Tech Report: Throughput Guarantees for TCP Flows Using Adaptive Two Color Marking and Multi-Level AQM

YC
Technical Note DACS008
Department of Mechanical and Industrial Engineering
University of Massachusetts Amherst

December 2001

Abstract

This report provides technical details missing from the INFOCOMM '02 paper [1]. The details are related to the design and analysis of the control system.

1 Model

Our starting point is the fluid-flow model developed in [2] for modeling TCP flows and AQM routers. In this section we will extend this model to account for two-color marking at the network edge and multi-level active queue management (AQM) running at the core. To begin, we assume a single edge router serving m sets of aggregate flows with each having N_i identical TCP flows. Each aggregate has a token bucket with rate A_i and size $b_i >> 1$. The aggregation of these TCP flows feed a core router with link capacity C. At time $t > 0$, this router has queue length $q(t)$. At time $t > 0$, each TCP flow is characterized by its window size $W_i(t)$ and average round-trip time

$$R_i(t) \stackrel{\triangle}{=} T_i + \frac{q(t)}{C}$$

where T_i is the propagation delay. The sending rate r_i of an edge is

$$r_i = \frac{N_i W_i(t)}{R_i(t)}.$$

The fluid flow model is described by $m + 1$ coupled differential equations; one equation for each of the m TCP window dynamics and one for the AQM router. The differential equation for the AQM router is given by

$$\frac{dq(t)}{dt} = -C + \sum_{i=1}^{m} r_i \tag{1}$$

while each TCP window satisfies

$$\frac{dW_i(t)}{dt} = \frac{1}{R_i(t)} - \frac{W_i(t)W_i(t - R_i(t))}{2R_i(t - R_i(t))} p_i(t - R_i(t)) \tag{2}$$

where $p_i(t)$ denotes the probability that a mark is generated for this aggregate flow.

To finish up, we model the color-marking process at the i-th edge and the multi-AQM action at the core. To model coloring, we let $f_i^g(t)$ be the fraction of fluid marked green; i.e.,

$$f_i^g(t) = \min\left\{1, \frac{A_i(t)}{r_i(t)}\right\}$$

and $1 - f_i^g(t)$ the red fraction. At the core, we let $p_g(t)$ and $p_r(t)$ denote the probabilities that marks are generated for the green and red fluids, respectively.[1] Consistent with Diffserv, we assume that $0 \leq p_g(t) < p_r(t) \leq 1$. Probability $p_i(t)$ is then related to the green and red marks by

$$p_i(t) = f_i^g(t)p_g(t) + (1 - f_i^g(t))p_r(t).$$

Let $\tilde{r}_i$ denote the minimum guaranteed sending rate (MGR) for the i-th edge (aggregate). We say that the router is over-provisioned if $\sum_{i=1}^m \tilde{r}_i \leq C$ and under-provisioned if $\sum_{i=1}^m \tilde{r}_i > C$. Last, we say that it is exactly-provisioned if $\sum_{i=1}^m \tilde{r}_i = C$. The objective of this paper is to develop control strategies at both the core and edges to ensure that the edge sending rates r_i $(1 \leq i \leq m)$ meet or exceed their respective MGRs when the system is over-provisioned. In the next section we address the steady-state feasibility problem; namely, determine whether values exist for $\{f_i^g\}$ and $[p_g, p_r]$ such that the sending rates meet the MGRs

[1] More precisely, marks are embedded in the fluid as a time varying Poisson process, and the product of p_g and p_r with the green and red fluid throughputs respectively determine the intensity of this Poisson process

2 An Architecture for Providing Minimum Throughputs

The purpose of this section is to suggest control architectures for realizing the potential of DiffServ. These controllers act upon measured send rates r_i and queue length q to produce updated values of bucket rates A_i and marking probabilities p_r and p_g. Finally, before delving into the structure and design of such controllers, it is worthwhile to note that this theorem also identifies networks that *cannot achieve* a given set of MGRs; regardless of control scheme.

