A GAIN SCHEDULING APPROACH TO ACTIVE QUEUE MANAGEMENT

Sabato Manfredi * **and Mario di Bernardo** *

* *Dipartimento di Informatica e Sistemistica*
Università di Napoli Federico II, Napoli, Italy.
E-mail:{sabato.manfredi,mario.dibernardo}@unina.it

Abstract: The aim of this paper is to discuss the use of gain scheduling to improve the performance of classical AQM controllers. We show that by using gain scheduling it is possible to synthesize controllers able to better cope with unwanted variations of the network operating regime; for example variations of the round-trip time, the load or the buffer capacities. Simulations are provided to compare the performance of the gain scheduled controller with that of PI-based AQM control scheme.

KeyWords: active queue management, communication network control.

1. INTRODUCTION AND PROBLEM MOTIVATION

Recently, novel congestion Active Queue Management (AQM) control strategies have been proposed to improve the performance of standard TCP-RED algorithms (see for example (C.V.Hollot *et al.*, 2002) and references therein). It has been suggested that feedback strategies more effective than simple drop-tail control mechanisms are required at intermediate routers to complement endpoint congestion control strategies. Active Queue Management (AQM) schemes have been proposed in order to deliver preemptively congestion notification to the source for reducing its transmission rate and therefore avoiding buffer overflow(C.V.Hollot *et al.*, 2002; Y.Chait *et al.*, 2001; Aweya *et al.*, 2001; Low *et al.*, 2002; Park *et al.*, 2003). Nevertheless, AQM schemes based on classical controllers (i.e. PI, PD, PID) perform poorly in the presence of variations of the network nominal operating regime. Several robust (S.Manfredi *et al.*, 2004; Quet and Ozbay, 2004) and adaptive AQM schemes (Zhang *et al.*, 2003; Kunniyur and Srikant, 2001; Y.Gao *et al.*, 2002; W.Wu *et al.*, 2001) have been proposed to deal with network uncertainties caused by variations of the load, N, the round trip time R, the link capacity, C, or the presence of unresponsive flows such as those generated by UDP sources.

Here we propose a gain scheduling approach to design adaptive AQM schemes to guarantee acceptable performance in different operative scenarios. The proposed controller is based on an integral action for nonlinear systems with its parameters continuously depending on a scheduling aggregate throughput variable, N/R, estimated locally at the congested router. We note that the proposed controller is derived by using the fluid model of N homogeneous sources accessing a single bottleneck presented in (C.V.Hollot *et al.*, 2002). The system of N sources with heterogeneous round trip times R_k, $k = 1..N$ on a single bottleneck behaves on average as a system with N flows each having an identical equivalent round trip time $R_{eq} = \frac{1}{N}\sum_{k=1}^{N}\frac{1}{R_k}$ (C.V.Hollot *et al.*, 2001). So the choice of load N/R as scheduling variable can also mitigate on average the effects of heterogeneous sources. Note that we assume R is not estimated online but its possible variations are taken into account in the controller stability margin design by considering its worst value.

We wish to emphasize that the main of this paper is to test the suitability of a gain scheduling approach for AQM control and propose its innovative use for congestion control. Thus simulations will be reported which were carried out in SIMULINK/Matlab. The NS2 implementation of the control law is currently under investigation and is not discussed in the paper.

2. A FLUID MODEL OF TCP BEHAVIOR

A fluid model of TCP dynamical behaviour was derived in (V.Misra *et al.*, 2000) using the theory of stochastic differential equations. The model describes the evolution of the average characteristic variables on the network such as the average TCP window size and the average queue length. Extensive simulations in *Network-Simulator (NS)*-2 have shown that the model captures indeed the qualitative behaviour of TCP traffic flows. Hence, it is particularly useful for the design of innovative Active Queue Management control schemes for TCP-controlled flows using a control theory approach.
Under the assumption of neglecting the TCP timeout, the model is described by the following set of nonlinear coupled ODEs:

$$\begin{aligned}\dot{W}(t) &= \frac{1}{R(t)} - \frac{W(t)}{2}\frac{W(t-R(t))}{R(t-R(t))}p(t-R(t)) \\ \dot{q} &= \begin{cases} -C + \dfrac{N(t)}{R(t)}W(t), & q > 0; \\ \max\{0, -C + \dfrac{N(t)}{R(t)}W(t)\}, & q = 0. \end{cases}\end{aligned} \tag{1}$$

