Proportional Bandwidth Allocation in DiffServ Networks

Eun-Chan Park and Chong-Ho Choi
School of Electrical Engineering and Computer Science, Seoul National University, Seoul, KOREA
Email: {ecpark|chchoi}@csl.snu.ac.kr

Abstract—By analyzing the steady state throughput of TCP flows in differentiated service (DiffServ) networks, we show that current DiffServ networks are biased in favor of those flows that have a smaller target rate, which results in unfair bandwidth allocation. In order to solve this unfairness problem, we propose an adaptive marking scheme, which allocates bandwidth in a manner which is proportional to the target rates of the aggregate TCP flows in the DiffServ network. This scheme adjusts the target rate according to the congestion level of the network, so that the aggregate flow can obtain its fair share of the bandwidth. Since it utilizes edge-to-edge feedback information without measuring or keeping any per-flow state, this scheme is scalable and does not require any additional signaling protocol or any significant changes to the current TCP/IP protocol. It can be implemented in a distributed manner using only two-bit feedback information, which is carried in the TCP acknowledgement. Using extensive simulations, we show that the proposed scheme can provide each aggregate flow with its fair share of the bandwidth, which is proportional to the target rate, under various network conditions.

Index Terms—Proportional bandwidth allocation, fairness, Quality of Service, DiffServ networks, scalability

I. INTRODUCTION

Differentiated service (DiffServ) architecture has been proposed in order to provide different levels of service to satisfy different service requirements in a scalable manner [1]. In DiffServ architecture, IP flows are classified and aggregated into different forwarding classes, marked with different levels of priority at the edges of a network and dropped with different dropping mechanisms at the core of a network. Therefore, DiffServ networks can provide Quality-of-Service (QoS) beyond the current best-effort service. In DiffServ networks, a customer makes a contract with the service provider for the establishment of a service profile, called the Service Level Agreement (SLA). The service profile specifies the minimum throughput (also called the committed information rate (CIR) or target rate) that should be provided to the customer, even in the case of congestion. In order to assure the conditions specified in the SLA, the necessary components are the packet marking mechanism administrated by profile meters or traffic conditioners at the edge routers and the queue management mechanism operated at the core routers. The packet marking mechanism monitors and marks packets according to the profile at the edge of the network. If the measured flow conforms to the service profile, the packets belonging to this flow are marked with high priority (e.g., marked as IN) and receive assured service. Otherwise, the packets belonging to the non-conformant part of a flow are marked with low priority (e.g., marked as OUT) and receive best effort service. The queue management mechanism, deployed at core routers, gives preferential treatment to high priority packets. During times of congestion, high priority packets are forwarded preferentially and low priority packets are dropped with a higher probability. The most prevalent profile meters are the Token Bucket (TB) marker and the Time Sliding Window (TSW) marker, and the most widely deployed queue management algorithm is RED with In/Out (RIO) [2], [3], [4].

Also, many mechanisms have been proposed to provide assured service [5]–[8], and there has been some recent research done on modelling TCP behavior in DiffServ networks [9], [10]. The previous studies performed in this area were focused on simply assuring the target rate. However, this assurance is not sufficient to satisfy the customer. Considering the fact that the target rate is determined by the terms of the SLA, and that the customer’s fee is calculated accordingly, the bandwidth should be allocated in proportion to the target rate, which we refer to as “proportional bandwidth allocation”. Note that the notion of proportional allocation of bandwidth is different from that of proportional fairness [11], [12]. When the target rates of aggregate flows are different, the assurance of relative throughput, as well as the assurance of minimum throughput, must both be considered. When the network is over-provisioned, the surplus bandwidth should be allocated to the aggregates in proportion to the target rates. When the network is over-subscribed, the service rates should also be allocated in proportion to the target rates, even if it is impossible to assure them completely. However, the existing mechanisms [5]–[8] do not offer any guarantees when it comes to dealing with surplus bandwidth or bandwidth deficit.

Studies based on simulations [13], [14] have shown that assuring the throughput in DiffServ networks depends on several factors, such as the round-trip time (RTT), the target rate, and the existence of non-responsive flows. In order to reduce the effects of RTT and target rate on throughput, a few mechanisms have been proposed [6], [14], [15]. The main idea behind these mechanisms is that packets belonging to flows which send packets more aggressively should be preferentially dropped. However, the mechanism in [6] requires that a per-flow state should be conveyed and maintained at the routers, which causes a scalability problem. The algorithm in [14] needs to measure the RTT and requires an additional signaling protocol for the purpose of communicating between the edge routers. Similarly, the algorithm in [15] also needs to estimate...
the RTT and packet loss rate, resulting in heavy computational overhead, which should be avoided in high-speed networks. In [16], to allocate bandwidth proportionally, every router assigns tickets, which represent a relative share of the bandwidth, and these tickets are reassigned at each hop based on the contractual agreements. Hence, this scheme requires that all of the routers should be involved in proportional bandwidth allocation.

Fair bandwidth allocation without per-flow state in the core routers was addressed in [17]–[19]. By keeping per-flow state in edge routers and carrying that information in packets to core routers, CSFQ [17] achieves max-min fairness in bandwidth allocation approximately while keeping the core routers stateless. Rainbow Fair Queuing [18] that avoids fair share rate calculation in the core routers reduces computational overhead in achieving the max-min fairness. Recently, SCALE-WFS [19] has been proposed, in the context of DiffServ network. It aims to achieve weighted fair bandwidth sharing, which is similar to the notion of proportional bandwidth allocation in this paper. SCALE-WFS calculates the fair rate in the core routers using per-aggregate state instead of per-flow state, and it requires a labelling mechanism to carry per-aggregate information in packets, which represents for the fair share rate.

The objective of this study is to propose a new marking scheme, whose role is to allocate bandwidth fairly among aggregate flows in a distributed manner without requiring any complex signaling protocol or any labelling mechanism. This study is an extended version of [20], which is based on the observation of simulation results. This paper makes the following contributions.

(i) By analyzing the steady state throughput of aggregate TCP flows in DiffServ networks, we reveal the unfairness problem in bandwidth sharing when aggregate flows with different target rates share a common bottleneck link.

(ii) We propose an adaptive marking scheme to solve this unfairness problem and achieve proportional bandwidth allocation. This scheme adapts the target rate in a completely distributed manner according to the congestion level of the network, so that the target rate matches to is fair share of the bandwidth. The proposed scheme does not require core routers to calculate the fair rates and to maintain any per-flow or per-aggregate states, because it utilizes only two-bit edge-to-edge feedback information using TCP acknowledgement (ACK). Hence, it is highly scalable and does not require the use of any additional signaling protocol or labelling mechanism.

(iii) Using simulations, we confirm that the proposed scheme allocates the bandwidth to aggregate flows in proportion to their target rates.

The rest of the paper is organized as follows. In Section II, we analyze the unfairness problem by considering the steady state behavior of TCP. Based on the results of this analysis, we propose an adaptive marking scheme in Section III. We also show that the proposed marking scheme achieves proportional bandwidth allocation and discuss issues such as its implementation, scalability. Section IV presents the ns-2 simulation results under various network conditions to show the effectiveness of the proposed scheme. The conclusions follow in Section V.

