Providing Throughput Differentiation for TCP Flows Using Adaptive Two-Color Marking and Two-Level AQM

Y. Chait , C.V. Hollot, Vishal Misra, Don Towsley , H. Zhang and John C.S. Lui

Abstract—**In this paper we propose a new paradigm for a Differential Service (DiffServ) network consisting of two-color marking at the edges of the network using token buckets coupled with differential treatment in the core. Using fluid-flow modelling, we present existence conditions for token-bucket rates and differential marking probabilities at the core that result in all edges receiving at least their minimum guaranteed rates. We then present an integrated DiffServ architecture comprising of an active rate management controller at the marking edge and a two-level active queue management controller at the core. The validity of the fluid flow model and performance of this new scheme are verified using `ns` simulations.**

I. Introduction

The differentiated services architecture (DiffServ) is under consideration for providing different services in a scalable manner to users in the Internet. It adheres to the basic Internet philosophy; namely that complexities should be relegated to the network edge while preserving the simplicity of the core network. Two per-hop behaviors (PHBs) have been standardized by IETF, expedited forwarding (EF) [1] and assured forwarding (AF) [2]. The former is intended to support low delay applications while the latter is intended to provide throughput differentiation among clients according to a negotiated profile.

We focus on services built on top of the AF PHB. Using token buckets, routers at the edge of the network monitor and mark packets green when they fall within a profile. Otherwise they remain unmarked (colored red). The core routers give preference to green packets. In the presence of congestion, red packets are more likely to be dropped (or have their congestion notification bit set in the presence of ECN [3]). This promises to allow a network provider to supply throughput differentiation to different users by appropriate setting of the edge markers.

In this paper we address the problem of providing users with minimum throughputs. One might expect this to be an easy problem to solve as it suffices to choose an edge marker appropriate for the desired throughput. Unfortunately, several studies have concluded that the throughput attained by a customer is affected, not only by the edge marker but by the presence of other customer flows, propagation delays, etc. [4], [5], [6]. This is because the predominance of traffic is carried by TCP and the TCP congestion avoidance mechanism reacts in a complex manner with its environment. In order to provide minimum throughputs to aggregates, we introduce an *Active Rate Management* (ARM) mechanism at edge markers that are responsible for setting the token bucket parameters in order to provide these minimum throughputs and to adapt to changes in the network. Our ARM mechanism is a classical, linear, time-invariant controller, e.g., [7]. We establish feasibility through a combination of analysis and simulation when it is coupled with a two-level active queue management (AQM) controller at a congested router. In particular, simulations demonstrate that the ARM mechanisms are able to maintain throughputs at or above minimum guaranteed rates (MGR) and are able to respond in a timely manner to fluctuations in traffic characteristics .

There does not appear to be other work on the problem of providing minimum throughput levels to customers within the AF framework that is based on control theory. However, Yeom and Reddy studied the related problem of how to fairly divide throughput among individual TCP flows passing through a common edge marker [8].

The rest of the paper is organized as follows. Section II describes a fluid model of the system. Section III presents conditions under which MGRs can be provided. Section IV presents an architecture based on ARM and two-level AQM for providing MGRs to aggregates. This architecture is evaluated through simulation in Section V and Section VI summarizes the paper.

II. Network Model

Our starting point is the fluid-flow model developed in [9] for modelling TCP flows and AQM routers. In this section we will extend this model to account for two-color marking at the network edge and two-level AQM at the core; see Figure 1. To begin, we assume m edge routers, each serving a number of aggregates consisting of N_i identical TCP flows with each having token buckets with rate $A_i(t)$ and size $b_i >> 1$, $i = 1, \ldots, m$. These edges feed a core router with link capacity C and queue length q. At time $t > 0$, each TCP flow is characterized by its average window size $W_i(t)$ and average round-trip time

$$R_i(t) \triangleq T_i + \frac{q(t)}{C}$$

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

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

This work is supported in part by the National Science Foundation under Grants CMS-9800612, ANI-9873328, ANI-0077039, and EIA0080119, and by DARPA under Contract DOD F30602-00-0554.

Y. Chait is with the MIE Department, University of Massachusetts, Amherst, MA 01003; chait@ecs.umass.edu

C.V. Hollot is with the ECE Department, University of Massachusetts, Amherst, MA 01003; hollot@ecs.umass.edu

Vishal Misra is with the Computer Science Department, Columbia University, New York, NY 10027; misra@cs.columbia.edu

Don Towsley and H. Zhang are with the Computer Science Department, University of Massachusetts, Amherst, MA 01003; {towsley,honggang}@cs.umass.edu

John C.S. Lui is with the Department of Computer Science & Engineering The Chinese University of Hong Kong Shatin, N.T. Hong Kong; cslui@cse.cuhk.edu.hk

