On Achievable Service Differentiation with Token Bucket Marking for TCP *

Sambit Sahu† Philippe Nain‡ Don Towsley† Christophe Diot* Victor Firoiu*

†Dept. of Computer Science, University of Massachusetts Amherst, MA 01003 {sahu,towsley}@cs.umass.edu

‡Inria, B.P. 93, Sophia Antipolis, France nain@sophia.inria.fr

*Sprint ATL, 1 Adrian Court, Burlingame, CA cdiot@sprintlabs.com

*Bay Architectures Lab, Nortel Networks, Billerica, MA 01821 vfiroiu@nortel.com

ABSTRACT

The Differentiated services (diffserv) architecture has been proposed as a scalable solution for providing service differentiation among flows without any per-flow buffer management inside the core of the network. It has been advocated that it is feasible to provide service differentiation among a set of flows by choosing an appropriate "marking profile" for each flow. In this paper, we examine (i) whether it is possible to provide service differentiation among a set of TCP flows by choosing appropriate marking profiles for each flow, (ii) under what circumstances, the marking profiles are able to influence the service that a TCP flow receives, and, (iii) how to choose a correct profile to achieve a given service level. We derive a simple, and yet accurate, analytical model for determining the achieved rate of a TCP flow when edge-routers use "token bucket" packet marking and core-routers use active queue management for preferential packet dropping. From our study, we observe three important results: (i) the achieved rate is not proportional to the assured rate, (ii) it is not always possible to achieve the assured rate and, (iii) there exist ranges of values of the achieved rate for which token bucket parameters have no influence. We find that it is not easy to regulate the service level achieved by a TCP flow by solely setting the profile parameters. In addition, we derive conditions that determine when the bucket size influences the achieved rate, and rates that can be achieved and those that cannot. Our study provides insight for choosing appropriate token bucket parameters for the achievable rates.

1. INTRODUCTION

The rapid growth of the Internet has been accompanied by an evolution of new applications, ranging from complex applications such as IP telephony, video on demand, interactive multimedia to simple data services. These new applications often require "better" service than single level of service provided by the current IP network. This new requirement calls for an architecture that can support multiple level of services while preserving the scalability and simplicity of the current Internet. The Differentiated services (diffserv) architecture [1] has been proposed as a scalable solution based on the same paradigm [3] as the Internet: that "complexity should be relegated to the end-points of the network while preserving the simplicity of the core network". This architecture advocates simple packet scheduling and buffer management at the core routers based on tags that are set at the routers at the edge of the network. The edge routers are allowed to perform traffic management on a per-flow basis whereas the core routers are not. With the assumption that no trust can be attached to the behavior of the end-hosts, the responsibility of end-to-end service assurance is primarily placed on the edge routers.

Although there have been several traffic management and packet marking mechanisms [2, 4, 8, 11, 13, 19] proposed for edge-routers, it is yet to be seen whether diffserv is able to deliver the promise of end-to-end service differentiation across applications. While these proposals differ from one another in the mechanism details, the solutions advocated have a common basic approach: packets of each flow are marked based on a chosen profile at an edge router; conforming and non-conforming packets are marked differently to receive different treatment from core routers that use active queue management mechanisms. These proposals have also proposed solutions for achieving service differentiation across a mix of responsive and non-responsive flows, i.e., TCP and UDP based flows. There remains, however, a lack of clear understanding of whether such profile based marking at an edge-router is sufficient to deliver *service differentiation even across a set of responsive flows*.

The goal of this paper is to examine (i) whether it is possible to provide service differentiation among a set of TCP flows that share a common bottleneck link based on their marking profiles, (ii) under what circumstances, the marking profiles are able to influence the service that a TCP flow receives, and (iii) how to set marking profile parameters to achieve a given rate, if the rate is feasible. In order to examine the above questions, we first derive a simple, yet accurate, analytical model for determining the send rate of a TCP flow when edge-routers use token bucket packet marking. The interference of other flows sharing the same bottleneck path is modeled by induced losses in the flow under study at the bottleneck router. Under the token bucket marking that we analyze, packets that conform to token bucket parameters, assured rate A and bucket size B, are marked as green and the excess packets are marked as red. We assume that active queue management at the core routers pref-

*This work was supported in part under National Science Foundation grant NCR 95-23807 and a gift from Sprint ATL. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

erentially drop packets such that green packets always incur lower loss as compared to red packets. Our model is validated through simulation using ns-2 [9].

Several simulation studies [5, 6, 19] have examined some of the above questions. Some of these studies have identified several useful "rules of thumb" such as the fact that it is easier to achieve a lower rate than a larger rate. But there is a general lack of understanding of when and how the several marking parameters influence the achieved rate. This is partly due to the fact that there are too many parameters that need to be chosen carefully, and most often, the answer to these questions are sensitive to the choice of parameters. Our analytical model provides a valuable tool to examine the effect of the marking parameters very easily over a wide range of parameters and provides guidance in selecting correct marking parameters.

In order to answer the question whether a set of TCP flows can be provided service differentiation based on their token bucket profile, we use our model to examine the effect of the token bucket parameters on the achieved rate of a TCP flow. Specifically we examine how, and under what conditions, bucket and rate parameters affect the achieved rate. We observe that the answer to these questions depend on the value of the token bucket parameters as well as the loss rates of green and red packets. From our study, we observe three important results: (i) the achieved rate is not proportional to the assured rate, (ii) it is not always feasible to achieve the assured rate and, (iii) there exist ranges of values of the achieved rate for which token bucket parameters have no influence.

In addition, we determine conditions under which a bucket helps improve the achieved throughput. We derive a boundary condition that can be used to determine whether a rate cannot be achieved irrespective of the token bucket parameters. Based on these conditions, for the achievable range of rates, we examine how to choose token bucket parameters to achieve a target rate and possible tradeoffs in choosing assured rate and bucket size parameters.

While some of our findings do match the results reported in other simulation based studies [2, 4, 5, 6], thus corroborating these claims, we do find significantly different conclusions in other cases. For example, it is not always possible to achieve an arbitrary rate by increasing the token bucket parameters. In [5], it has been observed that it is difficult to provide any meaningful service differentiation with a rate-based marking scheme. In [6], the impact of different marking schemes has been evaluated through simulation. In [19], it has been observed that it is difficult to provide service differentiation without modifying the congestion control mechanisms at the end-hosts, and several new marking schemes have been proposed. While these studies provide valuable insights regarding the behavior of marking schemes, it is very difficult to garner insight regarding setting appropriate marking parameters from simulation based studies. With the help of our analytical model, we are able to derive closed form conditions for understanding the impact of each token bucket parameter, and able to provide meaningful insights about parameter selection.

There have been proposals [8] advocating the use of more than two colors for marking packets. It has been shown in [6] that such marking schemes have some additional benefits over two color marking in the presence of non-responsive flows. However, it is not obvious whether such marking schemes help when the traffic consists solely of responsive flows. As our focus is to understand the achievable service differentiation among TCP flows, we restrict our investigation to two color marking. However, it is possible to extend our model to three color marking as well. Our model is inspired by the work in [18] which proposes a simple model for TCP with rate-based marking in the absence of a token bucket.

