

# Switching Resilient PI Controllers for Active Queue Management of TCP Flows

Deniz Üstebay

Department of Electrical and  
Electronics Engineering  
Bilkent University  
Ankara 06800, Turkey  
Email: deniz@ee.bilkent.edu.tr

Hitay Özbay

Department of Electrical and  
Electronics Engineering  
Bilkent University  
Ankara 06800, Turkey  
Email: hitay@bilkent.edu.tr

**Abstract**— Active Queue Management (AQM) is used in computer networks to increase link utilization with less queueing delays. The fluid flow model of TCP based on delay differential equations supplies the mathematical background for modelling the AQM as a feedback system. Recently various PI and PID controllers are designed for this feedback system, [7], [18]. In this paper, we consider the case for which the Round Trip Time (RTT) is time varying and we propose switching resilient PI controllers using the design method introduced in [18].

## I. INTRODUCTION

One of the most persistent problems confronted in the Internet is the congestion. When congestion is present in the computer network, buffers at the routers are filled with packets. Any packet that reaches to these routers are lost and to be resent. If lots of packets are lost and tried to be resent, a considerable amount of delay will be observed by the end users of the Internet, [4]. AQM is a congestion control mechanism that is used for preventing buffer overflows and such large delays. On the other hand, buffers that have an amount of packets less than a desired level are also unwanted because they signal under utilization of link capacities. In fact, AQM tries to maintain a certain desired level of queue length at the router buffers by avoiding both buffer overflow and emptiness. For this objective, AQM schemes mark the packets passing through the link according to a certain probabilistic rule. This packet marking probability can be a static function of queue length as in RED [3] and REM [1] or a dynamic function of queue length as in [7], [18]. In [11], a fluid flow model of AQM was developed. In [6], [7], the fluid flow model is linearized and a design method for PI control of AQM is proposed. In [18], new PI and PID controllers for AQM are developed using techniques introduced in [5],[13] and it is shown that the proposed PI controller performs better than the PI controller designed in [7] in terms of tracking and robustness.

Different methods of AQM have been proposed in [2], [9], [14], [16], [19], [20]. With the exception of [20], where  $H_\infty$  based AQM techniques are used, the papers mentioned above do not consider time varying propagation delays, which may occur due to changes in the communication channels. The proposed switched  $H_\infty$  control method of [20] is relatively

complicated to implement in real networks. Therefore we consider simpler PI controllers and switch among them.

In this paper, we apply the PI controller proposed in [18] to a network with time varying round trip times(RTT). We propose that when RTT (i.e. time delay) varies with time, dividing the operating range to smaller regions and designing PI controllers for each of these regions give better results than using a single PI controller for the entire operating range. This proposition is supported by ns-2 [12] simulation results.

Remaining of the paper is organized as follows. The AQM model and the details of PI controller design for AQM are given in Section II. The results and analysis of the ns-simulations can be found in Section III. Concluding remarks are made in Section IV.

## II. PI CONTROL FOR AQM

### A. Mathematical Model of AQM Supporting TCP Flows

The dynamical model of TCP was developed using fluid flow approximation in [11] and used by [2], [7], [10], [14], [15], [20]. In this paper we use a simplified version of this model introduced in [7]. The model consists of the following nonlinear differential, time-delayed equations:

$$\dot{W}(t) = \frac{1}{R(t)} - \frac{W(t)}{2} \frac{W(t-R(t))}{R(t-R(t))} p(t-R(t)) \quad (1)$$

$$\dot{q}(t) = N(t) \frac{W(t)}{R(t)} - c(t) \quad \text{when } q(t) > 0 \quad (2)$$

where  $W$  is the TCP window size,  $q$  is the queue length,  $N$  is the number of TCP flows,  $c$  is the link capacity, and  $p$  is the probability of packet mark. Here, RTT (total delay in the feedback path) is expressed by

$$R(t) = T_o(t) + \frac{q(t)}{c(t)} \quad (3)$$

where  $T_o(t)$  is the propagation delay. Note that we consider time varying propagation delay. The variation of  $T_o(t)$  will be taken to be slow compared to the variations of  $\frac{q(t)}{c(t)}$ , but the magnitude of the variations of the propagation delay is larger than the variations of queueing delay.

