A Robust Active Queue Management Algorithm for Wireless Network

Kanchan Chavan, Madhu N. Belur and Abhay Karandikar
Information Networks Lab, Department of Electrical Engineering
Indian Institute of Technology Bombay, Mumbai 400 076, India.
Email: kanchanvc@gmail.com, {belur,karandi}@ee.iitb.ac.in

***Abstract*— Active Queue Management (AQM) has been extensively studied in the context of wired Internet. In this paper, we propose an AQM algorithm for a bottleneck node with wireless link. Specifically, we propose a robust controller design method that maintains the queue length close to an operating point. We treat capacity variation as an external disturbance and design the controller using $\mathcal{H}_\infty$ control techniques. This results in the controller being robust to changes in the design parameters of the system, while it meets the performance objectives. We also consider the effect of round trip time in our model. Our method of incorporating the delay into the discretized model simplifies controller design.**

I. Introduction

Active Queue Management (AQM) techniques play an important role in controlling congestion in Transmission Control Protocol (TCP) networks. Random Early Detection (RED) [1] was one of the earliest AQM methods. However, RED suffers from several drawbacks like requirement of considerable effort in tuning its parameters to achieve good performance [2]. The development of RED, however, sparked off a lot of interest in AQM algorithms and subsequently several algorithms [3], [4], [5], [6], [7], that addressed the shortcomings of RED, were proposed. An AQM algorithm comprises of two basic components- one that monitors the fluctuations in queue length and the other that conveys any incipient congestion by dropping or marking a packet with some probability. TCP congestion control mechanism responds to this feedback by adaptively modifying its window size. Various AQM techniques differ in the way they perform queue measurements for detecting congestion and the mechanism for packet drop. See [8] for a comprehensive overview of AQM algorithms.

It was subsequently recognized that TCP with AQM router can be considered as a feedback control system. Thus control theoretic models of AQM have been developed in [9], [10], [11] among others and these models have helped in a better understanding of router queue dynamics in TCP networks. This has resulted in systematic and scalable techniques for controller design. (See [12], [13] and the references therein for a detailed overview and analysis.).

Despite a plethora of literature in AQM, most of these works have focussed on wired network. In this paper, we study AQM for the case of wireless links. Unlike wired link which is assumed to have a fixed capacity, a wireless link has a capacity that is time varying due to multipath fading and mobility [14]. Thus, the controller is required to meet performance objectives in the presence of these capacity variations.

An important feature of the method proposed in this paper is that we have not ignored the delay due to round trip time in the model, nor has the delay transfer function e^{-sT} been approximated by a real rational transfer function. Keeping in mind that the packet drop/mark probability has its effect on the TCP sources only after at least the round trip time, modifying the model by ignoring the delay results in the modified model suggesting an exaggerated ability to control the queue length using drop probability for control. The importance of considering the time delay has also been noted in [15], [16], [17]. In this paper, we assume that there are no fluctuations in the round trip time; this constant time delay in the system model is incorporated in a discretized model of the continuous time TCP network system (see Section II).

More recently, AQM for wireless networks has also been studied in [18]. Capacity variation of wireless link has been considered as a disturbance and an $\mathcal{H}_\infty$ control technique has been used for controller design to achieve disturbance attenuation. However, in [18], due to the (linearized) time-delay system that is eventually studied, the solution to the $\mathcal{H}_\infty$ control problem as formulated there requires the solution of a matrix inequality that is *not* a linear matrix inequality (LMI). As a result, a search/sweeping process is required to get a clue of suitable parameters (using heuristic methods described in [19]), after which an LMI problem is to be solved to find a controller that meets the performance objective.

In contrast to [18], our method of incorporating the delay into the discretized model simplifies controller design. In our approach, the controller can be designed to control the discrete time system for any of various performance objectives like Linear Quadratic Regulator (LQR), Linear Quadratic Gaussian (LQG), $\mathcal{H}_2$, $\mathcal{H}_\infty$, or just pole placement.

In order to ensure that the controller stabilizes the router queue despite inaccuracies in the linearized discretized model, we design a robust controller using $\mathcal{H}_\infty$ control methods. We treat variations in the capacity as disturbances to the system and seek to design a controller that reduces the maximum effect that the disturbances can cause to fluctuations in the queue length. The robustness of this controller is verified using simulations. In this sense the active queue management technique of this paper can be termed as Robust Queue Management (RQM). The same discrete time model of the TCP network with a wireless link can be used for meeting other control objectives.

