# Modeling TCP Behavior in a Differentiated Services Network

Ikjun Yeom and A. L. Narasimha Reddy, *Senior Member, IEEE*

***Abstract*—The differentiated services architecture has been proposed for providing different levels of services and has recently received wide attention. A packet in a diff-serv domain is classified into a class of service according to its contract profile and treated differently by its class. While many studies have addressed issues on the diff-serv architecture (e.g., dropper, marker, classifier and shaper), there have been few attempts to analytically understand a flow's behavior in a diff-serv network.**

**In this paper, we propose simple models of TCP behavior in a diff-serv network. Our models quantitatively characterize TCP throughput as functions of the contract rate, the packet-drop rate and the round-trip time in either two-drop precedence or three-drop precedence network. We also extend our models to aggregated flows. The models are validated through a number of simulations.**

***Index Terms*—AF PHB, differentiated service, TCP modeling.**

## I. Introduction

The differentiated services (DS or diff-serv) architecture has been recently proposed for providing different levels of services [1], [3], [5]. To support service differentiation for individual or aggregated flows, the architecture provides a *meter* and a *marker* at the edges of the network and a *dropper* with various dropping mechanisms in the core of the network. Fig. 1 shows a simple diagram of a differentiated services network. A meter measures the temporal rate of a flow, and a marker sets the DS field of a packet of the flow based on its contract rate (also referred to as reservation rate in this paper) and the current sending rate. A dropper discards packets of different flows according to the DS fields of the packets and the current load levels at the dropper. The current architecture defines expedited forwarding (EF) and assured forwarding (AF) per-hop behaviors (PHBs) to allow delay and bandwidth differentiation. It is expected that the diff-serv architecture will be utilized for differentiation of aggregated flows.

Several recent studies have shown that it is difficult to guarantee requested throughput to individual TCP flows in such networks [2], [6], [14], [24]. When a TCP flow detects packet loss, the flow assumes that the packet loss is due to congestion in the network. A TCP flow tries to avoid this congestion by halving its transmission rate. This reduced rate may be less than its reservation rate, and this results in loss of throughput. Earlier simulation results [2], [6], [14] have shown that flows with relatively higher reservations may realize less throughput than their target rates while flows with smaller reservations may realize throughput higher than their target rates. It has been also shown that it is difficult to guarantee absolute bandwidth with a simple marking and dropping scheme [24]. There is a clear need to understand the end-to-end performance of a TCP flow in a diff-serv network that is characterized by PHBs.

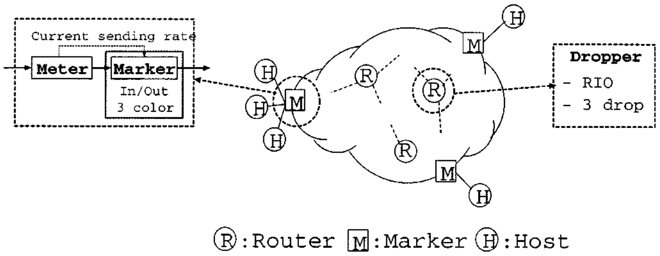


Fig. 1. Differentiated services network.

Many studies proposed throughput models for TCP flows by characterizing the behavior of the congestion avoidance schemes [7], [8], [12], [13]. With these models, we can analyze and predict TCP throughput in networks when packets are not marked or treated differently within the network. The objective of this paper is to propose steady-state throughput models for TCP flows in a differentiated services network as a function of reservation rates, packet drop rates and round-trip times. It is expected that such a model provides an insight into the end-to-end performance of TCP flows in a diff-serv network. It is also expected that such a model may lead to improvements/modifications of the basic PHBs in order to realize end-to-end performance goals. Some of the questions we expect to answer are: 1) How is the steady-state TCP throughput related to a flow's contract rate? 2) How does the realized bandwidth vary as a function of packet drop rate? 3) What is the realizable bandwidth for a best-effort flow? 4) What is the cut-off contract rate below which contracts can be met? and 5) Given a steady-state bandwidth goal, what contract rate should a flow request from the network (when these can be different)?

To develop the models, we first characterize the behavior of a TCP flow in a two-drop precedence network, called *RED In/Out (RIO)* [1], [5]. Then, we extend this model to a network with three-drop precedences [14], [17], [18]. To complete our study, we extend the models to aggregated marking schemes.

This paper makes the following significant contributions: 1) We present a simple analytical model for the steady-state throughput of a TCP flow in a differentiated services network. 2) We present extensive simulations to validate the model in different scenarios and different network conditions. 3) We derive the throughput of an individual flow within an aggregation when the aggregated source employs proportional marking.

Manuscript received September 21, 1999; revised January 5, 2000, June 8, 2000, and August 15, 2000; approved by IEEE/ACM Transactions on Networking Editor S. Floyd. This work was supported in part by a Texas ATP grant, a National Science Foundation Career Award, and by EMC Corporation.

The authors are with the Department of Electrical Engineering, Texas A&M University, College Station, TX 77843-3128 USA (e-mail: ikjun@ee.tamu.edu; reddy@ee.tamu.edu).

Publisher Item Identifier S 1063-6692(01)01315-2.

4) We show that the analytical model provides intuitive understanding of TCP behavior in a differentiated services network. The developed model shows that a TCP flow cannot always achieve its contract rate because of the sawtooth behavior in congestion avoidance.

The rest of this paper is organized as follows. In Section II, we describe the details of differentiated services schemes studied here. Section III proposes and derives our models and presents simulations to validate the models. In Section IV, we extend our models to aggregated flows. Section V discusses our models and presents the summary of related studies on TCP modeling and differentiated services. In Section VI, we conclude this paper and present directions for further research.

## II. Differentiated Services Schemes

The differentiated services architecture consists of many components. Among those components, the dropping policy associated with the marker has an important role in providing different levels of service. Current architecture allows the definition of many different PHBs that can provide different levels of service [1]. We study the assured forwarding PHB [3], [5], [2] which focuses on differentiation of provided bandwidth. In this section, we describe two and three drop precedence policies.

### *A. Two-Drop Precedence*

Two-drop precedence policy was originally proposed in [5] as RIO. In a RIO network, the edge devices of the network monitor and mark incoming packets of either individual or aggregated flows. A packet of a flow is marked IN if the temporal sending rate at the arrival time of the packet is within the contract profile of the flow. Otherwise, the packet is marked OUT. The temporal sending rate of a flow is measured using *Time Sliding Window (TSW)* [5] or a *token bucket* controller. In this paper, the TSW packet marker is studied.

The core routers in the network maintain a virtual queue for IN packets and a physical queue for both IN and OUT packets. When the network is congested, the routers begin dropping the OUT packets first. If the congestion persists even after discarding all the incoming OUT packets, the IN packets are dropped. With this dropping policy, the network gives preference to the IN packets and provides different levels of service to users based on their service contracts.

### *B. Three-Drop Precedence*

Three-drop precedence policy was proposed as an extension of two-drop precedence in [17], [18]. In a three-drop precedence network, the edge devices mark a packet as one of *green*, *yellow* or *red* depending on the sending rate and the reservation rate for each color. Generally, it is recommended that the reservation rate for *yellow* is set to be greater than or equal to the reservation rate for *green*. If the current sending rate is less than the reservation for *green*, the packet is marked as *green*. If the sending rate is greater than the reservation for *green* but less than the reservation for *yellow*, the packet is marked as *yellow*. Otherwise, the packet is marked as *red*. The core routers provide differentiation by dropping the *red* packets first, the *yellow* packets second and then the *green* packets. It is expected that the three color marking and dropping policies give better control in realizing performance goals.

```
Upon each packet arrival:
  $avg\_rate = ((avg\_rate * win\_len) + pkt\_size)$
             $/(win\_len + now - last\_arrival)$
  $last\_arrival = now$

$win\_len$ : a constant
$avg\_rate$: a flow's estimated sending rate
$pkt\_size$: the packet size of the arriving packet
```

Fig. 2. TSW algorithm.

## III. TCP Modeling in a Differentiated Services Network

In this section, we present models for TCP throughput in a diff-serv network and illustrate the difference between the achieved throughput of a TCP flow and its contract rate quantitatively. The models are developed for two-drop precedence and three-drop precedence policies. We assume that packets are dropped randomly rather than in bursts since the packets are randomly picked to be discarded in RED routers when the average queue length of the router is less than a certain threshold. TCP models for regular networks have considered bursty losses [7] and random losses [12]. Our assumption of random losses may not hold in all the situations in a diff-serv network. The implications of this assumption are discussed in Section V. We define IN (OUT) packet loss probability, $p_{\text{in}}$ ($p_{\text{out}}$) as the ratio of IN (OUT) packets dropped by the number of IN (OUT) packets sent. Our models are based on the IN (OUT) packet loss rates observed by individual flows.

### *A. Packet Marking with TSW Rate Estimator*

In this section, we discuss packet marking with the TSW rate estimator. Fig. 2 shows the TSW algorithm proposed in [5]. It is shown that the TSW marker remembers the history of the past $win\_len$ interval and smooths out the TCP's burstiness. $win\_len$ is recommended to be of the order of an RTT in [5]. A marker marks a packet IN when this $avg\_rate$ is less than the contract rate. When the $avg\_rate$ is greater than the contract rate, a packet is marked IN with the probability, $contract\_rate/avg\_rate$. The rest are marked OUT.

It is typically known that a TCP source sends packets in a burst within an RTT interval. More precisely, a TCP source sends packets until the total number of unacknowledged packets is $cwnd$[1] (congestion window) and waits for ACK arrival. The TCP receiver sends an ACK for every one or two packets received, and the TCP sender uses this ACK-clocking to send new data packets. Consequently, a TCP flow's sending rate averaged in each RTT interval is $cwnd * pkt_size/$RTT. Here we define *reservation window* ($rwnd$ or $R$) as

$$R = \frac{contract\_rate}{pkt\_size} \times \text{RTT}. \tag{1}$$

With $rwnd$, we can see that the ideal packet marking behavior as one of the following two cases:

[1] In real TCP, window size is in unit of bytes, but we use unit of packets for simplicity.

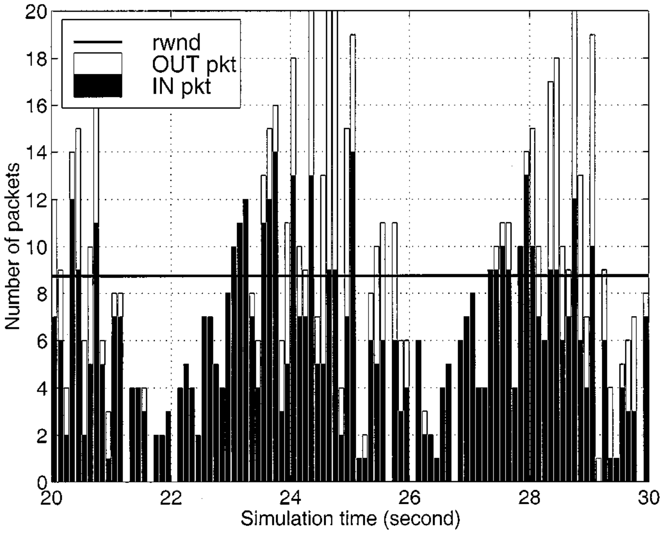


Fig. 3. IN/OUT packet distribution.

1) When $cwnd \le rwnd$: every packet is marked IN.
2) When $cwnd > rwnd$: a packet is marked IN with the probability $rwnd/cwnd$, and the rest are marked OUT.

It is not feasible, however, to measure RTT accurately. It is also not feasible to synchronize measuring intervals to packet sending intervals in a TCP source. To observe practical packet marking behavior, we present Fig. 3 observed from a simulation using ns-2 [20]. This figure shows packet marking of a TCP flow: The contract rate is 0.5 Mb/s; packet size is 1 KB; average RTT (including queueing delay) is 140 ms; IN packet drop rate is zero and OUT packet drop rate is 0.027; average achieved throughput is 0.69 Mb/s. The employed marker is a general TSW packet marker with fixed $win\_len$ (= 1 second). Here note that $rwnd = 0.5$ Mb/s $*$ 140 ms/1 KB = 8.75 packets.