Equation (1) specifies the TCP window dynamic incorporating the additive increase and multiplicative decrease (AIMD)

behavior of TCP. Likewise, equation (2) describes the queue length dynamic. It is possible to use these equations to describe TCP as a feedback control system, where  $p$  is the control input generated by feedback from  $q$ . This nonlinear feedback system can then be linearized around an equilibrium point using small signal analysis. Let  $q_o, W_o, c_o, N_o, p_o, R_o$  be the nominal values at the equilibrium point. For  $q(t) = q_o + \delta_q(t)$ ,  $W(t) = W_o + \delta_W(t)$ ,  $c(t) = c_o + \delta_c(t)$ ,  $N(t) = N_o + \delta_N(t)$ ,  $p(t) = p_o + \delta_p(t)$  and  $R(t) = R_o + \delta_R(t)$ , with  $R_o = T_o + \frac{q_o}{c_o}$  a transfer function  $G_{pq}(s)$  from input  $\delta_p$  to output  $\delta_q$  can be obtained, see e.g. [2], [6], [7], [15],

$$G_{pq}(s) = e^{-R_o s} \frac{R_o c_o K}{R_o s + K^{-1}} \frac{1}{R_o s + 1}, \quad K = \frac{R_o c_o}{2N_o}. \quad (4)$$

Now, with plant dynamics expressed by this transfer function, we can design a PI controller for the plant.

### B. Resilient PI Controller Design

Recently a study on allowable PI and PD control parameters for time delayed systems with at most two unstable poles is done in [5], [13]. The objective of this study was to find the largest allowable intervals for certain PI and PD controller parameters. Controllers obtained with this method is expected to work for a wide range of system parameters. Hence they will be resilient in the sense of [17]. In [18], this controller parameter design method is applied to the AQM problem by selecting the center of the largest allowable intervals as the optimal gains of the controllers. We now summarize the PI control design of [18].

For  $K \gg 1$  the transfer function of the plant can be rewritten as

$$G_{pq}(s) \approx \frac{c_o K}{s} f(s) \quad \text{where} \quad f(s) := \frac{e^{-R_o s}}{(R_o s + 1)}. \quad (5)$$

Let

$$\theta := \left\| \frac{f(s) - 1}{s} \right\|_{\infty}.$$

According to [5], [13], the optimal proportional gain that maximizes the allowable integral gain interval is  $\frac{1}{2\theta}$  and the maximum value of the maximal interval of the integral gain is  $\frac{1}{4\theta}$ . To make the controller robustly stable with respect to largest perturbations in the controller parameters, we choose the proportional gain as  $\frac{1}{2\theta}$  and the integral gain as  $\frac{1}{8\theta}$ , which is the midpoint of the maximal interval, [18]. Then the PI controller for the plant (5) is obtained as

$$K_{pi}(s) = \frac{1}{2\theta} \frac{1}{c_o K} \left( 1 + \frac{1}{8\theta s} \right). \quad (6)$$

In order to implement these controllers in ns-2 we use a digital implementation method suggested in [8], [6].

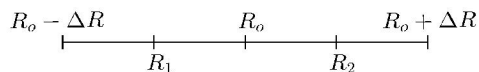


Fig. 1. RTT

### C. Switching Control

The PI controller design given above assumes that RTT is time invariant. However, in computer networks RTT is probably time varying or uncertain.

For the case shown in Fig. 1, the nominal value of RTT is  $R_o$  and we assume that RTT takes values between  $R_o - \Delta R$  and  $R_o + \Delta R$ . If we are to design a single PI controller for this plant we can assume

- (i) the plant is nominal, let  $R = R_o$ , and implement  $K_{pi0}$ ,
- (ii) for  $R_o - \Delta R < RTT < R_o$ , let  $R = R_1 = R_o - \frac{\Delta R}{2}$  and implement  $K_{pi1}$
- (iii) for  $R_o < RTT < R_o + \Delta R$ , let  $R = R_2 = R_o + \frac{\Delta R}{2}$  and implement  $K_{pi2}$ .