Another feature about capacity variations adopted in this

[0]Acknowledgement: The research was supported in part by IITB-TTSL Center for Excellence in Telecom and by SERC division, Department of Science and Technology, India.

paper is that these variations (about a nominal capacity value C_0), are made to influence the plant states through a low pass filter. This filter incorporates the situation that capacity variations are slower than queue length variations and other such variables of the system. Thus the controller is required to reduce the effect due to only low frequency capacity variations.

The rest of the paper is organized as follows. We study the system model and make certain approximations (like linearizations, discretizations) in the next section. The control problem and the performance objective of the controller is stated in Section III. Section IV contains simulation results of the proposed RQM controller and comparison with RED performance. A few conclusive remarks follow in Section V.

II. System Model and Problem Formulation

In this section, we first describe the fluid flow model of TCP window and router dynamics in a wireless network.

A. Fluid Flow Model of TCP for Wireless Network

We model the TCP network by a simple dumb-bell topology where TCP sources send data towards their destinations through a router which represents the bottleneck node in the network (see Figure 2). With receipt of acknowledgement packet, each source increase its rate of transmission and this eventually causes the outgoing link capacity of the bottleneck node to be exceeded. When a packet drop is detected by the TCP source, it reduces its window size by half. In this way, each source attempts to determine the available capacity in the network. We consider the fluid flow model of TCP behavior as proposed in the classical paper [10]. This model relates window and queue dynamics with capacity of the router outgoing link and the round trip time (RTT). Note that RTT is the time required for the packet to reach the destination plus the time it takes for an acknowledgement to reach the sender. Hence, it comprises of transmission time, queuing delay and propagation delay.

Though originally proposed for wired networks, we use the fluid flow model for the wireless scenario where we assume that the capacity of the bottleneck node is time varying. We further assume that these variations are relatively slow compared to queue dynamics. With these assumptions, we write the fluid flow model as follows

$$\begin{aligned} \dot{W}(t) &= \frac{1}{R(t)} - \frac{W(t)W(t-R(t))}{2R(t-R(t))}p(t-R(t)), \\ \dot{q}(t) &= \frac{W(t)}{R(t)}N(t) - C(t), \end{aligned} \tag{1}$$

where W is the TCP window size (packets), q is the queue length (packets), R is the round-trip time (seconds) $R = \frac{q}{C(t)} + T_p$, C is the link capacity (packets/s), T_p is the propagation delay (seconds), N is the load factor (number of TCP sources), p is the probability of packet drop, and $\dot{x}$ denotes the derivative of x with respect to time.

The first term on the right hand side of the first equation corresponds to additive increase of the window while the second term corresponds to multiplicative decrease in response to packet drop which occurs with probability p. The second equation captures the queue dynamics. The queue length q and window-size W are positive and bounded quantities, i.e. $q \in [0, \bar{q}]$ and $W \in [0, \bar{W}]$, where $\bar{q}$ and $\bar{W}$ denote buffer capacity and maximum window size respectively. The marking probability $p(.)$ takes values in $[0, 1]$.

We linearize the above nonlinear system about an equilibruim point. Suppose W_0, C_0, R_0, p_0, q_0, T_p and N satisfy suitable equations such that $\dot{W}(t)$ and $\dot{q}(t)$ are zero. Like in [10], the corresponding equations for an equilibruim operating point are $W_0^2 p_0 = 2$, $W_0 N = R_0 C_0$ and $R_0 = \frac{q_0}{C_0} + T_p$.

Consider deviations of variables W, q, p and C about such an equilibruim point to obtain the following linearized model; let δW, δq, δp and δC respectively denote these deviations, i.e. $\delta W := W - W_0$, $\delta q := q - q_0$, $\delta p := p - p_0$ and $\delta C = C - C_0$.

$$\begin{aligned} \delta\dot{W}(t) &= \frac{-NC_0}{(q_0 + C_0 T_p)^2}(\delta W(t) + \delta W(t - R_0)) \\ &\quad - \frac{C_0(q_0 + C_0 T_p)}{2N^2}\delta p(t - R_0), \end{aligned} \tag{2}$$

$$\delta\dot{q}(t) = \frac{C_0 N \delta W(t) - C_0 \delta q(t) + q_0 \delta C(t)}{q_0 + C_0 T_p}.$$

This model is utilized below in the following subsection for controller design after suitable discretization.