We now again consider the system of nonlinear differential equations written explicitly in terms of the bucket rate A_i:

$$\begin{aligned}
\dot{q} &= -C + \sum_{i=1}^{m} \frac{N_i W_i(t)}{R_i(t)} \\
&\stackrel{\triangle}{=} f(q, W_i, p_g, p_r, A_i); \\
\dot{W}_i &= \frac{1}{R_i(t)} - \frac{W_i(t) W_i(t - R_i(t))}{2 R_i(t - R_i(t))} p_i(t) \\
&\stackrel{\triangle}{=} g_i(q, W_i, p_g, p_r, A_i)
\end{aligned}$$

where

$$p_i(t) = \left(\frac{A_i}{r_i(t)} p_g(t - R_i(t)) + \ (1 - \frac{A_i}{r_i(t)}) p_r(t - R_i(t)) \right).$$

We follow the same design philosophy used in [4] and derive controllers based on linearized dynamics. At an operating point (q, W_i, p_g, p_r, A_i) we have

$$\begin{aligned}
0 &= -C + \sum_{i=1}^{m} \frac{N_i W_i}{R_i}; \\
0 &= 1 - 0.5 \left(\frac{A_i}{r_i} p_g + (1 - \frac{A_i}{r_i}) p_r \right) W_i^2; \\
R_i &= T_{pi} + \frac{q}{C}.
\end{aligned}$$

In the linearization process we make two approximation. First, we ignore the delay R in the term $W(t-R)/R(t-R)$ and secondly, assume that $\min\{1, \frac{A_i}{r_i}\} = \frac{A_i}{r_i}$. Linearization about the operating point gives

$$\begin{aligned}
\dot{\delta q}(t) &= \sum_{i=1}^{m} \frac{\partial f}{\partial W_i} \delta W_i(t); \\
\delta \dot{W}_i(t) &= \frac{\partial g_i}{\partial W_i} \delta W_i(t) + \frac{\partial g_i}{\partial p_g} \delta p_g(t - R_i) + \\
&\quad \frac{\partial g_i}{\partial p_r} \delta p_r(t - R_i) + \frac{\partial g_i}{\partial A_i} \delta A_i(t)
\end{aligned}$$

where

$$\delta q \equiv q(t) - q$$

$$\begin{aligned}
\delta W &\equiv W(t) - w \\
\delta p_g &\equiv p_g(t) - p_g \\
\delta p_r &\equiv p_r(t) - p_r \\
\delta A_i &\equiv A_i(t) - A_i.
\end{aligned}$$

The partial, evaluated at the operating point are:

$$\begin{aligned}
\frac{\partial f}{\partial q} &= -\sum_{i=1}^{m} \frac{r_i}{CR_i} \\
\frac{\partial f}{\partial W_i} &= \frac{N_i}{R_i} \\
\frac{\partial g_i}{\partial W_i} &= -\frac{A_i}{2N_i}(p_g - p_r) - \frac{W_i}{R_i} p_r \\
\frac{\partial g_i}{\partial p_r} &= \frac{W_i A_i}{2N_i} - \frac{W_i^2}{2R_i} \\
\frac{\partial g_i}{\partial p_g} &= -\frac{A_i W_i}{2N_i} \\
\frac{\partial g_i}{\partial A_i} &= -\frac{W_i}{2N_i}(p_g - p_r).
\end{aligned}$$

Taking the Laplace transform of the linearized equations yields

$$\begin{aligned}
\delta W_i(s) &= \frac{\frac{\partial g}{\partial A_i}}{s - \frac{\partial g}{\partial W_i}} \delta A_i(s) + \frac{\frac{\partial g}{\partial p_g}}{s - \frac{\partial g}{\partial W_i}} e^{-sR_i} \delta p_g(s) + \\
&\quad \frac{\frac{\partial g}{\partial p_r}}{s - \frac{\partial g}{\partial W_i}} e^{-sR_i} \delta p_r(s) \\
\delta q(s) &= \sum_{i=1}^{m} \frac{\frac{\partial f}{\partial W_i}}{s - \frac{\partial f}{\partial q}} \delta W_i(s).
\end{aligned}$$

These equations form the block diagram of the open-loop network shown in Figure 1.

2.1 Active Rate Management (ARM)

Similar to the introduction of the AQM in [4], we propose a feedback structure around the token bucket termed ARM. The need for this feedback is due to the result from [2] which showed that the resulting throughput may not be equal to the token bucket rate. The purpose of ARM is to regulate the token bucket rate A_i such that $r_i \geq \tilde{r}_i$ if capacity is available. Since our ARM compares an aggregate's sending rate to its bucket rate, it is necessary to construct an estimate for this sending rate. We follow the TSW (time slice window) procedure which consists of the following. The send rate is computed by measuring the number of sent packets over a fixed time period T. This value is then smooth by a low-pass filter. A fluid model for this dynamics is given by:

$$F(s) = \frac{a}{s+a} e^{-sT_{TSW}}$$

For this purpose, we introduce the feedback structure as shown in Fig. 2.