Since it is shown that considered PI controller is robust to the changes in RTT, [18], these three controllers are expected to have good performance in the neighborhood of RTT values they are designed for. In this paper, we illustrate that it is possible to improve the performance in the case of time varying RTT by applying switching control. Two different configurations are investigated:

- a) Using two of the PI controllers above, we perform mid-point switching. When RTT is in  $[R_o - \Delta R, R_o]$  interval  $K_{pi1}$  is active and when RTT is in  $[R_o, R_o + \Delta R]$  interval  $K_{pi2}$  is active.
- b) Instead of dividing  $[R_o - \Delta R, R_o + \Delta R]$  interval into two, we divide it into  $N \gg 1$  intervals. Therefore, we design  $N$  different PI controllers for each of these intervals and as RTT varies among these intervals, the controller parameters switch accordingly.

## III. SIMULATIONS

The performance of the designed PI controllers are tested with ns-2 simulations. The network topology (Fig. 2) of the simulations consists of a single bottleneck link and two routers at the ends of this link which support  $N$  TCP flows as in [19]. The buffers of both routers can hold 300 packets where the packets are of size 1000 Bytes. All the links in the network has the same capacity,  $C_0 = C_1 = 10$  Mbps. The propagation delays of the links vary with time. As RTT of the network varies in the interval  $[0.16s, 0.48s]$ , the propagation delay of the bottleneck link,  $T_0$ , takes values in the interval  $[4ms, 36ms]$ ; the propagation delay of the links between routers and TCP sources/destinations,  $T_1$ , takes values between

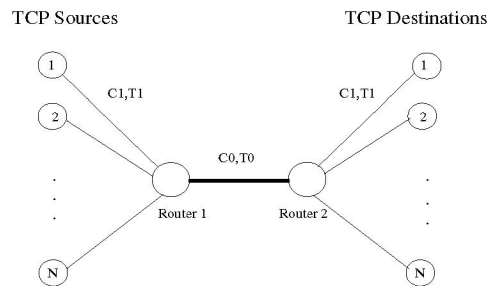


Fig. 2. Network Topology

TABLE I

THE ANALYSIS OF SIMULATION RESULTS FOR  $RTT_1$ 

Controller	Mean	Std	RMS error	$\mathcal{E}$
$K_{pi0}$	153.34	39.64	0.27	0.33
$K_{pi1}$	142.60	39.14	0.27	0.27
$K_{pi2}$	159.91	45.96	0.31	0.49
$K_{pi1}$ - $K_{pi2}$ Switching	152.52	35.07	0.23	0.22
16 Switching Controllers	148.77	34.63	0.23	0.23

TABLE II

THE ANALYSIS OF SIMULATION RESULTS FOR  $RTT_2$ 

Controller	Mean	Std	RMS error	$\mathcal{E}$
$K_{pi0}$	152.28	46.03	0.31	0.42
$K_{pi1}$	134.68	43.60	0.31	0.42
$K_{pi2}$	159.47	51.80	0.35	0.52
$K_{pi1}$ - $K_{pi2}$ Switching	150.67	41.69	0.28	0.32
16 Switching Controllers	150.57	40.31	0.27	0.32

[8ms,72ms]. The nominal values of the system parameters are:  $N_o = 30$  TCP flows,  $c_o = 1250$  packets/s,  $q_o = 150$  packets,  $R_o = 0.32$  s.

We investigate the performance of five different configurations:

- single controller,  $K_{pi0}$ ,
- single controller,  $K_{pi1}$ ,
- single controller,  $K_{pi2}$ ,
- two switching controllers,  $K_{pi1}$  and  $K_{pi2}$ ,
- N=16 switching controllers.

This experiment is done twice, first for an RTT function,  $RTT_1$ , as in Fig. 3(a) and then for a more quickly changing RTT function,  $RTT_2$ , as in Fig. 4(a). The plots of queue lengths are given in Fig. 3 and Fig. 4. For the purpose of evaluating the simulation results, we compare some properties of the queue lengths for all cases. We do the comparison with mean, standard deviation (abbreviated std), RMS error and  $\mathcal{E}$  as in [18]. RMS error can be formulated as

$$\text{RMS error} = \left( \frac{1}{M} \sum_{k=1}^M \left( \frac{q(k) - q_d}{q_d} \right)^2 \right)^{\frac{1}{2}}, \quad (7)$$

where  $q_d = 150$  packets is the desired queue length and  $M$  is the total number of the samples generated by network simulator. The last metric is given as  $\mathcal{E} = T_0/T_{\text{total}}$  where  $T_0$  is the total length of time intervals for which  $q(t) \notin [120, 180]$  and  $T_{\text{total}}$  is the total simulation time (200 seconds).