B. Digital AQM Controller Design

The objective of an AQM controller is to maintain the queue length of the router at the desired set value. The controller is also required to be robust with respect to variations in network parameters like queue set point and number of TCP sources. We can represent the entire system by the model as shown in Figure 1. The inputs to the plant are probability of packet drop δp and capacity variations δC while the output is the queue length deviation δq. The controller takes queue deviation as the input and produces probability of packet drop as its output.

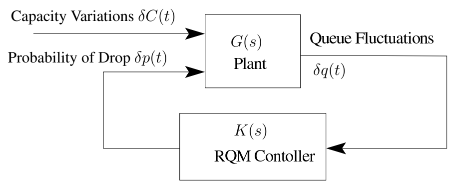

Fig. 1. AQM Controller Model for a Wireless Network

We now present a discretized linear model. We begin by first writing the above set of equations (Equation (2)) in state space like form. Here the state x at time t is $(\delta W(t), \delta q(t))$ and the input u is $(\delta p, \delta C)$.

$$\begin{aligned} \dot{x} &= \begin{bmatrix} \frac{-NC_0}{(q_0+C_0T_p)^2} & 0 \\ \frac{NC_0}{q_0+C_0T_p} & \frac{-C_0}{q_0+C_0T_p} \end{bmatrix} \begin{bmatrix} \delta W(t) + \delta W(t - R_0) \\ \delta q(t) \end{bmatrix} \\ &\quad + \begin{bmatrix} \frac{-C_0(q_0+C_0T_p)}{2N^2} & 0 \\ 0 & \frac{q_0}{q_0+C_0T_p} \end{bmatrix} \begin{bmatrix} \delta p(t - R_0) \\ \delta C(t) \end{bmatrix}, \end{aligned} \tag{3}$$

and $y = \delta q(t) = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} \delta W(t) \\ \delta q(t) \end{bmatrix}$. After defining

$a_{11} := -\frac{-NC_0}{(q_0+C_0T_p)^2}$, $a_{21} := \frac{NC_0}{q_0+C_0T_p}$, $a_{22} := -\frac{C_0}{q_0+C_0T_p}$, $b_{11} := -\frac{C_0(q_0+C_0T_p)}{2N^2}$ and $b_{22} := \frac{q_0}{q_0+C_0T_p}$, we get the following state space like set of equations

$$\begin{aligned} \overset{\bullet}{x} &= \begin{bmatrix} a_{11} & 0 \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} \delta W(t)+\delta W(t-R_0) \\ \delta q(t) \end{bmatrix} + \begin{bmatrix} b_{11}\delta p(t-R_0) \\ b_{22}\delta C(t) \end{bmatrix}, \\ y &= \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} \delta W(t) \\ \delta q(t) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta p(t-R_0) \\ \delta C(t) \end{bmatrix}. \end{aligned} \quad (4)$$

Consider the Laplace transform of the signals and incorporate the time delay R_0 by introducing the operator e^{-sR_0}. The transfer function $G(s)$ of the plant from input signals $(\delta p, \delta C)$ to the output signal δq comes out to be

$$G(s) = \frac{\begin{bmatrix} a_{21}b_{11}e^{-sR_0} & (s-a_{11}(1+e^{-sR_0}))b_{22} \end{bmatrix}}{(s-a_{22})(s-a_{11}-a_{11}e^{-sR_0})}. \quad (5)$$

($G(s)$ acts on input $\begin{bmatrix} \delta p \\ \delta C \end{bmatrix}$ to give output δq).

In order to circumvent the difficulty of incorporating the irrational transfer function e^{-sR_0} into controller design methods, we propose a discretization of the continuous time model with a sampling period T_s such that R_0 is an integer multiple of T_s (say $R_0 = nT_s$). Then, the operator e^{-sR_0} can be discretized to obtain the operator z^{-n}. Discretizing the continuous time system is reasonable given that eventually AQM techniques are implemented digitally.

We use the bilinear transformation to discretize the transfer function in Equation (5). This transformation corresponds to the trapezoidal rule for approximating integration. Discretization of the transfer function $G(s)$ using the bilinear transformation translates to replacing s with $\frac{2}{T_s}(\frac{z-1}{z+1})$, and, as explained above, e^{-sR_0} by z^{-n}.