where W is the average TCP window size (packets), q the average queue length (packets), T_p the propagation delay, R the transmission round-trip time ($R = \frac{q}{C} + T_p$), C the link capacity (packets/s), N the number of TCP sessions and p the probability of a packet being marked. All variables are assumed non negative. If we assume $N(t) = N_0$, $R(t) = R_0$ and $C = C_0$ to be the nominal values of R,N and C, we can then linearize the dynamic model (1) about the operating point (W_0, q_0, p_0) where W_0, q_0 are the state values of the equilibrium of interest when the input p is set equal to p_0 (see (C.V.Hollot *et al.*, 2002; C.V.Hollot *et al.*, 2001) for further details). Hence, we obtain the following set of delay differential equations (DDEs):

$$\begin{aligned}\delta\dot{W}(t) &= -\frac{N_0}{R_0^2 C_0}(\delta W(t) + \delta W(t-R_0)) \\ &\quad -\frac{R_0 C_0^2}{2N_0^2}\delta p(t-R_0) \\ \delta\dot{q}(t) &= \frac{N_0}{R_0}\delta W(t) - \frac{1}{R_0}\delta q(t),\end{aligned} \tag{2}$$

where $\delta W \doteq W - W_0$, $\delta q \doteq q - q_0$, $\delta p \doteq p - p_0$. In what follows we will consider the same parameter values used in (C.V.Hollot *et al.*, 2002), i.e. $C_0 = 3750$ packets/s, $R_0 = 0.246$ s, $N_0 = 60$, $q_{max} = 800$ packets corresponding to the steady-state operating regime $W_0 = 15$ packets, $q_0 = 175$ packets, $p_0 = 0.008$. As shown in (C.V.Hollot *et al.*, 2002), the delay R_0 in the state term $\delta W(t - R_0)$ in (2) can be neglected when $W_0 \gg 1$. This is a realistic assumption for a typical network operating condition and so for control design purposes we refer to the simplified model:

$$\delta\dot{W}(t) = -\frac{2N_0}{R_0^2 C_0}\delta W(t) - \frac{R_0 C_0^2}{2N_0^2}\delta p(t-R_0) \tag{3}$$

$$\delta\dot{q}(t) = \frac{N_0}{R_0}\delta W(t) - \frac{1}{R_0}\delta q(t). \tag{4}$$

The AQM control strategy introduced in the paper is designed on the previous model. The robust nature of the strategy presented here will not only reduce the sensitivity to network parameters but also eliminate inaccuracies due to the use of the linear model (4). Note that we will validate the control law designed using (4) on the nonlinear model (1).

3. GAIN SCHEDULING CONTROL

The main limitation of fixed linear controllers is that the controller guarantees the performances both in terms of stability margins and responsiveness only around a single operative equilibrium point. To overcome this limitation, we will synthesize a gain scheduling AQM controller for maintaining an acceptable performance also in the presence of network condition variations. Given the nonlinear model (1), we consider the static scheduled integral control (Khalil, 2002):

$$\begin{aligned}\dot{\sigma}(t) &= \mu \\ p(t) &= M(\alpha)\sigma + M_3(\alpha)e\end{aligned} \tag{5}$$

where $\mu = \begin{bmatrix} e \\ \dot{q} \end{bmatrix}$, $M = \begin{bmatrix} M_1 & M_2 \end{bmatrix}$ and $e = q - q_0$.

Under the action of such controller, the linearized closed loop matrix is

$$A_c = \begin{bmatrix} -\frac{2N_0}{R_0^2 C_0} & -\frac{R_0 C_0^2}{2N^2}(M_2+M_3) & -\frac{R_0 C_0^2}{2N^2} M_1 \\ \frac{N_0}{R_0} & -\frac{1}{R_0} & 0 \\ 0 & 1 & 0 \end{bmatrix};$$

and taking the load factor $\frac{N}{R}$ as the scheduling variable, we design the scheduling controller (5) to guarantee the unity gain crossover $w_g \simeq 0.5$rad/s and phase margin about 80 degree at each scheduled operating point. In so doing the controller gains depends continuously on the scheduling variable as:

$M_i = \frac{N}{R} * m_i + n_i$ with $i = 1, 2, 3,$

with m_i and n_i being quantities depending on the desired position of the closed-loop eigenvalues. The details of the derivation are given in Appendix A.