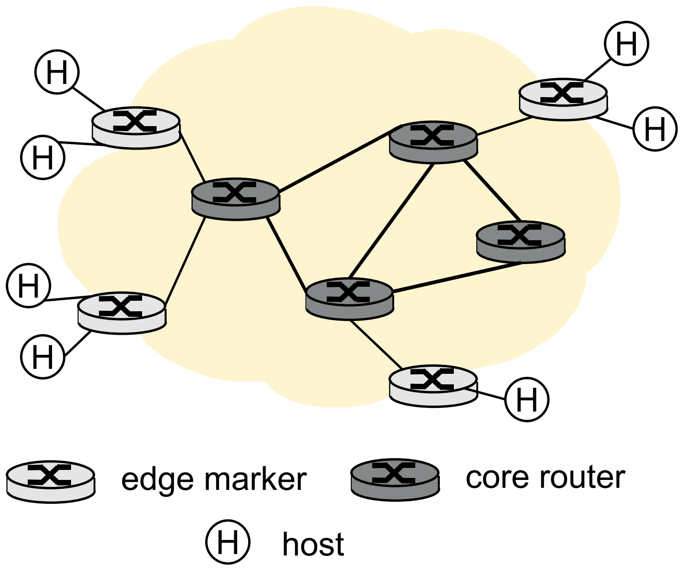


Fig. 1. The Differentiated Service Architecture.

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

$$\frac{dq(t)}{dt} = \begin{cases} -C + \sum_{i=1}^{m} r_i(t), & q(t) > 0 \\ \left[-C + \sum_{i=1}^{m} r_i(t)\right]^+, & q(t) = 0 \end{cases} \tag{1}$$

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 the fluid.

Next we model the color-marking process at the i-th edge and the two-level 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 $1 - f_i^g(t)$ be 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 $0 \le p_g(t) < p_r(t) \le 1$. 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 MGR for the i-th aggregate at an edge . We say that the router is over-provisioned if $\sum_{i=1}^m \tilde{r}_i < C$ 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 the edges to ensure that the send rates $\{r_i\}$ meet or exceed their respective MGRs when the system is exact or over-provisioned.

In the next section we address the Diffserv feasibility problem which amounts to finding $\{f_i^g\}$, p_g and p_r such that the sending rates $\{r_i\}$ meet the MGRs

III. Feasible Diffserv Networks

As shown in [4], it's not always possible to meet MGRs with token bucket marking and TCP. In the following, we will identify network conditions rendering this Diffserv problem feasible. To begin, the network dynamics (1) and (2), at equilibrium, yield:

$$\sum_{i=1}^{m} r_i = C \tag{3}$$

where

$$r_i = \frac{\sqrt{2}N_i}{R_i\sqrt{p_i}} = \frac{\sqrt{2}N_i}{R_i\sqrt{f_i^g p_g + (1 - f_i^g)p_r}}$$

Given network parameters $(\{N_i\}, \{R_i\}, C)$, and MGRs $\{\tilde{r}_i\}$ satisfying $\sum \tilde{r}_i \le C$, we say the *over-provisioned Diffserv network is feasible* if there exist $(\{f_i^g\}, p_g, p_r)$ such that (3) is satisfied with

$$0 \le f_i^g \le 1; \quad r_i \ge \tilde{r}_i; \quad 0 \le p_g < p_r \le 1$$

Theorem: *Given MGRs $\{\tilde{r}_i\}$, the over-provisioned Diffserv network is feasible if and only if*

$$\sum_{i=1}^{m} \frac{\sqrt{2}N_i}{R_i\sqrt{\overline{p}_i}} \le C \tag{4}$$

where

$$\overline{p}_i = \min\left\{1, \frac{2N_i^2}{\tilde{r}_i^2 R_i^2}\right\} \tag{5}$$

Proof. If the over-provisioned Diffserv network is feasible, then, necessarily, $p_i \le \overline{p}_i$ and

$$C = \sum_{i=1}^{m} \frac{\sqrt{2}N_i}{R_i\sqrt{p_i}} \ge \sum_{i=1}^{m} \frac{\sqrt{2}N_i}{R_i\sqrt{\overline{p}_i}} \tag{6}$$

Now, suppose that (4) holds. We will show that the Diffserv network is feasible. To this end, define $\epsilon = 2(\sum_{i=1}^m N_i/(R_i\sqrt{\overline{p}_i}C))^2$, and note that $0 \le \epsilon \le 1$, with $\epsilon = 1$ when

$$\sum_{i=1}^{m} \sqrt{2}N_i/(R_i\sqrt{\overline{p}_i}) = C$$

Now, set

$$\begin{aligned} p_r &= 1 \\ p_g &= \epsilon \min\{\overline{p}_i\} \\ f_i^g &= \frac{1 - \epsilon\overline{p}_i}{1 - p_g}, \quad i = 1, \ldots m \end{aligned}$$

Thus, $r_i \ge \tilde{r}_i$ since $p_i = \epsilon\overline{p}_i \le \overline{p}_i$. In addition, $\sum_{i=1}^m r_i = C$ follows immediately from the definition of ϵ. This completes the proof. □

Remark: Feasible, over-provisioned Diffserv networks can possess an infinite number of solutions $\{f_i^g\}$, p_g and p_r. We will say more about this In Section IV.