Fig. 3 shows packet distribution over time. Simulation time is divided by the average RTT interval (140 ms), and we count the number of IN/OUT packets sent in each interval. Each bar represents the total number of packets sent. The *black* portion indicates the number of IN packets, and the *white* portion indicates the number of OUT packets.[2] Here note that each interval may not be exactly synchronized to the individual RTT due to queueing delay variations. This figure shows that the most packets sent in an interval where the total number of packets sent is less than $rwnd$ are marked IN, and that the number of OUT packets increases as the total number of packets increases.

From the above observations, we confirm that the practical packet marking behavior is approximately the same as the ideal behavior for this scenario. Practical marking behavior will sometimes differ from this ideal behavior. However, additional simulations later in this paper, using TSW packet marking, give further evidence that the packet marking of TSW is a viable approximation of this ideal behavior. In the rest of this paper for deriving the TCP models, we assume the ideal marking behavior: 1) Every packet in a $cwnd$ is marked IN when $cwnd$ is less than $rwnd$. 2) A packet is marked IN with the probability $rwnd/cwnd$, and the rest are marked OUT when $cwnd$ is greater than $rwnd$.

### B. Modeling for Two-Drop Precedence

In a steady state, a flow going through a differentiated services network can experience different levels of congestion based on its contract rate and the network dynamics. A flow that experiences no IN packet drops ($p_{\text{in}} = 0$) is said to observe an *undersubscribed path*. A flow that does not transmit any OUT packets either because every OUT packet is dropped ($p_{\text{out}} = 1$) or because the sending rate is less than the contract profile is said to observe an *oversubscribed path*. These classifications let us analyze simpler network conditions, which in turn are used to develop the general model. We develop separate models for these two situations and combine these two to model a general situation where a flow may experience a nonzero drop rate for IN and OUT packets. It is to be noted that a single network may be classified differently by the nature of packet drops experienced by individual flows. Different routers may observe different levels of congestion and may drop IN and OUT packets at the same time. The general model is applied in such a situation.

While we develop the models, we consider the average round-trip time (RTT) and the average packet size of a flow. We assume that there are no ACK drops in the network and that window size is not limited by the advertised window of the receiver. Our model relies heavily on earlier work in [7].

*1) A Flow Through an Undersubscribed Path:* We assume that a flow through an undersubscribed path does not observe any IN packet drops. The steady state TCP throughput is defined as

$$B = \frac{\text{Total number of packets sent} \times k}{\text{Total transmission time}} \tag{2}$$

where $k$ is the packet size. It is noted that $B$ strictly means the sending rate rather than the throughput. We assume that impact of packet loss on throughput is negligible. It has been similarly assumed in [7].

We define a "period" $A_i$ as the time between immediately after one packet drop and just before the next packet drop. $N_i$ is defined as the number of packets sent in period $A_i$. In the steady state, we can assume that the number of packets sent, $N$, in each period is the same. Similar assumption has been made for modeling TCP throughput in [7], [8], [12], [13]. Then, we can express the throughput as

$$B = \frac{E[N] \times k}{E[A]}. \tag{3}$$

To derive $N_i$ and $A_i$, we define $W_i$ as the window size at the end of period $A_i$ and $X_i$ as the number of rounds in period $A_i$ as shown in Fig. 4. To model delayed-ACKs, we assume that $d$ packets are acknowledged by one ACK as in [7]. Then, we have

$$W_i = \frac{W_{i-1}}{2} + \frac{X_i}{d} \tag{4}$$

[2] In Fig. 3, the *black* portion is shown in bottom, and the *white* portion is shown on top. This is only for visual comparison of number of IN/OUT packets and does not mean that packets are marked IN first.

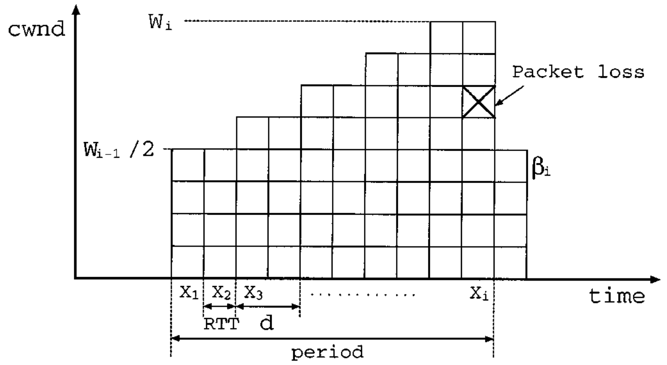


Fig. 4. Packet sent in each period.

since the window is increased by one every $d$ rounds. Now, $N_i$ is expressed by

$$N_i = \sum_{j=0}^{X_i/d-1} \left(\frac{W_{i-1}}{2} + j\right) d + \beta_i \tag{5}$$

$$= \left\{\frac{X_i W_{i-1}}{2d} + \frac{X_i}{2d}\left(\frac{X_i}{d} - 1\right)\right\} d + \beta_i \tag{6}$$

$$= \frac{X_i}{2}\left(\frac{W_{i-1}}{2} + W_i - 1\right) + \beta_i \tag{7}$$

where $\beta_i$ is the number of packets sent in the last round. We define $N_{\mathrm{out}(i)}$ and $N_{\mathrm{in}(i)}$ as the number of packets sent marked OUT and IN in each period, respectively. Then, the mean of $N_i$ is given by

$$E[N] = E[N_{\mathrm{out}}] + E[N_{\mathrm{in}}]. \tag{8}$$

If $p_{\mathrm{out}}$ is the probability that an OUT packet is dropped, then the probability that $i$th OUT packet is dropped is given as

$$P[N_{\mathrm{out}} = i] = (1 - p_{\mathrm{out}})^{i-1} p_{\mathrm{out}}. \tag{9}$$

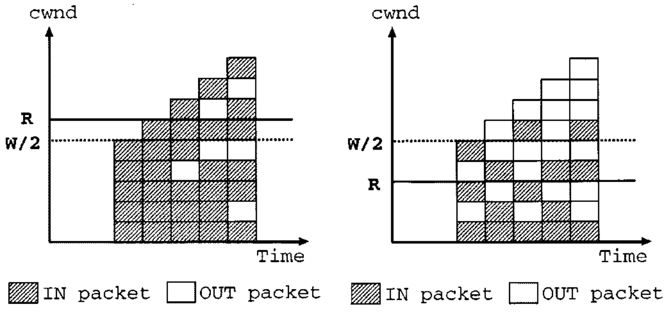
Then, the mean of $N_{\mathrm{out}}$ is

$$E[N_{\mathrm{out}}] = \sum_{i=1}^{\infty} i(1 - p_{\mathrm{out}})^{i-1} p_{\mathrm{out}} = \frac{1}{p_{\mathrm{out}}}. \tag{10}$$

To derive $E[N_{\mathrm{in}}]$, we consider two scenarios shown in Fig. 5. In Fig. 5(a), $R \geq W/2$ and as a result of an OUT packet drop, the sending rate has fallen below the reservation rate. In Fig. 5(b), $R < W/2$ and even after an OUT packet drop, the sending rate remains above the contract rate. Then, $N_{\mathrm{in}(i)}$ is given by

$$N_{\mathrm{in}(i)} = \sum_{j=0}^{X_i/d-1} \min\left\{\frac{W_{i-1}}{2} + j, R\right\} d \tag{11}$$

$$= \begin{cases} X_i R & \text{if } R \leq W_{i-1}/2 \\ \displaystyle\sum_{j=0}^{(x_i/d - W_i + R - 1)} \left(\frac{W_{i-1}}{2} + j\right) d & \\ \quad + dR(W_i - R) & \text{otherwise.} \end{cases} \tag{12}$$



(a) (b)

Fig. 5. Reservation and congestion windows. (a) $R \geq W/2$. (b) $R < W/2$.

After algebraic manipulations, we have

$$N_{\mathrm{in}(i)} = \begin{cases} dR\left(W_i - \dfrac{W_{i-1}}{2}\right) & \text{if } R \leq W_{i-1}/2 \\ d\left\{RW_i - \dfrac{R}{2}(R+1)\right. & \\ \quad \left. - \dfrac{W_{i-1}}{8}(W_{i-1} - 2)\right\} & \text{otherwise.} \end{cases} \tag{13}$$

From (7) and (8), we have

$$E[N] = E[N_{\mathrm{out}}] + E[N_{\mathrm{in}}] + E[\beta]$$
$$= \frac{E[X]}{2}\left(\frac{E[W]}{2} + E[W] - 1\right) + E[\beta]. \tag{14}$$

When $R \leq E[W]/2$, from (10) and (13), we have

$$E[N] = \frac{1}{p_{\mathrm{out}}} + \frac{E[W]d}{2} R + E[\beta]$$
$$= \frac{E[W]d}{4}\left(\frac{E[W]}{2} + E[W] - 1\right) + E[\beta]. \tag{15}$$

From (15), it follows that

$$E[W] = \frac{2R+1}{3} + \frac{4}{3}\sqrt{\frac{R^2}{4} + \frac{3}{2dp_{\mathrm{out}}} + \frac{1}{16} + \frac{1}{4}R}. \tag{16}$$

For small values of $p_{\mathrm{out}}$ and large values of $R$, it is observed that

$$E[W] \approx \frac{2R + \sqrt{4R^2 + \dfrac{24}{dp_{\mathrm{out}}}}}{3}. \tag{17}$$

When $R > E[W]/2$, we also have

$$E[N] = \frac{1}{p_{\mathrm{out}}} + d\left\{E[W]R - \frac{E^2[W]}{8} - \frac{R^2}{2}\right.$$
$$\left. - \frac{1}{2}\left(R - \frac{E[W]}{2}\right)\right\} + E[\beta]$$
$$= \frac{E[W]d}{4}\left(\frac{E[W]}{2} + E[W] - 1\right) + E[\beta]. \tag{18}$$

From (18), it follows that

$$E[W] = R + \frac{1}{2} + \sqrt{\frac{2}{dp_{\mathrm{out}}} + \frac{1}{4}}. \tag{19}$$

For small values of $p_{\text{out}}$ and large values of $R$, it is observed that

$$E[W] \approx R + \sqrt{\frac{2}{dp_{\text{out}}}}. \tag{20}$$

Now, we consider $E[A]$ in (3). From Fig. 4, $E[A]$ is given by

$$E[A] = (E[X]+1)\text{RTT} = \left(\frac{E[W]d}{2}+1\right)\text{RTT}. \tag{21}$$

We assume that $\beta_i$ is uniformly distributed between 1 to $W_i$ and that $X_i$ and $W_i$ are i.i.d. random variables as in [7]. Then we have $E[\beta] = E[X] = E[W]/2$. Thus, from (3), (7) and (21) the throughput of a TCP flow is expressed as follows,

$$B = \frac{\left(\frac{3d}{8}E[W] + \frac{2-d}{4}\right)E[W]k}{\left(\frac{dE[W]}{2}+1\right)\text{RTT}} \tag{22}$$

where $E[W]$ is given by (16) and (19).