II. ANALYSIS OF THE UNFAIRNESS PROBLEM

It has been shown through simulation that the profile meters which are currently in use are biased toward those aggregates that have a smaller target rate [13], [20]. An aggregate with a smaller target rate occupies more bandwidth than its fair share, while an aggregate with a larger target rate gets less than its fair share. By means of analysis, we show that this phenomenon is indeed true. First, we present a graphical analysis based on our intuition regarding the steady state behavior of TCP flows, this behavior being dominated by the Additive Increase Multiplicative Decrease (AIMD) algorithm [21] adopted in TCP congestion control. This analysis gives an insight into the problem of unfair bandwidth sharing. Then, we reinvestigate the unfairness problem by means of a mathematical analysis, from which we derive the conditions required to judge which aggregates get more/less than their fair share. Later, we confirm the validity of the analysis using simulation, which provides a clue to solving the unfairness problem.

Consider a case wherein aggregates with different target rates share a common bottleneck link whose capacity is $C$ [packets/sec]. Let us define the target rate of the $i$th aggregate as $R_{t,i}$ [packets/sec]. A network is under-subscribed or over-provisioned if $\sum R_{t,i} < C$, and is over-subscribed or under-provisioned if $\sum R_{t,i} > C$. Let us define the fair share of the $i$th aggregate, $R_{f,i}$, that achieves proportional sharing of bandwidth as:

$$R_{f,i} = R_{t,i} + (C - \sum_j R_{t,j}) \frac{R_{t,i}}{\sum_j R_{t,j}} = \frac{R_{t,i}}{C}. \tag{1}$$

It is important to note that the fair share, $R_{f,i}$ in (1), is dependent on the bottleneck link capacity and on the target rates of the other aggregates that share the bottleneck link. Therefore, in order to achieve fair allocation of bandwidth, the routers need to keep track of global information on link capacity and on the target rates of all aggregates. Our approach to assuring the fair allocation of bandwidth, which will be presented more fully in the next section, is feedback-based. It does not require keeping track of global information and can be implemented and performed in a distributed manner.

In order to compare the actual throughput of the $i$th aggregate $R_{t,i}$, with its fair share $R_{f,i}$, we define the relative gain of the $i$th aggregate, $G_i = R_{t,i}/R_{f,i}$.

A. Graphical analysis

For simplicity, let us consider a case wherein there are two aggregates which have the same characteristic and share a common bottleneck link. Let us assume that the initial sending rate is zero. The sending rate of aggregate flows is adjusted by the TCP congestion control mechanism. Without loss of generality, we also assume that $R_{t,1} < R_{t,2}$. Our goal here is to demonstrate that $G_1 > 1$ and $G_2 < 1$. Figure 1 illustrates the relationship between the target rate and the actual rate under DiffServ networks.

1) Under-subscription case ($R_{t,1} + R_{t,2} < C$): At first, the two aggregates increase their sending rates up to their target rates. Once these target rates are attained, the aggregates probe the surplus bandwidth, i.e., $C - (R_{t,1} + R_{t,2})$, and compete to occupy the surplus bandwidth by sending OUT packets.
These \textit{OUT} packets follow the TCP congestion avoidance mechanism [22], which is characterized by the AIMD algorithm. Because the AIMD algorithm tends to distribute the available bandwidth evenly to those aggregates participating in the competition [21], the surplus bandwidth is apportioned out \textit{evenly rather than proportionally}, as shown in Fig. 1(a). Consequently, \( R_1 \) becomes bigger than \( R_{f,1} \) while \( R_2 \) becomes smaller than \( R_{f,2} \), i.e., \( G_1 > 1 \) and \( G_2 < 1 \).

2) \textit{Over-subscription case} (\( R_{t,1} + R_{t,2} > C \)): We need to divide this case into two subcases, i.e., Case 1 and Case 2. In Case 1, neither of the two target rates is achievable (0.5\( C < R_{t,1} < R_{t,2} \)). In Case 2, one target rate is achievable, while the other is not (\( R_{t,1} \leq 0.5C < R_{t,2} \)).

Let us first consider Case 1. Since the sum of the two target rates exceeds the bottleneck link capacity and almost all of the packets sent by the two agrees are marked as \( IN \), the proper service differentiation, which is based on the target rate, cannot be realized. Therefore, the target rate does not affect the achievable rate, which is determined by the bottleneck link capacity and the TCP congestion control mechanism. Consequently, the throughputs of the two aggregates become equal, i.e., one half of the bottleneck link capacity. It is obvious that \( R_1 > R_{f,1} \) and \( R_2 < R_{f,2} \) from Fig. 1(b).

Now, let us consider Case 2 wherein \( R_{t,1} \leq 0.5C < R_{t,2} \). The two aggregates start to increase their sending rates until they reach \( R_{t,1} (< R_{t,2}) \). After \( R_{t,1} \) is achieved, the packets belonging to the first aggregate are marked as \( OUT \) while the packets belonging to the second aggregate are still marked as \( IN \), because the sending rate of the second aggregate is smaller than \( R_{t,2} \). Due to the preferential dropping in the core router, the first aggregate gets no extra bandwidth and the remaining bandwidth (\( C - 2R_{t,1} \)) is occupied by the second aggregate. In this case, we can see that \( G_1 > 1 \) and \( G_2 < 1 \) from Fig. 1(b).

\section*{B. Mathematical analysis}

Here, we extend the analysis to the general case of \( N \) aggregates, and derive the conditions for an aggregate to get more or less bandwidth than its fair share. Let us assume that the \( i \)th aggregate flow consists of \( N_i \) identical TCP flows and that the number of flows within each aggregate is the same. We use the token bucket algorithm [10] and the non-overlapping RIO algorithm [4] as the profile meter and queue management mechanism, respectively. We set the token bucket size for the \( i \)th aggregate, \( B_i \) [packets], to be equal to the product of the average RTT value of the flows (\( T \) [sec]) and the target rate (\( R_{t,i} \) [packets/sec]), i.e., \( B_i = T \cdot R_{t,i} \).

1) \textit{Under-subscription case} (\( \sum_i R_{t,i} < C \)): In [10], the achievable throughput of TCP flows in DiffServ networks is analyzed based on the steady state behavior of TCP flows. Ignoring the packet loss due to the time-out mechanism of TCP and setting \( B_i = T \cdot R_{t,i} \), the steady state throughput of the \( i \)th aggregate is

\[
R_i = \frac{1}{2} \left[ R_{t,i} + \sqrt{R_{t,i}^2 + \alpha} \right],
\]

where \( \alpha = 6N^2/(p_{out} T^2) \) and \( p_{out} \) is the loss rate of the \textit{OUT} packets [10].