[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

IV. A New Control Paradigm for DiffServ

In the previous section we have studied equilibria of this system independent of the core queuing and marking edge policies. In this section we present the control scheme that will maintain desired performance around this equilibrium in the face of changing session loads, propagation times and other network parameters. To this end, again consider the system of nonlinear differential equations where, now, we explicitly show dependence on the bucket rate A_i:

$$\begin{aligned}
\dot{q}(t) &= -C + \sum_{i=1}^{m} \frac{N_i W_i(t)}{R_i(t)} \\
&\triangleq f(q, W_i, p_g, p_r, A_i) \\
\dot{W}_i(t) &= \frac{1}{R_i(t)} - \frac{W_i(t) W_i(t - R_i(t))}{2R_i(t - R_i(t))} \cdot \\
&\quad \left(\frac{A_i}{r_i(t)} p_g(t - R_i(t)) \right. \\
&\qquad \left. + (1 - \frac{A_i}{r_i(t)}) p_r(t - R_i(t)) \right) \\
&\triangleq g_i(q, W_i, p_g, p_r, A_i)
\end{aligned}$$

We follow the same design philosophy used in [7]; namely, deriving controllers based on linearized LTI models. First, we identify the equilibrium point $(q_o, W_{io}, p_{go}, p_{ro}, A_{io})$ which satisfies

$$\begin{aligned}
0 &= -C + \sum_{i=1}^{m} \frac{N_i W_{io}}{R_i} \\
0 &= 1 - \left(\frac{A_{io}}{r_{io}} p_{go} + (1 - \frac{A_{io}}{r_{io}}) p_{ro} \right) \frac{W_{io}^2}{2} \\
R_i &= T_i + \frac{q_o}{C}.
\end{aligned}$$

In the linearization we make two approximations. Firstly, we ignore delay R_i in the term $W_i(t - R_i)/R_i(t - R_i)$, but deal with it in the probability terms $p_r(t - R_i)$ and $p_g(t - R_i)$. Secondly, we replace saturation terms $\min(1, \frac{A_i}{r_i})$ with $\frac{A_i}{r_i}$. Finally, linearization about the equilibrium point gives

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

where

$$\begin{aligned}
\delta q &\equiv q(t) - q_o \\
\delta W_{io} &\equiv W_i(t) - W_{io} \\
\delta p_g &\equiv p_g(t) - p_{go} \\
\delta p_r &\equiv p_r(t) - p_{ro} \\
\delta A_i &\equiv A_i(t) - A_{io}
\end{aligned}$$

and where evaluating the partial at this equilibrium point gives (partials not shown are zero)

$$\begin{aligned}
\frac{\partial f}{\partial q} &= -\sum_{i=1}^{m} \frac{r_{io}}{C R_i} \\
\frac{\partial f}{\partial W_i} &= \frac{N_i}{R_i} \\
\frac{\partial g_i}{\partial W_i} &= -\frac{A_{io}}{2N_i}(p_{go} - p_{ro}) - \frac{W_{io}}{R_i} p_{ro} \\
\frac{\partial g_i}{\partial p_r} &= \frac{W_{io} A_{io}}{2N_i} - \frac{W_{io}^2}{2R_i} \\
\frac{\partial g_i}{\partial p_g} &= -\frac{A_{io} W_{io}}{2N_i} \\
\frac{\partial g_i}{\partial A_i} &= -\frac{W_{io}}{2N_i}(p_{go} - p_{ro}).
\end{aligned}$$

Performing a Laplace transformation, we obtain a block diagram representation for the open-loop system shown in Figure 2. The open-loop plant, obtained from the above equation, is defined as:

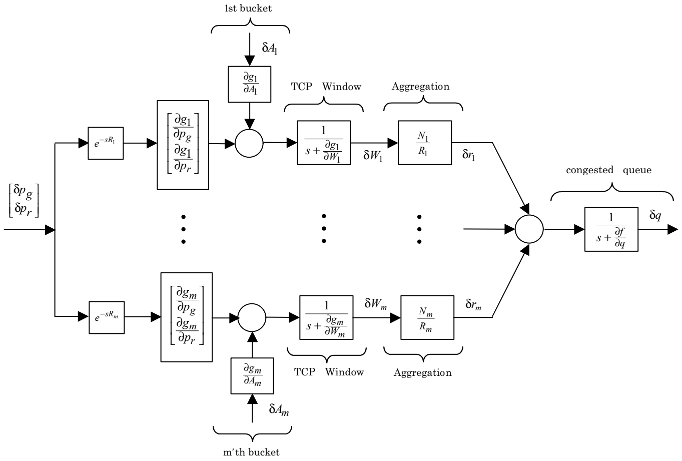


Fig. 2. Block diagram of an open-loop DiffServ network.

