Throughput Differentiation Using Coloring at the Network Edge and Preferential Marking at the Core

Yossi Chait, C. V. Hollot, Vishal Misra, *Member, IEEE*, Don Towsley, *Fellow, IEEE*, Honggang Zhang, and Yong Cui, *Student Member, IEEE*

***Abstract*—In this paper we introduce an innovation in differentiated services architecture consisting of adaptive two-level coloring at the edge and preferential marking at the core. We identify general properties of these two processes which, when met, guarantee a desirable fixed point for the network; i.e., one where aggregated flow rates meet or exceed given targets in an over-provisioned network. Specific mechanisms realizing the aforementioned properties lead to so-called active rate management controllers for edge coloring, and a preferentially-marking, active queue management controller at the core. We discuss stability of the fixed point for this network, and validate results using `ns` simulations.**

I. Introduction

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

We focus our attention on services built on the AF PHB which employ token buckets [3] at the network's edge routers to color packets green when rates fall within the rate profile and red otherwise. At the core, routers give preference to green packets; i.e., in the presence of congestion, red packets are more likely dropped (or have their congestion notification bit set in the presence of ECN [4]). These two actions, coloring at the edge and preferred marking at the core, allow an ISP to offer throughput differentiation.

In this paper, we develop coloring strategies that provide aggregate flows with minimum throughput guarantees in networks with sufficient resources. One might expect this to be straightforward since the freedom in coloring packets would appear sufficient to regulate throughputs. However, several studies have shown that achievable throughput is affected not only by edge coloring, but by the presence of other factors including competing flows and propagation delays [5]–[7]. This occurs since the predominance of network traffic is affected by TCP's congestion-control mechanism which reacts to the network's environment to adapt send rates. To be successful then, a throughput differentiation scheme must work within the TCP framework and coordinate with TCP's congestion-control mechanisms. In some recent work [8], we addressed this issue and introduced a so-called Active Rate Management (ARM) mechanism at edge routers to adaptively set token bucket parameters, and, together with preferentially-marking Active Queue Management (AQM) to drive throughputs toward target rates. However, the feasibility of this setup was only demonstrated using `ns` simulations. There appears to be limited analytical work dealing with guaranteeing minimum throughputs within the AF framework. Related work by Yeom and Reddy [9] studies the problem of fairly dividing throughput among individual TCP flows passing through a common edge. By providing analysis, our present work substantiates the observations made in [8]. For a recent comprehensive review of advances in Internet QoS see [10].

Our main result [Theorem 1, Section III] considers congestion-controlled flows,[1] edge routers equipped with ARMs, and a single congested link with preferentially-marking AQM. Under the assumption that the link's capacity exceeds the given target rates, this theorem characterizes coloring and marking behaviors sufficient for the ARMs and AQM to cooperatively guarantee that aggregate flows meet or exceed given target rates. This theorem goes on to identify those aggregates that exactly meet their target rates, and to quantify how excess capacity is meted out to the remaining.

In Section IV we consider implementable ARM and AQM schemes that satisfy the conditions set forth in Theorem 1. We introduce a proportional-integral ARM that incorporates integral action to adaptively set the token-bucket rates to their required values, and, a proportional gain to affect transient performance. For the core router we show that a variety of popular AQM schemes can be easily modified to meet Theorem 1's conditions. These include a RED AQM with nonoverlapping marking profile as shown in Fig. 4(b), a two-level PI (REM) [11] having two (green and red) buffer reference levels and two proportional-integral controllers (see Fig. 5(a)), and similarly, an adaptive virtual queue (AVQ) [12] utilizing two virtual queues and two utilization set-points; see Fig. 5(b). In Section V we address stability and discuss how ARMs and the AQM interact so that aggregate flows converge to their assigned target rates.

[1]Our results are stated for a general class of congestion-control algorithms which includes TCP-Reno and proportionally fair schemes.

Manuscript received June 14, 2004; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor S. Low. This work was supported in part by the National Science Foundation under Grants ANI-9873328, ANI-0077039, ANI-0125979, ANI-0238299, DOE DE-FG02-02ER25528, and by DARPA under Contract DOD F30602-00-0554.

Y. Chait, C. V. Hollot, D. Towsley, H. Zhang, and Y. Cui are with the University of Massachusetts, Amherst, MA 01003 USA (e-mail: chait@ecs.umass.edu; hollot@ecs.umass.edu; towsley@cs.umass.edu; honggang@cs.umass.edu; ycui@ecs.umass.edu).

V. Misra is with Columbia University, New York, NY 10027-7003 USA (e-mail: misra@cs.columbia.edu).

Digital Object Identifier 10.1109/TNET.2005.852890

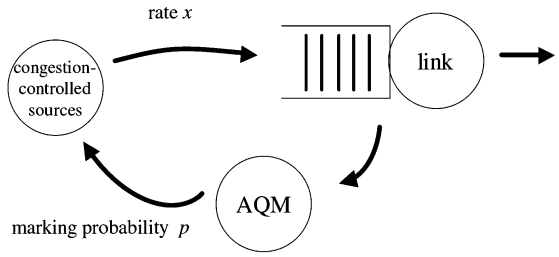


Fig. 1. Interaction between multiple TCP-controlled flows and a single congested AQM link.

Using fluid flow modeling [13] we give local stability conditions for a TCP-RENO network, relegating a complete stability analysis to [14]. Finally, ns simulations in Section VI demonstrate that our DiffServ architecture guarantees target rates in the face of realistic traffic variations.

II. Two-Level Coloring and Preferential Marking

The interaction of heterogenous, congested-controlled flows with a single congested link is depicted in Fig. 1. Under the congestion avoidance mechanism of TCP-Reno,[2] the equilibrium response function (corresponding to a fluid flow model, e.g., [14]), relating rate $\hat{x}$ to packet marking probability $\hat{p}$ is[3]

$$\hat{x} = \sum_{j \in J} \frac{N_j}{\hat{\tau}_j} \sqrt{\frac{2}{\hat{p}} - 2}$$

where J is a set of indexes keeping track of the aggregated homogenous flows, each comprised of N_j flows with $\hat{\tau}_j$ round-trip times. In this section we augment the setup in Fig. 1 with n edge routers, each handling an aggregated flow. Accordingly, we decompose J into n index sets $J_1, J_2, \ldots, J_n$ and write down the equilibrium response function for the ith aggregate flow x_i passing through the ith edge:

$$\hat{x}_i = \sum_{j \in J_i} \frac{N_j}{\hat{\tau}_j} \sqrt{\frac{2}{\hat{p}} - 2}. \tag{1}$$

Clearly, $x = \sum_i x_i$, where x_i may be heterogenous. In Fig. 2 we focus on the ith aggregate source x_i and show additional control mechanisms; namely, an ARM with corresponding target rate $\underline{x}_i$ at the edge, and, a preferentially-marking AQM at the core router. The role of these elements is to cooperatively guarantee that x_i achieves minimal rate $\underline{x}_i$ regardless of delay, marking probability and traffic load. This ARM mechanism, keyed to its target rate, colors the aggregate packets either green or red. Roughly speaking, packets are colored green if $x_i \leq \underline{x}_i$ and red otherwise. This produces a green flow rate $f_{gi}x_i$ with green fraction $f_{gi} \in [0,1]$. The corresponding red flow rate is $(1-f_{gi})x_i$. The AQM at the core marks green and red packets according to a preferential scheme to produce packet-marking probability pair (p_g, p_r). The packet-marking probability p_i experienced by the ith aggregate is then related to this green and red packet-marking probability by $p_i = f_{gi}p_g + (1-f_{gi})p_r$. Given target rates $\underline{x}_i$ and link capacity c, we classify a network to be *over-provisioned* if $c > \sum \underline{x}_i$, and *under-provisioned* when $c < \sum \underline{x}_i$. Our objectives are threefold:

i) In an over-provisioned network, to quantify ARM/AQM properties that guarantee target rates are achieved; i.e., $\hat{x}_i \geq \underline{x}_i$.
ii) In an over-provisioned network, to quantify the distribution of excess capacity amongst aggregates. Conversely, in an under-provisioned scenario, to quantify how the want of capacity is borne.
iii) To design/analyze ARM/AQM mechanisms that guarantee convergence to the rates $\hat{x}_i$ predicted in the above items.

In the next section, we tackle the first two objectives. The third objective is the focus of Sections IV–V.

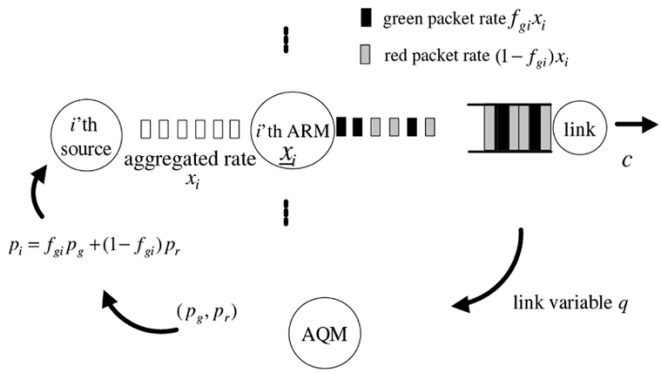


Fig. 2. An ARM mechanism colors packets green or red depending on how aggregate rate x_i compares to its target rate $\underline{x}_i$. This produces a green packet rate $f_{gi}x_i$ and red packet rate $(1-f_{gi})x_i$. The AQM marks these green and red packets with probability p_g and p_r, respectively.

III. ARMs and AQMs That Guarantee Target Rates

From Fig. 2 we see that an ARM amounts to a rule (possibly dynamic) from aggregates and target rates $(x_i, \underline{x}_i)$ to green flow ratio f_{gi}. Similarly, a AQM reacts to a link variable (such as arrival rate or buffer length) to produce a pair of green and red packet-marking probabilities (p_g, p_r). In this section we identify ARM/AQM properties that guarantee target rates are achieved. First we define the class of congestion-controlled flows considered here.