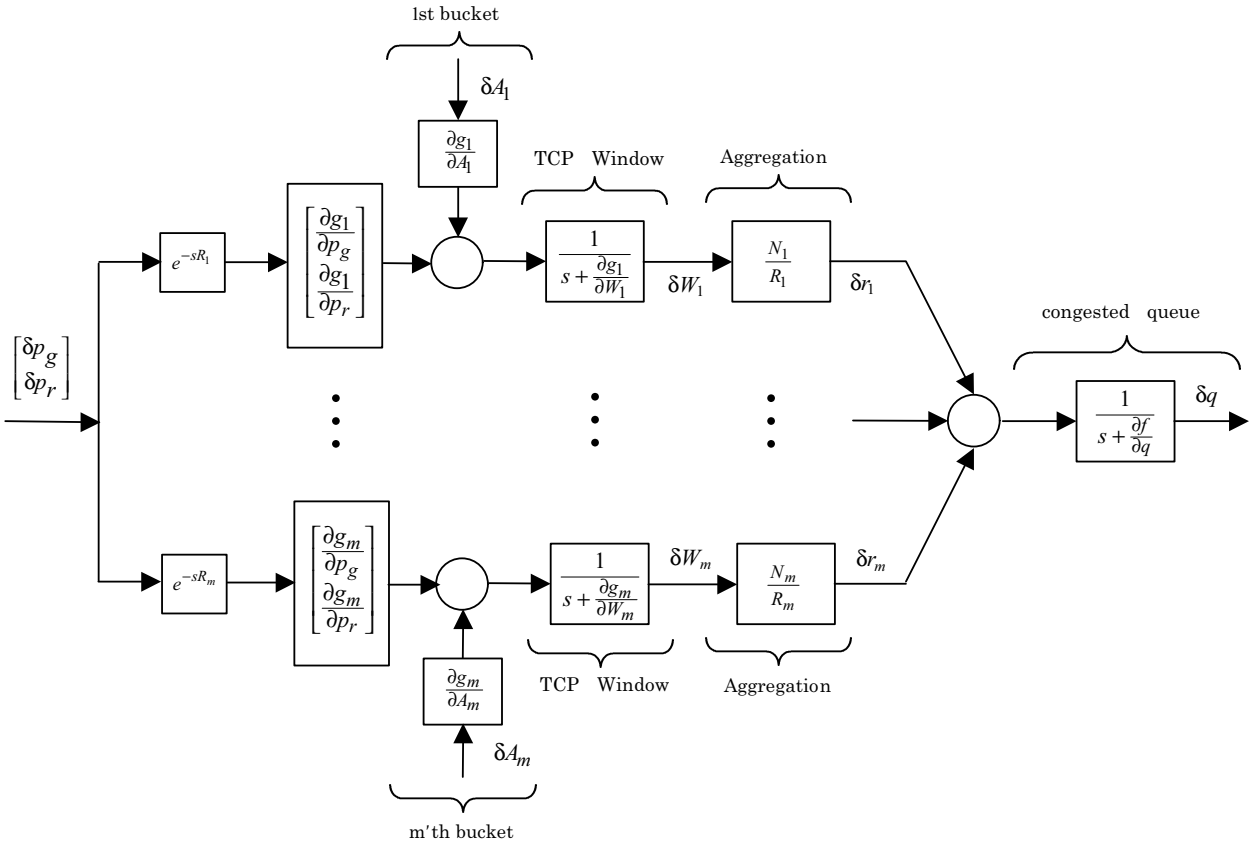

Figure 1: Block diagram of an open-loop DiffServ network.

2.2 The Multi-PI AQM

In a Diffserv network we modify the standard PI AQM by introducing two set points for the queue, q^g_{ref} and q^r_{ref} as shown in Fig. 3. In an under-provisioned case, q must converge to q^g_{ref} , otherwise to q^g_{ref} or q^r_{ref} . The marking probabilities, p_g and p_r, for the green and red fluid, respectively, are computed by the two AQM PI controllers, $AQM_g(s)$ and $AQM_r(s)$. To this end, we use the same controller in both loops, that is, $AQM(s) = AQM_g(s) = AQM_r(s)$.