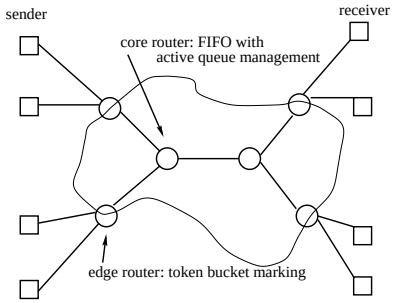


Figure 1: Typical Diffserv network model

The remainder of the paper is organized as follows. In Section 2, we formally introduce token bucket marking that is used at an edge router and the active queue management that is used at a core router. We derive analytical expressions for send rate of a TCP flow. In Section 3, we validate the derived model via simulation using ns-2. In Section 4, we examine the effect of token bucket parameters on the achieved TCP rate. Section 5 provides insight into how to best choose token bucket parameters for achieving a target rate for a TCP flow. Section 6 concludes our study.

2. TCP BEHAVIOR WITH LEAKY-BUCKET MARKING

In this section, we develop an analytical model for determining the send rate of a TCP flow when edge-routers use token bucket packet marking and core-routers use active queue management with FIFO scheduling for preferential packet dropping. We focus on a single TCP flow; the interference of other flows sharing the same bottleneck path is modeled by induced losses in the flow under study at the bottleneck path. First we introduce the network model, the details of the token bucket marker, and the simple loss model for the active queue management that we consider. We illustrate how the loss model can be used to model multi-RED, a generalization of RED [14] for multiple classes, and RIO [2]. Using this loss model, we then develop an analytical model to examine the effect of token bucket parameters on the send rate of a TCP flow. We use the term "send rate" and "achieved rate" synonymously in the rest of the paper.

2.1 Network model

Figure 1 shows a typical diffserv network model in which a sender gains access to the core network through an edge router. An edge-router marks packets, possibly on a per-flow basis, using a pre-negotiated "marking profile". If the sending rate of a flow conforms to its marking profile, the packets are marked as green. There are different variants as to how to mark (and treat) the packets that exceed the profile. For our study, we shall consider the 2-color marking where the excess packets are marked as red. Our study examines the behavior of the achieved rate when (i) a source uses TCP to send its packets, (ii) a token bucket marking profile is used for marking packets at an edge-router, and (iii) a core-router uses active queue management coupled with FIFO scheduling to provide preferential packet dropping. We focus on a single bottleneck core router in our analysis.

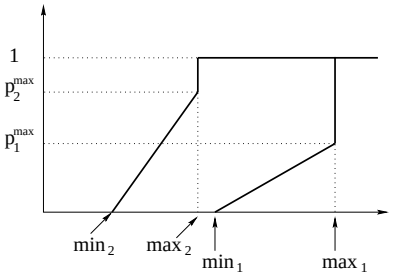


Figure 2: Multi-RED: generalization of RED for multiple classes

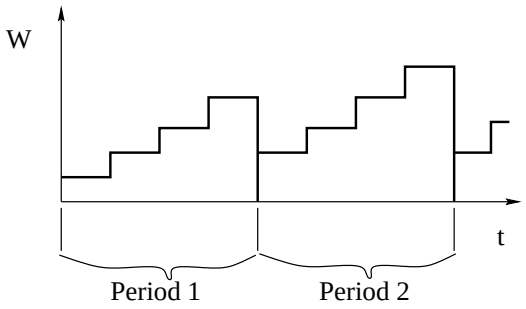


Figure 3: Evolution of TCP window size as a renewal process

A token bucket marking profile for 2-color marking is described by a pair of parameters, A and B, that denote the token rate and the token bucket size respectively. As long as the sending rate of a flow conforms to the token bucket with parameter (A, B), the packets are marked as green. When the sending rate does not conform to the token bucket parameters, the excess packets are marked as red. We define a marker that marks packets based on the above marking mechanism as a 2-color token bucket marker.

We consider the following active queue management at a core router: The core router maintains an estimate of the average queue length $\tilde{x}$. The drop probability for the incoming packet is a function of this average queue length and packet class (i.e., green or red). We assume that such a function is provided for each packet class. Under the assumption that the average queue length at the bottleneck router converges to a steady state value, we define p_1 and p_2 to be the loss probability of a green and red packet respectively at the core router. For our analysis, we make the following assumptions about p_1 and p_2 that we refer to as the "non-overlapping loss model" for the congested link: $p_1 > 0 \Rightarrow p_2 = 1$ and $p_2 < 1 \Rightarrow p_1 = 0$. This models the scenario where if a green packet is lost, a red packet is dropped with probability one and if a red packet loss is not equal to one, a green packet is never dropped. In order to understand TCP behavior with token bucket marking, first we derive analytical expressions for determining the send rate with the above loss model. Later we relax this condition and conjecture how our result can be extended to a loss model that is not restricted to the above conditions.

Let us now describe how multi-RED, a generalization of RED [14] active queue management for multiple packet classes, relates to the active queue management we described above. Figure 2 illustrates various parameters of multi-RED. The loss probability of a packet is given by: $p_i = \frac{\tilde{x} - min_i}{max_i - min_i} p_i^{max}$, for $min_i < \tilde{x} < max_i$, where $i = 1, 2$ for green and red packets respectively. With the help of extensive simulation in ns-2 [9], when $min_1 > max_2$, we observe the loss behavior consistent with the model we described above. If $min_1 > max_2$, we denote the parameter setting of multi-RED as non-overlapping. RIO [2] uses the similar loss model as multi-RED, except that, in the case of green packets, the average queue length is computed by accounting for the number of green packets in the queue.

2.2 Modeling TCP behavior

We consider that there is a single bottleneck link in the network and examine how congestion on this link affects the congestion control of TCP. We are interested in the congestion avoidance phase of TCP, which has often been modeled using renewal arguments [7, 10]. In such an approach, a suitable conditioning on loss events allows one to focus on TCP behavior during a renewal period and use the renewal reward theorem [17] to derive an analytical expression for long term TCP behavior.

We consider the Reno flavor of TCP [16], which is widely used in the Internet. The basic congestion avoidance mechanism works in the following way: TCP's congestion control window W, is increased by $1/W$ each time an ACK is received. Whenever a loss is detected, the window is decreased, with the amount of the decrease depending on whether packet loss is detected by duplicate ACKs or by timeout. If the loss is detected due to duplicate ACKs, the window size is reduced by half. If the loss is detected due to a timeout, the window size is reduced to one. The reader is referred to [15] for a detailed discussion on TCP congestion control mechanism.

For the moment we assume that all loss events are due to triple duplicate ACK (TD) notification. We shall include timeout (TO) events into our model later in Section 2.5. Figure 3 illustrates the evolution of the windowing behavior of a TCP source in congestion avoidance phase, where each renewal period is defined to begin immediately after a loss occurs and lasts until the next loss event [7]. Our approach is similar to the one in [7] except that we consider a deterministic model in which the window size is W at the beginning of a renewal period and $2W$ at the end of a renewal period. This simplified model is inspired by the work in [18]. Let T denote the average round-trip time for the TCP connection which we are interested in. For a given token bucket parameter (A, B), and T, we define W_a to be the "assured window size" associated with the connection that is expressed as $W_a = A \times T$. This approximation allows us to study the behavior of the marker at a round-trip time scale focusing on the current window size instead of the current send rate. Using the terminology in [18], we refer to $p_1 = 0, 0 < p_2 < 1$ as the under-subscripted case and $p_1 > 0, p_2 = 1$ as the over-subscripted case. We consider these cases separately when deriving the analytical model for characterizing TCP congestion control behavior. To simplify the presentation, we assume that the receiver does not delay acknowledgements [16], that is, it acknowledges every packet as soon as it receives it. We also assume that the receiver's window is sufficiently large such that it does not restrict the growth of sender's congestion window. Note that it is not difficult to accommodate delayed ACK behavior, and receiver window limitation into our model. The reader is referred to [12] where we have accounted for the above two factors in our model.