Note that the estimation of the load factor N/R is obtained using the rate formula given by

$$\frac{N}{R} = \frac{r}{W}.$$

We measure the incoming rate r at the router, while W is estimated by using the equilibrium formula

$$W = \sqrt{\frac{2}{p}}.$$

We note that W is the average window size of the flows accessing the bottleneck and it can be estimated also referring to the window observer introduced in (S.Manfredi *et al.*, 2004).

We note that from practical point of view the rate can be easily estimated at the router by using existing approach (for example Stoica and averaging algorithms) based on counting the packets incoming at the bottleneck.

3.1 Simulation Results

In what follows, the gain scheduling AQM strategy (GSAQM) is tested and compared with PI-AQM in different network scenarios. All the simulations presented here are carried out using Simulink on the network topology introduced in (C.V.Hollot *et al.*, 2002). The round trip time variations are those obtained by varying the round trip propagation delay T_p while the parameters of the PI controller are those suggested in (C.V.Hollot *et al.*, 2002). In particular we will validate the effectiveness of GSAQM on four different scenarios: 1) $N(t) = N_0$, $R(t) = R_0$ and $C = C_0$, 2) highly varying $N(t)$ 3) highly varying $N(t)$ and slowly varying $R(t)$, 4) presence of unresponsive UDP flows.

3.2 ***Case 1****: $N(t) = N_0$, $R(t) = R_0$ and $C(t) = C_0$*

We consider the GSAQM and PI performance when: $C = 3750$ pakets/s, $R_0 = 0.246$ s, $N = 60$, $q_{max} = 800$ packets, $W_0 = 15$ packets, $q_0 = 175$ packets, $p_0 = 0.008$. Fig. 1shows that the GSAQM presents no overshoot and fast convergence to set-point with respect to PI controller.

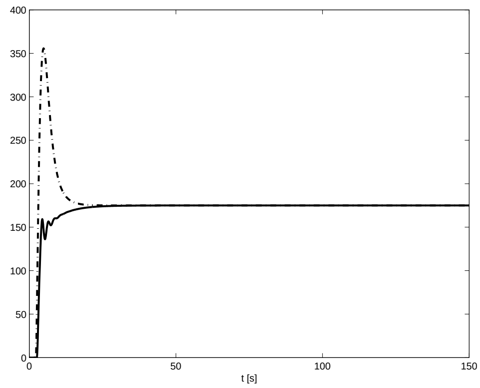

Fig. 1. Case 1: Time evolution of the queue length for the PI-AQM (dotted line) and GSAQM schemes applied to the nonlinear model;

3.3 ***Case 2****: highly varying $N(t)$*

Now we consider larger variations of the load N variable as normal random signal $\in (150, 450)$, period 10 sec.

The evolution of the queue length for the robust GSAQM scheme and the PI controller, depicted in Fig.2 shows that the GSAQM scheme presents an improved transient and steady-state performance. Namely, the PI controller shows consistent variations of the queue length (between minimum and maxim values) and so of the jitter delay. Moreover, we observe intervals where packets are lost due to queue overflows ($q \geq 800$). The GSAQM, instead, guarantees an acceptable steady-state response and eliminates all packet losses over the time range of interest while reducing queue variance. The resulting estimation of the aggregate throughput variable N/R presents a good agreement with real value (Fig. 3).

We note as the advantages of GSAQM with respect to PI are remarked for higher mean load N (500 packets) in particular in term of queue regulation and packet loss (Fig. 4) while maintaining acceptable the N/R estimation (Fig. 5).