Now, we extend this to include time-outs. When a packet loss is detected by a time-out, the sender does not send packets until the time-out. Thus, to consider time-outs, (3) is extended to

$$B = \frac{\left\{\left(\frac{3d}{8}E[W] + \frac{2-d}{4}\right)E[W] + P_{\text{TO}}E[N_{\text{TO}}]\right\}k}{\left(\frac{dE[W]}{2}+1\right)\text{RTT} + P_{\text{TO}}E[T_{\text{TO}}]}. \tag{23}$$

Here, $P_{\text{TO}}$ and $T_{\text{TO}}$ represent the probability that packet loss is detected by a time-out and the time duration taken for detecting a time-out, respectively. $N_{\text{TO}}$ is the number of packets sent in a time-out period and is typically one or two.[3]

From [11], time-out in TCP-Reno occurs 1) when two packets are dropped and the congestion window is less than ten packets, or 2) when three or more packets are dropped within a window and the number of packets between the first and the second packet drops is less than $2+3W/4$. In this paper, we will make the simplifying assumption that three or more packet drops result in a time-out. The probability that there are three packet drops and the number of packets between the first and the second packet drops is greater than $2+3W/4$ is very low. Then, $P_{\text{TO}}$ is the probability that three or more packets are dropped when there is at least one packet drop. Thus, $P_{\text{TO}}$ is expressed as is shown in (24), at the bottom of the page.

$$P_{\text{TO}} = \frac{1-(1-p_{\text{out}})^{E[W]-R} - (E[W]-R)p_{\text{out}}(1-p_{\text{out}})^{E[W]-R-1}}{1-(1-p_{\text{out}})^{E[W]-R}} - \frac{\frac{(E[W]-R)(E[W]-R-1)}{2}p_{\text{out}}^2(1-p_{\text{out}})^{E[W]-R-2}}{1-(1-p_{\text{out}})^{E[W]-R}}. \tag{24}$$

To model exponential backoff in time-out, we manipulate $E[T_{\text{TO}}]$ in (23). Exponential backoff is a scheme to avoid serious congestion by doubling waiting time for retransmission when consecutive packet losses occur. The waiting time for retransmission is doubled until six consecutive packet losses. Thus, $T_{\text{TO}}$ is given by

$$T_{\text{TO}} = \begin{cases} 2^l T_0, & \text{for } 0 \le l \le 6 \\ 64T_0, & \text{for } l > 6 \end{cases} \tag{25}$$

where $T_0$ is the time taken to detect the first time-out, and $l$ is the number of consecutive packet losses. The probability of more than six consecutive packet losses is very low, and we ignore it for simplicity. Then, the mean of $T_{\text{TO}}$ is

$$E[T_{\text{TO}}] = \frac{\sum_{l=0}^{6}(2p_{\text{out}})^l}{\sum_{l=0}^{6}p_{\text{out}}^l}T_0 = \frac{(1-(2p_{\text{out}})^7)(1-p_{\text{out}})}{(1-2p_{\text{out}})(1-p_{\text{out}}^7)}T_0. \tag{26}$$

*2) A Flow Through an Oversubscribed Path:* A flow through an oversubscribed path observes IN packet losses as well. There are two possible situations: 1) Every packet is marked as IN when the sending rate is less than the contract profile $(W_i < R)$. 2) Whenever a packet is marked as OUT, the packet is dropped $(W_i = R)$. In both the situations, OUT packets are not transmitted. Thus, the total number of packets sent in each period, $N_i$, is given by

$$N_i = N_{\text{in}(i)}. \tag{27}$$

Here note that we ignore one OUT packet sent when $W_i = R$. Then, $E[N_{\text{in}}]$ is calculated in a similar way in (10) as following

$$E[N_{\text{in}}] = \min\left\{\frac{1}{p_{\text{in}}}, \frac{3d}{8}R^2 + \frac{d}{4}R\right\} + E[\beta]. \tag{28}$$

Then, from (14), (28) and (27) we have

$$\begin{aligned} E[N] &= E[N_{\text{in}}] \\ &= \frac{3d}{8}E^2[W] + \frac{d}{4}E[W] + E[\beta] \\ &= \min\left\{\frac{1}{p_{\text{in}}}, \frac{3d}{8}R^2 + \frac{d}{4}R\right\} + E[\beta] \end{aligned} \tag{29}$$

$$E[W] = \min\left(\sqrt{\frac{1}{9} + \frac{8}{3dp_{\text{in}}}} - \frac{1}{3}, R\right). \tag{30}$$

[3] In [7], $E[N_{\text{TO}}]$ is given by $1/(1-p)$ where $p$ is the drop probability. If $p$ is less than 0.5, $E[N_{\text{TO}}]$ is in between 1 and 2. Refer [7] for detail.

Applying (30) to (23), we can model TCP throughput in an oversubscribed path where $P_{\text{TO}}$ and $E[T_{\text{TO}}]$ are given by

$$P_{\text{TO}} = \frac{1-(1-p_{\text{in}})^{E[W]} - E[W]p_{\text{in}}(1-p_{\text{in}})^{E[W]-1}}{1-(1-p_{\text{in}})^{E[W]}} - \frac{\dfrac{E[W](E[W]-1)}{2}\, p_{\text{in}}^2 (1-p_{\text{in}})^{E[W]-2}}{1-(1-p_{\text{in}})^{E[W]}} \tag{31}$$

$$E[T_{\text{TO}}] = \frac{(1-(2p_{\text{in}})^7)(1-p_{\text{in}})}{(1-2p_{\text{in}})(1-p_{\text{in}}^7)}\, T_0. \tag{32}$$

*3) Combined Model:* In this section, we combine the models for the undersubscribed and the oversubscribed paths so that our model can fit the general situation.

To combine the two models, we extend the definition of a period as follows: 1) Undersubscribed period $A_u$ is defined as a period ended by an OUT packet drop, and let $B_u$ be the throughput achieved in $A_u$ and $p'_{\text{out}}$ be the probability of an OUT packet loss in such a period. 2) Oversubscribed period $A_o$ is defined as a period ended by an IN packet drop, and let $B_o$ be the throughput achieved in $A_o$ and $p'_{\text{in}}$ be the probability of an IN packet loss in such a period. We assume that IN and OUT packet losses are random and not correlated to each other. Then, average throughput, $B$ in steady state is

$$B = Q_u B_u(p'_{\text{out}}) + Q_o B_o(p'_{\text{in}}) \tag{33}$$

where $Q_u$ and $Q_o$ are the probabilities that $A_u$ and $A_o$ occur, respectively, and $Q_u + Q_o = 1$. From the definition of $A_u$ and $A_o$, $Q_u$ is the ratio of the number of OUT packet losses and the total number of losses, and $Q_o$ is the ratio of the number of IN packet losses and the total number of losses.

$$Q_u = \frac{(1-p_m)p_{\text{out}}}{(1-p_m)p_{\text{out}} + p_m p_{\text{in}}} \tag{34}$$

$$Q_o = \frac{p_m p_{\text{in}}}{(1-p_m)p_{\text{out}} + p_m p_{\text{in}}} \tag{35}$$

where $p_m$ is the probability that a packet is marked IN. Since we assume that there is no OUT packet transmitted in $A_o$, $p'_{\text{out}}$ is equal to $p_{\text{out}}$. This assumption may not hold when the network changes quickly, and we can have both IN and OUT packet drops in a period. However, if a network is stabilized and queue length in RIO routers does not change quickly, OUT packet should be discarded first, and the assumption may be protected. Then, we can directly use (23) for $B_u$ in (33).

$p'_{\text{in}}$ is not equal to $p_{\text{in}}$ since $p'_{\text{in}}$ is the IN packet loss rate from IN packets sent in $A_o$ while $p_{\text{in}}$ is the IN packet loss rate from IN packets sent in both $A_o$ and $A_u$. For calculating $p'_{\text{in}}$, we divide packets into three groups, OUT packets sent in $A_u$, IN packets sent in $A_u$, and IN packets sent in $A_o$. Note that no OUT packets are sent in $A_o$. Then, we have

$$E[N] = Q_u(E[N_{\text{out}}] + E[N_{\text{in}}^u]) + Q_o E[N_{\text{in}}^o] \tag{36}$$

where $N_{\text{out}}$, $N_{\text{in}}^u$ and $N_{\text{in}}^o$ are the number of OUT packets sent in an $A_u$, the number of IN packets sent in an $A_u$, and the number of IN packets sent in an $A_o$, respectively. From the definition of $p_m$

$$p_m = \frac{Q_u E[N_{\text{in}}^u] + Q_o E[N_{\text{in}}^o]}{E[N]}. \tag{37}$$

From (36) and (37), and using $E[N_{\text{out}}] = 1/p_{\text{out}}$, $E[N]$ is expressed by

$$E[N] = \frac{Q_u}{1-p_m}\, E[N_{\text{out}}] \tag{38}$$

$$= \frac{1}{(1-p_m)p_{\text{out}} + p_m p_{\text{in}}}. \tag{39}$$

Now, $p'_{\text{in}}$ is

$$p'_{\text{in}} = \frac{Q_u E[N_{\text{in}}^u] + Q_o E[N_{\text{in}}^o]}{Q_o E[N_{\text{in}}^o]}\, p_{\text{in}} \tag{40}$$

$$= \frac{E[N] - Q_u E[N_{\text{out}}]}{E[N] - Q_u(E[N_{\text{out}}] + E[N_{\text{in}}^u])}\, p_{\text{in}} \tag{41}$$

$$= \frac{p_m p_{\text{in}}}{1 - p_{\text{out}}(1-p_m)(E[N_{\text{out}}] + E[N_{\text{in}}^u])}. \tag{42}$$

Here, note that $(E[N_{\text{out}}] + E[N_{\text{in}}^u])$ is the total number of packets sent in an $A_u$. We can reasonably assume that $R > E[W]/2$ since both IN and OUT packets are dropped [as in Fig. 5(a)]. Therefore, from (18) and (20), $(E[N_{\text{out}}] + E[N_{\text{in}}^u])$ is calculated as follows:

$$\begin{aligned} E[N_{\text{out}}] &+ E[N_{\text{in}}^u] \\ &= \frac{3d}{8}\left(R + \sqrt{\frac{2}{dp_{\text{out}}}}\right)^2 + \frac{2-d}{4}\left(R + \sqrt{\frac{2}{dp_{\text{out}}}}\right) \\ &= \frac{3d}{8}R^2 + \left(3d\sqrt{\frac{2}{p_{\text{out}}}} - d + 2\right)\frac{R}{4} + \frac{3d}{4p_{\text{out}}} \\ &\quad + \frac{2-d}{4}\sqrt{\frac{2}{p_{\text{out}}}}. \end{aligned} \tag{43}$$

Note that $E[\beta]$ in (18) is given by $1/2E[W]$. Applying (42) to the model for an oversubscribed path, we can finally compute (33).

*4) Simulations:* In this section, we validate our models through simulations. First, we use a simple network topology which has one bottleneck link so that we can understand the results of the models intuitively. Then, we use complex topologies to show that the models are accurate under various network conditions. We also compare the results with the simple model based on (17) and (20) instead of (16) and (19). The simple models are useful to observe and explain throughput of a TCP flow intuitively.

In the simulations, we use Network Simulator version 2 (ns-2) [20]. Our diff-serv implementation is validated in [14], [15]. TCP-Reno sources are used as senders. Receivers do not employ delayed-ACK $(d = 1)$ for simplicity. For droppers, we use RIO presented in [5] with parameters (`minTh/maxTh/pmax`) 20/40/0.5 for OUT packets and 40/80/0.02 for IN packets. We use the general TSW marker with a one-second window. It is to be noted that the analytical models assumed an ideal marker. We ran each simulation for five minutes and collected statistics after one minute in order to avoid data from transient state. One minute would be enough

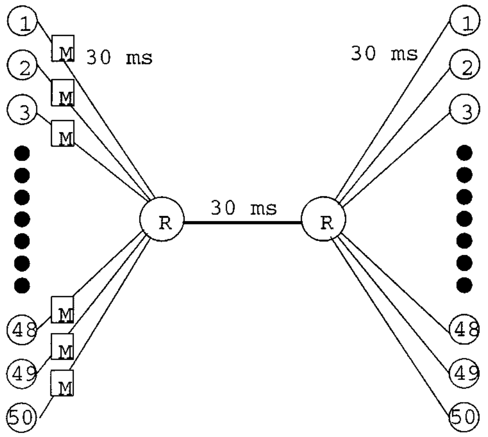


Fig. 6. Simple network topology.