2.3 Under-subscribed case

As discussed above, this refers to the case when $p_1 = 0, p_2 > 0$. There are two possible scenarios based on whether $W_a > W$ or $W_a \leq W$. Consider first the case when $W_a > W$, which is illus-

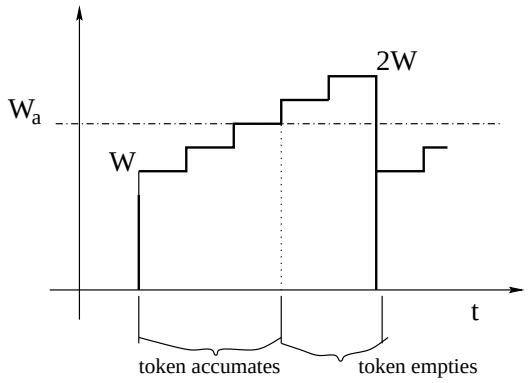


Figure 4: Effect of bucket size: $W_a > W$

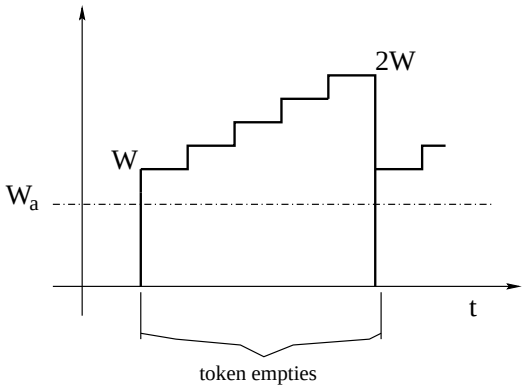


Figure 5: Effect of bucket Size: $W_a < W$

trated in Figure 4. Let W_i denote the current window size during the i^{th} round of a renewal period. As long as $W_i < W_a$, all the packets in that round are marked green. If the bucket size $B = 0$, then the moment $W_i > W_a$, $W_i - W_a$ packets are marked red. However due to the non-zero bucket size, the situation is trickier. We want to determine the number of red packets transmitted before the first red packet is lost. During a renewal period, tokens accumulate so long as $W_i < W_a$ due to the fact that the transmission rate is lower than the token generation rate. Using a fluid approximation of the staircase function W_i, the number of tokens accumulated during the current renewal period while $W_i > W_a$ is given by: [1]

$$N_b = \min\left\{B, \frac{(W_a - W)^2}{2}\right\} \tag{1}$$

If a loss event is a red packet loss, it is easy to show that the bucket was empty at the beginning of the renewal period. Thus we have:

LEMMA 1. *For the case $p_1 = 0, p_2 > 0$, the token bucket is always empty at the beginning of a renewal period when loss events are due to TD ACKs.*

As a result of token accumulation, some packets continue to be marked as green even though $W > W_a$, since these packets are matched against these accumulated tokens. The extra number of packets that are marked as green when the bucket size is B is given by (1). Thus, the number of packets marked red during a renewal period is given by:

$$N_{red} = \frac{(2W - W_a)^2}{2} - \min\left\{B, \frac{(W_a - W)^2}{2}\right\} \tag{2}$$

Given that a red packet is lost with probability p_2, the probability that i consecutive red packets are transmitted successfully before a loss occurs is $(1-p_2)^i p_2$. Thus the expected number of red packets transmitted before a loss occurs is $1/p_2$. From (2):

$$\frac{(2W - W_a)^2}{2} - \min\left\{B, \frac{(W_a - W)^2}{2}\right\} = 1/p_2 \tag{3}$$

Let us consider the case when $B \geq (W_a - W)^2/2$. From (3), solving for W, we get:

$$W = \frac{W_a + \sqrt{W_a^2 + 6/p_2}}{3} \tag{4}$$

Similarly, when $B < (W_a - W)^2/2$, (the bucket size determines the limit on the number of tokens accumulated), solving for W, we get:

$$W = \frac{W_a + \sqrt{2B + 2/p_2}}{2} \tag{5}$$

So far we have assumed that $W_a > W$. When $W_a < W$, the presence of a bucket does not help. This is because at the beginning of a renewal period the bucket is empty and the packet arrival rate is greater than the assured rate. Thus the number of red packets in this case is $3W^2/2 - WW_a$. Equating this to the expected number of red packets and solving for W yields:

$$W = \frac{W_a + \sqrt{W_a^2 + 6/p_2}}{3} \tag{6}$$

Thus (4), (5), and (6) determine the value of W for the three cases, i.e., (i) $W_a > W, B \geq \frac{(W_a - W)^2}{2}$, (ii)$W_a > W, B < \frac{(W_a - W)^2}{2}$ and (iii) $W_a < W$, respectively. It is possible to rewrite the conditions to remove the dependence on W as follows:

[1] We shall observe in Section 3 through the validation of our model that it is a reasonable approximation. Many previous studies have used similar fluid approximations [10, 18] for modeling TCP behavior.

$$W = \begin{cases} \left(W_a + \sqrt{W_a^2 + 6/p_2}\right)/3, & W_a \leq \tilde{W} \\ \left(W_a + \sqrt{2B + 2/p_2}\right)/2, & W_a > \tilde{W} \end{cases} \quad (7)$$

where $\tilde{W} = \sqrt{2(B + 1/p_2)} + 2\sqrt{2B}$. The sending rate of the TCP source is the ratio of the average number of packets sent during a renewal period and the average duration of the renewal period, i.e., $r = \frac{3W}{2T}$. Thus the sending rate is given as:

$$r = \begin{cases} \left(A + \sqrt{A^2 + \frac{6}{p_2 T^2}}\right)/2, & A \leq \tilde{W}/T \\ 3A/4 + \left(3\sqrt{B + 1/p_2}\right)/\left(2\sqrt{2}T\right), & A > \tilde{W}/T \end{cases} \quad (8)$$

2.4 Over-subscribed case

Note that in this case $p_1 > 0$ and $p_2 = 1$. The first packet that ends the current renewal period can be either a red or a green packet. We consider both cases below.

Let us consider the case when a red packet is lost. Using a similar argument as in Lemma 1, red packet loss implies that the bucket is empty at the beginning of the renewal period. Thus the number of packets that are marked green due to the presence of a non-zero bucket size is exactly equal to the number of tokens that are accumulated until $W_i > W_a$. This yields the following condition:

$$\frac{(2W - W_a)^2}{2} = \min\left\{B, \frac{(W_a - W)^2}{2}\right\} \quad (9)$$

Solving for W, we obtain:

$$W = \begin{cases} 2W_a/3 & : \quad W_a < 3\sqrt{2B} \\ \left(W_a + \sqrt{2B}\right)/2 & : \quad W_a \geq 3\sqrt{2B} \end{cases} \quad (10)$$

Next we consider the case when the lost packet is green. Given that a green packet is lost with probability p_1, the expected number of green packets transmitted before a loss occurs is $1/p_1$. Computing the number of green packets transmitted during a renewal period, and equating it to $1/p_1$ we get $W = \sqrt{\frac{2}{3p_1}}$.