\textbf{Proposition 1 (under-subscription case):} In under-subscribed DiffServ networks with the token bucket profile meter and the RIO algorithm, let us consider the case where \( N \) aggregates with different target rates share a common bottleneck link. If the target rate of the \( i \)th aggregate, \( R_{t,i} \), satisfies the condition described in (3), then the \( i \)th aggregate occupies less bandwidth than its fair share, i.e.,

\[
R_{t,i} > \frac{1}{\sqrt{N-1}} \left( \sum_{j=1,j \neq i}^N R_{t,j} \right) \quad \Rightarrow \quad G_i < 1. \tag{3}
\]

\textbf{Proof:} We assume that the utilization of the bottleneck link is equal to its capacity in the steady state, i.e., \( \sum_i R_i = C \). Let \( \Delta R_i = R_i - R_{f,i} \). From (2), \( \Delta R_i \) is

\[
\Delta R_i = \frac{1}{2} \sum_{j=1}^N R_{t,j} \left[ \left( \sum_{j=1,j \neq i}^N R_{t,j} \right) \sqrt{R_{t,i}^2 + \alpha} \right]

- R_{t,i} \left( \sum_{j=1,j \neq i}^N \sqrt{R_{t,j}^2 + \alpha} \right).
\]

If the following inequality holds, then \( \Delta R_i < 0 \), i.e., \( G_i < 1 \);

\[
\left( \sum_{j=1,j \neq i}^N R_{t,j} \right) \sqrt{R_{t,i}^2 + \alpha} < R_{t,i} \left( \sum_{j=1,j \neq i}^N \sqrt{R_{t,j}^2 + \alpha} \right) \tag{4}
\]
By squaring both sides of (4), this equation becomes
\[
R_{t,i}^2 \left( \sum_{(l,m) \in S} R_{t,l} R_{t,m} \right) + \alpha \left( \sum_{j=1}^{N} R_{t,j} \right)^2 < R_{t,i}^2 \left( \sqrt{(R_{t,i}^2 + \alpha)(R_{t,i}^2 + \alpha)} \right) + \alpha \left( (N-1) R_{t,i}^2 \right),
\]
where \( S = \{(l,m) | l,m = 1,2,\ldots,i-1,i+1,\ldots,N, \text{ and } l \neq m \} \). Hence, if the 4th target rate satisfies the condition in (3), (4) is satisfied and \( G_i < 1 \).  

When \( N = 2 \) in (3), proposition 1 confirms that an aggregate which has a smaller/larger target rate occupies more/less bandwidth than its fair share.

2) Over-subscription case (\( \sum_i R_{t,i} > C \)): Similarly to the under-subscription case, we can obtain the steady state throughput of TCP flows in the over-subscribed DiffServ network as \( R_i = \min(R_{t,i}, \beta) \), where \( \beta = N_i \sqrt{3/(2p_{im})} / T \) and \( p_{im} \) is the loss rate of the \( IN \) packets \([10]\). Note that some of the aggregates can achieve their target rates (i.e., \( R_i = R_{t,i} < \beta \)), while the others cannot (i.e., \( R_i = \beta < R_{t,i} \)). If we assume that \( R_{t,1} < R_{t,2} < \cdots < R_{t,N} \) without loss of generality, we can consider the following two possible cases, i.e.,

**Case 1:** \( R_i = \beta \), for \( i = 1,2,\ldots,N \), 

**Case 2:** \( R_i = \left\{ \begin{array}{ll} R_{t,i}, & \text{for } i = 1,2,\ldots,k, \\ \beta, & \text{for } i = k+1,1,\ldots,N. \end{array} \right. \)  

These two cases are analogous to the two subcases in the over-subscription case examined in the previous subsection when \( N = 2 \).

**Proposition 2.1 (over-subscription Case 1):** Let us consider the case wherein there are \( N \) aggregates competing for the common bottleneck link in an over-subscribed DiffServ network and none of the target rates are achievable. The 4th aggregate occupies more/less bandwidth than its fair share if and only if its target rate is smaller/larger than the average target rate of the other \( N-1 \) aggregates, i.e.,
\[
\begin{align*}
R_{t,i} &< \frac{1}{N-1} \left( \sum_{j=1,j \neq i}^{N} R_{t,j} \right) \quad \iff \quad G_i > 1, \\
R_{t,i} &> \frac{1}{N-1} \left( \sum_{j=1,j \neq i}^{N} R_{t,j} \right) \quad \iff \quad G_i < 1.
\end{align*}
\]

**Proof:** From (5) and the assumption of full-utilization of the bottleneck link, i.e., \( \sum_i R_i = C \), \( \Delta R_i = R_i - R_{f,i} \) is
\[
\Delta R_i = C \left( 1 - \frac{R_{t,i}}{\sum_{j=1}^{N} R_{t,j}} \right).
\]

Hence, (6) holds from (7).

**Proposition 2.2 (over-subscription Case 2):** Let us assume that there are \( N \) aggregates in an over-subscribed DiffServ network and that \( k < N \) aggregates can achieve their target rates while the others cannot. The \( i \leq k \)th aggregate occupies more bandwidth than its fair share if \( R_{t,i} < C/N \), i.e.,
\[
R_{t,i} < C/N \quad \rightarrow \quad G_i > 1, \quad (i \leq k).
\]

Furthermore, for the other \( N-k \) aggregates, the \( i > k \)th aggregate occupies less than its fair share if \( R_{t,i} > C/N \), i.e.,
\[
R_{t,i} > C/N \quad \rightarrow \quad G_i < 1, \quad (i < k).
\]

In Table I, we compare the results of the analysis and the ns-2 simulation for several sets of \( R_{t,1} \) and \( R_{t,2} \) when \( C = 10 \) Mb/s. The first and last three sets of \( R_{t,1} \) and \( R_{t,2} \) correspond to the under-subscription case and the over-subscription case, respectively.

<table>
<thead>
<tr>
<th>( R_{t,1} )</th>
<th>( R_{t,2} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.2</td>
<td>4.65 / 4.50</td>
</tr>
<tr>
<td>1.5</td>
<td>3.49 / 3.57</td>
</tr>
<tr>
<td>1.8</td>
<td>1.72 / 1.82</td>
</tr>
<tr>
<td>2.8</td>
<td>2.26 / 2.00</td>
</tr>
<tr>
<td>3.7</td>
<td>3.19 / 3.00</td>
</tr>
<tr>
<td>5.5</td>
<td>5.04 / 5.00</td>
</tr>
<tr>
<td>10</td>
<td>5.14 / 5.00</td>
</tr>
</tbody>
</table>

**TABLE I: THROUGHPUTS AND RELATIVE GAINS OBTAINED FROM ANALYSIS AND SIMULATION**

<table>
<thead>
<tr>
<th>( R_{t,1} )</th>
<th>( R_{t,2} )</th>
<th>( R_1 )</th>
<th>( R_2 )</th>
<th>( C_1 )</th>
<th>( C_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.2</td>
<td>4.65 / 4.50</td>
<td>5.06 / 5.50</td>
<td>1.40 / 1.36</td>
<td>0.76 / 0.83</td>
<td></td>
</tr>
<tr>
<td>1.5</td>
<td>3.49 / 3.57</td>
<td>6.50 / 6.43</td>
<td>2.09 / 2.14</td>
<td>0.78 / 0.77</td>
<td></td>
</tr>
<tr>
<td>1.8</td>
<td>1.72 / 1.82</td>
<td>8.26 / 8.18</td>
<td>1.55 / 1.64</td>
<td>0.93 / 0.92</td>
<td></td>
</tr>
<tr>
<td>2.8</td>
<td>2.26 / 2.00</td>
<td>7.73 / 8.00</td>
<td>1.13 / 1.00</td>
<td>0.97 / 1.00</td>
<td></td>
</tr>
<tr>
<td>3.7</td>
<td>3.19 / 3.00</td>
<td>8.60 / 7.00</td>
<td>1.07 / 1.00</td>
<td>0.97 / 1.00</td>
<td></td>
</tr>
<tr>
<td>5.5</td>
<td>5.04 / 5.00</td>
<td>4.96 / 5.00</td>
<td>1.01 / 1.00</td>
<td>0.99 / 1.00</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>5.14 / 5.00</td>
<td>6.84 / 7.00</td>
<td>1.36 / 1.30</td>
<td>0.89 / 0.91</td>
<td></td>
</tr>
</tbody>
</table>

Note that when \( N = 2 \), the aggregate that has the smaller/larger target rate always occupies more/less bandwidth than its fair share regardless of the subscription level, as was already shown by means of the graphical analysis in the previous subsection.