2.3 The Diffserv Network

The combined ARM/AQM Diffserv network is shown in Fig. 4. For control analysis and design, we model this network in a standard block diagram format as shown in Fig. 5. At equilibrium, if the network is under- or exact-subscribed then $p_g = 0$ and $0 < p_r < 1$, and if over-subscribed then $p_r = 1$ and $p_g > 0$. The plant, a matrix transfer function, becomes square by taking one of these

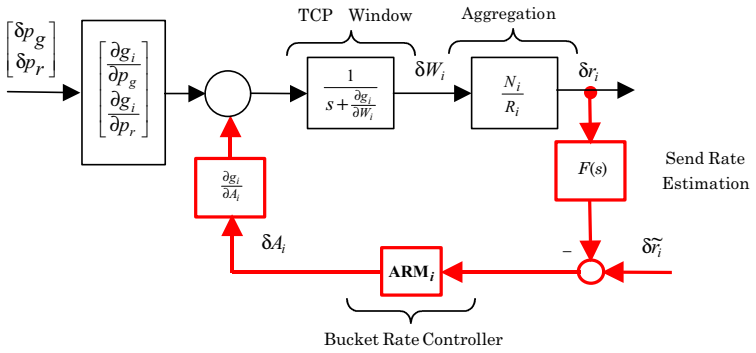

Figure 2: The ARM control system.

two forms:

$$\begin{bmatrix} \delta W_1(s) \\ \vdots \\ \delta W_m(s) \\ \delta q(s) \end{bmatrix} = P(s) \begin{bmatrix} \delta A_1(s) \\ \vdots \\ \delta A_m(s) \\ \delta p_r(s) \end{bmatrix}, \quad q = q_{ref,r},\ p_r = 0,\ p_g = \delta p_g = 0$$

or

$$\begin{bmatrix} \delta W_1(s) \\ \vdots \\ \delta W_m(s) \\ \delta q(s) \end{bmatrix} = P(s) \begin{bmatrix} \delta A_1(s) \\ \vdots \\ \delta A_m(s) \\ \delta p_g(s) \end{bmatrix}, \quad q = q_{ref,g},\ p_r = 1,\ \delta p_r = 0,\ p_g = 0;$$

For example, in a single marking edge with 3 aggregates and $p_g = 0$:

$$P(s) = \begin{bmatrix} \frac{\frac{\partial g_1}{\partial A_1}}{s-\frac{\partial g_1}{\partial W_1}} & 0 & 0 & \frac{\frac{\partial g_1}{\partial p_r}}{s-\frac{\partial g_1}{\partial W_1}} e^{-sR_1} \\ 0 & \frac{\frac{\partial g_2}{\partial A_2}}{s-\frac{\partial g_2}{\partial W_2}} & 0 & \frac{\frac{\partial g_2}{\partial p_r}}{s-\frac{\partial g_2}{\partial W_2}} e^{-sR_2} \\ 0 & 0 & \frac{\frac{\partial g_3}{\partial A_3}}{s-\frac{\partial g_3}{\partial W_3}} & \frac{\frac{\partial g_3}{\partial p_r}}{s-\frac{\partial g_3}{\partial W_3}} e^{-sR_3} \\ \frac{\frac{\partial f}{\partial W_1}}{s-\frac{\partial f}{\partial q}} \frac{\frac{\partial g_1}{\partial A_1}}{s-\frac{\partial g_1}{\partial W_1}} & \frac{\frac{\partial f}{\partial W_2}}{s-\frac{\partial f}{\partial q}} \frac{\frac{\partial g_2}{\partial A_2}}{s-\frac{\partial g_2}{\partial W_2}} & \frac{\frac{\partial f}{\partial W_3}}{s-\frac{\partial f}{\partial q}} \frac{\frac{\partial g_3}{\partial A_3}}{s-\frac{\partial g_3}{\partial W_3}} & \frac{\frac{\partial f}{\partial W_1}}{s-\frac{\partial f}{\partial q}} \frac{\frac{\partial g_1}{\partial p_r}}{s-\frac{\partial g_1}{\partial W_1}} e^{-sR_1} + \frac{\frac{\partial f}{\partial W_2}}{s-\frac{\partial f}{\partial q}} \frac{\frac{\partial g_2}{\partial p_r}}{s-\frac{\partial g_2}{\partial W_2}} e^{-sR_2} + \frac{\frac{\partial f}{\partial W_3}}{s-\frac{\partial f}{\partial q}} \frac{\frac{\partial g_3}{\partial p_r}}{s-\frac{\partial g_3}{\partial W_3}} e^{-sR_3} \end{bmatrix}$$