It is easy to show that these three conditions combined together yield the following:

$$W = \min\{\frac{2W_a}{3}, \frac{W_a + \sqrt{2B}}{2}, \sqrt{\frac{2}{3p_1}}\} \quad (11)$$

Thus the sending rate $r = \frac{3W}{2T}$ of the TCP source is given by:

$$r = \min\left\{A, \frac{3(A + \sqrt{2B}/T)}{4}, \frac{1}{T}\sqrt{\frac{3}{2p_1}}\right\} \quad (12)$$

2.5 Including timeout (TO) as loss events

So far our analysis has assumed all packet loss indications are due to triple duplicate ACKs. We now extend our model to include the case where a loss indication is due to a timeout event.

We consider the following model for the evolution of TCP window when a TO event occurs. This occurs when packets are lost and less than three duplicate ACKs are received [7]. Let $q(w)$ denote the probability that a loss event is due to a timeout (TO) occurrence when the current window size is w. Following a TO event, the congestion window is reduced to one, and one packet is resent in the first round after a TO. Let the current timeout value be T_0. If an ACK is not received by T_0, another TO event occurs, which doubles the duration of the timeout period; this doubling of the timeout period is repeated for each unsuccessful retransmission until $64T_0$ is reached. Let Z^{TO} define the duration of the sequence of TO events that occur until TCP recovers from timeout, i.e., successfully retransmits a packet. After this sequence of timeouts, a series of renewal periods due to TD events occur until another TO event occurs.

Consider the first TD renewal period that occurs following the end of timeout period. Let Z_1^{TD} the duration of this particular TD renewal period. It is important to observe that the bucket is not necessarily empty at the beginning of this renewal period. This is due to the fact that during the series of timeouts that occur before the first TD period, very few packets are transmitted. We assume that Z^{TO} is large enough such that the bucket is filled at the beginning of the first TD period. Let W_{TO} denote the window size for the first TD period that occurs after TCP source recovers from the sequence of TO periods. This is computed by taking into account that the bucket is filled at the beginning of this TD period. Let W_{TD} denote the window size that we computed in the previous section assuming only TD loss indications. With the above model for the evolution of congestion window of a TCP source, the achieved throughput is derived as:

$$r = \frac{3\left(qW_{TO}^2 + (1-q)W_{TD}^2\right)}{2\left(T\left(qW_{TO} + (1-q)W_{TD}\right) + qE[Z^{TO}]\right)} \quad (13)$$

where $q = q(2W_{TD})$. The details of the derivation for $q(w)$, $E[Z^{TO}]$, W_{TD}, W_{TO} can be found in [12].

2.6 Relaxing the overlapping losses

When the losses of green and red packets are overlapping (i.e., $p_1 > 0$ and $p_2 < 1$), we conjecture that the send rate of a TCP flow is determined by the minimum of the rates determined by considering the under-subscription and the over-subscription case. We validate this conjecture in Section 3.1 with the help of simulation using ns-2.

3. MODEL VALIDATION

Equations (8), (12), and (13) provide an analytical characterization of TCP send rate when packets are marked using a token bucket marker and conforming packets incur a lower loss rate than non-conforming packets. In this section we validate these formulae using a simulation model in ns-2 [9].

We have used the ns-2 code for the token bucket marker and modified RED to implement multi-RED. Figure 6 shows the network configuration we use for the simulation model. Each edge-router

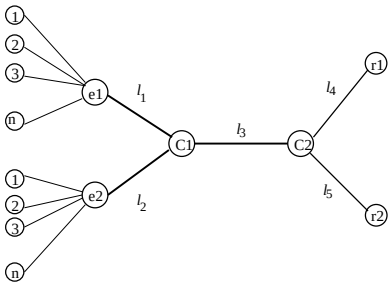


Figure 6: Simulation configuration

marks packets of each incoming flow i using a token bucket marking scheme and parameters (A_i, B_i). We assume that links l_1, l_2, l_4 and l_5 have capacity of 100Mb/s and employ droptail queue management. Link l_3 employs multi-RED queue management and has a link capacity 20Mb/s. In our simulation, we make sure that loss occurs only on l_3 and monitor this link to determine green and red packet losses. In addition, we monitor each TCP source to determine T and T_0 values.

In order to validate the analytical model for all the cases, it is important to perform experiments in which both under-subscription and over-subscription conditions are observed in the simulation. We observe from the simulation that it is often difficult to generate a wide range of loss rates when the background traffic consists of a set of TCP flows. This is partly because there are several parameters needed to be chosen in multi-RED that makes it extremely difficult to generate a desired loss rate.

In order to address this issue, we conduct two types of validation. In the first approach, we vary the parameters of multi-RED to generate different loss rates. We measure the loss rates of green and red packets and compute the predicted send rate from our model using measured loss rates. We again compare this against the measured send rate from the simulation. In the second method, we simulate a lossy link using a Bernoulli loss model on link l_3. By appropriately choosing the parameter of this loss process, one can easily generate a wide range of loss rates. With the help of such a loss model, we measure the send rate that we observe from the simulation and compare against the predicted rate from our model.

3.1 Validation I - multi-RED parameter settings

We experiment with different parameter settings with multi-RED and compare the measured send rate with the predicted send rate from our model. Note that while this experiment is closer to a real network scenario, it is not possible to generate an intended loss rate easily. We have chosen both non-overlapping and overlapping parameters for multi-RED (recall that when $min_1 < max_2$, we refer to this as overlapping parameter setting and non-overlapping otherwise). Table 1 illustrates the details of the multi-RED and token bucket parameters. We consider 60 TCP connections and run the simulation for 500 sec in each of the experiments. We compute the average send rate from the measured send rate of individual TCP connection. We measured the loss rate of both red and green packets. As conjectured in Section 2.6, we compute two send rates, one using the measured loss rate of red packets, and the other one using the measured loss rate of green packets. We take the minimum of these two computed rates as the send rate. We observe that in most of the cases, our model is able to predict the measured send rate. We conduct similar experiments where TCP flows have different round trip time and different token bucket parameters [12] and observe similar agreement of our model with the simulation.

Flow Id	A (kb/s)	B (pkt)	T (ms)
1 to 20	200	12	200
21 to 40	640	16	480
41 to 60	1000	24	100

Table 2: Parameters for Figure 8

3.2 Validation II - Bernoulli loss model

We make use of a Bernoulli loss model to validate our analysis over a wide range of loss rates. Figure 7 compares the achieved rate calculated from the analytical model with the measurements from the simulation for the under-subscribed case. We vary the loss probability of the red packets from 0.001 to 0.5. Figure 7 illustrates the validation with homogeneous parameter settings for all the TCP flows, i.e., an identical round trip time and an identical assured rate and bucket size for all senders. We consider 60 TCP connections for each of the illustration in Figure 7. For this simulation, we have chosen the round trip time to be 480 ms. The assured rate for each flow in Figure 7(a), (b), (c) are chosen to be 20 kb/s, 64 kb/s, and 128 kb/s respectively. The token bucket size is chosen to be $4, 8, 16$ packets for the above three illustrations. We assume that packet size is 1500 bytes. For the rest of the paper, we choose this packet size. We run the simulation for 500 seconds to eliminate any transient behavior. We measure the loss probability, round trip time, and TO period analyzing the simulation traces, and compute the send rate from our model. We determine the average rate of a TCP flow from the simulation and compare it with the computed send rate. Figure 7 shows that the analytical model accounting for both TD and TO events, is able to provide a quite accurate estimate of the achieved rate and matches the simulation results for the entire range of loss rate for red packets. We validate our model with heterogeneous TCP sessions where round trip time T and token bucket parameters of TCP connections are different. The details of this comparison can be found in [12].