Let $\hat{G}(z)$ denote the plant transfer function of the discretized system. This discrete time LTI model $\hat{G}(z)$ is used for the purpose of designing a causal controller $K(z)$ that meets a suitable control objective, and $K(z)$ is implemented in the time domain for computing the current incremental probability of packet drop, δp.

III. Robust Queue Management

The AQM problem can now be formulated for the linear system $\hat{G}(z)$ as a disturbance attenuation problem: design a controller $K(z)$ that takes input δq and gives output δp such that the 'effect' of the disturbance input (capacity variation) δC on δq is sufficiently small. There are several ways in which this effect can be quantified; one of the ways is to use the so-called induced ℓ_2 norm of the transfer function. We use the ratio of the ℓ_2 norm[2] of the output signal Pd to the input signal d to define the $\mathcal{H}_\infty$ norm of a stable transfer function $P(z)$ as follows

$$\|P\|_{\mathcal{H}_\infty} := \sup_{d\in\ell_2,\ d\neq 0} \frac{\|Pd\|_2}{\|d\|_2}.$$

Thus choosing a controller to reduce the closed loop system's $\mathcal{H}_\infty$ norm amounts to reducing the gain that the worst disturbance input signal δC can have on the queue length deviation δq in the closed loop system; this is called $\mathcal{H}_\infty$ control.

A controller that ensures that the closed loop system has a sufficiently small $\mathcal{H}_\infty$ norm tends to have good robustness properties. More precisely, the nominal plant $\hat{G}$ is only an approximation of the actual TCP network and hence the actual system can be considered as a perturbation of $\hat{G}$. Our controller (designed using the nominal plant) is required to be robust to these perturbations if the objective of queue stabilization is to be achieved for the actual plant also. Moreover, due to the time varying nature of wireless link capacity, we have considered capacity as an exogenous disturbance input. Thus this paper's proposed method provides for Robust Queue Management (RQM) of the buffer in the wireless router to address congestion.

In this paper, we assume slowly varying wireless channel. Thus instead of considering all disturbance sequences, we consider only low frequency variations in the capacity; this is achieved by including a digital low pass filter between δC and the plant. A simple first order filter with one pole and no zero with the pole location at $(1-\omega_c)$, where ω_c is the cutoff frequency, can be included. This allows capacity variations of frequencies at most ω_c to reach the plant.

Thus RQM scheme uses the instantaneous queue measurement and calculates a suitable probability of randomly dropping packets from the queue to achieve a robust control of queue length. We end this section with a quick summary of the major steps involved in RQM scheme and then proceed to evaluate performance results of simulations using this method in the following section.

1) Knowing the operating points (W_0, q_0, p_0, C_0), fix the desired queue length in buffer to q_0 and the probability of drop to p_0, using the equilibrium equations.
2) Design an $\mathcal{H}_\infty$ controller (using Scilab/Matlab) to get the digital controller $K(z)$.
3) Measure the instantaneous queue length of the router at every packet arrival and calculate the deviation from the set point value q_0 which gives rise to δq.
4) Use $K(z)$ to compute the (incremental) probability of drop δp corresponding to δq by $K(z)\widehat{\delta q}(z) =: \widehat{\delta p}(z)$ where $\widehat{\delta q}(z)$ and $\widehat{\delta p}(z)$ are the Z-transforms of signals δq and δp after sampling.
5) Update the probability of drop for the current instant, $p(k) = p_0 + \delta p(k)$, and drop the packets randomly from the router buffer with probability $p(k)$.

IV. Simulations and Performance Results

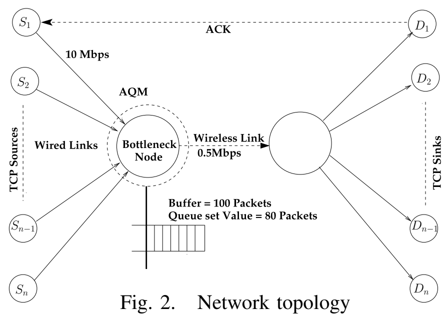

Fig. 2. Network topology

We now design the RQM controller for an example network. Figure 2 shows a dumb-bell shaped network topology, which we have used for simulation. The link marked 'Wireless link' in the figure is assumed to be a slowly varying link.