**C. Validity of the analysis**

In order to show the validity of the analysis, we performed an ns-2 simulation and compared the simulation results with the analysis results. Figure 4 in Section IV shows the network configuration used for the simulation, which is simple but sufficient to reveal the unfairness problem. Further details about the simulation configuration are provided in Section IV.

Note that when \( N = 2 \), the aggregate that has the smaller/larger target rate always occupies more/less bandwidth than its fair share regardless of the subscription level, as was already shown by means of the graphical analysis in the previous subsection.

In Table I, we compare the results of the analysis and the ns-2 simulation for several sets of \( R_{t,1} \) and \( R_{t,2} \) when \( C = 10 \) Mb/s. The first and last three sets of \( R_{t,1} \) and \( R_{t,2} \) correspond to the under-subscription case and the over-subscription case, respectively.
TABLE II
RELATIVE GAINS OBTAINED FROM SIMULATION AND ANALYSIS

<table>
<thead>
<tr>
<th>$R_{t,1}$</th>
<th>$R_{t,2}$</th>
<th>$R_{t,3}$</th>
<th>Simulation</th>
<th>Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>4</td>
<td>1.19 1.03 0.94</td>
<td>- - &lt; 1</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>4</td>
<td>1.22 0.97 0.97</td>
<td>- - &lt; 1</td>
</tr>
<tr>
<td>3.5</td>
<td>4</td>
<td>5</td>
<td>1.10 1.01 0.88</td>
<td>&gt; 1 &gt; 1 &lt; 1</td>
</tr>
<tr>
<td>3.5</td>
<td>4.5</td>
<td>5</td>
<td>1.12 0.95 0.91</td>
<td>&gt; 1 &lt; 1 &lt; 1</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>6</td>
<td>1.27 1.14 0.83</td>
<td>&gt; 1 &gt; 1 &lt; 1</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>6</td>
<td>1.28 1.19 0.75</td>
<td>&gt; 1 - &lt; 1</td>
</tr>
</tbody>
</table>

respectively, and the other three sets correspond to the exact-subscription case. In all cases, there is not much difference between the analysis results and the ns-2 simulation results, which confirms the validity of the analysis.

Remark1: For both the under-subscription case and the over-subscription case, as the difference between the total target rate and the link capacity, i.e., $\sum_i R_{t,i} - C$, decreases, the unfairness between the aggregates also decreases and the throughput of each aggregate approaches its fair share.

Remark2: When the network is exactly-provisioned, i.e., $R_{t,1} + R_{t,2} = C$, $R_1$ and $R_2$ are close to their target rates and the relative gains $G_1$ and $G_2$ are nearly equal to one.

Next, we compare the relative gains obtained from the simulation and those predicted in the analysis. When the conditions in the propositions are satisfied, relative gains can be predicted whether they are bigger or smaller than one. Table II lists the set of target rates and the relative gains when $N=3$ and $C=10$Mb/s. The first two rows in Table II correspond to the under-subscription case and the next two and the last two rows correspond to over-subscription Case1 and over-subscription Case2, respectively. As shown in Table II, the results predicted by the analysis match the simulation results well.

III. ADAPTIVE MARKING SCHEME

A. Design rationale

The remarks in Section III provide a clue to solving the problem of unfair bandwidth sharing; they show that if a network is exactly-provisioned, there is no bias in favor of an aggregate that has a smaller target rate. Taking this as the starting point of our proposition, we can infer that the unfairness problem can be solved by making the networks exactly-provisioned. We adjust the target rates, so that the sum of the adjusted target rates $R_{t,i}[n]$ at the $n$th update matches the bottleneck link capacity, while keeping the ratio of their original values fixed, i.e.,

$$R_{t,i}[n] = (1 \pm \delta) R_{t,i}[n-1] \text{ such that } \sum_i R_{t,i}[n] = C.$$  

Here, $\delta$ ($>0$) is an adjustment factor. If a network is under-subscribed/over-subscribed, we increase/decrease the target rates multiplicatively.

In order to accomplish proportional bandwidth allocation, we need to know whether the network is under-subscribed or over-subscribed, so that we can adjust the target rates accordingly. We look for a solution to this problem that is consistent with the philosophy of DiffServ, i.e., “moving complexity to the edges of the network” [1]. The solution should not require any per-flow state at the core routers, for the sake of scalability, or any critical changes either in the edge routers or the current transport-layer protocol, for the sake of compatibility. A solution that allocates the bandwidth proportionally to the target rates should have the following two properties.

- The target rates should be adjusted multiplicatively, so that their sum matches the bottleneck link capacity.
- The adjustment of the target rates should be performed at the edge routers in a distributed manner, without requiring any complex signaling protocol or per-flow state.

B. Architecture and algorithm

The preferential dropping taking place at the core routers provides a good indication of the state of congestion. If the network is far from being congested, the IN packets will rarely be dropped and the dropping probability for the IN packets, $p_{in}$, will be insignificant. If the network is heavily congested, almost all of the OUT packets will be dropped. Also, a certain proportion of the IN packets will be dropped and $p_{in}$ will not be negligible. Thus, by observing $p_{in}$, the edge router can infer the state of congestion at the core of the network and determine whether it should increase or decrease the target rate.

We propose an adaptive marking scheme that utilizes edge-to-edge feedback information. The egress edge router is in charge of estimating $p_{in}$, and based on the estimated value of $p_{in}$ it generates the feedback information required to adjust the target rate. Then, this feedback information can be carried in a two-bit flag in a packet header via TCP receivers and TCP senders, and finally it is utilized at the ingress edge router when adjusting the target rate. The feedback architecture of the adaptive marking scheme is shown in Fig. 2, and the role of each element is explained in the following paragraphs.

1) Core router: For the preferential dropping mechanism, we adopt the RIO active queue management algorithm [4]. Note that we do not make any changes in the core routers and that the core routers do not maintain any per-flow state. As shown in Fig. 3, the dropping probability of the OUT packets, $p_{out}$, is calculated using $q_{out}$, which is an exponentially weighted moving average (EWMA) of the queue length, consisting of both IN packets and OUT packets. The dropping probability of the IN packets, $p_{in}$, is computed in a similar manner using the parameters $q_{in}$, $q_{in}^{min}$, $q_{in}^{max}$, and $p_{in}^{max}$, as shown in Fig. 3. Here, $q_{in}$ is calculated by counting only the IN packets in the queue. By setting $q_{out}^{max}$, we can guarantee that the IN packets start to be dropped only after all of the OUT packets have been dropped. Using this preferential dropping mechanism, $p_{in}$ can be used to check whether there is any extra bandwidth available and whether the network is over-subscribed. A negligible value of $p_{in}$ means that there is surplus bandwidth available, while a value of $p_{in}$ close to $p_{in}^{max}$ implies that the network is over-subscribed.