Since the controlled variable is send rate, the actual plant is described by

$$P_T(s) = \begin{bmatrix} diag\left[\frac{N_1}{R_1}, \ldots, \frac{N_m}{R_m}\right] & 0_{m\times 1} \\ 0_{1\times m} & 1 \end{bmatrix} P(s)$$

The controller reflecting a single effective loop (either for red or green packets) is

$$C(s) = \begin{bmatrix} diag\left[C_{ARM_1}(s), \ldots, C_{ARM_m}(s)\right] & 0_{m\times 1} \\ 0_{1\times m} & -C_{AQM}(s) \end{bmatrix}$$

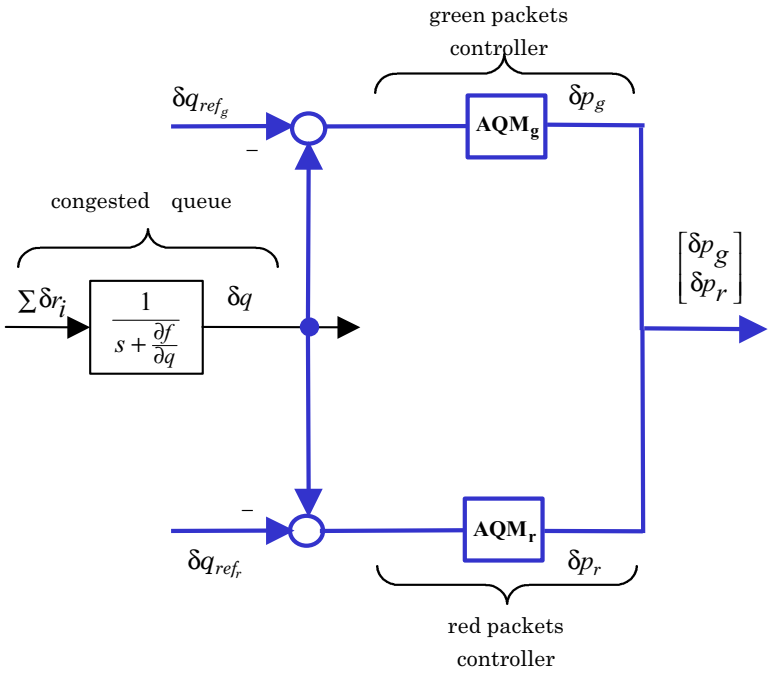

Figure 3: The multi-level AQM control system.

Specifically, the AQM controller is the same PI-type introduced in [4] with an added roll-off

$$C_{AQM}(s) = \frac{k_{aqm}(\frac{s}{z_{aqm}}+1)}{s(\frac{s}{p_{aqm}}+1)}$$

whereas the ARM controller has similar simplicity with an added roll-off

$$C_{ARM}(s) = \frac{k_{arm}(\frac{s}{z_{arm}}+1)}{s(\frac{s}{p_{arm}}+1)}$$

Finally, the rate estimator H is given by

$$H(s) = \begin{bmatrix} diag[F(s)]_{m\times m} & 0_{m\times 1} \\ 0_{1\times m} & 1 \end{bmatrix}$$

3 Stability Analysis

To verify validity of the fluid model and feasibility of our new ARM/AQM Diffserv paradigm, we constructed a network consisting of three set of senders, each served by a marking edge with a token bucket. These edges feed into a congested core with differentiation ability. The propagation delays T_{pi} are all uniform in the ranges: $T_{p1} \in [30-50]$ $msec$, $T_{p2} \in [15-25]$ $msec$ and $T_{p3} \in [0-10]$ $msec$. Each sender consists of n_i FTP flows, all starting uniformly in $[0, 50]$ sec, with $N_1 = 35$, $N_2 = 30$ and $N_3 = 25$. The Diffserv core queue has a buffer size of 800 packets, capacity of $C = 4500$ pkt/sec and ECN marking enabled. We used an average packet size of 500 Bytes.