Definition 1: Congestion-controlled sources adjust aggregate rates x_i as a function of packet-marking probability p_i. The equilibrium response functions $\hat{x}_i = h_i(\hat{p}_i)$ describing n such congestion-controlled sources are said to be *admissible and compatible* if for each $i \in \{1, 2, \ldots, n\}$:

i) $h_i(\rho)$ is nonnegative and nonincreasing over $\rho \in [0,1]$.
ii) $h_i(\rho) > 0 \Rightarrow \rho < 1$.
iii) $h_i(\rho)$ becomes unbounded as $\rho \to 0$.
iv) there exists numbers $\alpha_1, \alpha_2, \ldots, \alpha_n$ such that $\alpha_i h_i(\rho) = \alpha_j h_j(\rho)$ for all $j \in \{1, 2, \ldots, n\}$ and $\rho \in (0,1)$. $\triangle$

[2]In this paper we model only the congestion avoidance part (AIMD) of TCP-Reno, i.e., we do not model slow start or timeout.

[3]The "ˆ" notation denotes the equilibrium value of variables. This helps distinguish from instantaneous values used in the description of network dynamics considered for stability analysis in Section V.

TCP-Reno is an example of admissible and compatible congestion-controlled sources. Their response functions, described in (1), are

$$h_i(\hat{p}_i) = \sum_{j \in J_i} \frac{N_j}{\hat{\tau}_j} \sqrt{\frac{2}{\hat{p}_i} - 2} \tag{2}$$

which satisfies properties (i)–(iii) and (iv) with the α_i taken as

$$\alpha_i = \frac{1}{\sum_{j \in J_i} \frac{N_j}{\hat{\tau}_j}}.$$

The class of proportionally-fair congestion-controlled sources (see [15]) are also admissible and compatible where $h_i(\hat{p}_i) = w_i/\hat{p}_i$ and w_i is the constant associated with rate increase. A mixture of TCP-Reno and proportionally-fair, congestion-controlled sources are *not compatible*. Finally, the flow rates of TCP-Vegas flows respond to queuing delay $\hat{\tau}_q$ (instead of marking probability) and have response functions $\hat{x}_i \propto \hat{\tau}/\hat{\tau}_q$. In the liberal sense that queuing delay, like marking probability, is a form of link price, TCP-Vegas sources can be considered admissible and compatible.

Definition 2: An ARM is a mechanism for setting the fraction of green flow f_{gi}, and is said to *fully color* if the following holds at equilibrium:

$$\begin{cases} \hat{x}_i < \underline{x}_i \Rightarrow \hat{f}_{gi} = 1 \\ \hat{x}_i > \underline{x}_i \Rightarrow \hat{f}_{gi} = 0. \end{cases} \tag{3}$$

△

Definition 3: Consider an AQM that marks green and red packets at rates p_g and p_r, respectively. We say that such an AQM has *nonoverlapping* marking if, at equilibrium, the following holds:

$$\begin{cases} \hat{p}_r < 1 \Rightarrow \hat{p}_g = 0 \\ \hat{p}_g > 0 \Rightarrow \hat{p}_r = 1. \end{cases} \tag{4}$$

△

In words, a fully-coloring ARM colors the whole flow green (or red) when rates x_i *persistently* underachieve (or exceed) their targets $\underline{x}_i$. Definition 3 rules out AQMs that would *persistently* mark both red and green packets with nontrivial probabilities.

In summary, the steady-state interaction between admissible and compatible congestion-controlled sources, fully-coloring ARMs, and nonoverlapping AQMs are described by (3), (4), and

$$\begin{aligned} \hat{x}_i &= h_i(\hat{p}_i) \\ \hat{p}_i &= \hat{f}_{gi}\hat{p}_g + (1 - \hat{f}_{gi})\hat{p}_r \end{aligned} \tag{5}$$

where $\hat{f}_{gi}, \hat{p}_g, \hat{p}_r \in [0,1]$, and where the h_i satisfy the conditions of Definition 1. We now consider *over-provisioned* networks and show that rates $\hat{x}_i$, satisfying the above, do indeed meet or exceed their targets $\underline{x}_i$.

Theorem 1: (see Appendix for proof) Consider a network with n admissible and compatible congestion-controlled sources, each equipped with a fully-coloring ARM, and an AQM having nonoverlapping marking. Let the nonzero target rates $\underline{x}_1, \underline{x}_2, \ldots, \underline{x}_n$ be given, and take the α_i's as in condition (iv) of Definition 1. Assume the aggregates are ordered such that

$$\alpha_1\underline{x}_1 \geq \alpha_2\underline{x}_2 \geq \cdots \geq \alpha_n\underline{x}_n. \tag{6}$$

If the network is over-provisioned, then the steady-state rates $\hat{x}_i$ satisfying (3)–(5) meet or exceed their target rates. Specifically, let i^* denote the smallest integer in $\{1, 2, \ldots, n\}$ for which

$$\frac{c - \sum_{i=1}^{i^*-1} \underline{x}_i}{\sum_{i=i^*}^{n} \frac{1}{\alpha_i}} - \alpha_{i^*}\underline{x}_{i^*} > 0. \tag{7}$$

Then,

$$\hat{x}_i = \begin{cases} \underline{x}_i, & i = 1, \ldots, i^* - 1 \\ \dfrac{c - \sum_{j=1}^{i^*-1} \underline{x}_j}{\alpha_i \sum_{j=i^*}^{n} \frac{1}{\alpha_j}} > \underline{x}_i, & i = i^*, \ldots, n. \end{cases} \tag{8}$$

□

Remarks on Theorem 1

(i) The index i^* defined by (7) indicates which aggregates in ordering (6) meet or exceed their target rates. Aggregates ranked greater than i^* exceed their target rates while those ranked lower do not.

(ii) It is instructive to compare the behavior of TCP-Reno with and without ARM mechanisms. In both cases, assume aggregates are ordered as in (6). Then, (2) describes how the equilibrium window size $\alpha_i\hat{x}_i$ depends on the packet-marking rate $\hat{p}_i$. Without an ARM, $\hat{p}_i$ is the same for all aggregates. Said another way, the marking in a TCP-Reno/AQM network is described by (5) with $\hat{f}_{gi} \equiv 0$. Consequently,

$$\alpha_1\hat{x}_1 = \alpha_2\hat{x}_2 = \cdots = \alpha_n\hat{x}_n = \sqrt{\frac{2}{\hat{p}_r} - 2} \triangleq \widehat{\text{win}}$$

which shows that TCP-Reno networks regulate windows to the same size $\widehat{\text{win}}$. In the presence of a network with fully-coloring ARM, note that $\alpha_i\underline{x}_i$ in (6) represents the *window size required to achieve target rate* $\underline{x}_i$. Thus, with ARM, window sizes are regulated according to ranking (6); the first class has the largest target window size, while the nth aggregate in this ordering has the smallest target window $\alpha_n\underline{x}_n$. On the other hand, this nth aggregate is the *most aggressive* in grabbing excess capacity as seen from the following reformulation of (8):

$$1 = \frac{\hat{x}_1}{\underline{x}_1} = \cdots = \frac{\hat{x}_{i^*-1}}{\underline{x}_{i^*-1}} < \frac{\hat{x}_{i^*}}{\underline{x}_{i^*}} \leq \cdots \leq \frac{\hat{x}_n}{\underline{x}_n}. \tag{9}$$

The reason for (9) is that the ARM overrides TCP's effort to have all flows (in a single congested link case) experience the same packet marking rate. The ARM/AQM mechanisms do this by coloring and marking to modulate the marking rate as in (5). Without ARMs, e.g., $\underline{x}_i = 0, f_{gi} \equiv 0, p_i = p_r$ and all packets are implicitly red. In contrast, an ARM modulates p_i, via f_{gi}, to achieve target rates in spite of the differing delay/load ratios α_i. Nevertheless, an ARM will defer to TCP-Reno when $\hat{x}_i > \underline{x}_i$, setting $\hat{f}_{gi} = 0$ so that $\hat{p}_i = \hat{p}_r$. Thus, when an aggregate exceed its target rate, TCP-Reno kicks-in and distributes the share of excess capacity according to the delay/load ratios α_i. ♢

It is instructive to focus on a special case where some aggregates are not colored. Without loss of generality, we consider one such aggregate, say the $(n+1)$th where $\underline{x}_{n+1} = 0$. The result below shows how an ARM/AQM-equipped network apportions rates to this aggregate.

Corollary 2: (See [16] for proof) Consider the set-up in Theorem 1 with an additional $(n+1)$th aggregate having $\underline{x}_{n+1} = 0$. If

$$\frac{c - \sum_{j=1}^{n-1} \underline{x}_j}{\sum_{j=n}^{n+1} \frac{1}{\alpha_j}} - \alpha_n \underline{x}_n > 0 \tag{10}$$

take i^* as in Theorem 1. Otherwise, let $i^* = n+1$. Then,

$$\hat{x}_i = \begin{cases} \underline{x}_i, & i = 1, \ldots, i^* - 1 \\ \frac{c - \sum_{j=1}^{i^*-1} \underline{x}_j}{\alpha_i \sum_{j=i^*}^{n+1} \frac{1}{\alpha_j}}, & i = i^*, \ldots, n \end{cases} \tag{11}$$

and

$$\hat{x}_{n+1} = \frac{c - \sum_{j=1}^{i^*-1} \underline{x}_j}{\alpha_{n+1} \sum_{j=i^*}^{n+1} \frac{1}{\alpha_j}} \geq 0.$$

□

Remark on Corollary 2: Flows that are not colored by ARMs do not alter the ability of ARM-controlled flows from achieving their target rates; however, they influence the distribution of excess capacity. For example, recall from (9) that aggregates ranked below i^* do not exceed their target rates. Thus, when $i^* = n+1$, the $(n+1)$th aggregate grabs all excess capacity. This event depends on the α_i (the delay-load ratios α_i in the case of TCP-Reno) and target rates $\underline{x}_i$. ♢

We conclude this section with the under-provisioned dual to Theorem 1.

