing controllers for TCP/AQM networks. We could assure that the effectiveness of the proposed method was illustrated by comparing RED(Random Early Detection[7]) and the proposed method using *ns-2* and our developed network testbed.

REFERENCES

[1] S.Athuraliya, V.Li, S.Low and Q.Yin, "Rem:Active queue management," *IEEE Network*, 2001.

[2] T.Azuma, R.Watanabe, and K.Uchida, " An Approach to Solving Parameter Dependent LMI Conditions based on Finite Number of LMI Conditions," in *Proceedings of American Control Conference*, pp. 510-514, 1997.

[3] T.Azuma, R.Watanabe, K.Uchida, and M.Fujita, "A New LMI Approach to Analysis of Linear Systems Depending on Scheduling Parameter in Polynomial Forms," Automatisierungstechnik, Vol. 28, no. 4, pp. 119-204, 2001.

[4] T.Azuma, R.Watanabe and K.Uchida, "Constrained State Feedback H-infinity control of Nonlinear Systems," in *Proceedings of the 3rd Asian Control Conference*, pp. 476-479, 2000.

[5] T. Azuma, H. Naito and M. Fujita, "Stabilization of linear time-delay systems with self-scheduling parameters and the application to a stabilizing congestion control synthesis for TCP/AQM networks," in *Proceedings of IFAC Workshop on Time Delay Systems*, Session: Delays in Communication Networks, 2004.

[6] W.Feng, D. Kandlur, D. Saha, and K. Shin, "A Self-Configuring RED Gateway," in *Proceedings of IEEE/INFOCOM1999*, 1999.

[7] 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.

[8] R.Gibbson and F.Kelly, "Distributed connection acceptance control for a connectionless network," in *Proceedings of the 16th Intl. Telegraffic Congress*, 1999.

[9] C.Hollot, V.Misra, D.Towsley, and W.Gong, " On designing improved controllers for AQM routers supporting TCP flows," in *Proceedings of IEEE/INFOCOM*, 2001.
[10] C.Hollot, V.Misra, D.Towsley, and W.Gong, " A control theoretic analysis of RED," in *Proceedings of IEEE/INFOCOM*, 2001.
[11] C.Hollot and Y.Chait, "Nonlinear stanility analysis for a class of TCP/AQM networks," in *Proceedings of Conference on Decision and Control*, 2001.
[12] C.Hollot, V.Misra, D.Towsley, and W.Gong," Analysis and design of controllers for AQM routers supporting TCP flows," *IEEE Trans. on Automatic Control*, Vol. 47, No. 6, pp. 945-959, 2002.
[13] C.Jin, D.Wei, S.Low, G.Buhrmaster, J.Bunn, D.Choe, R.Cottrell, J.Doyle, W.Feng, O.Martin, H.Newman, F.Paganini, S.Ravot, S.Singh, " FAST TCP: From Theory to Experiments," *IEEE Communications Magazine*, 2003.(submitted)
[14] S.Kunniyur and R.Srikant, "Analysis and design of an adaptive virtual queue," in *Proceedings of ACM SIGCOMM*,2001.
[15] Xi Li, and C. de Souza,"Robust stabilization and H_∞ control of uncertain linear time-delay systems," in *Proceedings of 13th IFAC World Congress*, pp. 113-118, 1996.
[16] S.Low, F.Paganini and J.Doyle: Internet Congestion Control, *IEEE Control Systems Magazine*, Vol. 22, No. 2, pp. 28-43, 2002.
[17] S.Low, F.Paganini, J.Wang, S.Adlakha and J.Doyle,"Dynamics of tcp/aqm and a scalable control, in *Proceedings of IEEE INFOCOM*, 2002.
[18] W.Rugh and J.Shamma, "Research on gain scheduling," Vo. 32, pp. 1401-1425,2000.
[19] V.Misra, W.Gong, and R.Srikant, "Robust rate control for ABR sources," in *Proceedings of ACM/SIGCOMM*, 2000.
[20] F.Paganini, J.C.Doyle, and S. Low, "Scalable laws for stable network congestion control," , in *Proceedings of Conference on Decision and Control*, 2001.
[21] F.Paganini, Z.Wang, S.Low and J.Doyle, "A new TCP/AQM for stable operation in fast networks," in *Proceedings of IEEE INFOCOM*, 2003.
[22] G.Vinnicombe, "On the stability of networks operating TCP-link congestion control," in *Proceedings of IFAC*, 2002.
[23] R.Watanabe, K.Uchida, M.Fujita and E.Shimemura, "L^2 gain and H^∞ control of linear systems with scheduling parameter," in *Proceedings of of 33rd IEEE Conference on Decision and Control*, pp.1412-1414, 1994.

Fig. 8. Testbed hardware configuration(a photogragh).

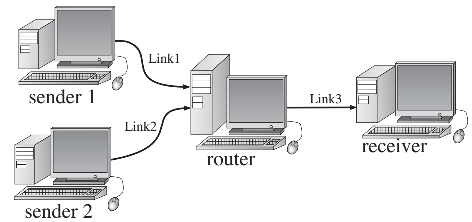

Fig. 9. Testbed hardware configuration(The network topology).

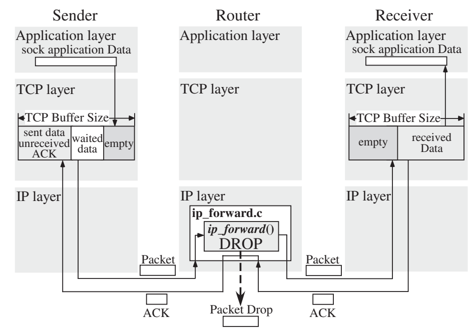

Fig. 10. Testbed software : Linux kernel configuration.

TABLE I

PARAMETERS OF LINKS.

	Link1–2	Link3(approach)	Link3(back)
Bandwidth	100[Mbps]	865[Kbps]	100[Mbps]
Delay	0.5[ms]	12.5[ms]	0.5[ms]

TABLE II

PARAMETERS OF COMPUTERS.

	sender 1–2	router	receiver
OS		Linux 2.4.18	
Distribution		Redhat	
LAN Standard		100Base-TX	
TCP Buffer Size	25[MByte]	65[KByte]	45[MByte]
Ring Buffer Size		65536[Byte]	

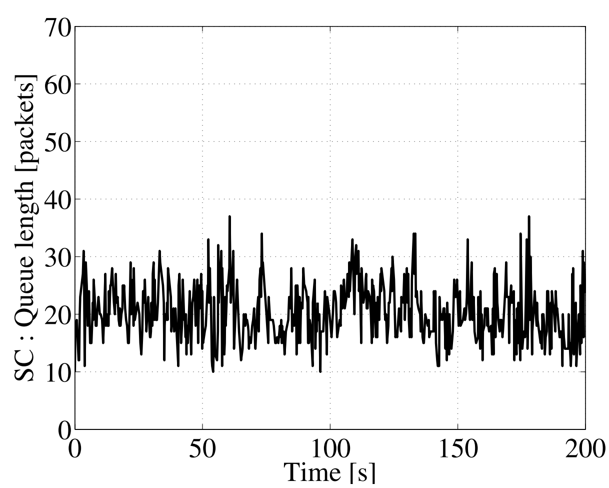

Fig. 11. The queue length of the router n21 using our developed network testbed