2) Egress edge routers: The egress edge routers estimate $p_{in}$ and generate feedback information which is used to adjust the target rate. Here, we assume that the network supports

\[p_{in} \approx q_{out}^{max},\]  

in another version of RIO, referred to as Decoupled-RIO, $q_{out}$ is computed only for OUT packets. In our study, we adopt Coupled-RIO, where $q_{out}$ is computed for both IN and OUT packets.
the Explicit Congestion Notification (ECN) mechanism [23], which has been proposed as a solution for signaling congestion rapidly and explicitly to TCP senders. Because the ECN mechanism marks packets instead of dropping them as a means of signaling congestion, we can make use of this congestion signaling information to estimate \( p_{\text{in}} \) at the egress edge router. Let us denote \( \hat{p}_{\text{in}} \) and \( \hat{\hat{p}}_{\text{in}} \) as the moving average and the estimate of \( p_{\text{in}} \), respectively. First, we calculate \( \hat{\hat{p}}_{\text{in}} \) as the fraction of ECN-marked packets in the recently arrived \( N_{w} \) IN packets. Next, we obtain \( \hat{p}_{\text{in}} \) as the weighted average of \( \hat{\hat{p}}_{\text{in}} \), in order to reduce the bursty nature of TCP, i.e.,

\[
\hat{p}_{\text{in}} = (1 - w)\hat{\hat{p}}_{\text{in}} + wp_{\text{in}}.
\]

Note that the two parameters, the window size \( N_{w} \) and the weight \( w \), are related to the responsiveness of the estimation algorithm. A large value of \( N_{w} \) or a small value of \( w \) results in a slow and smooth response to changes in \( p_{\text{in}} \). On the other hand, a small value of \( N_{w} \) and a large value of \( w \) results in a fast response, however, possibly leading to fluctuation in estimating \( p_{\text{in}} \) due to the burstiness of TCP.

Using \( \hat{\hat{p}}_{\text{in}} \), the egress edge routers generate feedback information which is used to adjust the target rate. If \( \hat{\hat{p}}_{\text{in}} \) is smaller than a given threshold value, \( p_{\text{th}}^{\min} \), which is close to zero, then the edge router sets the ITR (Increase Target Rate) bit in the packet’s header, i.e.,

\[
\text{if } (\hat{\hat{p}}_{\text{in}} < p_{\text{th}}^{\min}) \rightarrow \text{Set ITR bit,}
\]

\[
\text{else if } (\hat{\hat{p}}_{\text{in}} > p_{\text{th}}^{\max}) \rightarrow \text{Set DTR bit.}
\]

3) TCP receivers and TCP senders: Ideally, the feedback information should be conveyed from the egress edge router, where the information is generated, to the ingress edge router, where it is utilized. However, it is impossible to communicate information directly between these edge routers without the aid of an additional signaling protocol, because current IP networks do not have any signaling architecture for this feedback information. Such direct communication would cause extra traffic and overhead, which are both redundant and undesirable in high-speed networks. Hence, we have to find another way to convey the information to the ingress edge router. The TCP ACK packet can serve as a good messenger for this purpose. When TCP receivers receive a packet whose ITR or DTR bit is set, they simply extract these flags from the IP header and copy them into the unused field in the TCP header to be fed back to the TCP senders. Similarly, the TCP senders convey the information to the ingress edge router. Note that this mechanism of conveying feedback information is similar to the ECN mechanism [23].

4) Ingress edge routers: The ingress edge routers are in charge of adjusting the target rates. When the feedback information is conveyed in packet headers, the rate at which the information is transported to each ingress router is not identical. As the sender transmits packets faster, the ingress edge router receives this information and updates its target rate more frequently. In order to avoid this potential imbalance in the update rates among the ingress edge routers, we introduce a timer whose interval is \( T_{s} \). When the timer expires, the target rate is updated. The timer resides in each ingress router, and does not need to be synchronized. We introduce a variable \( n_{\text{ATR}} \) that is used to determine whether to increase or decrease the target rate. It is initialized at the expiration of the timer and is increased/decreased by one upon the receipt of a packet whose ITR/DTR bit is set. At each expiration of the timer, if \( n_{\text{ATR}} \) is positive/negative then the target rates are increased/decreased multiplicatively by \( (1 \pm \delta) \) i.e.,

\[
\text{if } (n_{\text{ATR}} > 0) \rightarrow R_{t,i}(nT_{s}) = (1 + \delta)R_{t,i}((n - 1)T_{s}),
\]

\[
\text{else if } (n_{\text{ATR}} < 0) \rightarrow R_{t,i}(nT_{s}) = (1 - \delta)R_{t,i}((n - 1)T_{s}).
\]

There is a trade-off when setting the values of \( T_{s} \) and \( \delta \). If \( T_{s} \) is too small or \( \delta \) is too big, the target rate will fluctuate and will not converge toward the level which corresponds to a fair allocation of the bandwidth. In the opposite case, the response to changes in the network will be slow.
C. Proportional bandwidth allocation

We have proposed the adaptive marking scheme based on the rationale that the target rates are adjusted multiplicatively so that their sum matches the bottleneck link capacity. In this subsection, we show that the proposed scheme achieves the proportional allocation of bandwidth.

For the sake of simplicity, we assume that (i) all of the flows have the same constant RTT, $T$ [sec], (ii) flows belonging to different aggregates can traverse different paths, (iii) the senders always have data to send, (iv) the buffer size is infinite. Let $l_i^b$ and $C_i^b$ denote the bottleneck link of the $i$th aggregate and its capacity, respectively. We define $L_i$ and $S_i^b$ as the set of links that the $i$th aggregate traverses in a DiffServ network and the set of aggregates that traverse the bottleneck link $l_i^b$, respectively. Also, we define $\hat{p}_{in} = 1 - \prod_{l_k \in L_i} (1 - p_{in,l_k})$, where $p_{in,l_k}$ is the dropping probability of $IN$ packets at the link $l_k$. We adopt the fluid-based TCP dynamic model [24], where the slow-start and time-out mechanisms of TCP are ignored. The DiffServ networks with the proposed marking scheme are controlled by the following three dynamics, i.e., TCP dynamics, target rate dynamics, and queue dynamics:

\[
\hat{R}_{i,j}(t) = \frac{1}{T^2} - \frac{1}{a} R_{i,j}(t) R_{i,j}(t - T) \hat{p}_{in}(t - T),
\]

\[
\hat{R}_{i,j}(t) = -\delta \hat{R}_{i,j}(t - T) \left[ -u(p^{th}_{min} - \hat{p}_{in}(t - T_s)) + u(\hat{p}_{in}(t - T_s) - p^{th}_{max}) \right],
\]

\[
d_i^b(t) = -C_i^b + \sum_{j \in S_i^b} \sum_{k=1}^{N_j} R_{j,k}(t).
\]

Here, $R_{i,j}$ and $\hat{p}_{in}$ are the sending rate and loss rate of the $j(\leq N_j)$th TCP flow belonging to the $i$th aggregate, respectively. We define $u(p)$ in (13) to be 1 if $p > 0$, and 0 otherwise, and $d_i^b$ as the queue length at the router which sends packets through the link $l_i^b$. For ECN-capable networks with infinite-size buffers, the sending rate of a TCP flow is equal to its throughput. We set the scaling constant $a$ in (12) to $3/2$ so that the steady state throughput becomes consistent with the results in [25], i.e., $\sqrt{3}/2 \{\sqrt{p_{i,j}} T\}$ where $p_{i,j}$ is the steady state value of $\hat{p}_{in}$.