Theorem 3: (see [16] for proof) \textit{Consider the set-up in Theorem 1 and suppose the network is under-provisioned. Let i^* denote the largest integer in $\{1, 2, \ldots, n\}$ for which

$$\alpha_{i^*} \underline{x}_{i^*} - \frac{c - \sum_{i=i^*+1}^{n} \underline{x}_i}{\sum_{i=i^*+1}^{n} \frac{1}{\alpha_i}} > 0.$$

Then, the steady-state rates $\hat{x}$ satisfy (3)–(5) as well as

$$\hat{x}_i = \begin{cases} \frac{c - \sum_{j=i^*+1}^{n} \underline{x}_j}{\alpha_i \sum_{j=1}^{i^*} \frac{1}{\alpha_j}} < \underline{x}_i, & i = 1, \ldots, i^* \\ \underline{x}_i, & i = i^* + 1, \ldots, n. \end{cases} \tag{12}$$

□

IV. Realization of ARM/AQMs

The results of the last section show that it is possible to regulate congestion-controlled flows provided the ARM/AQM mechanisms satisfy some basic properties. In steady-state it is sufficient for an ARM to *fully color* and the AQM to have *nonoverlapping marking*; i.e.,

$$\widehat{\text{ARM}}: \begin{cases} \hat{x}_i < \underline{x}_i \Rightarrow \hat{f}_{gi} = 1 \\ \hat{x}_i > \underline{x}_i \Rightarrow \hat{f}_{gi} = 0 \end{cases} \tag{13}$$

and

$$\widehat{\text{AQM}}: \begin{cases} \hat{p}_r < 1 \Rightarrow \hat{p}_g = 0 \\ \hat{p}_g > 0 \Rightarrow \hat{p}_r = 1. \end{cases} \tag{14}$$

A question remains on how to realize these steady-state behaviors in a network context. Also, it is essential that these structures contribute to an overall network dynamic that *converges* to desired steady-state behavior. This section deals with the first of these issues, while Section V addresses the convergence problem.

A. ARMs That Fully Color

One realization of (13) is the switching rule:

$$\text{ARM}: \begin{cases} x_i < \underline{x}_i \Rightarrow f_{gi} = 1 \\ x_i > \underline{x}_i \Rightarrow f_{gi} = 0. \end{cases}$$

Unlike (13), which is an equilibrium condition, the above rule holds at each instance and the discontinuous nature of this scheme, reminiscent of drop-tail AQM, could possibly lead to steady-state oscillation in regulating rates. A perhaps better-suited mechanism for achieving (13) employs integrator action as described by the differential equation

$$\frac{d}{dt} f_{gi} = \begin{cases} \underline{x}_i - x_i, & \text{if } f_{gi} > 0 \\ \underline{x}_i - x_i, & \text{if } f_{gi} = 0, \quad \underline{x}_i - x_i \geq 0 \\ 0, & \text{if } f_{gi} = 0, \quad \underline{x}_i - x_i < 0 \end{cases} \triangleq (\underline{x}_i - x_i)^+.$$

If $x_i > \underline{x}_i$ persists, then $f_{gi} \to 0$. Thus, this *integrating ARM* satisfies (13) and fully colors. For the purpose of affecting transient performance, the integrator is augmented with a proportional gain to give a general ARM structure

$$\begin{aligned} \frac{d}{dt} \zeta_i &= (\underline{x}_i - x_i)^+ \\ f_{gi} &= k_{Ii} \zeta_i + k_{pi} (\underline{x}_i - x_i). \end{aligned} \tag{15}$$

Parameters k_{Ii}, k_{pi} play a role in closed-loop stability and this proportional-integral ARM (PI-ARM) fully colors whenever $k_{Ii} > 0$. The PI-ARM is similar in structure to AQM schemes such as AVQ [12], PI [17] and REM [11]. The objective of these schemes is to regulate network variables exactly to target. For example, PI (or REM) regulate buffer lengths to target, while AVQ matches arrival rates to a desired level of utilization. The same holds true for PI-ARMs when they match rates x_i to targets $\underline{x}_i$. However, for rates x_i to exceed targets $\underline{x}_i$ (as in the over-provisioned case) the associated PI-ARMs become inactive allowing the native congestion-controlled protocol to regulate the excess throughput. A PI-ARM matches rates exactly by producing a green ratio $f_{gi} \in (0, 1)$. In allowing steady-state rates to exceed targets, a PI-ARM permits $f_{gi} \to 0$. This latter action distinguishes the integral action in ARM from that used in AQMs.

Coloring with a token bucket: Having discussed ways to insure that an ARM fully colors, we now consider implementation issues for an ARM including the coloring of flows. To produce the f_{gi} called for in (15), recall from [5] how colored flows are actually produced at the edge routers. The mechanism used there employs a so-called *token bucket*, as shown in Fig. 3, which is filled with tokens at rate ξ_i and are distributed to packets in x_i. These packets contain a *DiffServ bit* that is initially set to red. When a packet passes the bucket it "grabs" a token setting this bit to green. As long as x_i is less than ξ_i, tokens are available and packets colored green. If $x_i > \xi_i$, the bucket is eventually drained and only a fraction $f_{gi} = min\{\xi_i/x_i, 1\}$ of the

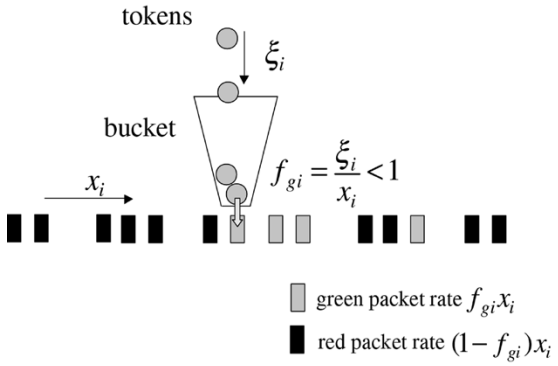


Fig. 3. Schematic of a token bucket mechanism.

flow is served tokens and colored[4]. As shown in [4], using a token bucket is not sufficient to regulate an aggregate rate to a target rate, and this observation was precisely the motivation for our introduction of an ARM which uses feedback to regulate these rates by measuring the actual flow x_i and then *adaptively* adjusting bucket rates ξ_i to guarantee target rates. Combining the token-bucket mechanism described above with the proportional-integral structure in (15) gives a complete description of a fully-coloring ARM:

$$\begin{aligned} \frac{d}{dt}x_{\text{avg},i} &= -kx_{\text{avg},i} + k\tilde{x}_i \\ \frac{d}{dt}\zeta_i &= (\underline{x}_i - x_{avg,i})^+ \\ \xi_i &= k_{Ii}\zeta_i + k_{pi}(\underline{x}_i - x_{\text{avg},i}) \\ f_{gi} &= \min\{\xi_i/x_{\text{avg},i}, 1\} \end{aligned} \tag{16}$$

where $\tilde{x}_i$ denotes an estimate of aggregate rate computed by counting total number of sent packets in a fixed period divided by that period. This estimate is passed through a low-pass filter with time constant $1/k$ to produced a smooth rate-estimate $x_{\text{avg},i}$. This estimate is the input to the proportional-integral part of the ARM. Note that a target rate $\underline{x}_i$ need not be a constant. Our scheme is effective as long as a time-varying target rate profile $\underline{x}_i(t)$ does so on a time scale slower than the responsiveness of the congestion-control protocol.

B. AQMS With Non-Overlapping Marking

Equation (14) describes an AQM with nonoverlapping marking. In this subsection we look at how proposed AQM schemes can be modified to meet this condition.

Preferentially-marking RED: RED establishes packet-marking rate p based on the core-router's queue length q. A continuous-time description of RED consists of a low-pass filter [13]:

$$\frac{d}{dt}q_{\text{avg}} = k_{\text{red}}q_{\text{avg}} + k_{\text{red}}q$$

where k_{red} is the filter pole, and an additional gain stage described by the profile in Fig. 4(a). A *preferentially-marking RED* [18] employs two gain profiles: a green profile $q_{\text{avg}} \rightarrow p_g$ and a red one $q_{\text{avg}} \rightarrow p_r$. This AQM satisfies the condition of Definition 3 if these profiles *do not overlap* as illustrated in Fig. 4(b).

[4]Token buckets are also equipped with a buffer to accumulate "unused" tokens allowing for the coloring of short bursts when $x_i > \xi_i$. We will not model this feature here and assume the buffer length is sufficiently large.

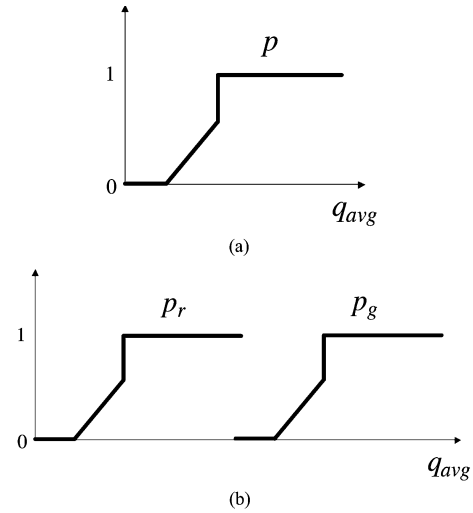


Fig. 4. (a) A typical marking profile for RED. (b) Marking profiles that qualify this RED as having nonoverlapping marking.

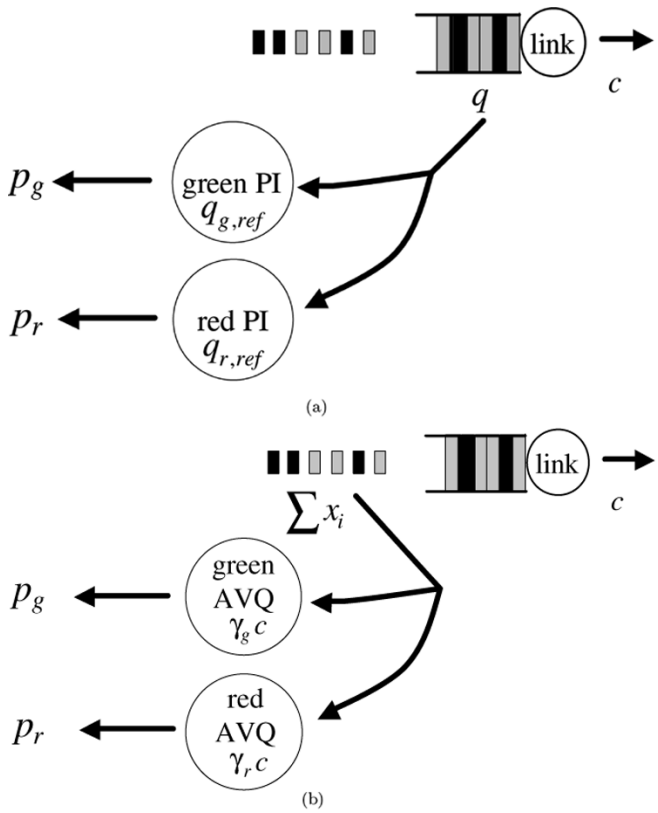


Fig. 5. (a) A PI (REM) AQM characterized by two proportional-integrator (PI) elements and separate green and red queue reference levels $q_{r,\text{ref}}$ and $q_{g,\text{ref}}$. A similar structure defines the AVQ in (b) which has separate green and red utilization targets.