TABLE I
LOSS RATES OF IN/OUT PACKET

| Bottleneck | 30 Mbps | | 20 Mbps | | 15 Mbps | |
|---|---|---|---|---|---|---|
| Loss rate | $p_{in}$ | $p_{out}$ | $p_{in}$ | $p_{out}$ | $p_{in}$ | $p_{out}$ |
| Max | 0 | 0.0645 | 0 | 0.3056 | 0.0089 | 1 |
| Min | 0 | 0.0366 | 0 | 0.1543 | 0 | 0.8971 |
| Mean | 0 | 0.0485 | 0 | 0.2262 | 0.0045 | 0.9485 |

for a flow to reach steady state since the RTT of a flow is set to less than one second in every simulation.

At first, we ran four simulations with a simple network topology shown in Fig. 6. In each simulation, we set the bottleneck link bandwidth to a different level so that the network experienced different subscription levels. The total contract rate of the flows is 25 Mb/s in each simulation. We set the bottleneck bandwidth to 40, 30, 20, and 15 Mb/s.[4] The contract rate of an individual flow is randomly picked from 0 to 1 Mb/s.

The loss rates of IN/OUT packets in each simulation are presented in Table I. "Max" and "Min" rows show the maximum and minimum loss rates among the 50 flows, respectively. "Mean" shows the average loss rate of the 50 flows. It is clearly observed that all the flows observe an undersubscribed path in the simulations with 30 and 20-Mb/s link bandwidth. It is to be noted that even when the total contract rate of 25 Mb/s exceeds the link bandwidth of 20 Mb/s, the packet losses indicate that we should apply the undersubscribed model in this situation. The OUT packet loss rates vary over a wide range from 1.49% to 30.56%. In the simulation with 15-Mb/s bottleneck link, a few IN packets are dropped, and some OUT packets are transmitted.

Fig. 7 compares the throughput observed in simulations with the estimated throughput from our models. In the figures, the square and the star indicate the simulated throughput and the estimated throughput, respectively. "×" is the estimation using the simple models. The dashed line shows $achieved\ rate = contract\ rate$.

In Fig. 7(b), we applied the undersubscribed model even though the bottleneck bandwidth is less than the total contract rate since no IN packet-drops were observed. It is clear that the model for an undersubscribed path estimates the individual throughput achieved by both IN and OUT packets quite accurately. It is observed that the estimation of the simple model is little higher than the simulations and the estimations of the complete model. With 15-Mb/s link capacity, we applied the combined model since there are both IN and OUT packet drops. It is shown that the combined model can estimate TCP throughput in a simple network topology very accurately.

Fig. 8 shows the relative error between the estimations and the simulations presented in Fig. 7. We here define the relative error as $|simulation\text{-}estimation|/simulation$. There are 200 samples from four simulations. It is observed that the relative errors of 93% of the samples are less than 10%, and that the maximum error is less than 25%. From Figs. 7 and 8, we confirm that the models are very accurate in a single bottleneck link network.

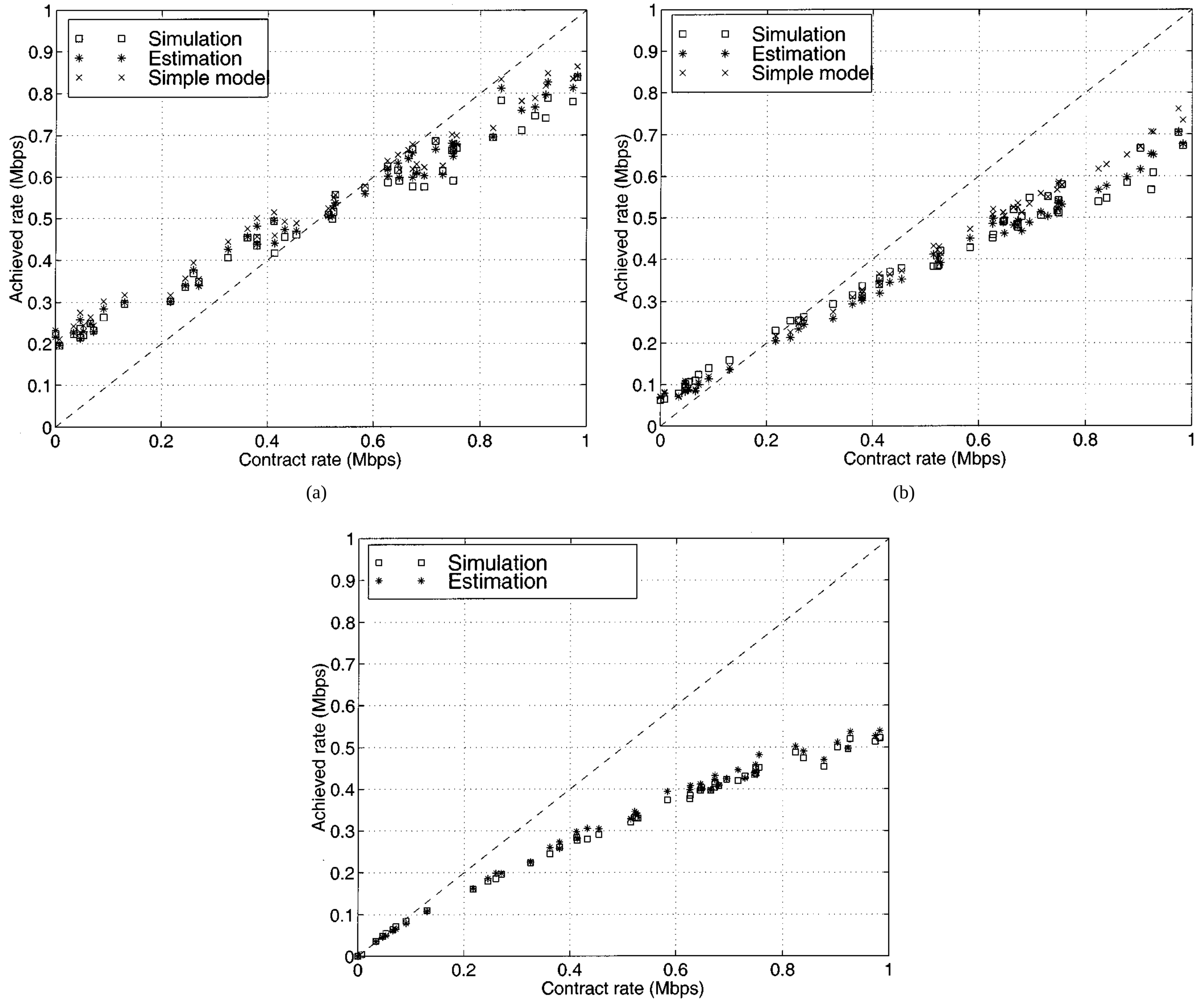
To observe TCP throughput with cross traffic, we conducted a simulation with a network topology as shown in Fig. 9(a). In this experiment, link delay and capacity of each link are set to 10 ms and 30 Mb, respectively. The contract rates of 30 TCP flows are randomly picked from 0 to 1 Mb/s, and the total contract rates add up to 15 Mb/s. At each switch, cross traffic consists of five exponential on/off sources with different on/off periods. The average sending rate of each source is 2 Mb/s, and thus the sending rate of the cross traffic at each switch is 10 Mb/s. Each cross traffic stream is injected into $R_i$ and leaves at $R_{i+1}$. Fig. 9(b) shows the results. This simulation shows that our model can be used to estimate throughput of a flow going through multiple links.

We conducted two other simulations with a merged topology and a split topology shown in Figs. 10(a) and 11(a). In the merged topology, flows are merged three times, and it is expected that flows will experience different levels of congestion. In the split topology, flows 1 to 10 and 11 to 20 reach $R4$ and $R5$, respectively, through $R1$ and $R3$, and flows 21 to 30 and 31 to 40 reach $R4$ and $R5$ through $R2$ and $R3$. Different capacity and delay characteristics are assigned to each link. The link capacity and the delay of each link are shown in the figures. The contract rate of each flow is set to 1 Mb/s in both the simulations. Figs. 10(b) and 11(b) show the results. In these simulations, both IN and OUT packets are dropped. Hence, we apply the combined model. It is shown again that the model keeps track of the different achieved rates under different RTTs and different link capacities.

The simulations presented in this section show that 1) The models based on (16), (19), and (33) are quite accurate in estimating individual TCP throughputs in diff-serv networks. 2) The simple models based on (17) and (20) are also accurately estimating TCP throughput. 3) The models work in multiple-link networks, merged and split network topologies with different levels of congestion. In the rest of this paper, we use the simple models instead of the original models since the simple models are easy to understand and give intuitive insight of TCP throughput in diff-serv networks.

*5) Throughput Analysis:* In this section, we discuss the excess bandwidth $B_e$ which is defined as the difference between realized throughput and its contract rate. In this discussion, we focus on the flows observing undersubscribed paths and use the

[4] In this paper, we present the results with 30, 20, and 15-Mb/s links due to space limitation. Please refer our longer tech. report [16].



(a)

(b)

(c)

Fig. 7. Throughputs of TCP-Reno flows with different reservation rate. (a) 30-Mb/s bottleneck link. (b) 20-Mb/s bottleneck link. (c) 15-Mb/s bottleneck link.

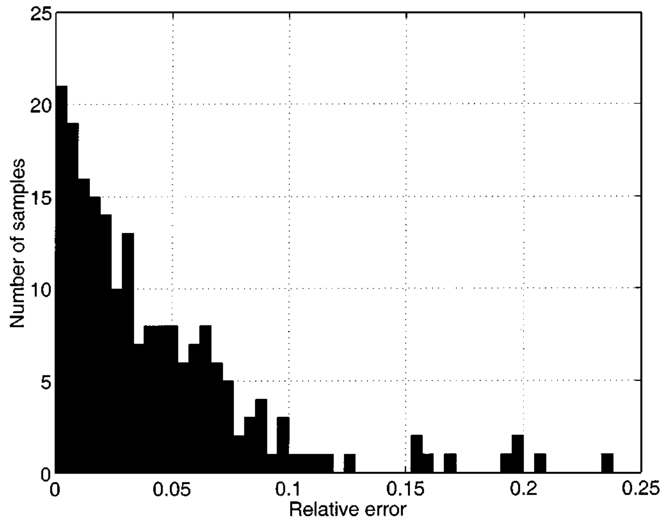


Fig. 8. Relative error between simulation and estimation.

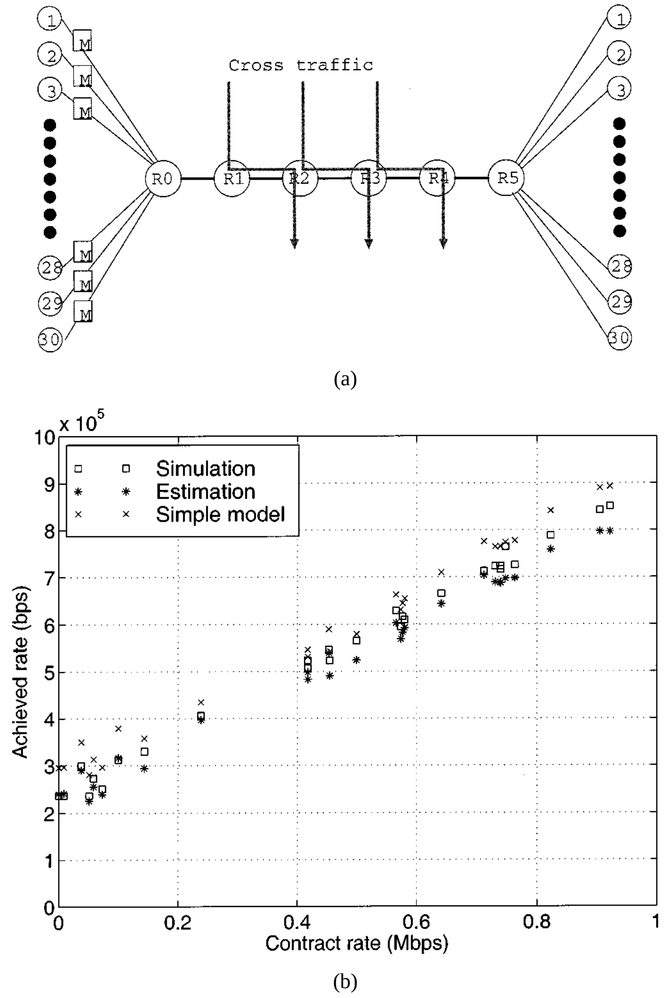
simple models based on (17) and (20). The contract rate is expressed by $(k \times R/\mathrm{RTT})$ from (1). Then, the excess bandwidth is

$$B_e = \begin{cases} \dfrac{k}{4\mathrm{RTT}}\left(3\sqrt{\dfrac{2}{p_{\mathrm{out}}}} - R\right) & \text{if } R \geq \sqrt{2/p_{\mathrm{out}}} \\ \dfrac{k}{2\mathrm{RTT}}\left(\sqrt{R^2 + \dfrac{6}{p_{\mathrm{out}}}} - R\right) & \text{otherwise.} \end{cases} \tag{44}$$

If $B_e$ of a flow is positive, that means the flow obtains more than its contract rate. Otherwise, it does not reach its contract rate. We present Fig. 12 to illustrate the following observations.