Figure 8 shows the comparison for the over-subscribed case, i.e., $p_2 = 1, p_1 > 0$ with heterogeneous configuration of parameters. We choose 60 TCP connections for this experiment. Table 2 shows the parameters that are chosen for the results in Figure 8. We assume that link l_3 is 40 Mb/s for this experiment to make sure that losses occur only due to the Bernoulli loss process. Figure 8(a), (b), (c) plot both the measured average send rate and computed send rate of a flow with flowid $1 \ldots 20$, $21 \ldots 40$, $41 \ldots 60$ respectively.

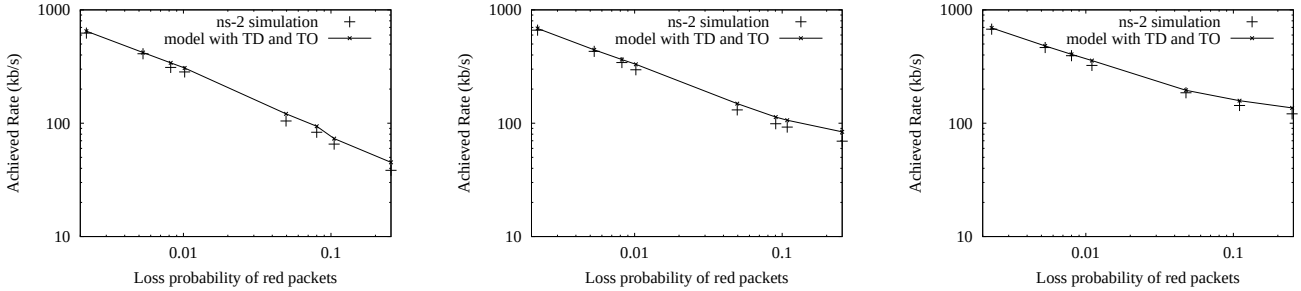
From Figure 8, we observe that our model predicts the send rate very accurately over a wide range of values of p_1, and for both small and large assured rate parameters. From the simulation we observed that at higher loss probability, there are many TO loss events. From Figure 8 we observe that our model accurately accounts for TO events. We have experimented with several different values of token bucket parameters A, B and round trip time T. The details can be found in [12].

4. EFFECT OF TOKEN BUCKET PARAMETERS

Having validated the model in the previous section, we now use it to examine the effect of token bucket parameters on the achieved rate. For the simplicity of the presentation, we use the model we derived with TD loss events for the rest of the paper, unless stated

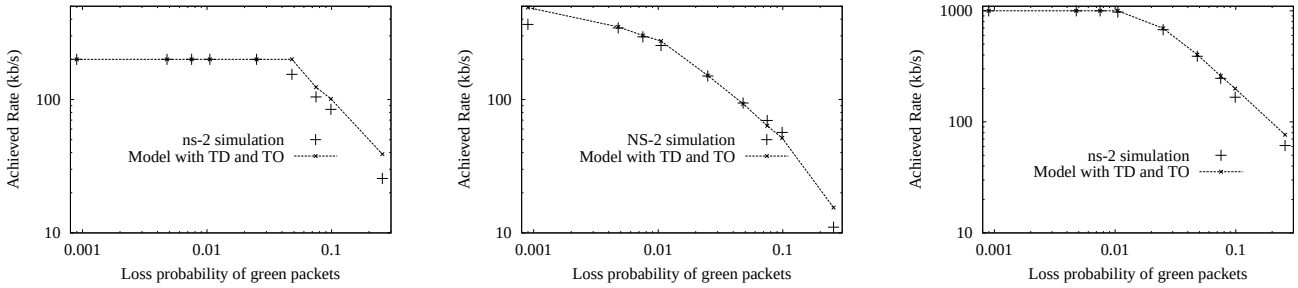
Multi-RED parameter			Multi-RED parameter			A	B	T	Send rate	Send rate	Error
min_1	max_1	p_1	min_2	max_2	p_2	(kb/s)	pkt	(ms)	(sim) kb/s	(model) kb/s	(%)
30	45	0.1	15	30	0.1	128	4	200	1068.3	1101.4	3.09
30	45	0.1	15	30	0.2	128	4	200	1013.3	1036.2	2.27
30	45	0.1	15	30	0.5	128	4	200	931.2	951.7	2.14
30	45	0.1	5	15	0.5	128	4	200	389.5	408.3	4.88
30	45	0.1	5	15	0.8	128	4	200	288.3	306.1	6.25
30	45	0.2	15	30	0.2	128	4	200	1036.7	1089.4	5.11
30	45	0.2	15	30	0.5	128	4	200	912.4	903.5	0.99
30	45	0.2	15	30	0.8	128	4	200	573.4	603.4	5.23
30	45	0.5	15	30	0.5	128	4	200	245.6	264.1	7.75
30	45	0.5	15	30	0.8	128	4	200	204.3	227.1	11.27
30	45	0.2	15	45	0.8	128	4	200	322.5	345.4	7.14
30	45	0.5	15	45	0.8	128	4	200	306.2	328.2	7.18

Table 1: Experiments with multi-RED parameters



(a) $A = 20$ kb/s (b) $A = 64$ kb/s (c) $A = 128$ kb/s

Figure 7: Validation of model with ns-2: under-subscribed case



(a) $A = 200$ kb/s, $T = 200$ ms (b) $A = 640$ kb/s, $T = 480$ ms (c) $A = 1000$ kb/s, $T = 100$ ms

Figure 8: Validation of Model with ns-2: Over-subscribed Case

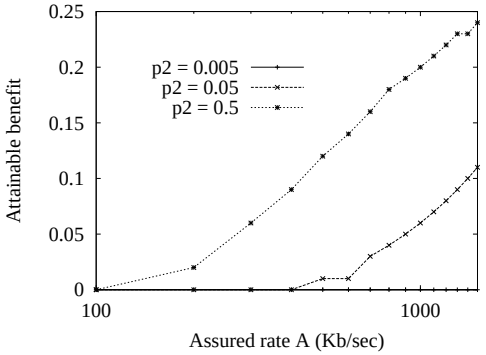


Figure 9: Effect of bucket size for under-subscription case

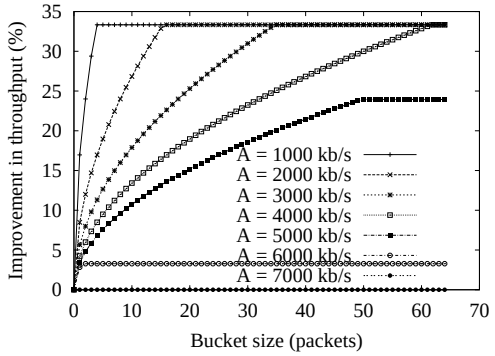


Figure 10: Effect of bucket size for over-subscribed case

otherwise. We provide a systematic examination by considering a wide range of choices for token bucket and loss parameters. In particular, (i) we determine conditions under which the achieved rate is sensitive to the choice of the bucket size, and, (ii) examine the effect of assured rate parameter on the achieved rate to determine the efficacy of token bucket marking in providing service differentiation. As we shall observe, some of the conclusions match what have been observed by other simulation studies [2, 5, 6, 19], thus reinforcing various observations. However, we will examine ranges of parameters that have not been considered in those previous studies and will observe that the conclusions are very different over these ranges.