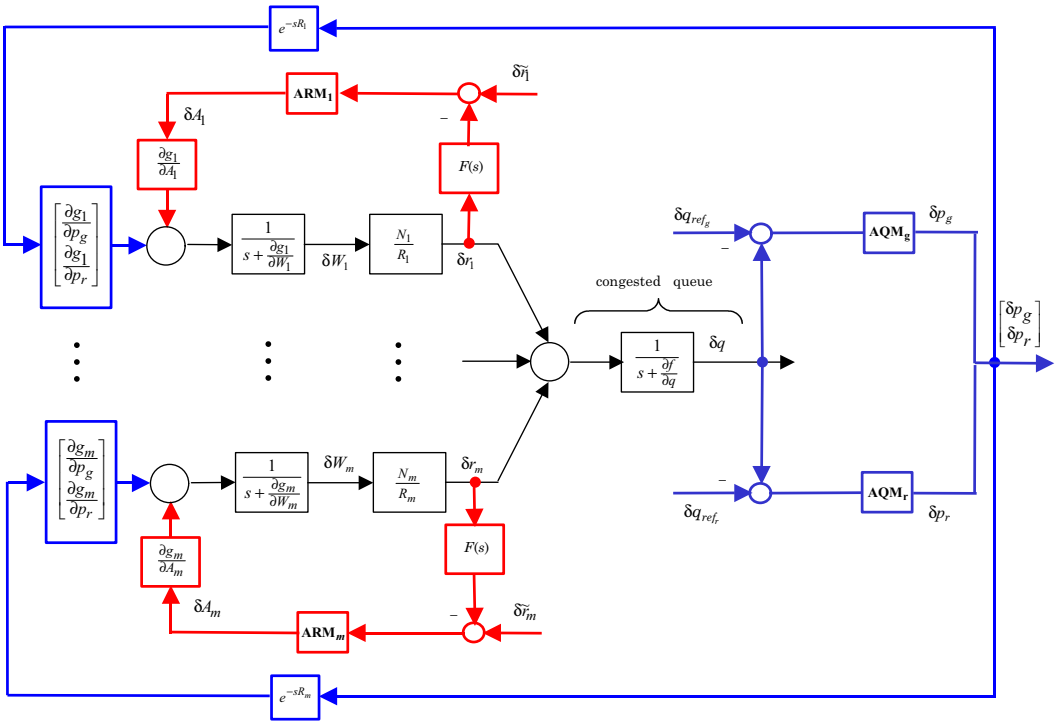

Figure 4: The combined ARM/AQM Diffserv network.

Since at this stage we are only interested in feasibility, no attempt is made at optimizing the controller. In fact, we simply adapt the same AQM controller from [4] which was designed for similar network parameters:

$$C_{AQM}(s) = \frac{9.6 \times 10^{-6}(\frac{s}{0.53} + 1)}{s}$$

This controller is used for both green and red flows. Note that the integrator's output, the marking probability (p_r or p_g), was limited to [0,1] to avoid windup. The set points for the red and green controllers were $q^r_{ref} = 100$ and $q^g_{ref} = 250$ packets. The idea behind this choice was to minimize the possibility of the queue oscillating between these points due to transients.

The ARM controller used for each aggregate has a similar structure to the above, but with different parameters to reflect the different dynamics of the send window and token bucket:

$$C_{ARM}(s) = \frac{0.25(\frac{s}{0.1} + 1)}{s(s+1)}$$

The specific parameters were used based on empirical data and our design experience.

These controllers were discretized with a sampling rate of 37.5 Hz. This rate is even slower than the one suggested in [4] making implementation possibly cheaper. This rate implies that 100 packets will pass the queue in between two sampling instances. The resolution for implementing marking or dropping packets is therefore 1%. This resolution is far finer that what is typically found RED AQMs, and is 10 times finer than the one in [4]. The implication is that we will achieve more accurate marking/dropping during transients.