The results are given in Tables I and II. According to the figures and tables, switching between two controllers improves the tracking and mean, standard deviation, RMS and  $\mathcal{E}$  values for both RTT functions. Using 16 controllers does not provide a significant improvement according to the results in Tables I and II. However for  $RTT_2$  case, around  $t \in [30, 40]$  there is a sudden drop in queue length which cannot be compensated by two switching controllers, see Fig. 4. This sudden drop in  $q(t)$  is not seen when we use  $K_{pi1}$  (designed for smallest nominal RTT) or 16 switched controllers. Since the numerical results show that  $K_{pi1}$  is not able to track the queue length properly, the sudden drop problem can be solved by 16 controllers with good performance.

#### IV. CONCLUSION

In this paper we proposed switching PI controllers for time varying time delayed single bottleneck network. We designed PI controllers according to the design method given in [18] (In [18] it is shown that the proposed PI design method performs

better than the benchmark PI design [7] which is available in the current version of ns-2). The performances of the designed controllers were tested via ns-2 simulations. Simulations show that switching between two controllers gives better results compared to a single controller case.

Note that theoretical proof of performance improvement by using switched controllers in such a complicated nonlinear system (packet level simulation setting) is not easy to obtain. In fact, even for simplified flow model it can be shown that arbitrary switching between two controllers may even destabilize the feedback system. Therefore our simulation results illustrate the value of mid-point switched PI controllers for this AQM problem.

#### ACKNOWLEDGMENT

This work is supported in part by the European Commission (contract no. MIRG-CT-2004-006666) and by TÜBİTAK (grant no. EEEAG-105E156).

#### REFERENCES

- [1] S. Athuraliya, S. H. Low, V. H. Li, Q. Yin, "REM: Active Queue Management," *IEEE Network*, vol. 15, no. 3, May-June 2001, pp. 48-53.
- [2] M. di Bernardo, F. Garofalo and S. Manfredi, "Performance of Robust AQM controllers in multibottleneck scenarios," *Proc. of the 44th IEEE Conference on Decision and Control, and the European Control Conference 2005* Seville, Spain, December 2005, pp. 6756-6761.
- [3] S. Floyd, V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Transactions on Networking*, vol. 1, no. 4, Aug. 1993, pp. 397-413.
- [4] S. Floyd, K. Fall, "Promoting the use of end-to-end congestion control in the Internet," *IEEE/ACM Transactions on Networking*, vol. 7 (1999), pp. 458-472.
- [5] A. N. Gündes, H. Özbay, A. B. Özgüler, "PID Controller Synthesis for a Class of Unstable MIMO plants with I/O Delays," *Automatica*, vol. 43 (2007), pp. 135-142.
- [6] C. Hollot, V. Misra, D. Towsley, W. Gong, "On designing improved controllers for AQM routers supporting TCP flows," in *IEEE INFOCOM*, Alaska, April 2001, pp. 1726-1734.
- [7] C. Hollot, V. Misra, D. Towsley, W. Gong, "Analysis and design of controllers for AQM routers supporting TCP flows," *IEEE Transactions on Automatic Control*, vol. 47, no. 6, June 2002, pp. 945-959.
- [8] R. Isermann, *Digital Control Systems Volume 1: Fundamentals, Deterministic Control* 2nd revised Ed., Springer-Verlag, 1989.
- [9] S. Kunniyur, R. Srikant, "Analysis and design of an adaptive virtual queue (AVQ) algorithm for active queue management," in *Proc. ACM SIGCOMM*, San Diego, CA, Aug. 2001, pp. 123-134.
- [10] S. Low, F. Paganini, J. Wang, S. Adhalka, J. Doyle, "Dynamics of TCP/RED and a scalable control," in *Proc. IEEE INFOCOM*, New York, June 2002, pp. 239-248.
- [11] V. Misra, W. Gong, D. Towsley, "A fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED," in *Proc. ACM SIGCOMM*, Stockholm, Sweden, Sep. 2000, pp. 151-160.
- [12] ns-2 Network Simulator, version: 2.27. [Online]. Available: <http://www.isi.edu/nsnam/ns/>.