[2]The ℓ_2 norm of a sequence $\{d(\cdot)\}$ is defined as $\|d\|_2^2 := \sum_{k\in\mathbb{Z}} |d(k)|^2$, where $\mathbb{Z}$ denotes all integers. The space of all such square summable sequences is also denoted by ℓ_2.

$(S_1, ..., S_n)$ are TCP sources which are connected to router with links having capacity of 10 Mbps and propagation delay of 10 ms. $(D_1, ..., D_n)$ are TCP sink nodes. S_1 and D_1 form a TCP source-sink pair, S_2 and D_2 form another source-sink pair and so on. TCP sinks are connected to intermediate node with links having capacity of 10Mbps and propagation delay of 10 ms. The outgoing link capacity of the wireless link is set with maximum value of 0.5 Mbps. Router node is the bottleneck node with limited buffer size of 100 packets and the set value for buffer is 80 packets. The packet size is fixed and set at 1500 bytes. Simulations have been carried out on network simulator NS version 2.1b2. During the simulations, the capacity of the wireless link varies randomly from 0.1 Mbps to 0.5 Mbps with nominal value being 0.3 Mbps and this capacity changes once a second. From the network parameters, the round-trip time can be computed to be 200 ms. The number of TCP sources N is assumed to be 50. With 1500 bytes as packet size, C_0 is 200 packets/second. Desired queue set point is 80 packets. This gives rise to equilibrium point $W_0 = 2.4$ and $p_0 = 0.34$. For these values, the RQM controller is designed using Matlab. We have used a low pass filter with pole placed at $p = 0.6$. The controller transfer function we obtained is

$$K(z) = \frac{28.63z^3 - 2.337z^2 - 23.92z + 7.053}{1000(z^3 + 2.491z^2 + 2.04z + 0.5325)}.$$

which has all its poles within the unit circle.

A. Performance of RED

For illustrating the efficacy of our proposed algorithm, we first perform the simulations with RED in the presence of capacity variations. RED performance has been tested for three values of $\max_p$: $0.1, 0.2$ and 0.5. The minimum and maximum thresholds have been set to 30 and 90 packets, respectively.

Figure 3 illustrates fluctuations in queue length for $\max_p = 0.1$. Notice that the queue fluctuations are high and RED is not able to achieve the desired queue control. (Our simulations with other settings of $\max_p$ show worse queue control.)

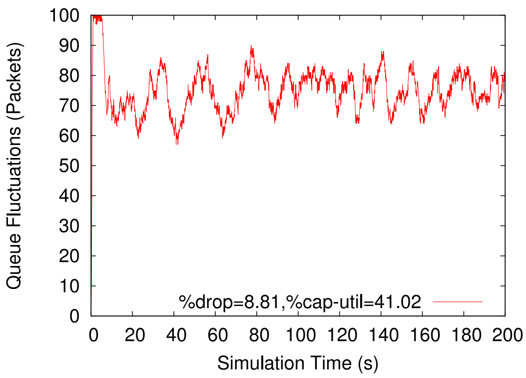

Fig. 3. RED Queue Fluctuations with $\max_p$=0.1

B. Performance of RQM

In this subsection, we depict the simulation results for the proposed RQM. In order to test the robustness of our scheme, we also check how the controller designed for one set of parameters performs when changes in these parameters take place. Figures 4 to 7 correspond to the RQM controller (designed actually for queue length 80) being utilized to stabilize the queue length at 90, 80, 70, and 45 packets, respectively. The controller's ability to control the queue fluctuations for this wide range of queue set points: from 90 to 45 packets, demonstrates its robustness.

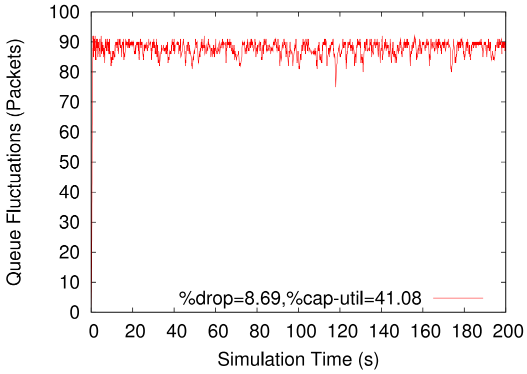

Fig. 4. RQM: Queue Fluctuations (desired queue length: 90)