Because we set the update interval for adjusting each target rate to the same value, $T_s$, the ratio of the target rates is maintained, i.e.,

\[
\frac{R_{i,j}(t)}{R_{i,j}(t - T)} = \frac{R_{i,j}^0}{R_{i,j}^0} \quad \forall i, j \text{ and } t \geq 0,
\]

where, $R_{i,j}^0$ is the initial value of $R_{i,j}(t)$ at $t=0$. By summing up both sides of (15) with respect to $j(\in S_i^b)$, we can see that the portion of the $i$th target rate among the total target rates is kept fixed at the steady state, i.e.,

\[
\frac{R_{i,j}^0}{\sum_{j \in S_i^b} R_{i,j}^0} = \frac{R_{i,j}^0}{\sum_{j \in S_i^b} R_{i,j}^0},
\]

where, $R_{i,j}^0$ is the steady state value of $R_{i,j}$. Hence, the proposed marking scheme attempts to allocate bandwidth in proportion to the target rates.

**Proposition 3:** Let us assume that the size of the steady state target window for each TCP flow is sufficiently large, i.e., $R_{i,j}^0 T/N_i > 1$. If the adaptive marking scheme is used in a DiffServ network with the non-overlapping RIO algorithm, it allocates bandwidth in proportion to the target rates. Hence, the steady state throughput of the $i$th aggregate, $R_i^\star$, converges to its fair share, which is proportional to the initial target rates, i.e.,

\[
R_i^\star = \left( \frac{R_{i,j}^0}{\sum_{j \in S_i^b} R_{i,j}^0} \right) C_i^b.
\]

Proof: The details of the proof are given in Appendix A.

D. Implementation and scalability

The feedback information is generated at the egress edge router by marking the ITR or DTR bit in the IP header of a packet. When assigning these bits, we can make use of the currently unused two bits in the IPv4 Type-Of-Service (TOS) field or IPv6 Traffic Class (TC) field. Also, there is an unused field of 6 bits in the current TCP header, the ITR or DTR bit can be copied into this unused field. Hence, the proposed scheme can be incorporated into the current TCP/IP protocol with this minor modification in the protocol stack. Note that the proposed scheme utilizes the ECN mechanism when generating the feedback information. If the network does not support ECN, it needs to be modified. In this case, TCP receivers should generate the feedback information on behalf of the egress edge routers by inspecting the sequence numbers of the packets received. Each TCP receiver monitors and estimates the loss rate and sets the ITR or DTR bit in the TCP header based on the estimated loss rate.

We can implement the proposed scheme at the edge of any provider network that makes a contract with its customer in the form of an SLA specifying the target rate. The proposed scheme implemented at the edge of a provider network can guarantee the fair allocation of bandwidth among the customer networks to which the provider network is connected, because it simply adjusts the target rate in a distributed manner. For example, consider a tier-2 network that is a customer network of the tier-1 network and is also a provider network of the tier-3 network. The proposed scheme can be implemented at the edge of the tier-2 network, and guarantees the proportional allocation of bandwidth among the tier-2 networks that are connected to the tier-1 provider network. Similarly, it can also be implemented at the edge of the tier-2 network and guarantees the proportional allocation of bandwidth among the tier-3 networks.

The proposed scheme does not make any changes in the core routers and it produces only a small amount of computational overhead in the edge routers. The operations required to implement the scheme are very few. At the edge of the network, only a few addition, multiplication and comparison operations are required, while no additional operation is required at the core of the network. Consequently, the architecture of the proposed scheme is incrementally deployable and highly scalable. Also, the proposed scheme is practical, and constitutes a cost-effective solution for fair bandwidth allocation in DiffServ networks.
IV. SIMULATION

A. Simulation setup

We consider the case of $N$ aggregate flows sharing a common bottleneck link in a DiffServ network. The network used for the simulation consists of $2N$ AS’s, $N$ ingress edge routers, $N$ egress edge routers, and two core routers. Figure 4 shows the configuration of this DiffServ network for $N=2$. The link between two core routers, which has a capacity of $C$, is a bottleneck link. The propagation delays and the capacities of the links are shown in Fig. 4. Each AS contains many TCP senders/receivers, and we consider one-directional aggregate flows. We use greedy FTP applications over the TCP connections and CBR (Constant Bit Rate) applications over the UDP connections. The sending rate of CBR application is set to one tenth of the initial target rate, i.e., $0.1R_t$. The packet size is set to $1$ KByte.

The edge router queues implement the drop-tail policy. The core router queues are managed by the non-overlapping RIO algorithm [4]. We set the scheduling algorithm of RIO to the round-robin algorithm. The parameters for RIO are set as follows: $(q_{min}, q_{max}, p_{out}) = (10, 40, 0.1)$ for the OUT packets and $(q_{min}, q_{max}, p_{in}) = (40, 80, 0.02)$ for the IN packets. We set the two thresholds, $p_{th}^{min}$ and $p_{th}^{max}$, described in (11), to 0.001 and 0.02, respectively. The parameters of the window size $N_w$ and the weight $w$ are set to 10 and 0.05, respectively. We set the update interval of the target rate, $T_s$, to $20$ ms and the adjustment factor, $\delta$, to 0.001.

In order to quantify the fairness, we define a fairness index as $F = \left( \sum_{i=1}^{N} G_i \right)^2 / \left( N \sum_{i=1}^{N} G_i^2 \right)$, which is similar to Jain’s fairness index [21]. The fairness index $F$ is less than or equal to one, and is equal to one when the throughputs of all aggregates are equal to their fair shares.

B. Simulation 1: Performance comparison with other algorithms

In this simulation, we consider two aggregate flows that have different initial target rates, $R_{i,1}$ and $R_{i,2}$. We fix $R_{i,1}$ at $5$ Mb/s and vary $R_{i,2}$ from $1$ Mb/s to $15$ Mb/s, and we set $C$ to 10 Mb/s. Note that the network is under-subscribed when $R_{i,2} < 5$ Mb/s, and over-subscribed when $R_{i,2} > 5$ Mb/s. We incorporated the proposed scheme into both the TB and TSW algorithms, which are the most prevalent profile meters, and we refer to the resulting schemes as the adaptive token bucket (ATB) algorithm and the adaptive time sliding window (ATSW) algorithm, respectively. For the TB algorithm, the token bucket size of each aggregate is set to the product of the target rate and the average RTT of the flows. For the TSW algorithm, the monitoring interval of the arrival rate is set to $0.1$s. Figures 5 and 6 show a comparison of the throughputs and relative gains of the TB and ATB algorithms, and the TSB and ATSW algorithms, respectively.

In the case of the TB algorithm, the fairness is degraded significantly when the difference between $R_{i,1}$ and $R_{i,2}$ is large. Figure 5(a) shows that once $R_{i,2}$ exceeds 5 Mb/s, $R_{1, TB}$ and $R_{2, TB}$ tend to share the bandwidth almost equally, even when $R_{i,2}$ is three times higher than $R_{i,1}$. The difference between the throughput and its fair share, i.e., $|R_{i, TB} - R_{fair,i}|$, exceeds 2 Mb/s in some cases. However, in the case of the ATB algorithm, the $R_{i, ATB}$’s are close to their fair shares, whether the network is under-subscribed or over-subscribed; both $R_{1, ATB}$ and $R_{2, ATB}$ are within about 0.3 Mb/s of their fair shares in all cases. Also, Fig. 5(b) shows that $G_{1, TB}$ and $G_{2, TB}$ increase up to 1.9 in some cases, which means that their throughputs are 90% higher than their fair shares. In contrast to TB, if the ATB algorithm is adopted, $G_{1, ATB}$ and $G_{2, ATB}$ are between 0.96 and 1.18 in all cases.