Specifically, if $p_r < 1$, then $p_g = 0$. Conversely, if $p_g > 0$, then $p_r = 1$. We note this is more stringent than (14) where nonoverlapping is required only in the steady state. The nonoverlapping in Fig. 4(b) holds for all instances.

Preferentially-marking PI and REM: The PI and REM AQMs are buffer-based and are distinguished by their ability to regulate buffers to desired queue lengths. Their structure is similar to that of the ARM structure where $(x_i, \underline{x}_i, f_{gi})$ is replaced by $(q, q_{g,\text{ref}}, q_{r,\text{ref}}, p_g, p_r)$ where $q_{\bullet,\text{ref}}$ denotes the

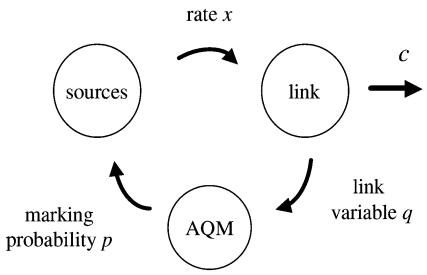


Fig. 6. Schematic of an AQM congestion-controlled network.

desired red or green queue length. A preferentially-marking PI (REM) consists of two standard PIs (REMs), one producing green marking rate p_g and one for p_r. The structure is given in Fig. 5(a) where the desired queue lengths satisfy $q_{r,\mathrm{ref}} < q_{g,\mathrm{ref}}$. We claim this two-level PI (REM) has nonoverlapping marking. To see this, assume $\hat{p}_r < 1$. We need to show that $\hat{p}_g = 0$. Since $\hat{p}_r < 1$, then, due to red integrator action, $\hat{q} \leq q_{r,\mathrm{ref}}$. Indeed, if $\hat{q} > q_{r,\mathrm{ref}}$, the red PI integrates toward $+\infty$ so that $\hat{p}_r \to 1$. Therefore, $\hat{q} \leq q_{r,\mathrm{ref}} < q_{g,\mathrm{ref}}$ which then implies that the green PI integrates toward $-\infty$ so that $\hat{p}_g \to 0$. The second part of (14) follows by similar logic.

Preferentially-marking AVQ: An adaptive virtual queue (AVQ) is a proportional-integral AQM acting on the link's arrival rate $\sum x_i$. Consequently, its structure is again similar to the ARM structure where $(x_i, \underline{x}_i, f_{gi})$ is replaced by $(x_i, \gamma_g c, \gamma_r c, p_g, p_r)$ with $\gamma_g, \gamma_r \in (0,1)$ denoting desired utility. The notion of *virtual queue* derives its name from the fact that input to the AVQ's integrator is arrival rate, and hence this integrator's output is a virtual queue length. A two-level AVQ is shown in Fig. 5(b). When $\gamma_r < \gamma_g$ this AVQ has nonoverlapping marking for exactly the same reason as the two-level PI (REM).

V. Stability: Convergence to Target Rates

In previous sections we have established the steady-state rates of networks employing active rate management and preferentially-marking active queue management. Now, we address how these schemes interact to *converge* to this equilibrium. Global convergence will be evaluated from network simulations conducted in the next section. Presently, we are concerned with the local attractivity of the equilibrium. Recent research has focused on the stability of congestion-controlled networks as illustrated in Fig. 6. We modify this setup to include ARMs and a two-level AQM, and to evaluate stability, we will consider a simple case where two congestion-controlled aggregates are controlled by fully-coloring ARMs as shown in Fig. 7(a). These congestion-controlled sources produce aggregate rates x_i in response to packet-marking rates p_i while the ARMs control the fraction of green-colored flows f_{gi}. Together with the two-level AQM, this ARM modulates the marking probabilities per $p_i = (1 - f_{gi})p_r + f_{gi}p_g$. The feedback loop is completed with the AQM setting marking rates (p_g, p_r) as a function of link variable q. Our present objective is to provide a conceptual argument for the local stability of this feedback connection. A more detailed stability proof for the general case can be found in [14]. To this end, suppose the network in Fig. 7(a) is over-provisioned

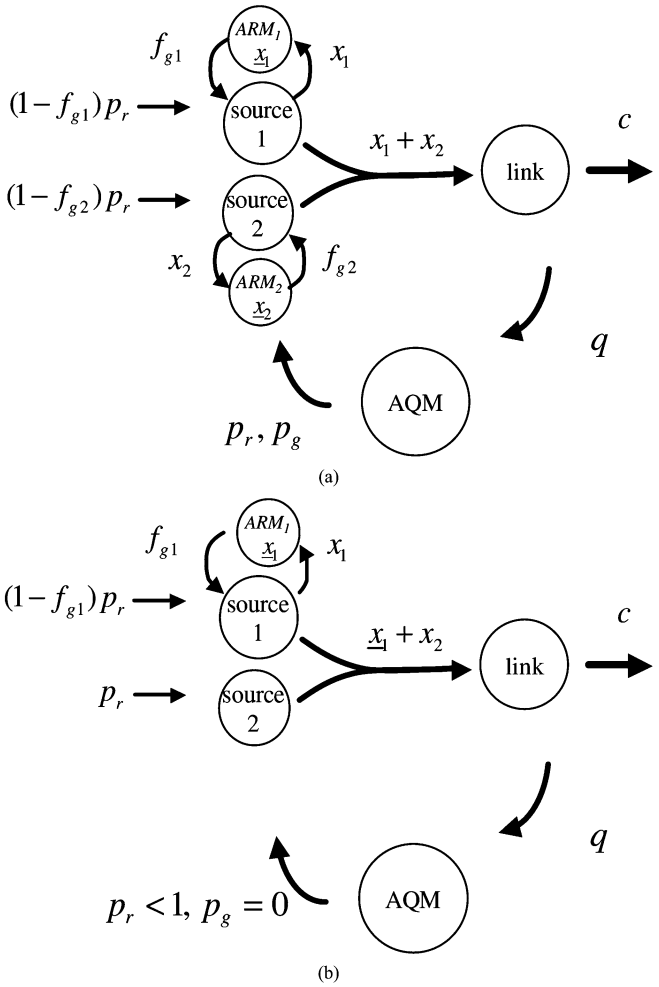


Fig. 7. In (a) the network is over-provisioned and parameters are such that $i^* = 1$ (see Theorem 1). Thus, at equilibrium, the first aggregate exactly meets its target $\hat{x}_1 = \underline{x}_1$, while for the second aggregate exceeds it $\hat{x}_2 > \underline{x}_2$. (b) shows that ARM_1 remains *active* while ARM_2 has *deactivated* allowing aggregate 2 to be controlled solely by the AQM's red marking probability p_r.

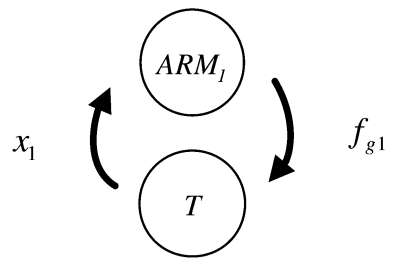


Fig. 8. Fig. 7(b) reduces to a negative feedback connection between ARM_1 and a dynamic T which models the interaction between aggregates and AQM *in the absence* of ARMs.

and parameters are such that index i^*, introduced in Theorem 1, satisfies $i^* = 1$. Thus, in the steady state, the first aggregate exactly meets its target, $\hat{x}_1 = \underline{x}_1$, while the second aggregate rate $\hat{x}_2$ exceeds $\underline{x}_2$. In achieving this, ARM_1 remains *active* while the second ARM has *deactivated* allowing rate x_2 to be controlled solely by the AQM's red marking rate p_r. Specifically, when $\hat{x}_i > \underline{x}_i$, the ARM enforces $f_{gi} = 0$ so $p_i = p_r$ implying the aggregate is under TCP control. This effect is shown in Fig. 7(b) which can then be expressed as a negative feedback connection between ARM_1 and a dynamic T as shown in Fig. 8.

To address local stability we let transfer function $T(s)$ represent the linearization of this T and likewise model the small-signal equivalent of the integrating ARM_1 in (16) by

$$\mathrm{ARM}_1(s) = \kappa \frac{s/z+1}{s} \frac{k}{s+k} \tag{17}$$

where $\kappa = k_{I1}$ and $z = k_{p1}/k_{I1}$. T models the interaction between aggregate and AQM *in the absence* of ARMs. It seems plausible to assume that an AQM stabilizes the aggregates in this situation. In this case, when $T(s)$ is a stable transfer function, we now show that there exists a stabilizing ARM_1 for the system in Fig. 8.

Proposition 4: Suppose $T(s)$ is proper[5] and stable with $T(0) > 0$. Then, the feedback system in Fig. 8 with $\mathrm{ARM}_1(s)$ from (17) is locally stable for $k > 0$ and some sufficiently small, positive gain κ.