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

In a compact matrix transfer-function form, we write:

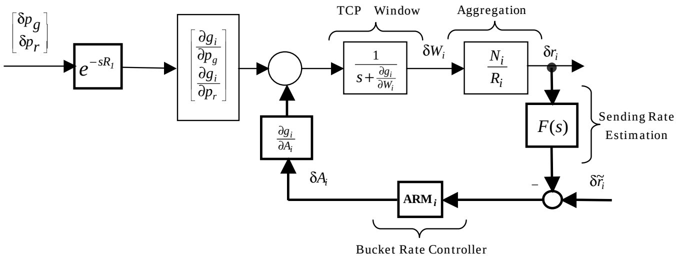
$$\begin{bmatrix} \delta W_1(s) \\ \vdots \\ \delta W_m(s) \\ \delta q(s) \end{bmatrix} = P(s) \begin{bmatrix} \delta A_1(s) \\ \vdots \\ \delta A_m(s) \\ \delta p_g(s) \\ \delta p_r(s). \end{bmatrix} \tag{7}$$

A. Active Rate Management (ARM)

Similar to the introduction of the AQM in [7], we propose a feedback structure around the token bucket termed ARM. 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 send rate to its MGR, it is necessary to construct an estimate for this send rate. We follow the TSW procedure which consists of the following. The send rate is computed by measuring the number of sent packets over a fixed time period T_{TSW}. This value is then smoothed by a low-pass filter. A fluid model for this dynamic is given by:

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

For this purpose, we introduce the feedback structure as shown in Fig. 3.



$\begin{bmatrix}\delta p_g\\ \delta p_r\end{bmatrix}$
e^{-sR_i}
$\begin{bmatrix}\frac{\partial g_i}{\partial p_g}\\ \frac{\partial g_i}{\partial p_r}\end{bmatrix}$
TCP Window
Aggregation
$\frac{1}{s+\frac{\partial g_i}{\partial W_i}}$
δW_i
$\frac{N_i}{R_i}$
δr_i
Sending Rate Estimation
$F(s)$
$\frac{\partial g_i}{\partial A_i}$
δA_i
ARM_i
–
$\delta\tilde{r}_i$
Bucket Rate Controller

Fig. 3. The ARM control system.

B. The Two-Level AQM

In a DiffServ network we modify the standard PI AQM by introducing two set points for the core queue, q^g_{ref} and q^r_{ref} as shown in Fig. 4. 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 two AQM controllers, $AQM_g(s)$ and $AQM_r(s)$. To this end, we use the same parameter in both loops, that is, $AQM(s) = AQM_g(s) = AQM_r(s)$.

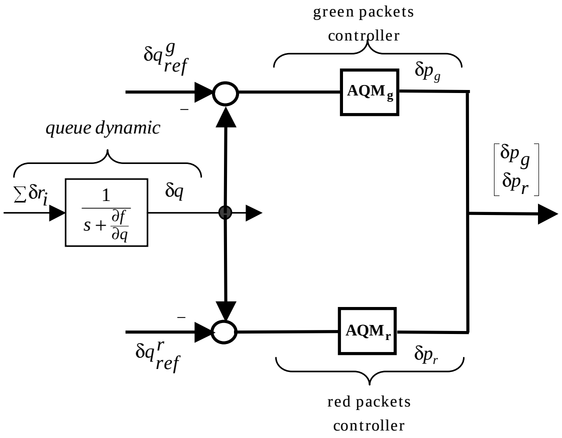


Fig. 4. Multilevel AQM controller.

C. Feasible Bucket Rates

In Section III we gave sharp conditions for an over-provisioned Diffserv network to be feasible. With the two-level AQM in place, it is now possible to be more explicit and to describe feasible bucket rates. To begin, assume the equilibrium queue length is either q^r_{ref} or q^g_{ref}. If $q_o = q^r_{ref}$, then, due to the integrator in the AQM controller, $p_{go} = 0$ and $0 < p_{ro} < 1$. Thus, the set of feasible marking probabilities of red fluid is

$$\mathcal{P}_r = \left\{ p_{ro} : \max_i \frac{2N_i^2}{\tilde{r}_i^2 R_i^2} < p_{ro} < 1 \right\}$$

As long as $\mathcal{P}_r$ is non-empty, the bucket-rates A_{io} solving the Diffserv problem are non-unique. Indeed, the set of feasible bucket rates is

$$\mathcal{A}_i = \left\{ A_{io} : \ 1 < A_{io} < r_{io} \left(1 - \frac{2N_i^2}{\tilde{r}_{io}^2 R_i^2 p_{ro}} \right), \ \ p_{ro} \in \mathcal{P}_r \right\}.$$

Conversely, if $q_o = q^g_{ref}$, then, due to the integrator in the AQM, $p_{ro} = 1$ and $0 < p_{go} < 1$. The set of feasible bucket rates can be expressed in terms of p_{go} as follows:

$$\mathcal{A}_i = \{ A_{io} : A_{io} = \min \{1, \gamma(p_{go})\}, p_{go} \in (0,1) \}$$

where

$$\gamma(p_{go}) = \frac{r_{io}}{1 - p_{go}} \left(1 - \frac{2N_i^2}{\tilde{r}_{io}^2 R_i^2} \right)$$

Using these parameterizations, we will analyze the stability of these feasible points (equilibria) in Section V.