4.1 Effect of bucket size

First we turn our attention to the study of the effect of bucket size on the achieved rate. We derive the achieved throughput for different assured rate A and loss parameter p_2 as a function of bucket size B. Let us examine the under-subscribed case first, where the achieved rate is given by (8). Note that the condition $A > \tilde{W}/T$ leads to the case where the achieved rate is influenced by the choice of bucket size B. This condition refers to the case where the bucket size imposes a constraint, i.e., more tokens are generated than can be stored in the bucket. On the other hand, whenever $A \leq \tilde{W}/T$, the achieved rate is not affected by the choice of bucket size. It is evident from the above condition that the effect of B on the achieved rate r very much depends on the values of parameters such as T, p_2, B. Let r_0 and r_∞ denote the achieved rate when $B = 0$ and $B = \infty$ respectively. In order to explore the maximum attainable gain due to a bucket, we derive the increase in the achieved rate $\delta r = (r_\infty - r_0)/r_0$ as a function of assured rate A for different values of p_2. Figure 9 plots δr as a function of assured rate A for $p_2 = 0.005, 0.05, 0.5$ and $T = 200$ ms.

There are two interesting observations to be made from Figure 9. First, when p_2 is small, the presence of a bucket has little effect on the achieved rate. Second, the need for a bucket becomes more pronounced as the assured rate, A, increases. These trends can be explained as follows: when unmarked packet loss rate is high, increasing B essentially protects a flow by increasing the number of packets that will be marked as green. Secondly, the reduction in the window size at the advent of a loss affects a flow with higher assured rate more than one with a lower assured rate. This can be shown easily from (8).

Next we consider the over-subscribed case where the achieved throughput is given by (12). The following argument determines the conditions under which B impacts the achieved rate. Consider a loss

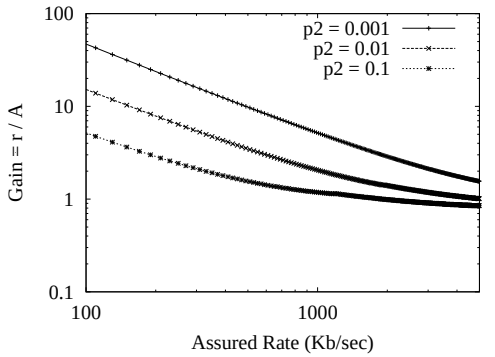


Figure 11: Service differentiation with under-subscribed case

rate p_1 and round trip time T. Given these two values, there are three possible cases:

(i) $\frac{1}{T}\sqrt{\frac{3}{2p_1}} > A$**:** In this case, as long as $B \geq A^2T^2/32$, the achieved rate is A. When the bucket size is zero, the throughput is $3A/4$. Thus increasing bucket size helps improve the throughput.

(ii) $\frac{1}{T}\sqrt{\frac{3}{2p_1}} < 3A/4$**:** Here the achieved rate is $\frac{1}{T}\sqrt{\frac{3}{2p_1}}$ regardless of the bucket size.

(iii) $3A/4 < \frac{1}{T}\sqrt{\frac{3}{2p_1}} < A$**:** Increasing the bucket size increases the achieved rate from $3A/4$ to $\frac{1}{T}\sqrt{\frac{3}{2p_1}}$. In this case, the achieved rate is lower than the assured rate A.

Figure 10 shows the improvement in throughput as a function of bucket size. We have chosen $p_1 = 0.001$, $T = 100$ ms. We vary the assured rate A between 1000 and 7000kb/s. These parameters are chosen carefully to illustrate all three cases identified above. From Figure 10 we observe that when $A \leq 4000$, the throughput increases from $3A/4$ to A, thus achieving the maximum possible increase of 33% in the achieved rate. When $A = 5000$, we see that the maximum possible benefit is limited to 22%. This is the case when $3A/4 < \frac{1}{T}\sqrt{\frac{3}{2p_1}} < A$. We find when $A = 7000$, increasing the bucket size has no effect on the achieved throughput. way as for the under-subscribed case.

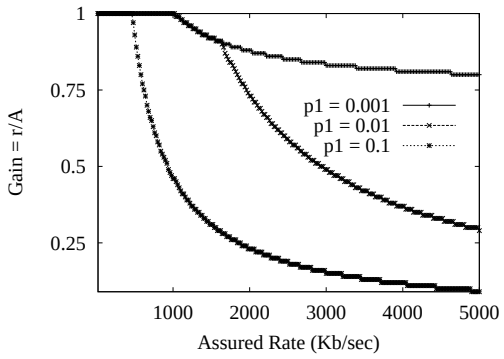


Figure 12: Service differentiation with over-subscribed case

4.2 Effect of assured rate A

In this section, we study the effect of assured rate A on the achieved rate of a TCP session. In particular, we examine whether it is possible to achieve service differentiation across a set of TCP sessions by choosing different assured rates. In this study we define "ideal" service differentiation to be the case when *achieved rate is proportional to the assured rate*. We define $\mathcal{G} = r/A$ to be the "gain factor" that we use as an indicator to understand the achieved service differentiation. We say that a marking mechanism is able to achieve ideal service differentiation when $r_1/A_1 = r_2/A_2$, for any assured rate A_1, A_2, where r_1, r_2 are the achieved rates respectively. In such a case, $\mathcal{G}$ is independent of A.

DEFINITION 1 (IDEAL SERVICE DIFFERENTIATION). *Let $\mathcal{G}$ be defined as $\mathcal{G} = r/A$, where A is the assured rate and r is the achieved rate. If $\mathcal{G}$ is a constant function of A, we say that ideal service differentiation is achieved.*

First we consider the under-subscribed case, i.e., $p_1 = 0, p_2 > 0$. Observe from (8) that as long as $A \leq \tilde{W}/T$, the achieved rate is greater than the assured rate, $r > A$. If $A > \tilde{W}/T$, then $r > A$ when $A \leq \left(6\sqrt{B + 1/p_2}\right)/\sqrt{2}T$ and $r < A$ otherwise. Now we determine $\mathcal{G}$ as a function of assured rate A to examine the level of service differentiation achieved in this case. We explore a wide range of values for assured rate A by varying A from 20Kb/s to 5Mb/s. Figure 11 shows $\mathcal{G}$ as a function of A when $T = 100$ ms, $B = 32$, and $p_2 = 0.001, 0.01, 0.1$. It is observed that the gain $\mathcal{G}$ is much higher for lower assured rates. For example, when $p_2 = 0.01$, $\mathcal{G}$ is as high as 30 for $A = 50$Kb/s while it is as low as 2.05 when $A = 1000$Kb/s. Such a disparity in $\mathcal{G}$ still exists for different value of p_2, although it decreases for large p_2. Another observation is that when $A > 1000$ kb/s, when $p_2 = 0.1$, $\mathcal{G}$ is less than 1, meaning that, the achieved rate is lower than the assured rate A. Following are the two immediate conclusions: (i) it is difficult to achieve assure profiles with higher rates, and (ii) for profile with lower rates, it is difficult to restrict the achieved rate to the smaller assured rate. Figure 11 shows that it is not possible to achieve proper service differentiation with this token bucket marker as it favors flows with smaller assured rate more than it does to flows with higher assured rate.