Proof: First, write $\mathrm{ARM}_1(s)T(s) = \kappa \tilde{T}(s)/s$. From the assumptions, $\tilde{T}(s)$ is proper, stable and $\tilde{T}(0) > 0$. The negative feedback system in Fig. 8 is locally stable provided

$$1 + \kappa \frac{\tilde{T}(s)}{s} = 0 \tag{18}$$

has solutions only in the open left-half of the complex plane (OLHP). We claim this is so for some sufficiently small and positive κ. We will use root locus arguments to prove this; see [19]. Since $\tilde{T}(s)$ is proper and has only poles in the open left-half of the complex plane[6], then for sufficiently small, positive root-locus parameter κ, the solutions to (18) are sufficiently close to the poles of $\frac{\tilde{T}(s)}{s}$ and clearly in the OLHP, except possibly for the small real solution emanating from the pole at the origin. To investigate, let this solution be $s = \epsilon$. We claim that $\epsilon < 0$. To see this, evaluate (18) at $s = \epsilon$ to obtain

$$1 + \kappa \frac{\tilde{T}(\epsilon)}{\epsilon} = 0.$$

Since $\tilde{T}(\epsilon) > 0$ (recall $\tilde{T}(0) > 0$) and $\kappa > 0$, then ϵ must be negative. We have thus shown that (18) has solutions only in the OLHP when κ is sufficiently small and positive. This proves the proposition. □

VI. Illustrative Simulations

To validate the proposed ARM/AQM DiffServ architecture, we consider a network simulation consisting of three aggregated flows, each served by an edge with fully-coloring ARM as shown in Fig. 9. These edges feed into a core router with AQM having nonoverlapping marking. The propagation delays are fixed except between sources and edges where they range uniformly as shown in Fig. 9. The ith aggregate consists of n_i FTP flows with starting times uniformly distributed over [0, 50] seconds. The core link is OC-3 with 155 Mb/s capacity, 20.6 ms buffer, ECN marking and 500 Bytes-sized packets.

[5]A transfer function is said to be *proper* if its number of zeros is equal to its number of poles.

[6]Typically, the dynamic T will contain delay terms (corresponding to the end-to-end delay in congestion-controlled aggregates receiving marked packets), and hence the transfer function $T(s)$ has an infinite number of poles. Since these poles are determined by the zeros of a delay polynomial with *principal part*, these poles are strictly bounded away from the imaginary axis of the complex plane.

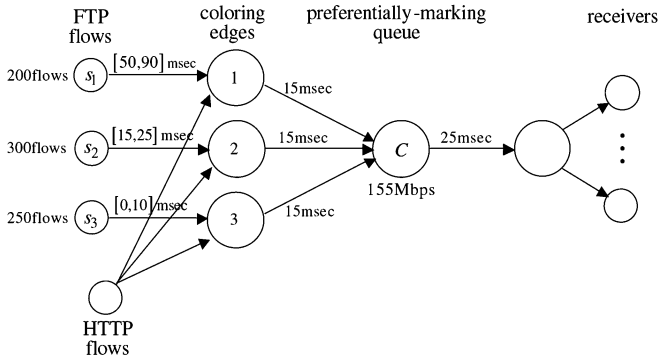


Fig. 9. The ns simulated DiffServ network.

A. Description of ARM/AQM Controllers

As described in Section IV, we use PI structures to implement the ARM/AQM controllers. Our AQM is buffer-based, utilizing two PI controllers (for red and green flows). In terms of the Laplace transforms of queue length $q(s)$ and marking rates $(p_g(s), p_r(s))$, these PI controllers are defined by

$$p_g = \mathrm{AQM}_{\mathrm{green}}(q - q_{g,\mathrm{ref}})$$
$$p_r = \mathrm{AQM}_{\mathrm{red}}(q - q_{r,\mathrm{ref}}),$$

where the transfer functions $\mathrm{AQM}_{\mathrm{green}}(s)$ and $\mathrm{AQM}_{\mathrm{red}}(s)$ are equivalent and designed according to the rules in [17] as

$$\mathrm{AQM}_{\mathrm{green}} = \mathrm{AQM}_{\mathrm{red}} = \frac{9.6 \times 10^{-6}\left(\frac{s}{0.53}+1\right)}{s}.$$

The green and red set points are $q_{g,\mathrm{ref}} = 12.9$ ms (500 packets) and $q_{r,\mathrm{ref}} = 5.16$ ms (200 packets), respectively. By design, these AQMs have nonoverlapping marking. The three ARM controllers have structure similar to the above PIs, and are described by (16) where $k = 1$ s, $k_{Ii} = 0.05$ and $k_{pi} = 0.5$. Consequently, these ARMS fully color. Estimated aggregate rates, the $\hat{x}_i$ in (16), are generated by counting total number of incoming packets in a period of 1 s. Token buckets have a size of 0.2 Mb. Finally, both ARM and AQM designs are discretized with a sampling rate of 37.5 Hz for ns implementation[7].

B. ns Packet-Level Experiments

We now present a series of experiments performed with ns to demonstrate feasibility and performance of our ARM/AQM design. Experiment 1 investigates performance of a 20% over-provisioned network under varying conditions such as transient FTP flows and HTTP flows. The effect of an aggregate with a zero target rate is studied in Experiment 2. Experiment 3 involves a 20% under-provisioned network, while Experiment 4 studies the division of bandwidth in a heterogeneous aggregate in an over-provisioned network. Finally, we apply our DiffServ scheme to a network with multiple congested links in Experiment 5. All FTP and HTTP sources use the TCP Reno protocol.

Experiment 1: The target rates for this over-provisioned network are $\underline{x}_1 = 70.5$ Mb/s, $\underline{x}_2 = 17$ Mb/s and $\underline{x}_3 = 42.5$ Mb/s. We introduce load variations and transients to each aggregate to

[7]Complete ns code can be downloaded from http://www.ecs.umass.edu/mie/labs/dacs/index2.html.

TABLE I
TIME AND FTP LOAD VARIATION

source	(time, $\pm\Delta$ FTP flows)		
1	$(100, +40)$	$(150, -80)$	$(200, +40)$
2	$(125, -60)$	$(175, +120)$	$(220, -60)$
3	—	$(190, +50)$	$(240, +50)$

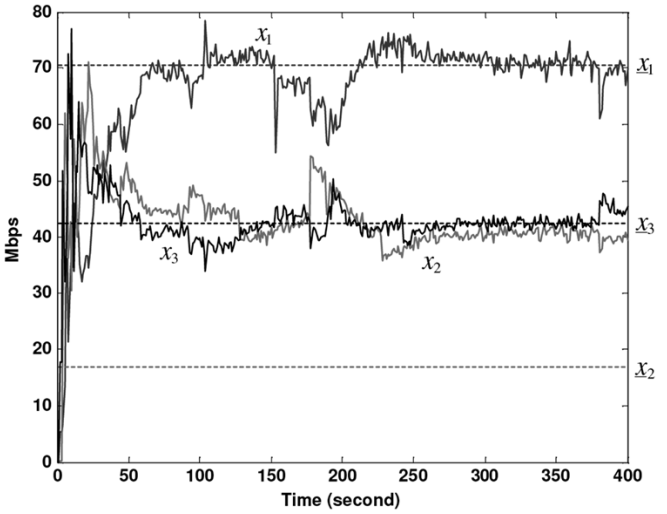


Fig. 10. Aggregate throughput (solid) and target rates (dashed) in Experiment 1.

study the ARM's ability to adapt to such variations. Table I lists the time and FTP load variation by sources.

Transients are introduced by short-lived flows at each edge: 2000 HTTP clients with exponential starting distribution; each client opening a connection containing one document (500 Bytes) and one image (2000 Bytes). These HTTP flows have zero target rates and consume less than 20% of link capacity. To predict performance of the DiffServ controls we form the ranking in (6):

$$\alpha_1 \underline{x}_1 > \alpha_3 \underline{x}_3 > \alpha_2 \underline{x}_2$$

(numerically: $79\,312 > 16\,150 > 7083$). From Theorem 1 we then have $i^* = 3$ which means that aggregates 1 and 3 should achieve their target rates while aggregate 2 exceeds its, and grabs excess capacity. The simulations confirm this as shown in Fig. 10.

Experiment 2: This experiment illustrates the behavior of an over-provisioned network having an aggregate flow that is not under DiffServ control. We model this as in Experiment 1 with an additional fourth aggregate having $\underline{x}_4 = 0$ consisting of 300 FTP flows traversing a path similar to that of aggregate 3. Fig. 11 supports Corollary 2 and shows that aggregates, not party to DiffServ, are allowed share of excess capacity in an over-provisioned network. Quantitatively, (10) in Corollary 2 gives $i^* = 3$ implying that aggregates 1 and 3 meets their target rates, while the second aggregate exceeds target and shares the over-capacity with the fourth aggregate. Moreover, from (11), Figs. 10 and 11, we see that all the bandwidth taken by this fourth aggregate comes from aggregate 2's excess.

Experiment 3: Here we return to the setting of Experiment 1 but make it an under-provisioned network by increasing target

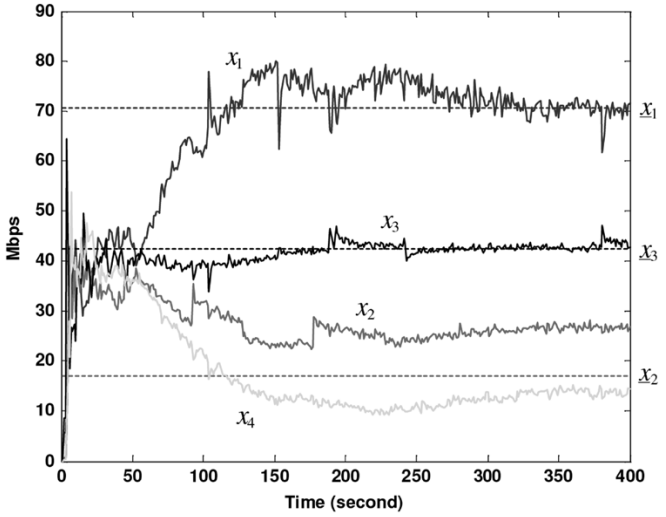


Fig. 11. Aggregate throughput (solid) and target rates (dashed) in Experiment 2.

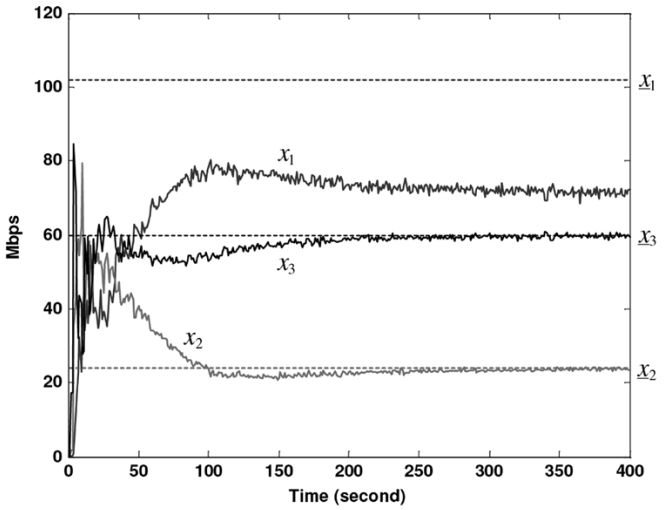


Fig. 12. Aggregate throughput (solid) and target rates (dashed) Experiment 3.