3.4 ***Case 3****: highly varying $N(t)$ and slowly varying $R(t)$*

Now variations of the round-trip propagation delay ($T_p \in (0.015, 0.42)$) are considered together

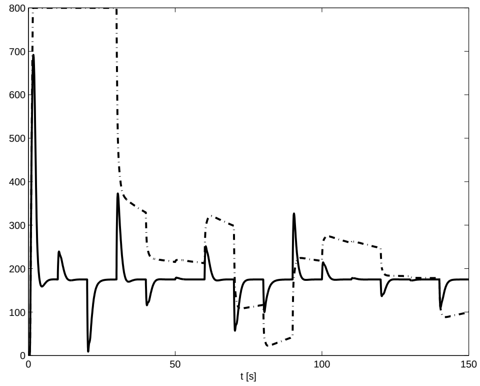

Fig. 2. Case 2 - Highly N variations: Time evolution of the queue length for the PI-AQM (dotted line) and GSAQM schemes applied to the nonlinear model

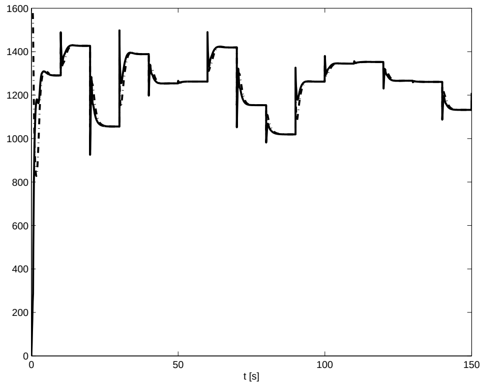

Fig. 3. Case 2 - Highly N variations: Time evolution of the real (solid line) and estimated aggregate throughput N/R (dotted line) on nonlinear system.

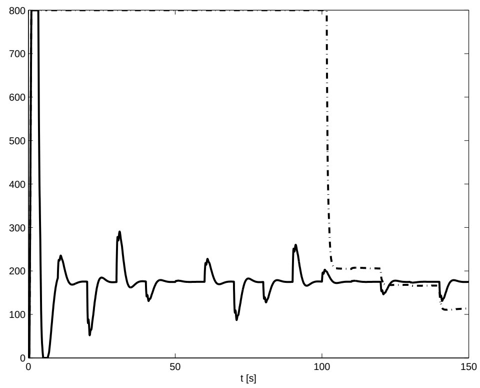

Fig. 4. Case 2 - Highly mean $N = 500$ variations: Time evolution of the queue length for the PI-AQM (dotted line) and GSAQM schemes applied to the nonlinear model

with load variations (N variable $\in (250, 750)$ representing a strong congestion scenario.

We can observe in Fig. 6 that also in this case RRAQM performs significantly better than the PI scheme with no packet losses, low queue excursion, better queue utilization and resulting good N/R estimation (Fig.7).

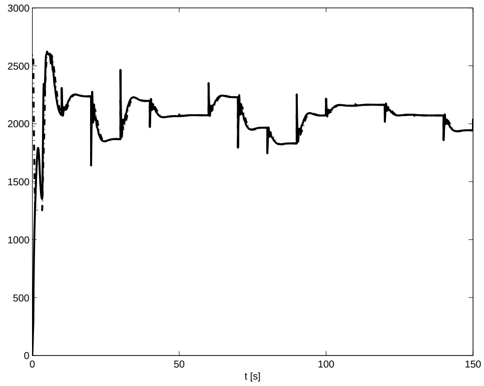

Fig. 5. Case 2 - Highly $N = 500$ variations: Time evolution of the real (solid line) and estimated aggregate throughput N/R (dotted line) on nonlinear system.

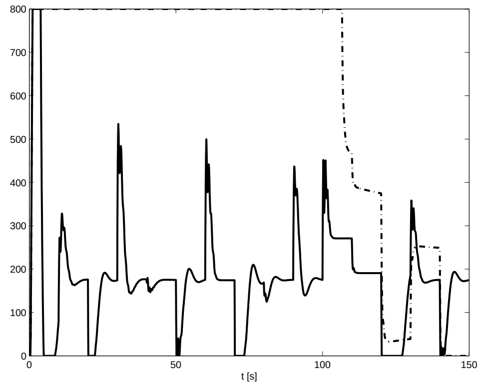

Fig. 6. Case 3 - Highly mean $N = 500$ variations: Time evolution of the queue length for the PI-AQM (dotted line) and GSAQM schemes applied to the nonlinear model