- [13] H. Özbay, A. N. Gündes, "Resilient PI and PD Controllers for a Class of Unstable MIMO plants with I/O Delays," in *CDROM Proceedings of 6th IFAC Workshop on Time Delay Systems*, L'Aquila, Italy, July 2006.
- [14] E-C. Park, H. Lim, K-J. Park, C-H. Choi, "Analysis and design of the virtual rate control algorithm for stabilizing queues in TCP networks," *Computer Networks*, vol. 44 (2004), pp. 17-41.
- [15] P-F. Quet, H. Özbay, "On the Design of AQM Supporting TCP Flows Using Robust Control Theory," *IEEE Transactions on Automatic Control*, vol. 49, no. 6, June 2004, pp. 1031-1036.
- [16] S. Ryu, C. Rump, C. Qiao, "A predictive and robust active queue management for Internet congestion control," in *Proc. of the 8th IEEE Symposium on Computers and Communications*, Kemer, Antalya, Turkey, June-July 2003, pp. 991-998.
- [17] G.J. Silva, A. Datta, S. P. Bhattacharyya, "PID Controllers for Time-Delay Systems," Birkhäuser, Boston, 2005.
- [18] D. Üstebay and H. Özbay, "A New PI and PID Control Design Method for Integrating Systems with Time Delays," to appear in *Proc. of the the Sixth WSEAS International Conference on Signal Processing, Robotics and Automation*, Corfu, Greece, February 2007.
- [19] P. Yan, Y. Gao, H. Özbay, "A Variable Structure Control Approach to Active Queue Management for TCP with ECN," *IEEE Transactions on Control Systems Technology*, vol. 13, no. 2, March 2005, pp. 203-215.
- [20] P. Yan and H. Özbay, "Robust Controller Design for AQM and  $H_\infty$  Performance Analysis," in *Advances in Communication Control Networks*, S. Tarbouriech, C. Abdallah, J. Chiasson Eds., Springer-Verlag LNCIS, Vol. 308, 2004, pp. 49-64.