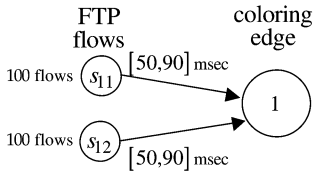


Fig. 13. Modified network topology at edge 1 in Experiment 4.

rates 20%: $\underline{x}_1 = 84.6$ Mb/s, $\underline{x}_2 = 20.4$ Mb/s and $\underline{x}_3 = 51$ Mb/s. Since all $\underline{x}_i$ are scaled by 20%, the aggregate ranking (6) in Experiment 1 remains the same: $\alpha_1 \underline{x}_1 > \alpha_3 \underline{x}_3 > \alpha_2 \underline{x}_2$. Using Theorem 3 we compute $i^* = 1$, that is, only aggregate 2 should achieve its target rate while aggregates 1 and 3 should not meet theirs due to insufficient core capacity as in (12). This is confirmed by the simulation results shown in Fig. 12.

Experiment 4: The purpose of this test is to study how bandwidth of a single aggregate is distributed between two of its heterogenous constituents. The setting is the over-provisioned network shown in Fig. 9 with 155 Mb/s link capacity. Now, however, aggregate 1 consists of two heterogenous flows as shown in Fig. 13. The target rates for this over-provisioned network are $\underline{x}_1 = 80$ Mb/s, $\underline{x}_2 = 20$ Mb/s and $\underline{x}_3 = 50$ Mb/s. In this case,

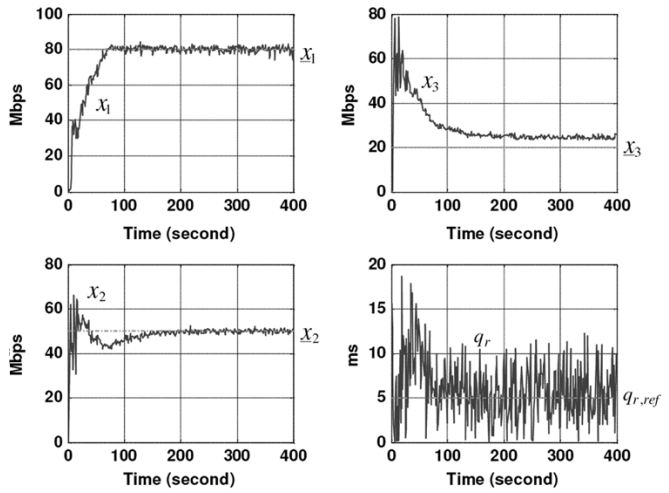


Fig. 14. Aggregate throughput (solid) and target rates (dashed) in Experiment 4.

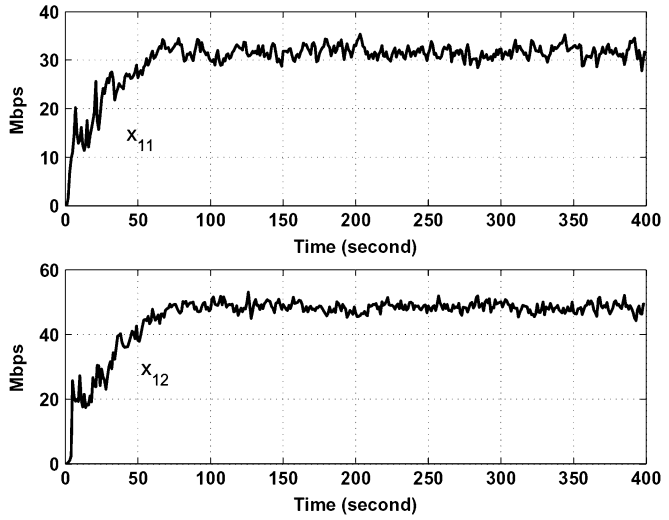


Fig. 15. Bandwidth distribution among the two heterogenous sets of flows in aggregate 1 ($x_1 = x_{11} + x_{12}$).

we compute the equivalent aggregate parameters feeding into edge 1 using (1): $\alpha_1 = (n_{11}/\hat{\tau}_{11} + n_{12}/\hat{\tau}_{12})^{-1} = (100/.225 + 100/.145)^{-1} = 8.8176e^{-4}$. The aggregate ranking in (6) remains the same as in Experiment 1. Thus, in Fig. 14 we see that aggregate 2 grabs all excess capacity. The bandwidth distribution among the two constituent flows of aggregate 1 is governed by TCP; i.e., aggregates with longer round-trip-time receive smaller fraction of resources. Specifically, the fraction of the first flow x_{11} in aggregate 1 is

$$\frac{\frac{100}{.225}}{\left(\frac{100}{.225} + \frac{100}{.145}\right)} \times 100 = 40\%$$

as observed in Fig. 15.

Experiment 5: The objective of this experiment is to examine the feasibility of our DiffServ architecture for multiple congested core routers. We consider a network with multiple congested links as shown in Fig. 16. There are two OC-3 core links (R1 and R2) each with capacity of 155 Mb/s. We utilize here the AQM and ARM parameters used in single congested link network in Experiments 1–4. The reference queue levels for

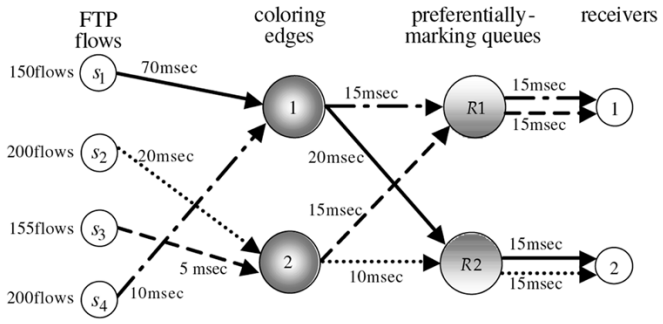


Fig. 16. Simulation topology for Experiment 5.

TABLE II
TARGET RATE SCHEDULE

time (secs)	$\underline{x}_1$ (Mbs)	$\underline{x}_2$ (Mbs)
0	80	60
100	160	160
400	168	168
700	176	176
1000	184	184
1300	80	60

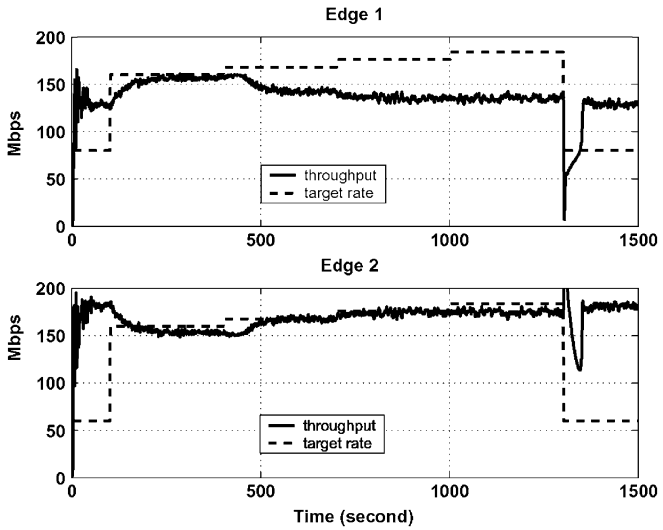


Fig. 17. Edges throughput (solid) with dynamically changing target rates (dashed).

the AQMs are 130 ms for green flow and 50 ms for red flow. Our test begins with an over-provisioned network, and by introducing time-varying target rates at the edges, i.e., target profiles, we transition the network from being over-provisioned, to exactly-provisioned, to under-provisioned, and, finally returning to an over-provisioned network. This is accomplished with the target rate schedule shown in Table II.

These time-varying target rates and resulting throughputs are shown in Fig. 17. At $t = 100$ s where $\underline{x}_1 = \underline{x}_2 = 160$, both throughputs are essentially equal to their minimum guaranteed rates and so the network is almost exactly provisioned. As we further increase target rates, edge 2 is able to maintain its increasing target rate at the expense of edge 2's throughput which is below its target. In general, the network can become under-provisioned if either of the links has limited capacity. In this test, both links become under-provisioned as observed in

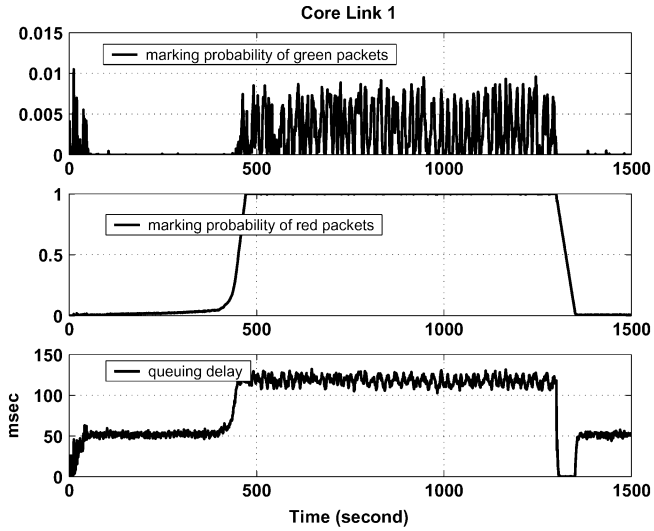


Fig. 18. Green and red packet marking probabilities and queuing delay at core link R1.

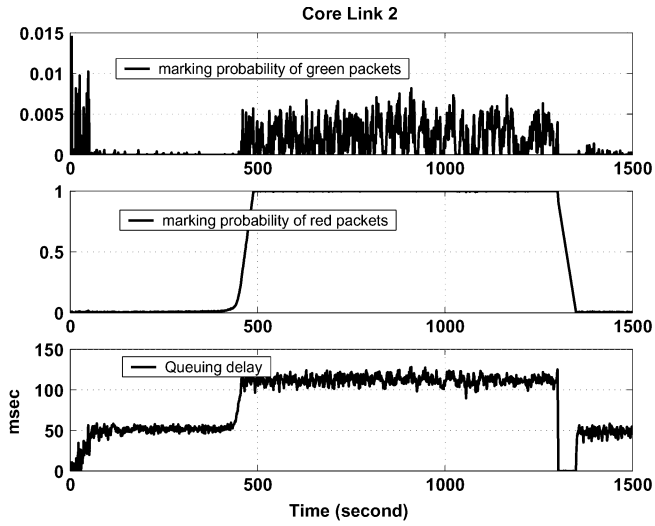


Fig. 19. Green and red packet marking probabilities and queuing delay at core link R2.