From (44) and Fig. 12, we can observe that:

1) When a flow reserves relatively higher bandwidth ($R \geq \sqrt{2/p_{\mathrm{out}}}$), $B_e$ is decreased as the reservation rate is increased. Moreover, if $R$ is greater than $3\sqrt{2/p_{\mathrm{out}}}$ (see line C in Fig. 12), the flow cannot reach its reservation rate.

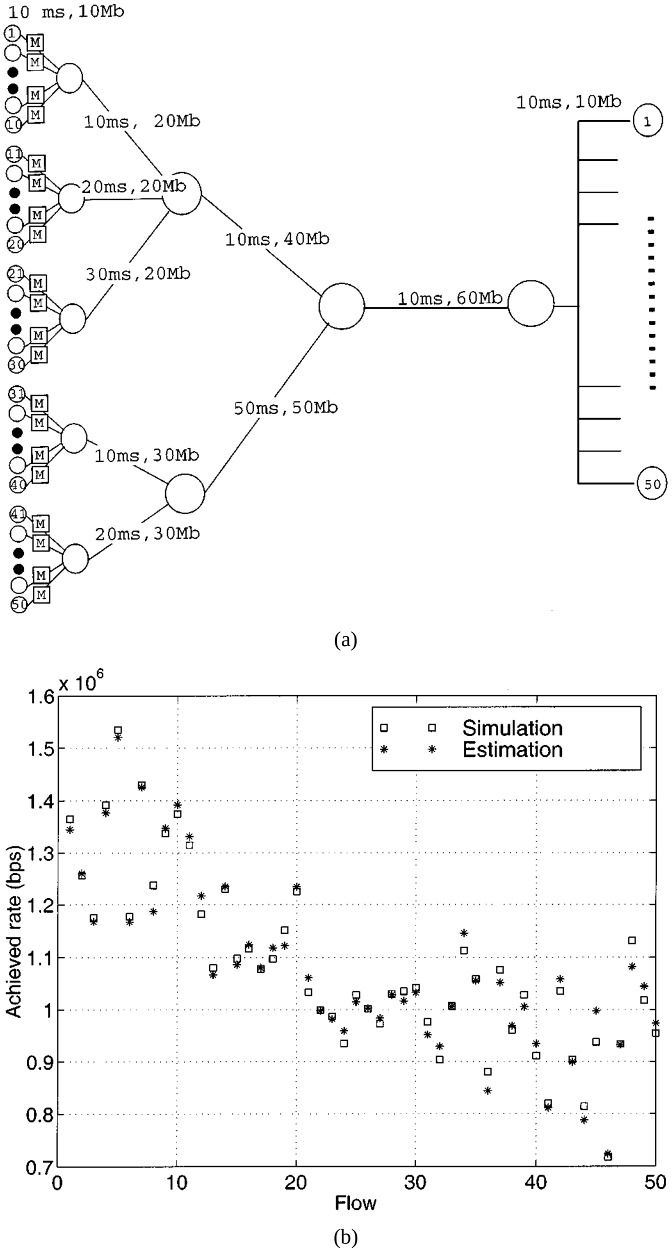


(a)

(b)

Fig. 9. Simulation with cross traffic. (a) Simulation topology. (b) Throughputs.

2) When a flow reserves relatively lower bandwidth ($R < \sqrt{2/p_{\text{out}}}$, see line B in Fig. 12), it always realizes at least its reservation rate. As it reserves less bandwidth, it obtains more excess bandwidth. These observations correspond to the simulated results in [2], [14]. TCP's multiplicative decrease of sending rate after observing a packet drop results in a higher loss of bandwidth for flows with higher reservations. This explains the observed behavior.
3) The above equation also shows that as the probability of OUT packet drop decreases, the flows with smaller reservation benefit more than the flows with larger reservations. This again validates the difficulty in providing service differentiation between flows of different reservations observed in [2], [14] when there is plenty of excess bandwidth in the network.
4) The realized bandwidth is observed to be inversely related to the RTT of the flow.
5) For best-effort flows, $R = 0$. Hence, $B_e$ ($= k\sqrt{6/p_{\text{out}}}/2\text{RTT}$, see line D in Fig. 12) gives the bandwidth likely to be realized by flows with no reservation.
6) Comparing the above best-effort bandwidth and when $R \geq \sqrt{2/p_{\text{out}}}$, we realize that the reservation rates larger than 3.5 times the best-effort bandwidth cannot be met.



(a)

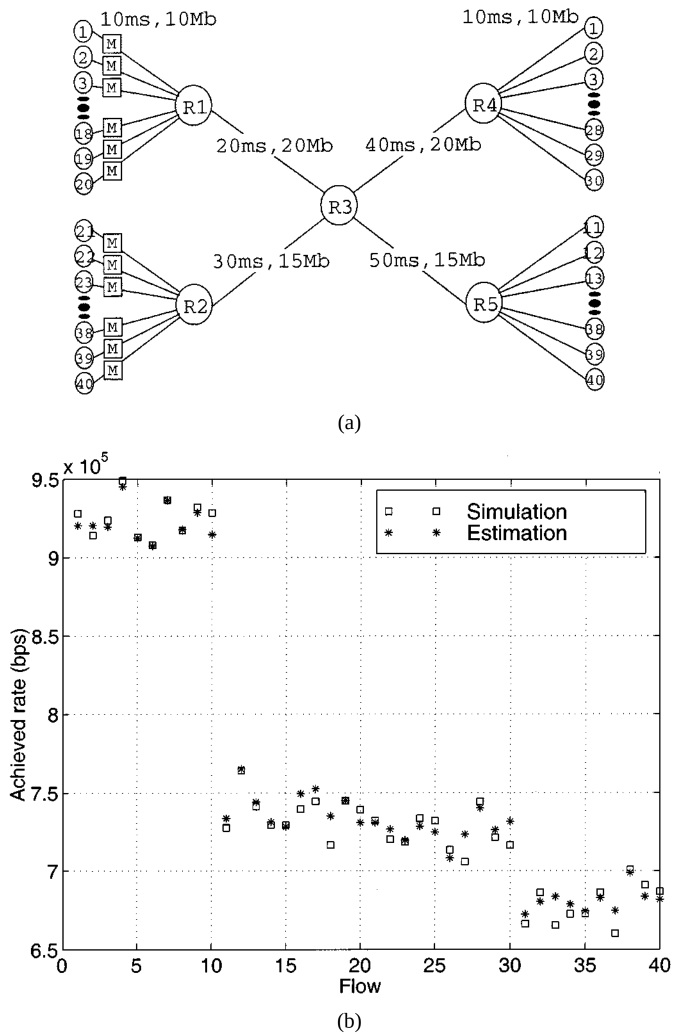
(b)

Fig. 10. Simulation with merged network topology. (a) Simulation topology. (b) Throughputs.

7) Equation (44) clearly shows that excess bandwidth cannot be equally shared by flows with different reservations (a goal of recent simulation studies [2], [4], [14]) without any enhancements to basic RIO scheme or to TCP's congestion avoidance mechanism.

### C. Modeling for Three-Drop Precedence

In this section, we extend the models to the three-drop precedence policy. In a three-drop precedence network, there may be three possible subscription levels: 1) only *red* packets are dropped [Fig. 13(a)]; 2) every *red* packet is dropped, and some of *yellow* [Fig. 13(b)] packets are dropped; and 3) every *red* and *yellow* packet is dropped, and some *green* packets are dropped [Fig. 13(c)]. To develop the models, we define $p_{\text{color}}$ and $R_{\text{color}}$



(a)

(b)

Fig. 11. Simulation with split network topology. (a) Simulation topology. (b) Throughputs.

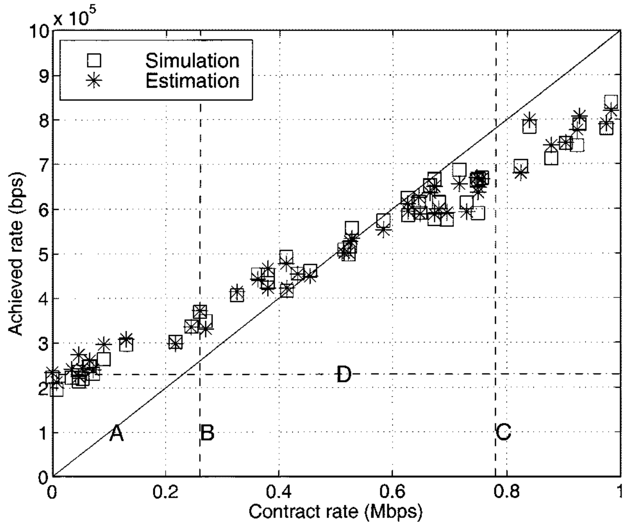


Fig. 12. Observations from the model.

as the drop probability and the reservation window of packets marked as that $\mathrm{color}(r,\, y,\, g)$, respectively.

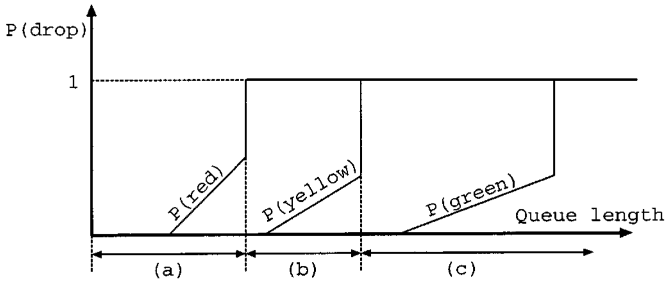


Fig. 13. Drop probabilities of three-drop precedence.

1) When $p_r > 0$ and $p_y,\, p_g = 0$.

This case is the same as the undersubscribed model of two-drop precedence. A *red* packet is treated as an OUT packet and both *yellow* and *green* packets are treated as an IN packets. Therefore, we can directly use (16), (19), (23) and (24) replacing $p_{\text{out}}$ and $R$ to $p_r$ and $R_y$, respectively.

2) When $p_r = 1,\, 1 > p_y \geq 0$ and $p_g = 0$.