Similarly, Fig. 6 shows that nearly the same improvement in fairness with the adaptive marking scheme is observed for TSW. These simulation results confirm that the proposed adaptive marking scheme can be incorporated with either the TB or the TSW algorithm, and that doing so greatly alleviates the problem of unfair bandwidth allocation. Hereafter, we focus our attention on a performance evaluation and comparison of the TB and ATB algorithms. This is because the incorporation of the adaptive marking scheme into the TSW algorithm would be expected to produce similar results as in the case of the TB algorithm.

C. Simulation 2: Performance evaluation of the adaptive marking scheme

We consider a simulation scenario wherein three aggregates share a common bottleneck link. For the under-subscription case, the initial target rates are set to $(R_{i,1})=(1, 2, 5)$ Mb/s and $C=15$ Mb/s. Similarly, for the over-subscription case, $(R_{i,2})$ are set to $(2.5, 8)$ Mb/s and $C=10$ Mb/s. Note that $(R_{i,1})=(1.875, 3.75, 9.375)$ Mb/s for the under-subscription case and $(R_{i,2})=(1.33, 3.33, 5.33)$ Mb/s for the over-subscription case.
Figure 7 shows the target rates (bold lines) and the throughputs (normal lines) of the three aggregates when the network is under-subscribed. From Fig. 7(a) which shows the results with the TB algorithm, we can see that the excess bandwidth \( i.e., C - \sum R_{t,i} \) is distributed evenly among the three aggregates. Hence, the throughput of each of the three aggregates is approximately \( 2\text{Mb/s} \) higher than its target rate, \( \{R_{i,TB}\}=(3.26, 4.13, 7.28)\text{Mb/s} \). The first aggregate gets 74% more bandwidth than its fair share, while the third aggregate gets 22% less bandwidth than its fair share, \( G_{1,TB}=1.74, G_{3,TB}=0.78 \). However, the ATB algorithm alleviates this unfairness by increasing the target rates so that they approach their fair allocations of bandwidth. As shown in Fig. 7(b), each throughput is close to its fair share, \( \{R_{i,ATB}\}=(2.25, 3.94, 5.07)\text{Mb/s} \), and \( \{G_{i,ATB}\}=(1.20, 1.05, 0.93) \). The fairness index increases from 0.901 to 0.990 due to the effect of the adaptive marking scheme.

For the over-subscription case, when the TB algorithm is used, a severe unfairness problem occurs, as shown in Fig. 8(a). The first aggregate, which has the smallest target rate, achieves its target rate and exceeds its fair share by 55% \( \text{i.e., } R_{1,TB}=2.06\text{Mb/s} > R_{f,1}=1.33\text{Mb/s} \) and \( G_{1,TB}=1.55 \) even though the throughput of the third aggregate, which has the largest target rate, is at most one half of its target rate and is 24% smaller than its fair share \( \text{i.e., } R_{3,TB}=4.08\text{Mb/s} < R_{f,3}=5.33\text{Mb/s} < R_{t,3} \) and \( G_{3,TB}=0.76 \). In contrast to TB, Fig. 8(b) shows that the target rates in the case of the ATB algorithm are reduced proportionally, and each aggregate nearly achieves its fair share, \( \{R_{i,ATB}\}=(1.57, 3.30, 5.07)\text{Mb/s} \), and \( \{G_{i,ATB}\}=(1.18, 0.99, 0.95) \).

Note that the target rates are adjusted using only a two-bit feedback signal, which is not informative enough to match them perfectly to their fair allocations of bandwidth. In spite of this limitation, the target rates are very close to the corresponding fair allocations, as shown in Fig. 7(b) and 8(b). The simulation results confirm that the adaptive marking scheme achieves proportional bandwidth allocation. Furthermore, it has been shown that the adaptive marking scheme is robust to the variations in the RTT and the number of flows in the aggregates [20].

**D. Simulation 3: Under dynamic traffic scenario**

In this simulation, we focus on the performance of the adaptive marking scheme under dynamic and more realistic traffic scenario.

For the dynamic traffic scenario, we introduced web-like short-lived flows, as well as unresponsive UDP flows and persistent long-lived TCP flows. We generated web-like mice traffic using \( \text{on/off} \) traffic, whose burst time and idle time were taken from the Pareto distributions, in order to mimic the self-similar property of web traffic [26]. Both the average burst time and the average idle time were set to 1s. During the \( \text{on} \)
Initially, the three aggregates had 10 TCP connections each. We also generated greedy and unresponsive CBR (Constant Bit Rate) traffic in each aggregate, whose sending rate is set to one tenth of the original target rate. Moreover, (Constant Bit Rate) traffic in each aggregate, whose sending rate is set to one tenth of the original target rate. Moreover, we changed the number of long-lived TCP flows dynamically.

Periods, packets were generated at a constant burst rate (e.g., 64Kb/s), whereas no packets were generated during the off periods. We set the number of web-like short-flows to 10 in each aggregate. We also generated greedy and unresponsive CBR (Constant Bit Rate) traffic in each aggregate, whose sending rate is set to one tenth of the original target rate. Moreover, we changed the number of long-lived TCP flows dynamically.

Initially, the three aggregates had 10 TCP connections each. At t=35s, 5 connections belonging to the first aggregate were dropped, and another 9 connections were established at t=51s. For the second aggregate, 7 additional connections were established at t=42s and lasted until t=83s. For the third aggregate, 12 connections were established randomly during the period between t=23s and t=29s and then were randomly dropped during the period between t=68s and t=73s. The other conditions are not changed from the Simulation 2.

The simulation results under these traffic scenarios are shown in Fig. 9(a) (under-subscription case) and 9(b) (over-subscription case). For the under-subscription case, $G_{1,ATB}$’s are calculated to be 1.15, 1.08, and 0.95, respectively, and $F_{ATB}$=0.994. For the over-subscription case, $G_{1,ATB}$=(1.16, 1.02, 0.93) and $F_{ATB}$=0.993. Compared with the results in the Simulation 2, $G_{1,ATB}$’s and $F_{ATB}$’s are almost the same. Furthermore, if we compare Fig. 9(a) with Fig. 7 and Fig. 9(b) with Fig. 8, we can see that the performance regarding throughput and fairness is not degraded due to dynamic traffic although the target rates are slightly affected by the changes in the traffic load. Figure 9 confirms that even when the traffic load changes dynamically and the unresponsive UDP flows and short-lived web-like flows coexist with persistent TCP flows, the proposed adaptive marking scheme works well.

**E. Simulation 4: When aggregates have different bottleneck links**

Until now, we have tested the performance of the adaptive marking scheme when all of the aggregates have the common bottleneck link. Here, we perform the simulation under conditions where some aggregates have different bottleneck links.

Let $C_i$ denote the capacity of the link between the $i$th ingress edge router and the core router $C_1$ depicted in Fig. 4. We set $C_i$ differently as $C_1$=(10, 20, 5)Mb/s, and set $C$ to 14Mb/s. The initial target rates are set as $R_{f,i,1}$=(1, 2, 4)Mb/s. If each $C_i$ were larger than $C$, then the link between the two core routers would become the common bottleneck link, and the fair shares would be twice the values of the initial target rates, i.e., $R_{f,i,3}$=(2, 4, 8)Mb/s. However, $C_3$ is smaller than $R_{f,3}$, and $R_3$ is bounded by $C_3$, i.e., $R_3 \leq C_3 \leq 5$Mb/s. Hence, the remaining bandwidth (i.e., $C - \sum_i \min(R_{f,i}, C_i) = 3$Mb/s) should be reallocated to the first and the second aggregates in proportion to their target rates, and the fair shares become $R_{f,i,3}$=(3, 6, 5)Mb/s.