Figs. 18 and 19. Specifically, at $t = 400$ s, target rates are increased to 168 Mb/s and the links settle at their green queuing delay set points, red marking probabilities are 1 and core routers begin marking green packets. Eventually, with further increase in target rates, the network cannot even achieve $\underline{x}_2$. It is interesting to note that behind the edges, bandwidth distribution is governed by TCP. In general, a aggregate with a smaller round-trip-time will grab a larger share of the bandwidth. However, such aggregates need not traverse the same path and may experiences different marking probabilities. For example, aggregates 1 and 4 enter the network via edge 1, with aggregate 1 having a larger round trip time. The marking probabilities of both links are similar, hence aggregate 4 grabs the lion's share of edge 1's throughput as seen in Fig. 20.

VII. Conclusion

In this paper, we presented a new control architecture for achieving target rates in congestion-controlled networks. It consists of edge coloring using token buckets whose rates are adjusted dynamically, and preferential treatment in the core. Assuming a general class of source-link algorithms, we derived formulae for steady-state aggregate rates in over-provisioned and under-provisioned networks. We established general rules for edge coloring and core active queue management, sufficient to guarantee target rates. We presented realizations of active rate management controllers for edge coloring and an active queue management controller consistent with the above rules. Local stability for TCP RENO networks with this new ARM/AQM architecture was discussed. The validity of our results using the above realizations was verified using `ns` simulations. Our aggregate rate equilibrium formulae could be used by bandwidth brokers in a DiffServ network to distribute excess capacity in some prescribed manner.

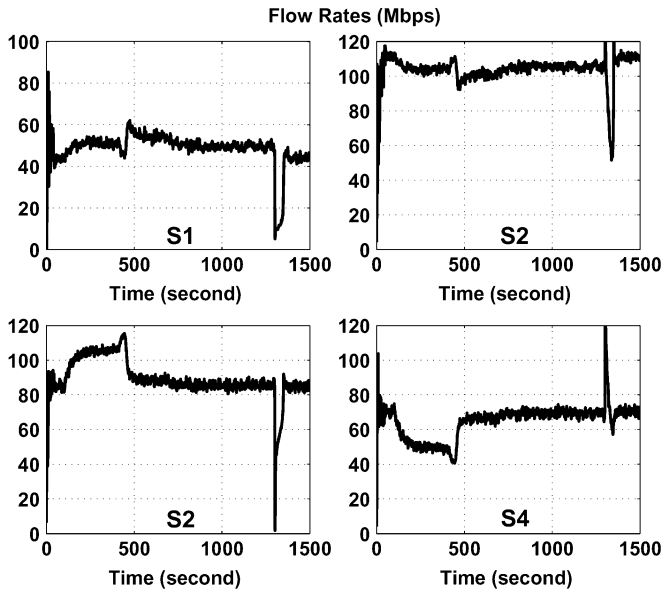


Fig. 20. Individual aggregate throughputs.

Future research on this DiffServ architecture can be taken in a number of directions. The allocation of excess bandwidth in over-provisioned networks is an important issue, and we are interested in finding out whether coloring edges can distribute this excess according to some fairness measure using TCP-like probing mechanisms. A similar question can be posed for the under-provisioned case, and, related to both is the possible benefit of adding a third color at the edge. Another challenge is to extend the equilibrium analysis in this paper to cover the case of multiple congested links—where even the notion of an *over-provisioned* network is not immediately evident.

Appendix I
Proof of Theorem 1

Before we prove the theorem, we need the following lemma.

Lemma 1: Consider an over-provisioned network with n admissible and compatible congestion-controlled aggregates, an ARM that fully colors and an AQM with nonoverlapping marking.

Then,

i) The red packet-marking rate satisfies $\hat{p}_r < 1$.

ii) All aggregates achieve their target rates; i.e.,

$$\hat{x}_i \geq \underline{x}_i, \quad i = 1, \ldots, n.$$

iii) If $\hat{x}_{i^*} > \underline{x}_{i^*}$ for some $i^* \in \{1, \ldots, n-1\}$, then

$$\hat{x}_i > \underline{x}_i, \quad i \in \{i^*, \ldots, n\}.$$

iv) If $\hat{x}_{i^*} = \underline{x}_{i^*}$ and $\hat{x}_{i^*+1} > \underline{x}_{i^*+1}$ for some $i^* \in \{1, \ldots, n-1\}$, then

$$\alpha_{i^*} \underline{x}_{i^*} \geq \alpha_j \hat{x}_j, \quad j \in \{i^*+1, \ldots n\}.$$

Proof: For each i, recall from (5) that

$$\hat{p}_i = (1 - \hat{f}_{gi})\hat{p}_r + \hat{f}_{gi}\hat{p}_g. \tag{19}$$

Now, at least one aggregate exceeds its target rate, say $\hat{x}_{i^*} > \underline{x}_{i^*}$ for some $i^* \in \{1, \ldots, n\}$. Since the ARM fully colors, then $\hat{f}_{gi^*} = 0$ so that from (19), $\hat{p}_{i^*} = \hat{p}_r$. Thus, $0 < g_i(\hat{p}_i) = g_i(\hat{p}_r)$ which implies that $\hat{p}_r < 1$ and proves (*i*). Since the AQM has nonoverlapping marking, $\hat{p}_g = 0$ and, as a result (19) becomes

$$\hat{p}_i = (1 - \hat{f}_{gi})\hat{p}_r \tag{20}$$

for all i. (20) will be in force for the remainder of the proof.

To show (*ii*) we proceed by contradiction and assume $\hat{x}_{i^*} < \underline{x}_{i^*}$ for some $i^* \in \{1, \ldots, n\}$. Since the ARM colors fully, $\hat{f}_{gi^*} = 1$ so that from (20), $\hat{p}_{i^*} = 0$. But, from Definition 3, $g_{i^*}(\rho)$ becomes unbounded as $\rho \to 0$ which contradicts the assumption $\hat{x}_{i^*} < \underline{x}_{i^*}$.

To prove (*iii*), we first conclude $\hat{f}_{gi^*} = 0$ since the ARM fully colors. Then, from (20), $\hat{p}_{i^*} = \hat{p}_r$ so that $g_{i^*}(\hat{p}_{i^*}) = g_{i^*}(\hat{p}_r)$. Again from (20) and the nonincreasing property of the g_i's:

$$\hat{x}_i = g_i(\hat{p}_i) = g_i(1 - \hat{f}_{gi}\hat{p}_r) \geq g_i(\hat{p}_r), \quad i \in \{1, \ldots n\}.$$

Since the aggregate are compatible

$$\hat{x}_i \geq g_i(\hat{p}_r) = \frac{1}{\alpha_i}\alpha_{i^*} g_{i^*}(\hat{p}_r) = \frac{1}{\alpha_i}\alpha_{i^*} g_{i^*}(\hat{p}_{i^*}) = \frac{1}{\alpha_i}\alpha_{i^*} x_{i^*}.$$

Let $i > i^*$. Then, from $\hat{x}_{i^*} > \underline{x}_{i^*}$ and (6) we have

$$\hat{x}_i \geq \frac{1}{\alpha_i}\alpha_{i^*}\hat{x}_{i^*} > \frac{1}{\alpha_i}\alpha_{i^*}\underline{x}_{i^*} \overset{(6)}{\geq} \frac{1}{\alpha_i}\alpha_i \underline{x}_i = \underline{x}_i.$$

Finally, to prove (*iv*) we start with $\hat{x}_{i^*} = \underline{x}_{i^*}$. Then,

$$\alpha_{i^*}\underline{x}_{i^*} = \alpha_{i^*}\hat{x}_{i^*} \geq \alpha_{i^*} g_{i^*}(\hat{p}_r).$$

Since $\hat{x}_{i^*+1} > \underline{x}_{i^*+1}$, then $\hat{f}_{gi^*+1} = 0$ and $\hat{x}_{i^*+1} = g_{i^*+1}(\hat{p}_r)$. Thus,

$$\alpha_{i^*}\underline{x}_{i^*} \geq \alpha_{i^*} g_{i^*}(\hat{p}_r) \overset{(6)}{\geq} \alpha_{i^*+1} g_{i^*+1}(\hat{p}_r) = \alpha_{i^*+1}\hat{x}_{i^*+1}.$$

We invoke (*iii*) and repeat the proof for $j \in \{i^*+1, \ldots, n\}$. □

Proof of Theorem 1: First, we show that (7) is nonvacuous. Since $c > \sum_{i=1}^{n} \underline{x}_i$, it follows that

$$\alpha_n \left(c - \sum_{i=1}^{n-1} \underline{x}_i \right) - \alpha_n \underline{x}_n > 0.$$

Hence, $i^* = n$ satisfies (7). Next, we prove (8). From (*iii*) of Lemma 1, there exists an $\hat{i} \in \{1, 2, \ldots n-1\}$ such that

$$\hat{x}_i \begin{cases} = \underline{x}_i, & i = 1, \ldots, \hat{i}-1 \\ > \underline{x}_i, & i = \hat{i}, \ldots, n. \end{cases} \tag{21}$$

We claim that $i^* = \hat{i}$. Proceeding by contradiction, assume $\hat{i} \neq i^*$. We consider two cases.