This case is oversubscribed for *yellow* packets and undersubscribed for *green* packets. A *red* packet is treated as an OUT packet, and a *green* packet is treated as an IN packet. A *yellow* packet is considered an IN packet compared to a *red* packet and considered as an OUT packet compared to a *green* packet. Therefore, we combine the undersubscribed model and the oversubscribed model. As the undersubscribed model, we have

$$E[W] = \begin{cases} R_g + \sqrt{\dfrac{2}{dp_y}} & \text{if } R_g > E[W]/2 \\ \dfrac{2}{3}\left(R_g + \sqrt{R_g^2 + \dfrac{6}{dp_y}}\right) & \text{otherwise.} \end{cases} \tag{45}$$

As the oversubscribed model, $W_i$ is limited by $R_y$. Thus, we also have

$$E[W] \leq R_y. \tag{46}$$

Therefore, finally, $E[W]$ is expressed by

$$E[W] = \begin{cases} \min\left\{R_g + \sqrt{\dfrac{2}{dp_y}},\, R_y\right\} & \\ & \text{if } R_g > \dfrac{E[W]}{2} \\ \min\left\{\dfrac{2}{3}\left(R_g + \sqrt{R_g^2 + \dfrac{6}{dp_y}}\right),\, R_y\right\} & \\ & \text{otherwise.} \end{cases} \tag{47}$$

Now, with (47), we can use (23) and (24) after replacing $p_{\text{out}}$ and $R$ to $p_y$ and $R_{\text{green}}$.

3) When $p_r,\, p_y = 1$, and $p_g > 0$.

This case is the same as the case in which every OUT packet is dropped in the two-drop precedence. So, (30) and (31) are applied by changing $p_{\text{in}}$ and $R$ into $p_g$ and $R_g$, respectively.

*1) Simulations:* The models derived in the previous section have been extended directly from our two-drop precedence model. It has been already shown that the model can estimate individual TCP throughput in diff-serv network environment

TABLE II
LOSS RATES OF $green/yellow/red$ PACKET

| Bottleneck | 40 Mbps | | 20 Mbps | | | 10 Mbps | |
|---|---|---|---|---|---|---|---|
| Loss rate | $p_g$,$p_y$ | $p_r$ | $p_g$ | $p_y$ | $p_r$ | $p_g$ | $p_y$ |
| Max | 0 | 0.096 | 0 | 0.140 | 1 | 0.016 | 1 |
| Min | 0 | 0.050 | 0 | 0.087 | 0.937 | 0 | 0.915 |
| Mean | 0 | 0.074 | 0 | 0.119 | 0.964 | 0.006 | 0.958 |

through a number of simulations. In this section, we show that the models can be extended to estimate TCP throughput when droppers employ three-drop precedence through a simple network topology.

In the simulations, we used the same topology presented in Fig. 6 except three-drop precedence is employed with parameters 20/40/0.5 for the *red* packets, 40/60/0.1 for the *yellow* packets and 60/80/0.02 for the *green* packets. Three simulations with different bottleneck bandwidths (40, 20, and 10 Mb/s each) were conducted to reflect the three situations described in the previous section. In each simulation, the contract rate for *yellow* of each flow is randomly picked from 0 to 1 Mb/s, and the contract rate for *green* is set to a half of the contract rate for *yellow* of that flow to follow [17] which recommends that the contract rate for *green* should be set less than the contract rate for *yellow*. The total reservation rate for *yellow* is set to 25 Mb/s, and the total reservation rate for *green* is set to 12.5 Mb/s. Each simulation ran for five minutes, and we collected statistics in the last four minutes.

Table II shows the loss rate of $green/yellow/red$ packets in each simulation. N/A for $p_{\text{color}}$ in the table means that no packet is marked as that color at all. The table shows that the simulations reflect the three different cases effectively. Individual throughputs from the simulations and the models are presented in Fig. 14. From Fig. 14(a), it is shown that the simple model for an undersubscribed path can estimate the throughput accurately in a three-drop precedence network. Fig. 14(b) and (c) also show that our models for an oversubscribed path can be applied in a three-drop precedence network.

## IV. THROUGHPUT MODEL FOR AGGREGATED FLOWS

The diff-serv architecture is being proposed for aggregate reservation. So far, our models have assumed that individual flows can reserve bandwidth. In this section, we develop a throughput model for an individual flow within an aggregated reservation.

### A. Packet Marking of Aggregated Flows

In Section III-A, we have described packet marking behavior of an individual TCP flow with the TSW rate estimator. We now discuss the marking behavior when the same marker marks TCP packets from an aggregation. Fig. 15 shows the conceptual model which we look at in this section. The figure shows $n$ individual flows being marked by a single aggregate marker.

When several TCP flows are aggregated, the impact of an individual TCP sawtooth behavior is reduced, and the aggregated sending rate is stabilized (even though there still exist some variations) as illustrated in Fig. 15. If the marker does neither maintain per-flow state nor employ other specific methods for distinguishing individual flows, an arriving packet is marked IN with the probability $(contract\_rate/aggregated\_sending\_rate)$.[5] Here we define $p_m$, the probability of a packet marked IN as

$$p_m = \frac{contract\_rate}{aggregated\_sending\_rate}. \tag{48}$$

In the steady state, $p_m$ is approximately equal for all the individual flows. A flow sending more packets then gets more IN packets, and consequently, the contract rate consumed by individual flows is roughly proportional to their sending rates.[6] We call this marking behavior *proportional marking*. It is noted that this proportional marking behavior is a direct result of not maintaining any flow state, and not dependent on the actual marking algorithm. In the following section, we propose a throughput model for individual TCP flows sharing a contract rate with a general TSW marker (showing proportional marking behavior), which does not maintain per-flow state nor employ any sophisticated mechanism for aggregated flows.

### B. Modeling for Throughput of Aggregated Flows

In this section, we develop a simple model for an individual flow in an aggregation in a two-drop precedence network. In the model, we assume that all the packets of aggregated flows are of the same size, $k$, a receiver does not employ delayed-ACK ($d = 1$) and the network is not oversubscribed. We define $r_A$ as the reservation rate which is contracted for the aggregation and $r_i$ as the marking rate achieved by the $i$th individual flow.

$$r_A = \sum_{i=1}^{n} r_i \tag{49}$$

where $n$ is the number of flows.

For simplicity, we consider that there is no time-out and $r_i$ is greater than $\sqrt{2/p_{\text{out}}}$. Then, from (1) and the simple model presented in (20) and (22), the throughput of the $i$th flow is given by

$$B_i = \frac{3k}{4\text{RTT}_i}\left(\frac{\text{RTT}_i}{k} r_i + \sqrt{\frac{2}{p_i}}\right) = \frac{3}{4} r_i + \frac{3k}{4\text{RTT}_i}\sqrt{\frac{2}{p_i}} \tag{50}$$

where $p_i$ is the OUT packet drop probability of the $i$th flow. Then, aggregated throughput $B_A$ is

$$B_A = \sum_{i=1}^{n} B_i = \sum_{i=1}^{n}\left\{\frac{3}{4} r_i + \frac{3k}{4\text{RTT}_i}\sqrt{\frac{2}{p_i}}\right\} \tag{51}$$

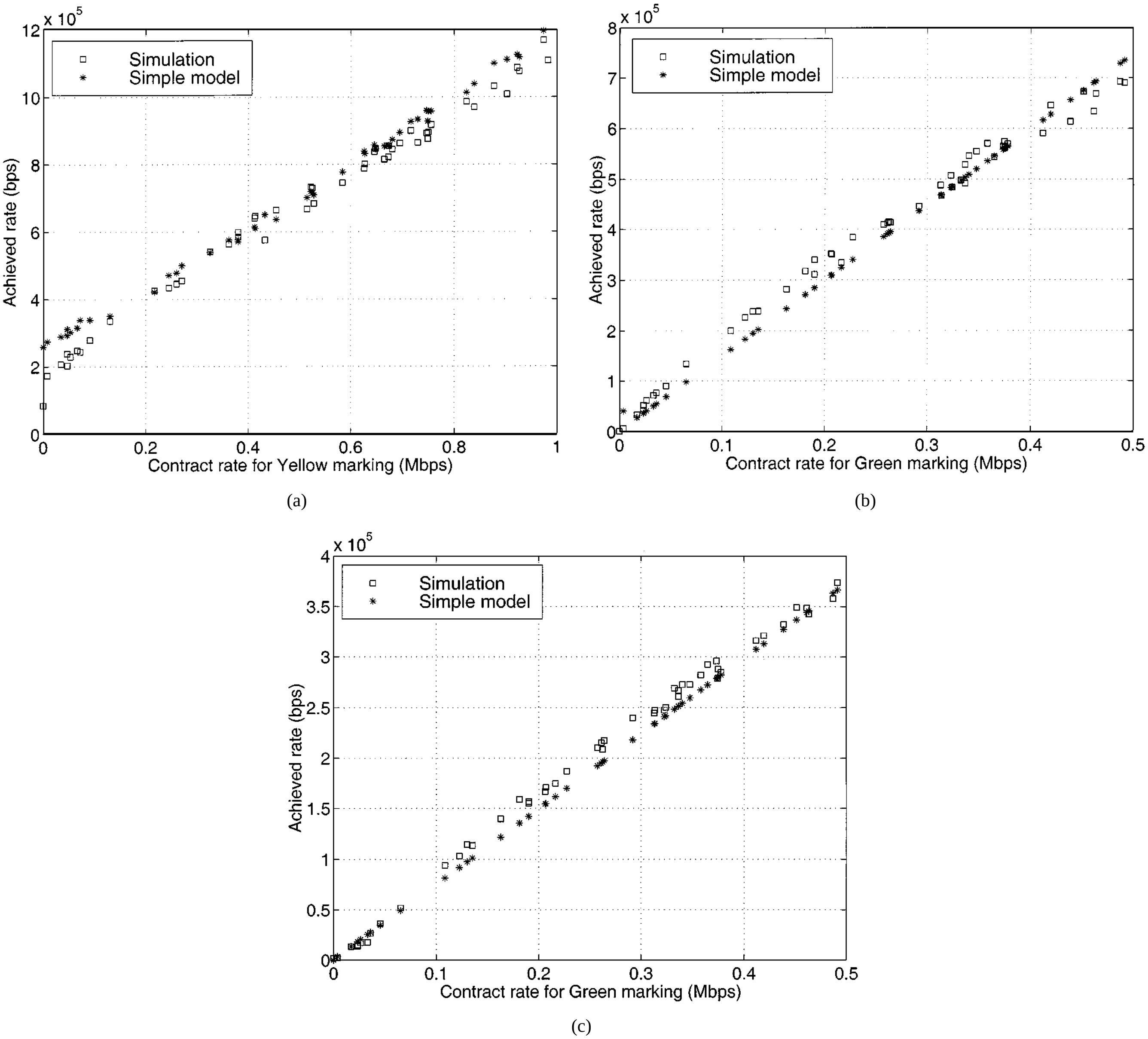
$$= \frac{3}{4} r_A + \frac{3k}{4}\sum_{i=1}^{n}\frac{1}{\text{RTT}_i}\sqrt{\frac{2}{p_i}}. \tag{52}$$

When a marker exhibits proportional marking, $r_i$ is roughly linearly proportional to $B_i$.

$$r_i = p_m \cdot B_i \tag{53}$$

[5] When the aggregated sending rate is less than the contract rate, every packet is marked IN.

[6] This behavior is different from the marking of individual flows in Section III. In the marking of individual flows, the contract rate for an individual flow is fixed. In the marking of aggregated flows, however, the contract rate consumed by an individual flow is not fixed even though the aggregated contract rate is fixed.



(a)

(b)

(c)

Fig. 14. Throughputs of TCP-Reno flows with different reservation rate in three-drop precedence network. (a) $p_{red} > 0$, (b) $p_{red} = 1, p_{yellow} > 0$, and (c) $p_{red}, p_{yellow} = 1, p_{green} > 0$.

where $p_m$ gives the probability that a flow's packet is marked IN. Then, (50) is rewritten by

$$B_i = \frac{3p_m}{4} B_i + \frac{3k}{4\mathrm{RTT}_i}\sqrt{\frac{2}{p_i}} \tag{54}$$

$$= \frac{3k}{4\mathrm{RTT}_i - 3p_m\mathrm{RTT}_i}\sqrt{\frac{2}{p_i}}. \tag{55}$$

From (53), $r_A = p_m \cdot B_A$ and therefore

$$p_m = \frac{r_A}{B_A} = \frac{r_A}{\frac{3}{4}r_A + \frac{3k}{4}\sum_{i=1}^{n}\frac{1}{\mathrm{RTT}_i}\sqrt{\frac{2}{p_i}}}. \tag{56}$$

Substituting $p_m$ with (56) and after some manipulations, we have

$$B_i = \frac{m_i}{\sum_{j=1}^{n} m_j} \cdot \frac{3r_A}{4} + \frac{3k}{4} m_i \tag{57}$$

where $m_i$ is given by

$$m_i = \frac{1}{\mathrm{RTT}_i}\sqrt{\frac{2}{p_i}}. \tag{58}$$

Equation (57) relates the realized bandwidth of an individual flow to the aggregate reservation $r_A$ and the network conditions ($\mathrm{RTT}_i$ and $p_i$) observed by various flows within the aggregation.