D. The DiffServ Network

The combined ARM/AQM DiffServ network is shown in Fig. 5. For control analysis and design, we put this network in a standard block diagram format as shown in Figure 6. Around the equilibrium queue length $q_0 = q^r_{ref}$, the linearized dynamics in (7) becomes

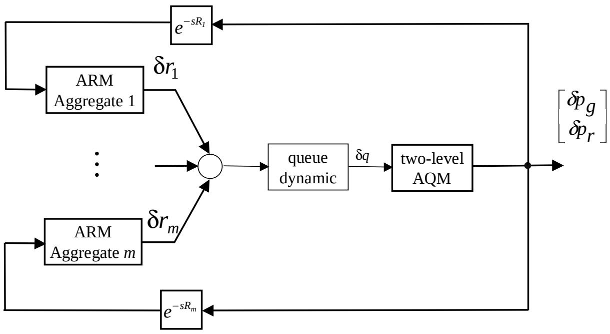


Fig. 5. The combined ARM/AQM DiffServ network.

$$\begin{bmatrix} \delta W_1(s) \\ \vdots \\ \delta W_m(s) \\ \delta q(s) \end{bmatrix} = P(s) \begin{bmatrix} \delta A_1(s) \\ \vdots \\ \delta A_m(s) \\ \delta p_r(s) \end{bmatrix}$$

while, for $q_0 = q^q_{ref}$, we have

$$\begin{bmatrix} \delta W_1(s) \\ \vdots \\ \delta W_m(s) \\ \delta q(s) \end{bmatrix} = P(s) \begin{bmatrix} \delta A_1(s) \\ \vdots \\ \delta A_m(s) \\ \delta p_g(s) \end{bmatrix}$$

Since the variables of interest are send rates $\{r_i\}$, we form

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

The controller is described by

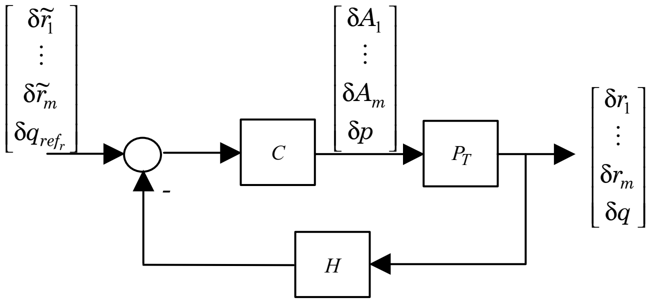


Fig. 6. A block-diagram representation of the ARM/AQM DiffSserv control system.

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

Specifically, the AQM controller has the same PI structure introduced in [7]

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

The ARM controller has similar structure with additional low-pass filtering

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

Finally, the rate estimator H is given by

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

V. ns Studies

To validate 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 as shown in Fig. (7). These edges feed into a congested core with differentiation ability. The propagation delays T_{pi} are all uniform in the ranges: $T_{p1} \in [50 - 90]$ sec, $T_{p2} \in [15 - 25]$ msec and $T_{p3} \in [0 - 10]$ msec. Each sender consists of N_i FTP flows, all starting uniformly in $[0, 50]$ sec, with $N_1 = 20$, $N_2 = 30$ and $N_3 = 25$. The differentiating core queue has a buffer size of 800 packets, capacity of $C = 3750$ pkt/sec and ECN marking enabled. We used an average packet size of 500 Bytes.

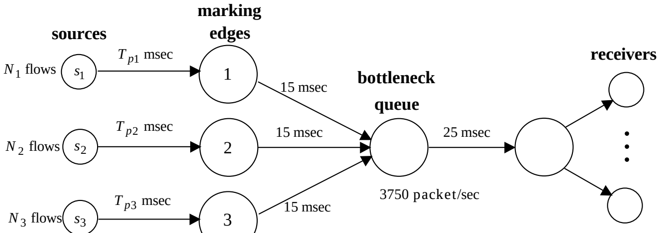


Fig. 7. The simulated DiffServ network.

A. Control Design and Analysis

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

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

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.

The two-level AQM controllers are taken from [7]:

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

where its output state, a marking probability (p_r or p_g), was appropriately limited to [0,1] to avoid integrator windup. This controller was discretize with a sampling rate of 37.5 Hz. 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 allow the queue, if possible, to converge to the lower queue level where $p_g = 0$.

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

$$C_{ARM}(s) = \frac{0.05(\frac{s}{0.1} + 1)}{s(s+1)}$$

This controller was discretized with a sampling rate of 37.5 Hz.

The send rate estimator used the Time Slice Window (TSW) algorithm with a $T_{TSW} = 1$ second time slice. This was smoothed used a first-order, low-pass filter with a corner frequency of $a = 1$ rad/sec.

Since the queue level at equilibrium may be either 100 or 250 packets, we analyze stability around each point. Using frequency response concepts (e.g., [10]), it can be shown that the DiffServ system is stable around each of these points over the feasible ranges of marking probabilities discussed in Section IV-C. The design of the two-level AQM and the ARM controllers, as well as stability analysis details can be found in [11].

B. ns Experiments

We now present a series of experiments performed with ns to demonstrate various aspects of the performance of our system. Experiment 1 demonstrates the inability of token buckets to achieve MGRs in certain situations in an exact provisioned core. The validity of our fluid model is also established by comparing the responses of ns and the nonlinear fluid model (using Simulink [12]). In experiments 2-6 we study the performance of our ARM/AQM DiffServ network under varying conditions such as transient FTP flows, HTTP flows and exact or over provisioned core.