Figure 12 illustrates a similar result for over-subscribed case, i.e., $p_2 = 1, p_1 > 0$. An important difference is that in this case the maximum value of $\mathcal{G}$ is 1. We observe that for larger assured rates, the achieved rate is much smaller than the assured rates, especially when loss rate is higher. This implies that it is difficult to achieve the assured rate when there is a loss of green packets. A much higher rate parameter A for the token bucket marker is necessary to achieve a desired rate r.

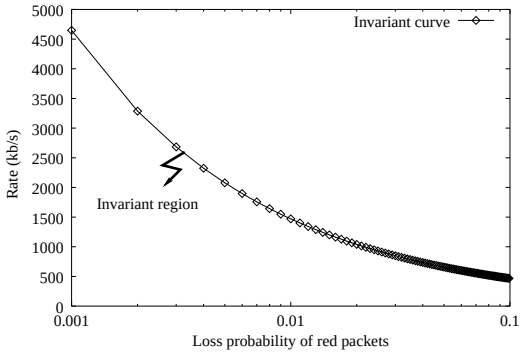


Figure 13: Invariant region

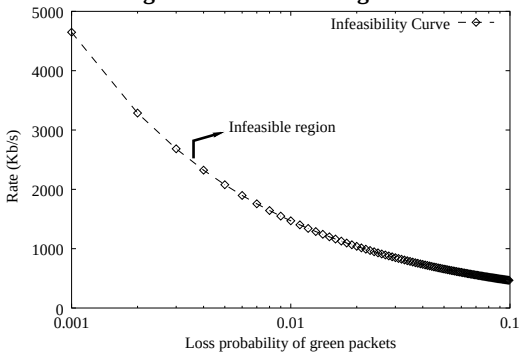


Figure 14: Infeasible region

5. CHOICE OF TOKEN BUCKET PARAMETERS

Our examination of the effect of token bucket parameters indicated that the achieved rate depends on the assured rate and bucket size in a non-linear manner. In this section, (i) we determine the conditions for which token bucket parameters can help achieve (regulate) a rate, and, (ii) how to best set token bucket parameters to achieve a rate that is within the control of token bucket parameters. This is motivated by our quest to understand whether profile based marking is sufficient to regulate the achieved rate for a TCP flow. As we shall observe in this section, there are conditions under which token bucket marking has no effect on the achieved throughput. However, our study provides insight on how to best choose the marking parameters to achieve a desired rate, when the rate is achievable.

5.1 Invariant (infeasible) region

We have observed that it is difficult to achieve a throughput $r \geq A$ in several situations, especially when the system is over-subscribed. We introduce $\tilde{r}$ to be the target rate of a TCP flow. We determine the token bucket parameters that are required to achieve $\tilde{r}$. Rewriting (8) in terms of this target rate $\tilde{r}$ to determine the required assured rate A, we get the following for the under-subscribed case:

$$A = \begin{cases} \tilde{r} - \frac{3}{2\tilde{r}p_2T^2}, & \tilde{r} \leq \frac{3\tilde{W}}{2T} \\ \frac{4}{3}(\tilde{r} - \frac{3}{2\sqrt{2}T}\sqrt{B + \frac{1}{p_2}}), & \tilde{r} > \frac{3\tilde{W}}{2T} \end{cases} \quad (14)$$

Observe that we need additional condition that $A \geq 0$ for (14). Thus when (i) $\tilde{r} \leq \frac{3\tilde{W}}{2T}$, we require that $\tilde{r} \geq \frac{1}{T}\sqrt{3/2p_2}$ and (ii) $\tilde{r} \geq \frac{3}{2\sqrt{2}}\sqrt{B + 1/p_2}/T$ when $\tilde{r} > \frac{3\tilde{W}}{2T}$. But (ii) is always guaranteed as $\frac{3\tilde{W}}{2T} < \tilde{r} < \frac{3}{2\sqrt{2}}\sqrt{B + 1/p_2}/T$ does not arise. This is because $\frac{3\tilde{W}}{2T} \geq \frac{3}{2\sqrt{2}}\sqrt{B + 1/p_2}/T$ is always true. Note that (i) arises from the fact that even when $A = 0$, $B = 0$, a rate $\tilde{r}_0 = \frac{1}{T}\sqrt{3/2p_2}$ is automatically achieved. For the over-subscribed case, when $\tilde{r} \leq \frac{1}{T}\sqrt{\frac{3}{2p_1}}$, we get the following from (12):

$$A = \begin{cases} \tilde{r}, & \tilde{r} \leq 3\sqrt{2B}/T \\ \frac{4\tilde{r}}{3} - \frac{\sqrt{2B}}{T}, & \tilde{r} > 3\sqrt{2B}/T \end{cases} \quad (15)$$

Note that the value of A is always positive under the condition that $\tilde{r} > 3\sqrt{2B}/T$. In addition, observe from (15) that there are several choices of (A, B) that can achieve the rate $\tilde{r}$ when $\tilde{r} \leq \sqrt{\frac{3}{2p_1}}/T$. When $\tilde{r} > \sqrt{3/2p_1}/T$, no parameter setting will achieve the rate $\tilde{r}$ as apparent from (12).

An interesting inference that we derive from (14) and (15) is that there are certain ranges of $\tilde{r}$ for which the token bucket parameters have no influence on the achieved rate. For the under-subscribed condition, we have the following result:

RESULT 1 (INVARIANT RANGE). *When $p_1 = 0, p_2 > 0$ (under-subscribed), if $\tilde{r} < \frac{1}{T}\sqrt{3/2p_2}$, the token bucket parameters have no influence on the achieved rate r. Moreover, r is always larger than $\tilde{r}$ under this condition.*

The proof of Result 1 follows directly from (14). Figure 13 illustrates the above result for $T = 100$ ms. The "invariant curve" represents $\tilde{r} = \frac{1}{T}\sqrt{3/2p_2}$. The values of $\tilde{r}$ and p_2 that satisfy $\tilde{r} < \frac{1}{T}\sqrt{3/2p_2}$ are marked in Figure 13 as invariant region where token bucket parameters do not have any impact on the achieved rate. The following is the result for over-subscribed case:

RESULT 2 (INFEASIBLE RANGE). *When $p_1 > 0, p_2 = 1$ (over-subscribed), no rate $\tilde{r} > \frac{1}{T}\sqrt{3/2p_1}$ can be achieved with any assured rate and/or bucket size. However there always exists some combination of (A, B) that can achieve a target rate $\tilde{r}$ when $\tilde{r} < \frac{1}{T}\sqrt{\frac{3}{2p_1}}$.*

Figure 14 illustrates the above result for $T = 100$ ms and $B = 10$. The results in Result 1 and 2 imply that with token bucket marking, it is not always possible to regulate the achieved throughput of a TCP flow.