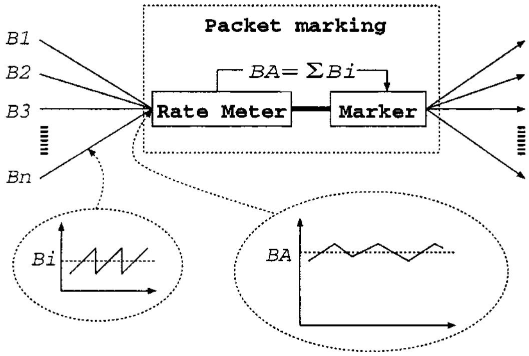


Fig. 15. Packet marking of aggregated flows.

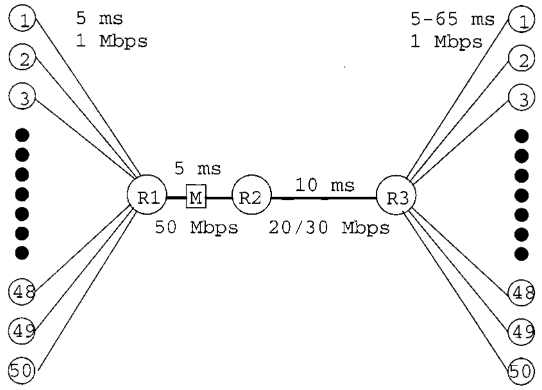


Fig. 16. Aggregated network topology with different RTTs.

From (52), $B_e$ (the excess bandwidth) of aggregated flows is calculated as follows,

$$B_e = \tfrac{3}{4}r_A + B_s - r_A = B_s - \tfrac{1}{4}r_A \tag{59}$$

where $B_s = (3k/4)\sum_{i=1}^{n}(1/\mathrm{RTT}_i)\sqrt{2/p_i}$, and it is approximately the throughput which the aggregated flows can achieve with zero contract rate ($r_A = 0$). Based on the above analysis, the following observations can be made:

1) The total throughput realized by an aggregation is impacted by the contract rate. Larger the contract rate, the smaller the excess bandwidth claimed by the aggregation [referring to (59)].
2) When the contract rate is larger than 4 times $B_s$, the realized throughput is smaller than the contract rate [referring to (59)].
3) The realized throughput of a flow is impacted by the other flows in the aggregation (as a result of the impact on $p_m$) when proportional marking is employed [referring to (56) and (57)].

### C. Simulations

In this section, we present simulations to validate the model for aggregated flows. To observe behaviors of individual flows within a large aggregation, we conducted two simulations with an aggregation of 50 individual flows. The simulation topology is shown in Fig. 16. The aggregate reservation rate is 10 Mb/s, and the bottleneck capacity is set to 20 or 30 Mb/s so that the bottleneck is not oversubscribed. The RTT without queueing delay of each flow is randomly picked from 50 to 160 ms.

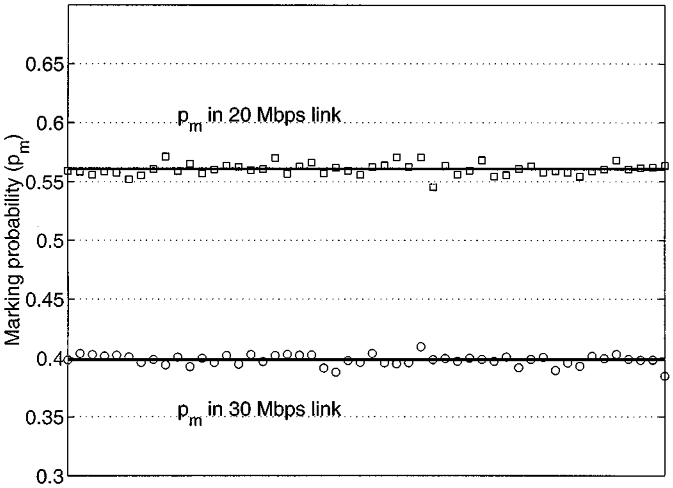


Fig. 17. The probability of a packet marked IN.

To confirm the assumption that $p_m$ is the same for all the flows within an aggregation, we present Fig. 17. The solid lines represent $p_m$ of aggregations in both simulations, which is the contract rate divided by average aggregated sending rate. The square and the circle show $p_m$ of individual flows, which is measured by the number of IN packets divided by the number of IN and OUT packets sent by that flow. It is observed that $p_m$ measured in simulation with 20 Mb/s is higher than $p_m$ measured with 30 Mb/s since more OUT packets were sent in the simulation with 30 Mb/s while the number of IN packets remained the same. It is clearly shown that individual $p_m$ is approximately equal to aggregated $p_m$, and we confirm that individual flows within an aggregation have roughly the same $p_m$.

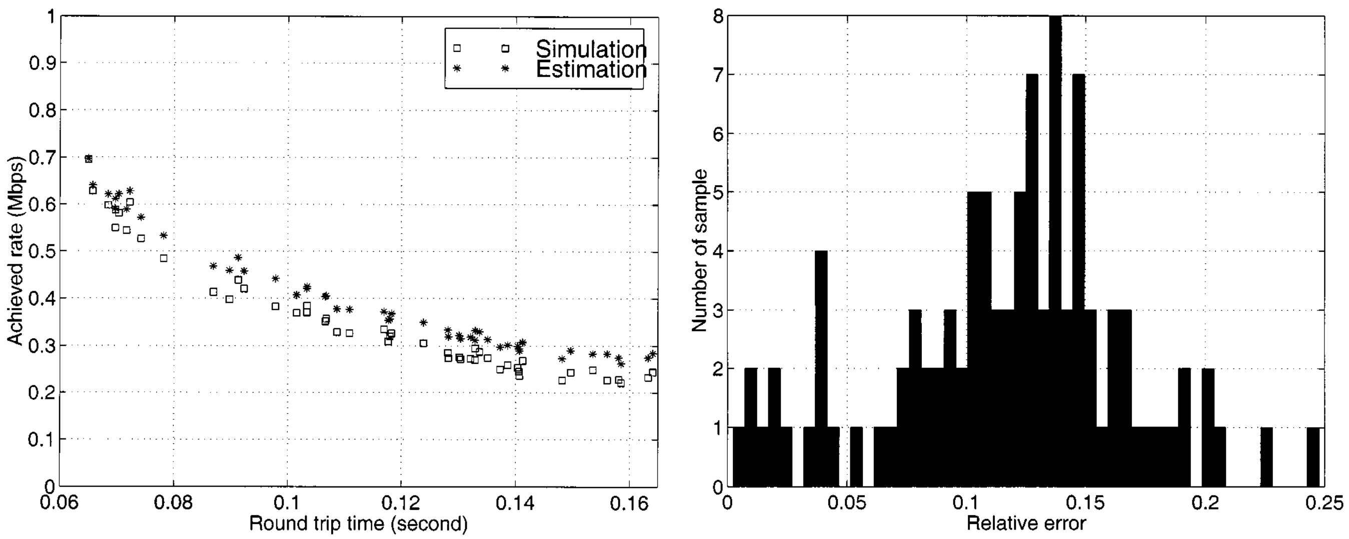
Fig. 18(a) shows the simulated and estimated throughputs with 20-Mb/s link. We also present relative errors in Fig. 18(b). It is clearly shown that the model can estimate the individual throughput of aggregated flows very accurately (maximum error is less than 25%). It is observed that the estimated throughputs are slightly higher than the simulation results.

We present another simulation to show how the model works in a complicated network topology. Fig. 19 shows the network topology used in the simulations. There are five markers and aggregated sources, each aggregated source consists of ten individual sources. Bandwidth of every link except the link between the two routers is 10 Mb/s, and bandwidth of the link between two routers is limited to 8 Mb/s. With this topology, we conducted two simulations. In the first simulation, each aggregated source reserves 1 Mb/s. We assign $\mathrm{RTT}_j^i$, RTT of the $i$th individual source in the $j$th aggregated source excluding queueing delay as

$$\mathrm{RTT}_j^i = 130 + 4 \times (i-1) \times (j-5.5)\ \text{(ms)}. \tag{60}$$

This results in five aggregated sources with varying differences in RTTs. For example, the aggregated source 1 has a (min RTT, max RTT) = (130 ms, 130 ms) compared to that of the aggregated source 5 with (58 ms, 202 ms). In the second simulation, the RTT distribution of each aggregation is the same as each other and given by

$$\mathrm{RTT}_j^i = 130 + 16 \times (i-5.5)\ \text{(ms)}. \tag{61}$$



(a) (b)

Fig. 18. Simulations with aggregated flows with different RTTs. (a) Bottle neck link = 20 Mb/s. (b) Relative error for aggregated model.

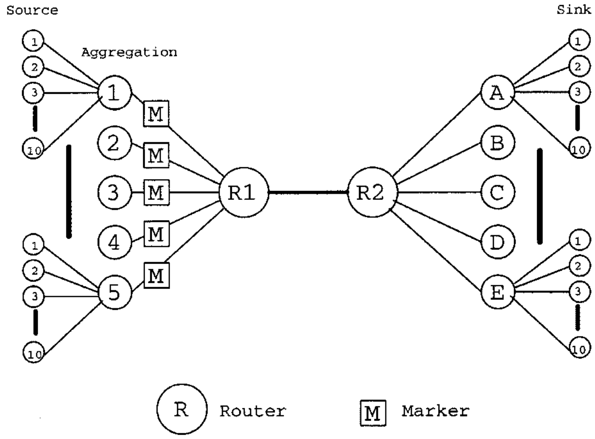
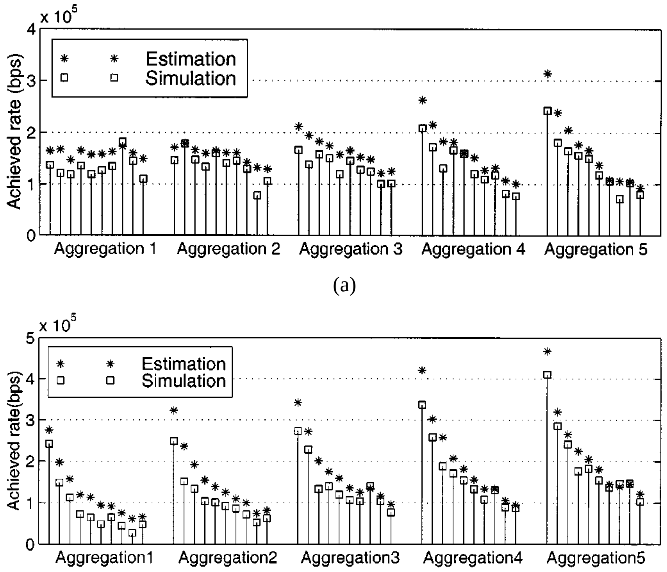


Fig. 19. Aggregated network topology with multiple aggregations.



(a)

(b)

Fig. 20. Dealing with multiple aggregations. (a) Different RTT distributions. (b) Different aggregate reservations.