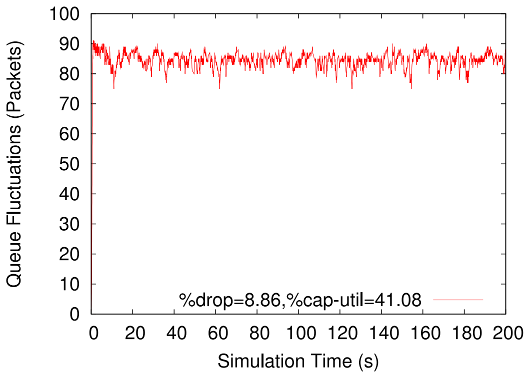

Fig. 5. RQM: Queue Fluctuations (desired queue length: 80)

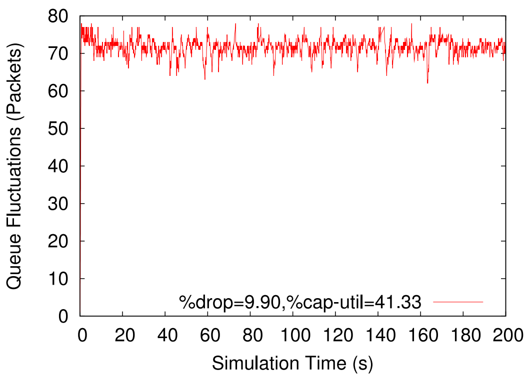

Fig. 6. RQM: Queue Fluctuations (desired queue length: 70)

It can be seen that queue control is not lost even when the queue set point is kept at the extreme end of the buffer: 90 packets. On the other hand, queue length is very well-regulated at the lower operating point values also like 45

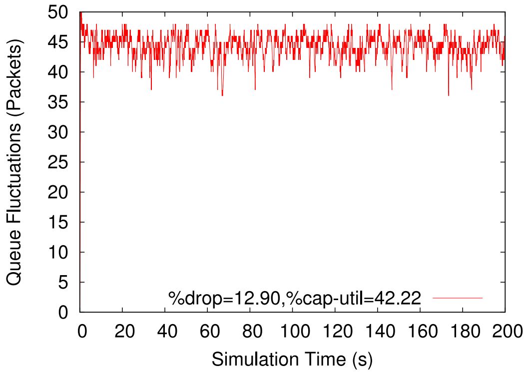

Fig. 7. RQM: Queue Fluctuations (desired queue length: 45)

packets. However, when we set the operating point at lower value, there is a slight increase in the packet loss.

We also check the RQM controller's performance in the scenario that the number of TCP sources N changes from $N_0 = 50$, the number for which the controller was designed. During simulations, N is varied from 50 to 250. The queue set point has been kept at $q_0 = 80$ packets. A study of queue length fluctuations reveals that queue control is not lost even when the number of sources is increased to as high as 250. For the lack of space, we have not plotted these graphs but they tend to be similar to Figures 4 to 7 albeit with different packet loss and capacity utilization. Figures 8 and 9 capture how packet loss and capacity utilization depend on the number of sources (while using a controller designed for $N_0 = 50$).

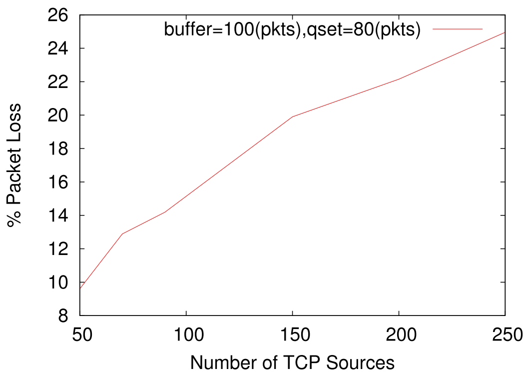

Fig. 8. RQM: Packet Loss (controller designed for $N = 50$)

V. Conclusions

Active Queue Management has been studied extensively for wired network. In this paper, we have proposed the design of robust AQM for wireless link. Specifically, we have proposed a way to address capacity variations of the wireless link by treating it as an external input. The round trip delay has been incorporated by discretizing the system and using the discrete time system for the design of a digital controller; we designed an $\mathcal{H}_\infty$ controller to achieve robust queue control.