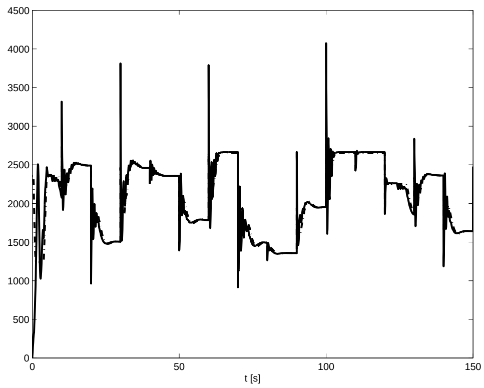

Fig. 7. Case 3 Time evolution of the real (solid line) and estimated aggregate throughput N/R (dotted line) on nonlinear system.

Similar results are observed when a slow link capacity variation of $C \in (3100, 3750)$ the variation in C is obtained by adding a square wave signal (period 30 s, amplitude of 700 packets) to a normal random signal with mean equal to 3750 packet, variance equal to 300 and sample time 0.1. is added to previous scenario.

3.5 ***Case 4***:*presence of unresponsive flows*

We consider the presence of ON-OFF unresponsive UDP flows simulated adding a square wave signal (period 30 s, amplitude of 200 packets) to a existing tcp/ip flows varying as normal random signal with mean equal to 400 packet, variance equal to 200 and sample time 10. We note as also the gain scheduling AQM controller is effective for compensating the effects of UDP flows on queue dynamics, reduced queue variance while avoiding packet loss (Fig.8).

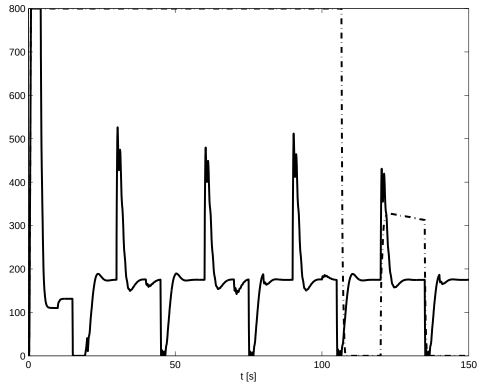

Fig. 8. Time evolution of the queue length for the PI-AQM (dotted line) and GSAQM schemes applied to the nonlinear model

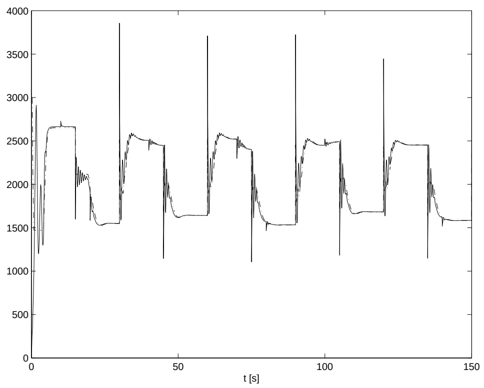

Fig. 9. Case 4: Presence of UDP flows. Time evolution of the real (solid line) and estimated aggregate throughput N/R (dotted line) on nonlinear system.

4. CONCLUSIONS

We have shown that it is possible to improve the performance of classical PI AQM control strategies under strong variations of the the load, round trip time and in the presence of unresponsive flows. As such variations are bound to occur in practical applications, the need was outlined for adaptive control scheme such as gain scheduling control. This paper has given a contribution in this direction presenting a possible approach to introduce an element of adaptivity in the AQM control strategy. The numerical simulations confirmed that the new scheme exhibits a better performance when compared to the available PI-based AQM control strategy. In particular, the queue length dynamics was shown to exhibit less fluctuations, no packet losses and good compensation to unresponsive UDP flows. We remark that the controller is implementable locally at the router and needs only of the rate estimation easy obtainable by Stoica or averaging algorithms. Besides the presented approach can be implemented as a distributed algorithm in a multi-bottleneck scenario. Current work is aimed at testing the strategy presented in this paper both numerically (via NS-2 simulations) and experimentally and extending the approach to multi-bottleneck scenario. Preliminary simulations in NS-2 present encouraging results.