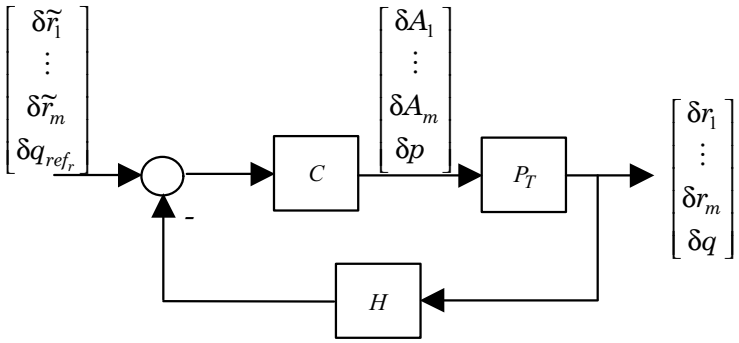

Figure 5: A block diagram of the ARM/AQM Diffserv control system.

The sending rate estimator used the TSW algorithm with a $T_{TSW} = 1$ seconds time slice. This was smoothed used a first-order, low-pass filter with a corner frequency of $a = 1$ rad/sec.

The closed-loop matrix transfer function $T(s)$

$$\begin{bmatrix} \delta r_1(s) \\ \delta r_2(s) \\ \delta r_3(s) \\ \delta q(s) \end{bmatrix} = T(s) \begin{bmatrix} \delta\tilde{r}_1(s) \\ \delta\tilde{r}_2(s) \\ \delta\tilde{r}_3(s) \\ \delta p(s) \end{bmatrix}$$

is given by

$$T(s) \doteq P_T(s)C(s)(I + P_T(s)C(s)H(s))^{-1}$$

where I denotes a 3×3 identity matrix.

Since the queue level at equilibrium can be either 100 or 250 packets, stability should be analyzed around each equilibrium point. There're several techniques available for this purpose: (1) state-space formulation where we study the eigenvalues of the closed-loop $T(s)$, (2) modern optimal control techniques such as mu-analysis, and (3) classical frequency domain ideas. The first two would require approximation of the pure time delay. However, we chose to use the third because we are interested only in quick analysis which is especially suited for a decentralized control scheme.

To evaluate stability of our system at a queue operating point of 100 packets, we ran a simulation. The desired rates were $MGR_1 = 2000, MGR_2 = 500, MGR_3 = 1250$. Since we have an over-provisioned system, the actual rates at equilibrium are $r_1 = 2000, r_2 = 1216, r_3 = 1284$. The generalized Nyquist stability criterion [6] says that this open-loop stable system is closed-loop stable if the origin is not encircled by the Nyquist plot of $\det(I + P_t tCH(s))$, $s \in \Gamma$, where Γ is the Nyquist contour with an appropriate indentation around the origin due to the integrator in C_{AQM}. The Nyquist plots corresponding to various propagation delays in the ranges $T_{p1} \in [30 - 50]$ $msec$, $T_{p2} \in [15 - 25]$ $msec$ and $T_{p3} \in [0 - 10]$ $msec$, are shown in Figure 6. No encirclements implies closed-loop stability. A proper design of this decentralized control system can executed using the multivariable stability margins ideas in [5].

Figure 6: Plots of $\det(I + P_T CH(s),\ s \in \Gamma$ for various propagation delays.

References

[1] Y. Chait, C.V. Hollot, V. Misra, D. Towsley, H. Zhang, C.S. Lui, "Throughput Guarantees for TCP Flows Using Adaptive Two Color Marking and Multi-Level AQM," to appear in *Proc. INFOCOMM '02*.

[2] V. Misra, W. Gong, D. Towsley. "Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED," to appear in *Proc. SIGCOMM'00*, Aug. 2000.

[3] S. Sahu, P. Nain, D. Towsley, C. Diot, V. Firoiu. "On Achievable Service Differentiation with Token Bucket Marking for TCP," ACM SIGMETRICS 2000, pg. 23-33, Santa Clara, CA, June 2000.

[4] C. V. Hollot, V. Misra, D. Towsley, and W. B. Gong, "On designing improved controllers for aqm routers supporting tcp flows." in *Proceedings of INFOCOM 2001*, Anchorage.

[5] O. Yaniv *Quantitative Feedback Design of Linear and Nonlinear Control Systems*, Kluwer Academic Publishers, Massachusetts, USA, 1991.

[6] J.M. Maciejowski *Multivariable Feedback Design*, Addison Wesley, New York, USA, 1989.