Experiment 1. In this experiment we compare the dynamics of an exact-provisioned ($C = 3750$ pkt/sec) DiffServ system employing a differentiating core queue and token buckets with fixed rates equal to the MGRs. All token bucket have a size of $b_i = 50$ packets. The MGRs in pkt/sec are: $\tilde{r}_1 = 2000$, $\tilde{r}_1 = 500$ and $\tilde{r}_1 = 1250$. We observe in Figure 8 that, as reported in [4], the send rate do not always converge to their corresponding token bucket rates, edge 1 in this case. We also observe good agreement between ns and the nonlinear differential equation fluid model, providing a sense of confidence in our analysis and design. Finally, the ARM also appear to drive the sending rates to their steady-state values faster. This is a result of the design of the ARM response time which is tunable via the controllers $G_{ARM}(s)$. Note that in all the experiments reported here the network and controller's initial conditions are zero.

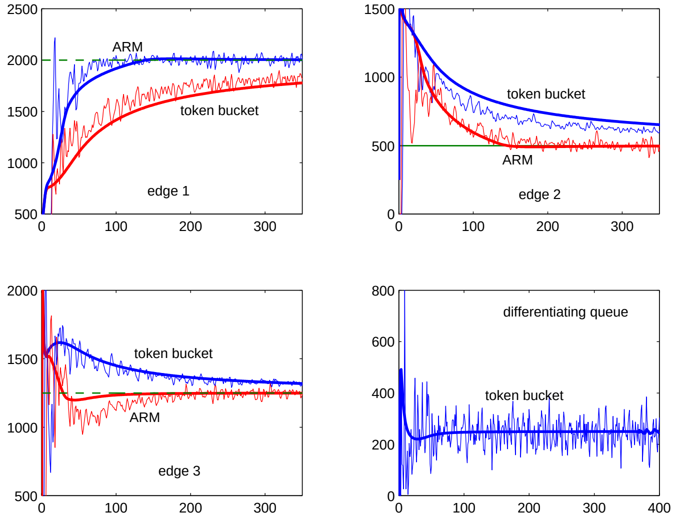


Fig. 8. Send rates with token bucket, ARM, MGRs (dashed) and differentiating queue dynamics in Experiment 1. Fluid model solution is depicted by thick lines
.

Experiment 2. In this experiment we repeat the setting of Experiment 1. We add some transient FTP flows as follows. In edge 1: add 4 flows at $t = 100$ sec, remove 8 flows at $t = 150$ and add 4 flows at $t = 200$. In edge 2: remove 6 flows at $t = 125$ sec, add 12 flows at $t = 175$ and remove 6 flows at $t = 220$. In edge 3: add 5 flows at $t = 190$ sec and take out 5 flows at $t = 240$. The ability of the ARM to regulate the send rates about their corresponding MGRs in observed in Figure 9.

Experiment 3. In this experiment we repeat the setting of Experiment 1 and add short-lived HTTP flows as follows: each edge has 100 HTTP clients with exponential starting distribution, each client opens 4 connections with each containing 1 doc base (500 Bytes) and 1 image (2000 Bytes). Again, the ARM is quite capable in achieving and maintaining its MGRs as shown in Figure 10.

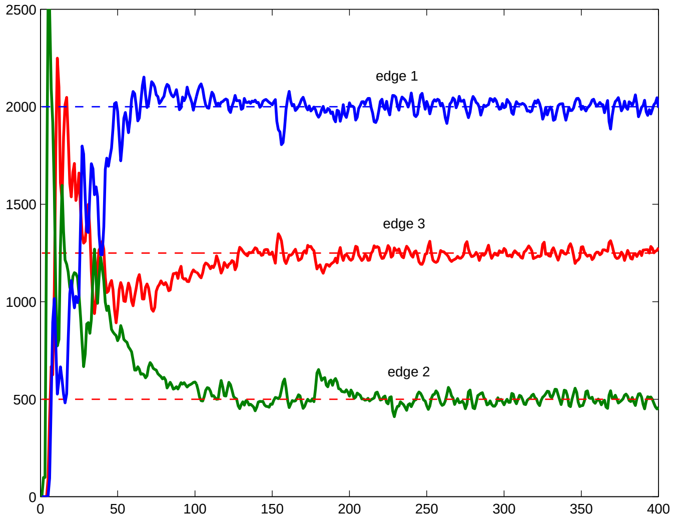


Fig. 9. Sendrates (solid) and MGRs (dashed) in Experiment 2.

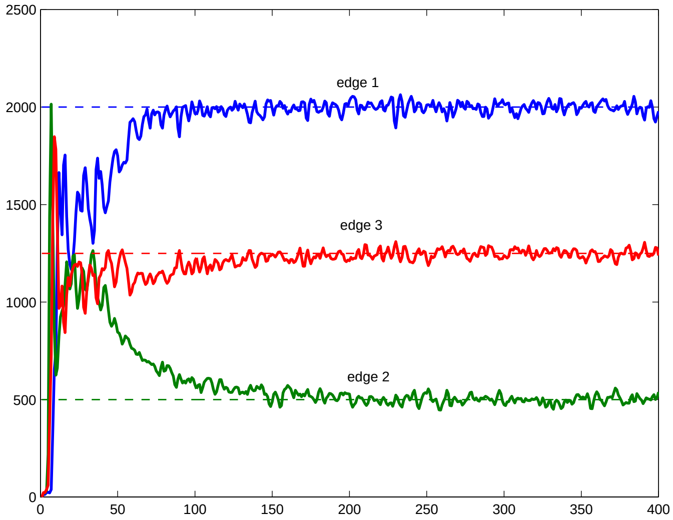


Fig. 10. Send rates (solid) and MGRs (dashed) in Experiment 3.