(*Case 1*: $\hat{i} < i^*$) In this situation

$$c = \sum_{i=1}^{\hat{i}-1} \underline{x}_i + \sum_{i=\hat{i}}^{n} \hat{x}_i. \tag{22}$$

Since $\hat{x}_i > \underline{x}_i$ for $i \in \{\hat{i}, \ldots, n\}$, then $\hat{f}_{gi} = 0$ for this same range of i since the ARM fully colors. It then follows from (20) that $\hat{p}_i = \hat{p}_r$ for all $i \in \{\hat{i}, \ldots, n\}$. Hence, since the aggregates are compatible

$$\alpha_{\hat{i}}\hat{x}_{\hat{i}} = \alpha_{\hat{i}} g_{\hat{i}}(\hat{p}_{\hat{i}}) = \alpha_{\hat{i}+1} g_{\hat{i}+1}(\hat{p}_{\hat{i}+1}) = \cdots = \alpha_n \hat{x}_n.$$

Combining this with (22) gives

$$\frac{c - \sum_{i=1}^{\hat{i}-1} \underline{x}_i}{\sum_{i=\hat{i}}^{n} \frac{1}{\alpha_i}} = \alpha_{\hat{i}}\hat{x}_{\hat{i}}.$$

By assumption $\hat{x}_{\hat{i}} > \underline{x}_{\hat{i}}$, so

$$\frac{c - \sum_{i=1}^{\hat{i}-1} \underline{x}_i}{\sum_{i=\hat{i}}^{n} \frac{1}{\alpha_i}} - \alpha_{\hat{i}}\underline{x}_{\hat{i}} > 0.$$

Using (7) we conclude $\hat{i} \geq i^*$, which is a contradiction.

(*Case 2*: $\hat{i} > i^*$) It follows from (21) that $\hat{x}_{i^*} = \underline{x}_{i^*}$. Thus, from (*iv*) of Lemma 1,

$$\alpha_{i^*}\underline{x}_{i^*} \geq \alpha_i \hat{x}_i \tag{23}$$

for $i = \hat{i}, \ldots, n$ so that

$$\begin{aligned} c &= \sum_{i=1}^{\hat{i}-1} \underline{x}_i + \sum_{i=\hat{i}}^{n} \alpha_i \frac{\hat{x}_i}{\alpha_i} \\ &\overset{(23)}{\leq} \sum_{i=1}^{i^*-1} \underline{x}_i + \sum_{i=i^*}^{\hat{i}-1} \alpha_i \frac{\underline{x}_i}{\alpha_i} + \sum_{i=\hat{i}}^{n} \frac{\alpha_i^* \underline{x}_{i^*}}{\alpha_i} \\ &\overset{(6)}{\leq} \sum_{i=1}^{i^*-1} \underline{x}_i + \alpha_{i^*}\underline{x}_{i^*} \sum_{i=i^*}^{\hat{i}-1} \frac{1}{\alpha_i} + \alpha_{i^*}\underline{x}_{i^*} \sum_{i=\hat{i}}^{n} \frac{1}{\alpha_i} \\ &\leq \sum_{i=1}^{i^*-1} \underline{x}_i + \alpha_{i^*}\underline{x}_{i^*} \sum_{i=i^*}^{n} \frac{1}{\alpha_i} \end{aligned}$$

which contradicts (7). This completes the proof. □

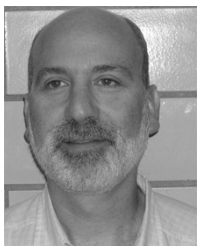
REFERENCES

[1] V. Jacobson, K. Nichols, and K. Poduri, "An expedited forwarding PHB," RFC 2598, Jun. 1999.

[2] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski, "Assured forwarding group," RFC 2597, Jun. 1999.

[3] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An architecture for differentiated services," RFC 2475, Dec. 1998.

[4] K. K. Ramakrishnan and S. Floyd, "A proposal to add explicit congestion notification (ECN) to IP," RFC 2481, Jan. 1999.
[5] S. Sahu, P. Nain, D. Towsley, C. Diot, and V. Firoiu, "On achievable service differentiation with token bucket marking for TCP," in *Proc. ACM SIGMETRICS*, Santa Clara, CA, Jun. 2000, pp. 23–33.
[6] I. Yeom and A. L. N. Reddy, "Modeling TCP behavior in a differentiated-services network," in *IEEE/ACM Trans. Netw.*, vol. 9, Feb. 2001, pp. 31–46.
[7] M. Goyal, A. Durresi, P. Misra, C. Liu, and R. Jain, "Effect of number of drop precedences in assured forwarding," in *Proc. GLOBECOM*, Dec. 1999, pp. 188–193.
[8] Y. Chait, C. V. Hollot, V. Misra, D. Towsley, H. Zhang, and C. S. Lui, "Providing throughput differentiation for TCP flows using adaptive two-color marking and two-level AQM," in *Proc. IEEE INFOCOM*, 2002, pp. 837–844.
[9] I. Yeom and A. L. N. Reddy, "Marking for QoS improvement," *J. Comput. Commun.*, pp. 35–50, Jan. 2001.
[10] V. Firoiu, J.-Y. Le Boudec, D. Towsley, and Z.-L. Zhang, "Theories and models for Internet quality of service," in *Proc. IEEE, Special Issue on Internet Technology*, vol. 90, Sep. 2002, pp. 1565–1591.
[11] S. Athuraliya, V. H. Li, and S. H. Low, "REM: Active queue management," *IEEE Network*, vol. 15, no. 3, pp. 48–53, May/Jun. 2001.
[12] S. Kunniyur and R. Srikant, "Analysis and design of an adaptive virtual aueue (AVQ) algorithm for active queue management," in *Proc. ACM SIGCOMM*, 2001.
[13] 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 *Proc. ACM SIGCOMM*, Aug. 2000.
[14] Y. Cui, Y. Chait, and C. V. Hollot, "Stability analysis of a diffserv network having two-level coloring at the network edge and preferential dropping at the core," presented at the Amerian Control Conf., Boston, MA, 2004.
[15] F. P. Kelly, A. K. Maulloo, and D. Tan, "Rate control for communication networks: Shadow prices, proportional fairness and stability," *J. Oper. Res. Soc.*, vol. 49, pp. 237–252, 1998.
[16] Y. Chait and C. V. Hollot. (2002) Fixed Point of a Diffserv Network with a Single Congested Link. [Online]. Available: http://www.ecs.umass.edu/mie/labs/dacs/index2.html#publications
[17] C. V. Hollot, V. Misra, D. Towsley, and W. B. Gong, "On designing improved controllers for AQM routers supporting TCP flows," in *Proc. IEEE INFOCOM*, Anchorage, AK, 2001, pp. 1726–1734.
[18] D. D. Clark and W. Fang, "Explicit allocation of best effort packet delivery service," in *IEEE/ACM Trans. Netw.*, vol. 6, Aug. 1998, pp. 362–373.
[19] E. O. Doebelin, *Control System Principles and Design*. New York: Wiley, 1985.



Yossi Chait received the B.S. degree in mechanical engineering from The Ohio State University, Columbus, in 1982, and the M.S. and Ph.D. degrees in mechanical engineering from Michigan State University, Lansing, in 1984 and 1988, respectively.

Currently he is a Professor in the Mechanical Engineering Department, University of Massachusetts, Amherst. He has been active in quantitative feedback theory teaching and research for the past 15 years. His recent research focuses on congestion control of the Internet and modeling of feedback mechanisms in biological systems such as the hypothalamus-pituitary-thyroid axis and circadian rhythms. He has consulted for industry in a broad range of applications, for example, in automatic welding, real-time particle analyzers, and vibrations reduction.

C. V. Hollot (S'79–M'82) received the Ph.D. degree in electrical engineering from the University of Rochester, Rochester, NY, in 1984.

Since 1984, he has been with the Department of Electrical and Computer Engineering at the University of Massachusetts, Amherst. He has served as Associate Editor for several control journals. His research interests are in the theory and application of feedback control.

Dr. Hollot received the National Science Foundation Presidential Young Investigator Award in 1988.

Vishal Misra (S'98–M'99) received the B.Tech. degree from the Indian Institute of Technology, Bombay, and the Ph.D. degree from the University of Massachusetts, Amherst, in 1992 and 2000, respectively.

Since 2001, he has been an Assistant Professor in the Computer Science Department of Columbia University, New York, where he also holds a joint appointment in the Electrical Engineering Department. His research interests include the modeling, analysis, and design of algorithms for communication networks.

Don Towsley (M'78–SM'93–F'95) received the B.A. degree in physics and the Ph.D. degree in computer science from the University of Texas, Austin, in 1971 and 1975, respectively.

From 1976 to 1985, he was a Member of the Faculty of the Department of Electrical and Computer Engineering, University of Massachusetts, Amherst, where he is currently a Distinguished Professor in the Department of Computer Science. He has held visiting positions at IBM T. J. Watson Research Center, Yorktown Heights, NY (1982–1983), Laboratoire MASI, Paris, France (1989–1990), INRIA, Sophia Antipolis, France (1996), and AT&T Labs–Research, Florham Park, NJ (1997). His research interests include networks, multimedia systems, and performance evaluation.

Dr. Towsley has served on the editorial boards of the IEEE TRANSACTIONS ON COMMUNICATIONS and the IEEE/ACM TRANSACTIONS ON NETWORKING, and is currently the Editor in Chief of the IEEE/ACM TRANSACTIONS ON NETWORKING. He currently serves on the editorial boards of *Performance Evaluation* and *Journal of the ACM*. He received the 1998 IEEE Communications Society William Bennet Paper Award and three Best Conference Paper Awards from ACM SIGMETRICS. He was a Program Co-Chair of the joint ACM SIGMETRICS and PERFORMANCE'92 Conference. He is a Member of ORSA and Chair of IFIP Working Group 7.3. He is a Fellow of the ACM.

Honggang Zhang received the B.Sc. degree from the Central South University, China, and the M.Sc. degree from Tianjin University, China. He also received a M.Sc. degree from Purdue University, West Lafayette, IN. In 2000, he joined the Ph.D. program in the Computer Science Department, University of Massachusetts, Amherst.

He is now working in the area of computer networks.

Yong Cui (S'98) received the B.E., M.E., and M.S. degrees, all in mechanical engineering, from the University of Science and Technology Beijing, Beijing, China, and Vanderbilt University, Nashville, TN, in 1990, 1993, and 2000, respectively. He is currently pursuing the Ph.D. degree at the University of Massachusetts, Amherst.

He worked as a Senior Controls Engineer at Corning Inc., Corning, NY, from 2001 to 2002. He was an instructor at the University of Science and Technology Beijing, Beijing, China, from 1993 to 1998. His research interests cover system dynamics and control, electromechanical systems design, smart sensing, and wireless sensor networks.

Mr. Cui is a member of Sigma Xi.