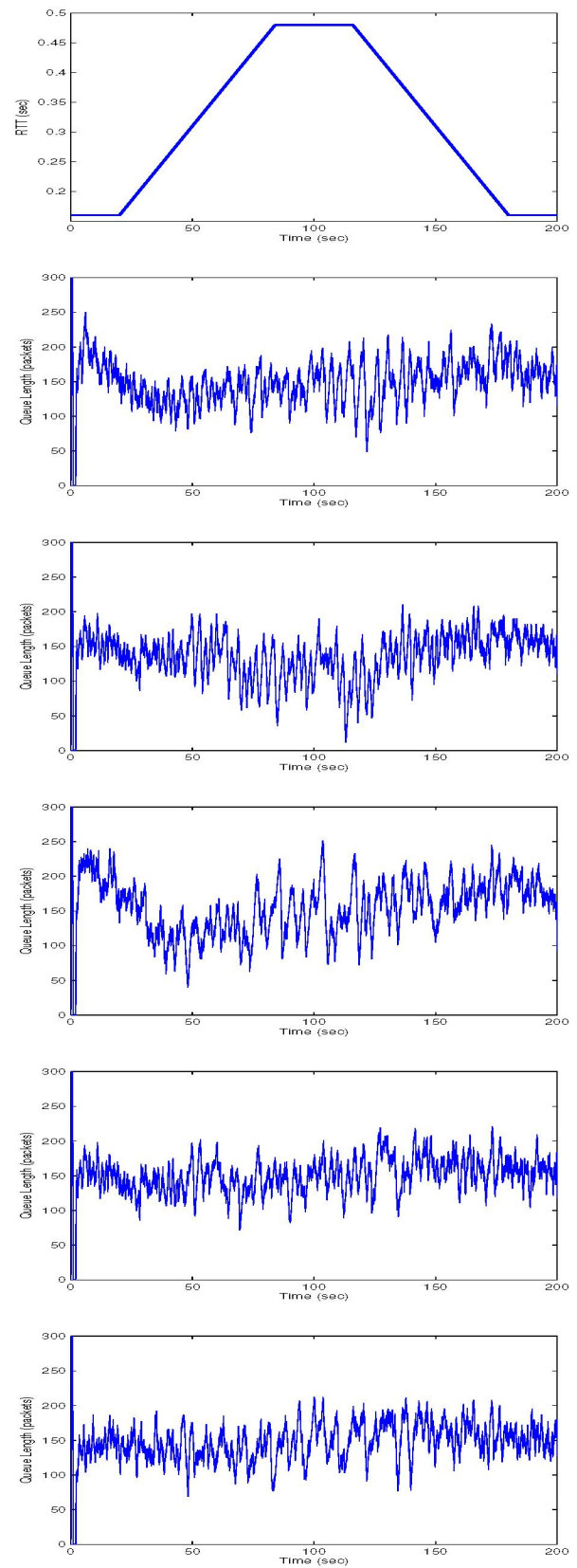


Fig. 3. ns-2 simulations: (a)  $RTT_1$  (b) single controller,  $K_{pi0}$  (c) single controller,  $K_{pi1}$  (d) single controller,  $K_{pi2}$  (e) two switching controllers,  $K_{pi1}$  and  $K_{pi2}$ , (f) N=16 switching controllers

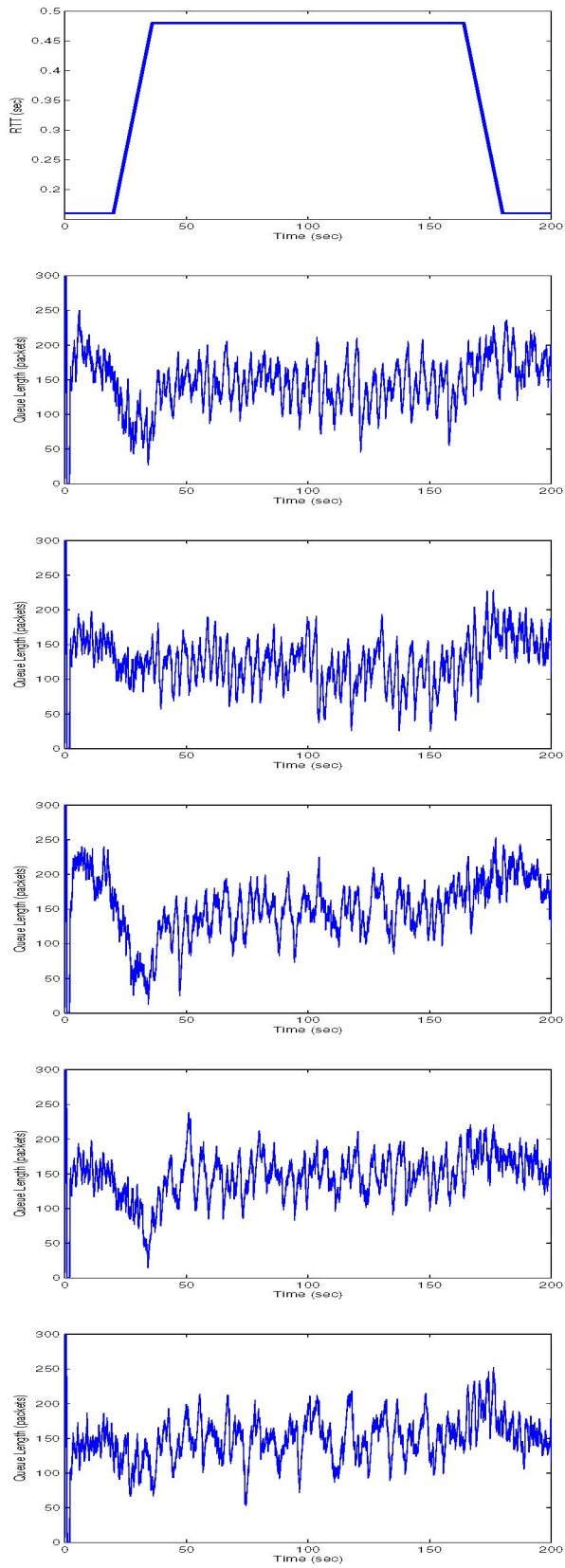


Fig. 4. ns-2 simulations: (a)  $RTT_2$  (b) single controller,  $K_{pi0}$  (c) single controller,  $K_{pi1}$  (d) single controller,  $K_{pi2}$  (e) two switching controllers,  $K_{pi1}$  and  $K_{pi2}$ , (f) N=16 switching controllers