Experiment 4. In this experiment we repeat the setting of Experiment 3, add the transient FTP flows in Experiment 2, and increase the core capacity by 20% to 4500 pkt/sec. It is seen (Figure 11) that the ARM achieves at least the MGRs, however, as expected, some aggregates will grab the available excess capacity. By studying the steady-state window equation, it is possible to predict which aggregates will consume that extra capacity.

Experiment 5. In this experiment we repeat the setting of Experiment 2. The system has exact provisioning. We introduce a background flow (4th edge) that feeds into the differentiating queue but since it does not have a DiffServ contract, all of its packets are marked red. There are no HTTP or transient FTP flows. Since both systems used a similar two-level AQM, the red marking probability approaches 1 ($p_r \to 1, \quad t \to \infty$) due

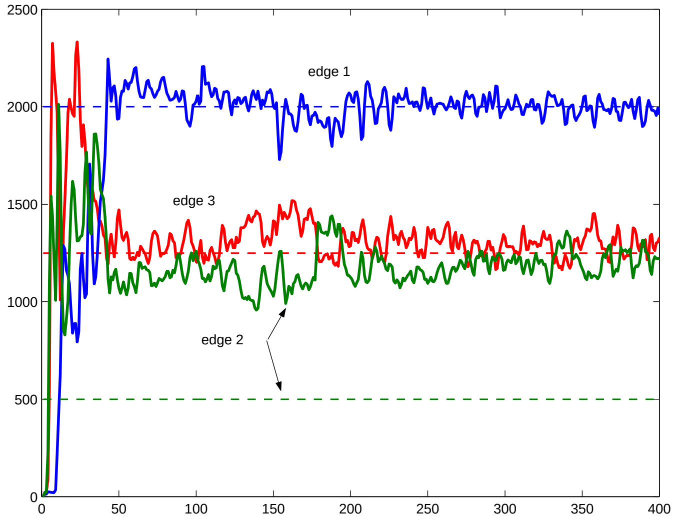


Fig. 11. Send rates (solid) and MGRs (dashed) in Experiment 4.

to the integrator in the AQM. The end results is that the background flow is completely rejected[2] and the ARM/AQM system is able to meet the MGRs for the aggregates with the contract. The results are shown in Figure 12.

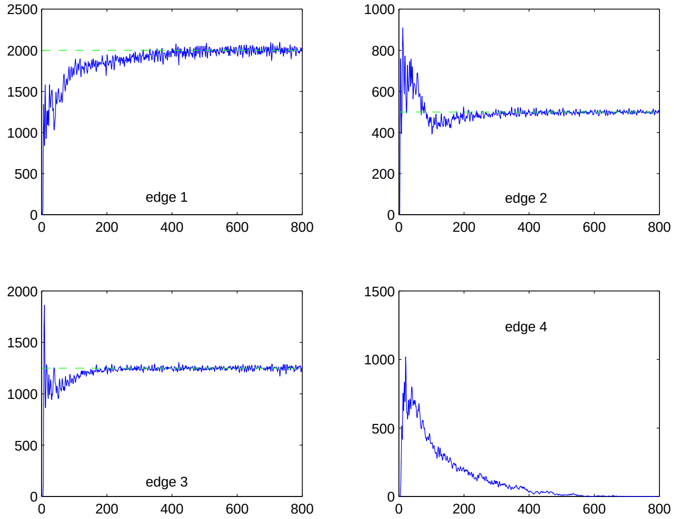


Fig. 12. Send rates with token bucket, ARM, and the MGRs in Experiment 5.

Experiment 5a. In this experiment we repeat the setting of Experiment 5 but increase the network capacity by 20% to 4600 pkt/sec. This should test the ability of our DiffServ system to insure the MGRs for those aggregates with contracts as well as allow non-contract aggregates to share over capacity. Indeed, Figure 13 indicates this versatility.

Experiment 6. In this experiment we repeat the setting of Experiment 1 with the ARM active but reduce the network's capacity by 20% (C=3000 packets). Clearly, in this case the Theorem no longer applies and there does not exist parameters that can achieve the MGRs. The results are shown in Figure 14. As is the case in Experiment 4, some aggregates will be more aggressive in seeking send rates. This can be studied from the steady-state window equation.