Appendix A

Diagonalizing the linearized closed loop matrix A_c, and imposing the equality between the closed loop eigenvalues and the desired one, $\lambda_i, i = 1 \dots 3$, we can derive the controller gains M_1, M_2 and M_3 as function of scheduling variable N/R (for the sake of brevity we omit here the detailed algebraic manipulation but report only the final equalities):

$$0.41\frac{N}{R}[\frac{1.92}{RC}(\frac{1}{R} - 3.46) + \frac{RC^2}{N^2}0.48(M_2+$$
$$M_3 - 0.14M_1)] + 18.26(0.0857 - \frac{0.0754}{R}) - 0.13 = \lambda_1$$
$$4\frac{N}{R}[-\frac{1.06}{RC}(\frac{1}{R} - 0.62) + \frac{RC^2}{N^2}0.43(M_2+$$
$$M_3 - 0.27M_1)] + 18.15(2.856 - \frac{0.716}{R}) + 3.78 = \lambda_2$$
$$3.7\frac{N}{R}[\frac{0.96}{RC}(\frac{1}{R} - 0.52)+$$
$$\frac{RC^2}{N^2}0.23(M_2 + M_3 - 0.44M_1)]+$$
$$1.32(2.9154 - \frac{0.7155}{R}) - 3.66 = \lambda_3$$

After some algebra, the above expressions can be rewritten a $M_i = \frac{N}{R} * m_i + n_i$, $i = 1, 2, 3$, with m_i and n_i being quantities depending from the desiderated eigenvalues λ_i.

REFERENCES

Aweya, Ouelette and D.Y. Montuno (2001). A control theoretic approach to active queue management.. *Computer Newtworks* pp. 36, 203–235.

C.V.Hollot, V.Misra, D.Towsley and W.Gong (2001). A control theoretic analisys of red.. *IEEE INFOCOM 2001.*

C.V.Hollot, V.Misra, D.Towsley and W.Gong (2002). Analysis and design of controllers for aqm routers supporting tcp flows. *IEEE ACM Trans on Automatic Control* pp. Vol 47, no. 6.

Khalil (2002). Non linear systems. *Prentice Hall*.

Kunniyur, S. and R. Srikant (2001). Analysis and design of an adaptive virtual queue (avq) algorithm for active queue management.. *In Proc. of ACM SIGCOMM*.

Low, S, F.Paganini and J.C.Doyle (2002). Internet congestion control. *IEEE Control Systems Magazine* pp. 28–43.

Park, E.C., K.J. Park H. Lim and C.H. Choi (2003). Analysis and design of the virtual rate control alghorithm for stabilization queues on tcp networks. *Comput. Networks,* pp. Vol.44, N.1, pp.17–41.

Quet, P.F. and H. Ozbay (2004). On the design of aqm supporting tcp flows using robust control theory. *Transactions on Automatic Control* pp. Vol 49, no. 6, pp. 1031–1036.

S.Manfredi, M. di Bernardo and F. Garofalo (2004). Robust output feedback active queue management control in tcp networks. *Accepted for publication, Conference on Decision and Control 2004, Bhamas,*.

V.Misra, W.B.Gong and D.Towsley (2000). Fluid-based analysis of a network of aqm routers supporting tcp flows with an application to red. *ACM SIGCOMM*.

W.Wu, Y.Ren and X.Shan (2001). A self-configuring pi controller for active queue management.. *In Asia-Pacific Conference on Communications (APCC)*.

Y.Chait, C.V.Hollot, V.Misra, S.Oldak, D.Towsley and W.Gong (2001). Fixed and adaptive model-based controllers for active queue management. *American Control Conference* pp. pp 2981–2986.

Y.Gao, G.He and J.C.J.Hou (2002). On leveraging traffic predictability in active queue management.. *In Proceedings of IEEE/INFOCOM*.

Zhang, H., C.V. Hollot and V. Misra (2003). A self-tuning structure for adaptation in tcp/aqm networks. *Figure 3: Instantaneous Queue Lengths. Technical Report, University of Massachusetts Amherst, ftp://gaia.cs.umass.edu/pub/Zhang03 staqm.pdf*.