Once designed, the controller can be used for fairly wide perturbations in the network parameters like load variations,

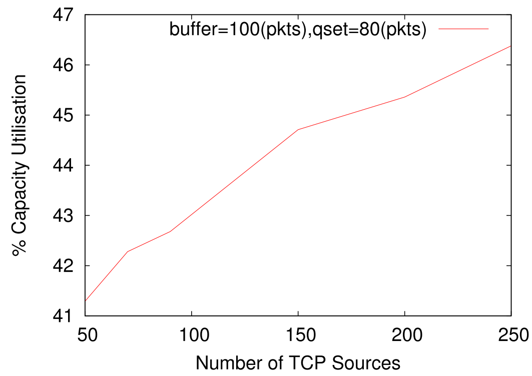

Fig. 9. RQM: Capacity Utilisation (controller designed for $N = 50$)

without requiring any online tuning or adjustment of parameters by the user. This feature gives a significant advantage in the context of wireless link.

References

[1] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Transactions on Networking*, vol. 1, no. 4, pp. 397–413, 1993.
[2] M. May, J. Bolot, C. Diot, and B. Lyles, "Reasons not to deploy RED," in *Proceedings of IWQoS*, pp. 260–262, 1999.
[3] T. Ott, T. Lakshman, and L. Wong, "SRED: Stabilized RED," in *Proceedings of IEEE INFOCOM*, pp. 1346–1355, 1999.
[4] W. Feng, D. Kandlur, D. Saha, and K. Shin, "BLUE: An alternative approach to active queue management algorithm," in *Proceedings of NOSSDAV*, pp. 41–50, 2001.
[5] W. Feng, D. Kandlur, D. Saha, and K. Shin, "A self-configuring RED gateway," in *Proceedings of IEEE INFOCOM*, pp. 1320–1328, 1999.
[6] S. L. S. Athuraliya, V.H. Li and Q. Yin, "REM: Active queue management," *IEEE Network*, vol. 15, pp. 48–53, 1999.
[7] S. Kunniyur and R. Srikant, "An adaptive virtual queue (AVQ) algorithm for active queue management," *IEEE/ACM Transactions on Networking*, vol. 12, pp. 286–299, 2004.
[8] S. Bohacek, K. Shah, G. Arce, and M. Davis, "Signal processing challenges in active queue management," *IEEE Signal Processing Magazine*, pp. 69–79, September 2004.
[9] 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 *Proceedings of SIGCOMM*, pp. 151–160, 2000.
[10] C. Hollot, V. Misra, D. Towsley, and W. Gong, "Analysis and design of controllers for AQM routers supporting TCP flows," *IEEE Transactions on Automatic Control*, vol. 47, pp. 945–959, June 2002.
[11] F. Kelly, "Mathematics unlimited - 2001 and beyond," in *Mathematical Modelling of the Internet* (B. Engquist and W. Schmid, eds.), Berlin: Springer-Verlag, 2001.
[12] S. Low, F. Paganini, and J. Doyle, "Internet congestion control," *IEEE Control Systems Magazine*, vol. 22, pp. 28–43, December 2004.
[13] P. Quet and H. Ozbay, "On the design of AQM supporting TCP flows using robust control theory," *IEEE Transactions on Automatic Control*, vol. 49, pp. 1031–1036, 2004.
[14] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge University Press, 2005.
[15] R. Johari and D. Tan, "End-to-end congestion control for the Internet: delays and stability," *IEEE/ACM Transactions on Networking*, vol. 9, pp. 818–832, December 2001.
[16] L. Massoulie, "Stability of distributed congestion control with heterogeneous feedback delays," Technical Report, Microsoft Research, 2000.
[17] G. Vinnicombe, "On the stability of end-to-end congeston control for the Internet," Technical Report, University of Cambridge, 2001.
[18] F. Zheng and J. Nelson, "An $\mathcal{H}_\infty$ approach to congestion control design for AQM routers supporting TCP flows in wireless access networks," *Computer Networks: the International Journal of Computer and Telecommunications Networking*, vol. 51, pp. 1684–1704, 2007.
[19] F. Zheng, Q. Wang, and T. Lee, "A heuristic approach to solving a class of bilinear matrix inequality problems," *Systems & Control Letters*, vol. 47, pp. 111–119, 2002.