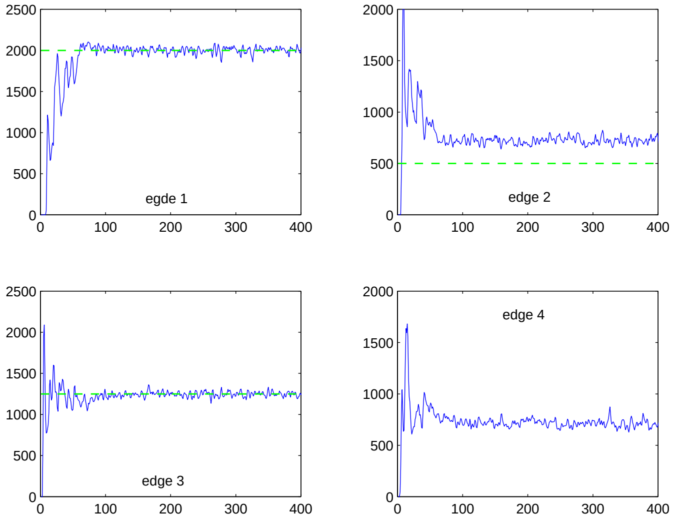


Fig. 13. Send rates and the MGRs in Experiment 5a.

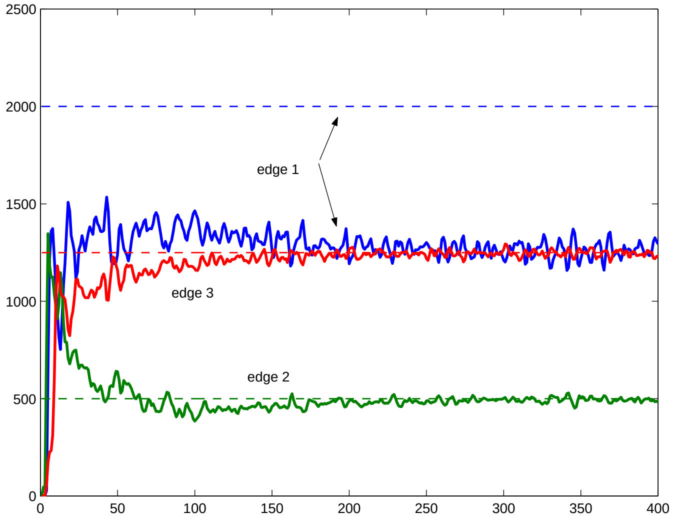


Fig. 14. Send rates (solid) and MGRs (dashed) in Experiment 6.

VI. Conclusions

In this paper we presented a design for a minimum throughput service based on the AF per hop behavior. The constituent components of this design include two-color token bucket edge markers coupled with a two-level AQM controller embedded in the core routers. The interactions between TCP flows and these components are captured through a simple fluid model, whose behavior is described by a set of ordinary differential equations that are readily solved. These equations are further analyzed to derive necessary and sufficient conditions under which the minimum throughput requirements of various flow aggregates can be supported. The equations can also be used to derive conditions for the stability of the design along with guidelines for setting parameters.

[2] A two-level RED AQM may not completely reject this flow due to lack of integration

We verified, through simulation, that our design does a good job at providing minimum throughputs, is robust, and that it adapts to fluctuations in traffic loads in a timely manner even when the model assumptions are not satisfied (e.g., packet flows instead of fluid flows). Thus our design appears quite promising as a mechanism for providing minimum throughput levels to flow aggregates.

There are several aspects of the design that can stand improvement and will be subject of future work. First, our fluid model is valid when the token bucket is large. We would like to extend the model to account for small token buckets as well. Second, our mechanisms do not provide for the fair sharing of excess or lack of bandwidth, when the network is over-subscribed or under-subscribed, respectively. Instead, the allocation is determined by the dynamics of TCP congestion control mechanism. Third, we would like to introduce an additional component that can divide the aggregate flow throughput among the constituent flows according to a policy specified by the administrator of the aggregate flow.

References

[1] V. Jacobson, K. Nichols, K. Poduri. “An expedited forwarding PHB,” RFC2598, June 1999.

[2] J. Heinanen, F. Baker, W. Weiss, J. Wroclawski. ” Assured forwarding goup,” RFC2597, June 1999.

[3] K.K. Ramakrishnan, S. Floyd. “A Proposal to add Explicit Congestion Notification (ECN) to IP,” RFC 2481, Jan. 1999.

[4] 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.

[5] I. Yeom, A. L. Narasimha Reddy, “Modeling TCP behavior in a Differentiated-Services Network,” *ACM/IEEE Transactions on Networking*, Feb. 2001.

[6] M. Goyal, A. Durresi, P. Misra, C. Liu, R. Jain, “Effect of Number of Drop Precedences in Assured Forwarding,” *Proc. GlobeCom 99*, Dec. 1999

[7] 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.

[8] I. Yeom, A.L.N. Reddy. “Marking for QoS Improvement,” *Journal of Computer Communications*, Jan. 2001, pp 35-50.

[9] V. Misra, W. Gong, D. Towsley. “Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED,” *Proc. SIGCOMM’00*, Aug. 2000.

[10] M. Y. Park,Y. Chait, and M. Steinbuch “Inversion-free design algorithms for multivariable Quantitative Feedback Theory: an application to robust control of a Cd-ROM,” *Automatica*, Vol. 33(5), pp. 915-920, 1997.

[11] Y. Chait, “Analysis and Design of Multi-Level AQM and Adaptive Token Bucket Marking,” DACS Lab, MIE Department, Technical Reprot, http://www.ecs.umass.edu/mie/labs/dacs/index2.html#publications.

[12] *Simulink Dynamic System Simulation for MATLAB*, The MathWorks, Inc., Natick, MA, USA.