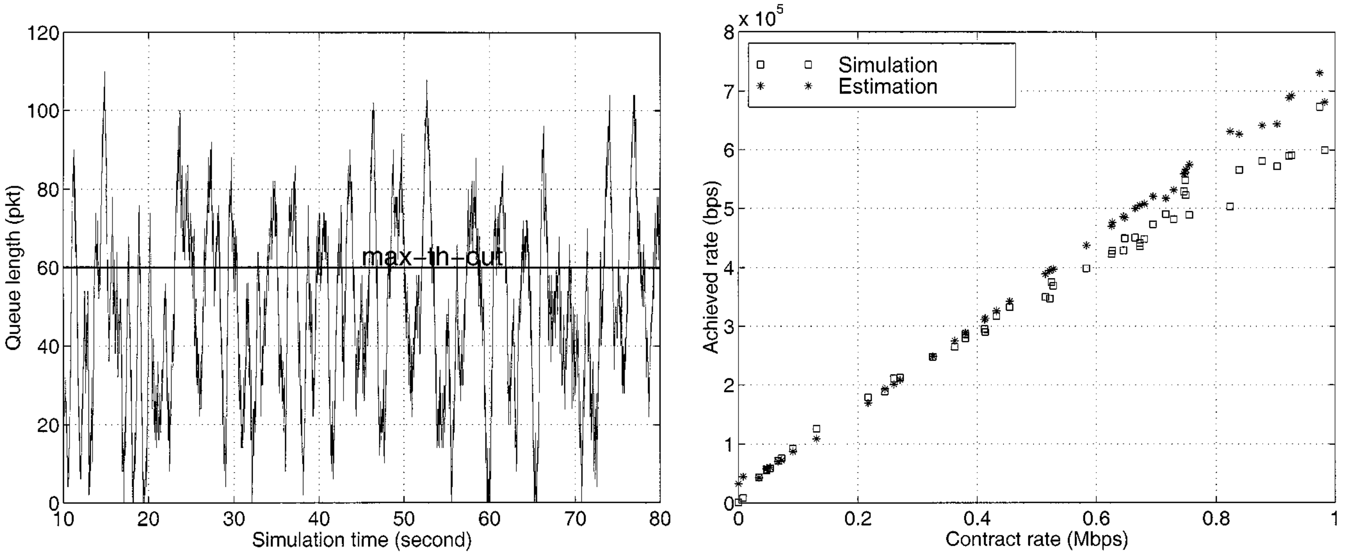
The reservation rate of the $j$th aggregation is given by $0.3 \times j$ Mb/s.

Fig. 20 shows the results. Fig. 20(a) shows the results of the first simulation with different RTTs and Fig. 20(b) shows the results of the second simulation with different aggregate reservations. It is observed that individual flows achieve different bandwidths within an aggregation. These differences increase with increased differences in the RTTs of the individual flows. It is shown that the model can estimate the individual throughputs in the network with several aggregated sources.

## V. Discussion and Related Work

While developing the models, we assumed that the packets are dropped randomly. This assumption is valid in undersubscribed networks since a packet is randomly picked to be discarded in a router when queue length is less than max_th. However, when the queue length builds up, OUT packets may be dropped in a burst. This could be a potential source of error in our models. This will result in an inaccurate estimation of time-outs. However, operational dynamics help from this becoming a significant source of error since 1) a flow through a congested path may not send too many OUT packets, and 2) OUT packets are not sent out in a burst. When all the OUT packets are being dropped, a flow can rarely increase its window to such an extent that it sends out significant number of OUT packets within a window. It is possible to enhance the presented model to consider this correlation to further improve the accuracy.

To understand the possible inaccuracy due to correlated losses, we ran a simulation (with simulation topology in Fig. 6 and 17-Mb/s bottleneck link capacity) where the queue length oscillated around max_th_out. The queue length and the throughput comparison of the model with simulation are shown in Fig. 21. Even in this situation where packet losses are forced



(a)

(b)

Fig. 21. Simulation with correlated packet losses. (a) Queue length. (b) Throughputs.

to be correlated, the model is within 20%–30% of the actual throughput. Deriving a more accurate model that considers this correlation is part of our future work.

After the congestion avoidance scheme for TCP was proposed in [19], several models of TCP behavior have been proposed in [7], [8], [11], [12], [13]. Each model estimates the steady state throughput of an individual TCP flow in terms of RTT and the probability of a packet-drop. In [12], a performance model for a TCP flow was proposed with the assumption that the TCP source avoids retransmission time-outs and has a sufficient receiver window. In [13], Floyd *et al.* derived a similar model to the model in [12]. Padhye *et al.* proposed a new TCP throughput model in [7], [8]. The model characterizes not only the behavior of fast retransmission mechanism but also the effect of TCP's timeout mechanism. Recently, it has been shown that some simplified assumptions and mathematical simplifications of the model in [7] result in errors in estimating individual flow throughputs [23].

A number of studies on quantitative analysis of differentiated services have been recently presented [9], [21], [22]. May *et al.* [22] presented models of packet behavior at a switch as a function of load within a differentiated services network. In [21], Dovrolis *et al.* addressed the issues of packet schedulers for differentiated services. They described the impact of scheduling schemes including weighted fair queueing (WFQ) and early deadline first (EDF) and proposed new scheduling schemes for differentiated services. Sahu *et al.* [9] characterized packet behaviors (delay and loss) of TCP flows through Markov analysis. In this study, a stochastic Markov model is used for deriving a model for TCP throughput in a differentiated services network. Our modeling provides a simpler and a more intuitive characterization of the TCP behavior in differentiated services networks. Our model has been recently extended to token-bucket marking in [10]. We have extended our model to consider two-window TCP in [16].

## VI. Conclusion

In this paper, we have derived throughput models of individual TCP flows in a differentiated services network. Our models estimate throughput in terms of RTT, packet drop rates and the reservation rate of a TCP flow. Our models are developed for various conditions in a differentiated services network including two-drop precedence, three-drop precedence and aggregated marking. We presented a number of simulations to validate our models. The simulation results show that the models can predict the throughput of individual and aggregated TCP flows quite accurately in various conditions.

Our model makes the following observations: 1) flows with larger contract rates are at a relative disadvantage compared to flows with smaller contract rates; 2) throughputs achieved by flows with larger contract rates may not reach their contract rates due to TCP's sawtooth behavior; 3) the contract rate shared by aggregated flows is consumed unfairly within the aggregation; and 4) TCP's throughput is impacted by RTT even in a diff-serv network.

## Acknowledgment

The comments and input from the referees and the editor have greatly contributed to improvement in the presentation of this paper.

## References

[1] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An architecture for differentiated services," Network Working Group, RFC2475, Dec. 1998.

[2] J. Ibanez and K. Nichols. (1998, Aug.) Preliminary simulation evaluation of an assured service. [Online]. Available: http://ecwww.eurecom.fr/~Ibanez/draft-Ibanez-diffserv-assured-eval-00.txt

[3] V. Jacobson, K. Nichols, and K. Poduri, "An expedited forwarding PHB," Network Working Group, RFC2598, June 1999.

[4] A. Basu and Z. Wang, "A comparative study of schemes for differentiated services,", Bell Labs Tech. Rep., Aug. 1998.

[5] D. D. Clark and W. Fang, "Explicit allocation of best-effort packet delivery service," *IEEE/ACM Trans. Networking*, vol. 6, pp. 362–373, Aug. 1998.

[6] W. Feng, D. D. Kandlur, D. Saha, and K. G. Shin, "Adaptive packet marking for providing differentiated services in the Internet," in *Proc. 6th Int. Conf. Network Protocols*, Oct. 1998, pp. 108–117.

[7] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP throughput: A simple model and its empirical validation," in *Proc. ACM SIGCOMM*, 1998, pp. 303–314.

[8] J. Padhye, V. Firoiu, and D. Towsley, "A stochastic model of TCP Reno congestion avoidance and control," Univ. of Massachusetts, Amherst, MA, CMPSCI Tech. Rep. 99-02, 1999.

[9] S. Sahu, D. Towsley, and J. Kurose, "A quantitative study of differentiated services for the Internet," in *Proc. Global Internet*, Dec. 1999, pp. 1808–1817.

[10] 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*, 2000, pp. 23–33.

[11] K. Fall and S. Floyd, "Simulation-based comparisons of Tahoe, Reno, and SACK TCP," *Comput. Commun. Rev.*, pp. 5–21, July 1996.

[12] M. Mathis, J. Semke, J. Mahdavi, and T. Ott, "The macroscopic behavior of the TCP congestion avoidance algorithm," *Comput. Commun. Rev.*, vol. 27, no. 3, pp. 67–82, July 1997.

[13] S. Floyd and K. Fall. (1998, Feb.) Promoting the use of end-to-end congestion control in the Internet. [Online]. Available: http://www-nrg.ee.lbl.gov/floyd/end2end-paper.html

[14] I. Yeom and A. L. N. Reddy, "Realizing throughput guarantees in a differentiated services network," in *Proc. ICMCS*, June 1999, pp. 372–376.

[15] ——, "Impact of marking strategy on aggregated flows in a differentiated services network," in *Proc. IWQoS*, May 1999.

[16] ——, "Modeling TCP behavior in a differentiated services network," Texas A&M Univ., College Station, TX, Tech. Rep..

[17] J. Heinanen, T. Finland, and R. Guerin. (1999, Feb.) A three-color marker. [Online]. Available: ftp://ftp.isi.edu/in-notes/rfc2697.txt

[18] ——, "A two-rate three-color marker,", draft-Heinanen-diffserv-trtcm-00.txt, Internet Draft, Mar. 1999.

[19] V. Jacobson and M. J. Karels, "Congestion avoidance and control," in *Proc. SIGCOMM*, Stanford, CA, Aug. 1998.

[20] Univ. of California at Berkeley. (1997) Network Simulator v.2 (ns-2). [Online]. Available: http://www-nrg.ee.lbl.gov/ns-2

[21] C. Dovrolis, D. Stiliadis, and P. Ramanathan, "Proportional differentiated services: Delay differentiation and packet scheduling," in *Proc. ACM SIGCOMM*, Aug. 1999, pp. 109–120.

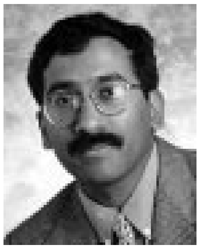
[22] M. May, J.-C. Bolot, A. Jean-Marie, and C. Diot, "Simple performance models for differentiated services schemes for the Internet," in *Proc. INFOCOM*, Mar. 1999, pp. 1385–1394.

[23] L. Qui, Y. Zhang, and S. Keshav, "On individual and aggregate TCP performance," in *Proc. ICNP*, Nov. 1999.

[24] I. Stoica and H. Zhang, "LIRA: An approach for service differentiation in the Internet," in *Proc. NOSSDAV*, June 1998, pp. 115–128.

**Ikjun Yeom** received the B.S. degree in electronic engineering from Yonsei University, Seoul, South Korea, in 1995, and the M.S. degree in computer engineering at Texas A&M University, College Station, TX, in 1998. He is currently working toward the Ph.D. degree in computer engineering at Texas A&M University.

He was with DACOM Company, Seoul, in 1995 and 1996. His research interests are in Internet QoS provisioning, the differentiated services network, and Internet congestion control.



**A. L. Narasimha Reddy** (S'86–M'90–SM'98) received the B.Tech. degree in electronics and electrical engineering from the Indian Institute of Technology, Kharagpur, India, in 1985, and the M.S. and Ph.D degrees in computer engineering from the University of Illinois, Urbana-Champaign, in 1987 and 1990, respectively. His work at the University of Illinois was supported by an IBM Fellowship.

He is currently an Associate Professor in the Department of Electrical Engineering at Texas A&M University. He was a Research Staff Member at IBM Almaden Research Center in San Jose, CA, from 1990 to 1995. His research interests are in multimedia, I/O systems, network QoS and computer architecture.

Dr. Reddy is a member of ACM SIGARCH and is a Senior Member of the IEEE Computer Society. He received a National Science Foundation CAREER award in 1996 and an Outstanding Professor award at Texas A&M during 1997–1998.