Figure 10 shows that the $R_{t,i}$’s increase almost twice of their initial values. While $R_{t,3}$ increases beyond 8Mb/s, the
分配得到的带宽不能充分利用。一个应用程序的带宽分配情况如图11(b)所示。这个应用程序的带宽分配情况和图11(a)中的情况相同，这表明分配的带宽不是按照比例分配的。平均分配的带宽为(R1, ATB) = (3.79, 5.06, 4.92)Mb/s。另一方面，当使用TB算法时，虔诚分配的带宽为(R1, TB) = (3.13, 5.84, 4.91)Mb/s。由于自适应标记方案，相对收益，G1和G2，从1.26增加到0.84和1.04和0.97，分别增加，公平性指数从0.972增加到0.999。对比这些结果，可以发现与仿真2和仿真3的结果相比，G1, ATB的和F, TB的相对收益相同，这表明自适应标记方案在即使当 aggregates 流量有不同瓶颈连接时，也能工作得很好。此外，R1和R2的波动显著降低。标准偏差(R1, ATB)从1.25Mb/s和1.32Mb/s降低到0.70Mb/s和0.73Mb/s，这对于需要一致比特率的应用程序（如VoIP或音频流）很重要。

第五章 结论

这篇论文关注的是分配给 aggregate TCP 流在 DiffServ 网络的公平带宽分配问题。我们分析性地展示了当前的DiffServ网络不能很好地分配带宽。当分配给 aggregate TCP 流的带宽不能充分利用，而分配给 aggregate TCP 流的带宽不能充分利用，这表明分配的带宽不是按照比例分配的。平均分配的带宽为(R1, ATB) = (3.79, 5.06, 4.92)Mb/s。另一方面，当使用TB算法时，虔诚分配的带宽为(R1, TB) = (3.13, 5.84, 4.91)Mb/s。由于自适应标记方案，相对收益，G1和G2，从1.26增加到0.84和1.04和0.97，分别增加，公平性指数从0.972增加到0.999。对比这些结果，可以发现与仿真2和仿真3的结果相比，G1, ATB的和F, TB的相对收益相同，这表明自适应标记方案在即使当 aggregates 流量有不同瓶颈连接时，也能工作得很好。此外，R1和R2的波动显著降低。标准偏差(R1, ATB)从1.25Mb/s和1.32Mb/s降低到0.70Mb/s和0.73Mb/s，这对于需要一致比特率的应用程序（如VoIP或音频流）很重要。

V. CONCLUSION

This paper focuses on the issue of fair bandwidth allocation among aggregate TCP flows in DiffServ networks. We analytically showed that the current DiffServ networks allocate bandwidth unfairly. An aggregate with a smaller target rate occupies more bandwidth than its fair share, while an aggregate with a larger target rate gets less than its fair share. Based on this analysis, we proposed the adaptive marking scheme that can allocate bandwidth in proportion to the target rates. The main idea of this scheme is to adjust the target rates to their fair shares according to the congestion level of the network. If the network is severely congested or, conversely, if it is far from being congested, the target rates are decreased or increased proportionally. This scheme can be implemented simply and in a distributed manner using only two-bit feedback information conveyed in the packet headers, without maintaining any per-flow state at the core routers or requiring any additional signaling protocol. The proposed scheme is scalable and compatible with the existing TCP/IP protocol. The extensive simulation performed as part of this study confirmed that the proposed scheme achieves proportional bandwidth allocation under various network conditions.

REFERENCES

Fig. 11. Throughputs of three aggregates when aggregates have different bottleneck links.


**APPENDIX**

A. Proof of Proposition 3

By setting $R_{l,i,t}(t) = 0$ in (13), we can obtain $\bar{p}_{in}$ in the steady state, as $p_{in}^{th} < \bar{p}_{in} < p_{in}^{th}$. Note that

$$ p_{in}^{th} = 1 - \prod_{l_{k} \in L} (1 - \bar{p}_{in,l_{k}}) \approx \sum_{l_{k} \in L} p_{in,l_{k}}^{*}. $$

Thus, due to the preferential dropping mechanism in RIO as shown in Fig. 3, $\bar{p}_{in} > p_{in,t}^{b} > 0$ results in $p_{out,t}^{b} = 1$.

Hence,

$$ \bar{p}_{out} = 1 - \prod_{l_{k} \in L} (1 - p_{out,l_{k}}^{*}) = 1. $$

Let us consider that $R_{i}^{*}$ is made up of two components, $R_{in,i}^{*}$ and $R_{out,i}^{*}$, which are composed of IN packets and OUT packets, respectively, i.e., $R_{i}^{*} = R_{in,i}^{*} + R_{out,i}^{*}$. Based on the assumption that TCP senders always have data to send, $R_{in,i}^{*}$ is equal to the corresponding target rate $R_{i}$. Then, the steady state value for the overall dropping probability of the $i$th aggregate becomes

$$ p_{i}^{*} = 1 - \frac{1}{R_{i}^{*}} \left[ R_{in,i}^{*}(1 - \bar{p}_{in}^{*}) + R_{out,i}^{*}(1 - \bar{p}_{out}^{*}) \right] $$

$$ = 1 - \frac{R_{in,i}^{*}}{R_{i}} (1 - \bar{p}_{in}^{*}). $$

Because TCP flows are assumed to be homogeneous, the steady state throughput of the $i$th aggregate is obtained from (12) as

$$ R_{i}^{*} = a N_{i} / (\sqrt{p_{in}^{*}} T). $$

From (A-1) and (A-2), $R_{i}^{*}$ becomes

$$ R_{i}^{*} = \frac{(1 - \bar{p}_{in}^{*}) R_{i}^{*,t}}{2} \left[ 1 + \sqrt{1 + \left( \frac{\sqrt{6 N_{i}}}{(1 - \bar{p}_{in}^{*}) R_{i}^{*,t}} \right)^{2}} \right]. $$

From the assumption of $R_{i}^{*} T / N_{i} \gg 1$, we approximate $4 R_{i}^{*}$ in (A-3) as

$$ R_{i}^{*} = (1 - \bar{p}_{in}^{*}) R_{i}^{*}. $$

Now, let us consider the queue dynamics. By setting $\bar{q}_{l}(t)$ = 0 in (14), we can see that the total throughput of the aggregates that traverse the bottleneck link is equal to the corresponding link capacity, i.e.,

$$ \sum_{j \in S_{i}} R_{j}^{*} = C_{i}^{b}. $$

By combining (A-4) and (A-5), $R_{i}^{*}$ becomes

$$ R_{i}^{*} = \left( \frac{R_{i}^{*}}{\sum_{j \in S_{i}} R_{j}^{*}} \right) C_{i}^{b}. $$

From (A-6) and (16), we can show that $R_{i}^{*}$ achieves its fair share.

Note that for the typical range of system parameters, the approximation error of (A-4) is less than 1.2% when $R_{i}^{*} > 100$Mbs (12500 packet/s), $T > 100$ms, $N_{i} < 100$, and $p_{in} < 0.01$. 