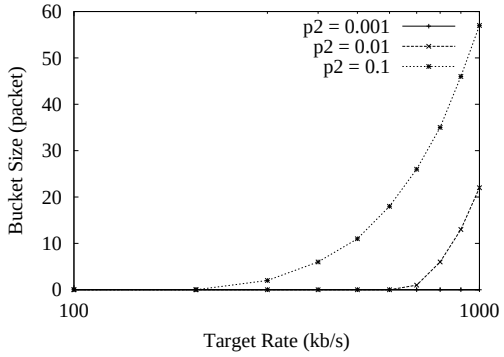


Figure 15: Maximum bucket size: for a target rate $\tilde{r}$, the above figure shows the maximum bucket size beyond which B has no effect on the achieved rate

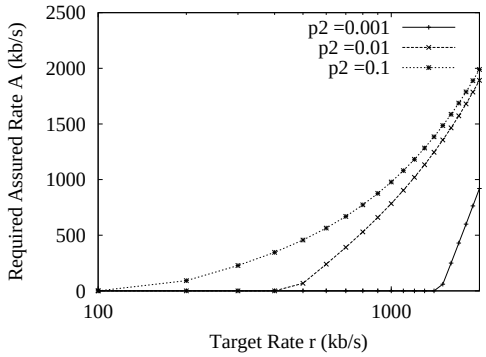


Figure 16: Under-provision case: required assured rate A to get a rate $\tilde{r}$

5.2 Parameter selections for achievable rate

We have seen that a certain range of values for $\tilde{r}$ are either not achievable or not influenced by the choice of token bucket parameters. Now we discuss how to best choose token bucket parameters to achieve a rate $\tilde{r}$ when it is achievable.

Consider the under-subscribed case. From (14), observe that when $\tilde{r} > 3\tilde{W}/2T$, A depends on the choice of B; otherwise it does not. As a result, B is likely to have greater effect on $\tilde{r}$ when $\tilde{r}$ is higher as compared to when $\tilde{r}$ is lower. Another observation is that, by choosing a lager bucket size B, it is possible to reduce the assured rate A and vice versa. We have seen this result partly in Figure 9. Also note that when $\tilde{r} < 3\tilde{W}/2T$, there is no added benefit in increasing B. Figure 15 illustrates the condition on the bucket size beyond which there is no gain in increasing the bucket size. When $T = 400$ ms, from Figure 15 we observe that for $p_2 = 0.001$, $B = 0$ is sufficient. For $p_2 = 0.01, 0.1$, we observe that increasing B helps, especially at higher value of $\tilde{r}$. We examine the effect of bucket size under a wide range of round trip time T and find the result sensitive to the choice of T. We also consider the effect of bucket size for the over-subscribed case. The details of the results can be found in [12].

Next we examine how to best choose the assured rate A to achieve $\tilde{r}$ for different loss conditions. Figure 16 plots the required A from (14) for $p_2 = 0.001, 0.01, 0.1$. We have chosen $B = 20$ for this

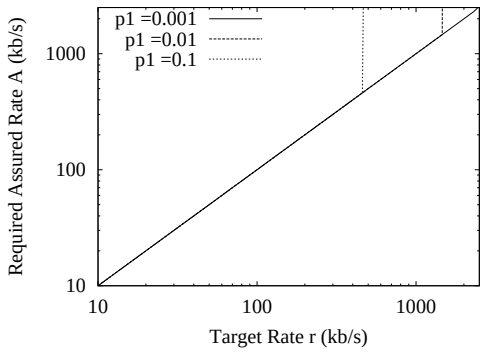


Figure 17: Over-provision case: required assured rate A to get a rate $\tilde{r}$

illustration. Observe from Figure 16 that when $T = 100$ ms, we do not need any assured rate until $\tilde{r} > \tilde{r}_0$, where $\tilde{r}_0$ can be calculated from (14). Note that $\tilde{r}_0$ depends upon p_2, B, T. For $\tilde{r} > \tilde{r}_0$, a much higher assured rate is required to achieve a higher $\tilde{r}$, i.e., $A_1/\tilde{r}_1 > A_2/\tilde{r}_2$ for $\tilde{r}_1 > \tilde{r}_2$.

We examine the over-subscribed case now, i.e., $p_1 > 0, p_2 = 1$, when $\tilde{r} < \frac{1}{T}\sqrt{\frac{3}{2p_1}}$. Figure 17 plot the required assured rate as a function of target rate $\tilde{r}$ for $T = 100$ ms. The main result Figure 17(a) illustrates that for the feasible range ($\tilde{r} < \frac{1}{T}\sqrt{\frac{3}{2p_1}}$), $A = r$ as long as $A \leq 3\sqrt{2B}/T$. It is possible to choose a larger B so that $A = r$ is satisfied instead of $A = \frac{4\tilde{r}}{3} - \frac{\sqrt{2B}}{T}$. This also indicates that there are several pair of values A, B which can achieve $\tilde{r}$ in this case. Also note from Figure 17 that when $p_1 = 0.01$, for $\tilde{r} > 1460$ kb/s, no value of A and B can achieve this rate.

The above illustration provides us insight regarding the (i) tradeoffs between the choice of A and B to achieve a target rate $\tilde{r}$, (ii) when it is not useful to increase B, and (iii) what rates are not achievable. This has immediate application as to how a user should choose a profile to get a desired rate. Also these results can be used by a service provider for allocating resources among different users to provide them assured rates when feasible, and deny services when it is not feasible.

6. CONCLUSION

In this work, we examined whether it is feasible to achieve service differentiation across a set of TCP flows using token bucket marking at the edge and active queue management at the core routers. We derived an analytical model for computing the send rate as a function of token bucket parameters. Our study indicates that it is not feasible to achieve ideal service differentiation across a set of TCP flows by setting their token bucket parameters for packet marking. This is mainly because the achieved rate is not proportional to the assured rate. In addition, there are ranges of parameters when (i) token bucket parameters have no effect on the achieved rate and, (ii) the target rate is not possible to achieve. We identified the conditions that determine when a rate is achievable. For the set of achievable rates, we examined how to best set the token bucket parameters to achieve these rates.

Acknowledgement

We would like to thank Prof. Jim Kurose for his invaluable guidance and many insightful discussions during the course of this work. And sincere thanks to Jitu Padhye and Tian Bu of Univ. of Massachusetts for their help with ns-2. The first author would like to thank Yun Wang of Concordia University for making his ns-2 implementation of token bucket marking available.

7. REFERENCES

[1] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss. An Architecture for Differentiated Services, RFC 2475, December 1998.

[2] D.D. Clark, and W. Fang. Explicit Allocation of Best-Effort Packet Delivery Service, *ACM Transaction on Networking,* Vol. 6, Number 4, August 1998, pp. 362-373.

[3] D. Clark. The Design Philosophy of the DARPA Internet Protocols, *Proceedings of Sigcomm,* pp. 106-114, August 1988.

[4] W. Feng, D.D. Kandlur, D. Saha, and K.G. Shin. Adaptive Packet Marking for Providing Differentiated Services in the Internet, *Proc. of Intl. Conf. on Network Protocols,* October 1998.

[5] J. Ibanez, and K. Nichols. Preliminary Simulation Evaluation of an Assured Service, IETF Draft, August 1998.

[6] M. Goyal, A. Durresi, R. Jain, C. Liu. Performance Analysis of Assured Forwarding, IETF Draft October 1999.

[7] J. Padhye, V. Firoiu, D. Towsley, J. Kurose. Modeling TCP Throughput: A Simple Model and its Empirical Validation, *Proceedings of ACM Sigcomm,* October 1998, pp. 303-314.

[8] J. Heinanen, R. Guerin. A Two Rate Three Color Marker, IETF Draft, May 1999.

[9] NS Simulator, Version 2.1b5, available from http://www-mash.cs.berkeley.edu/ns.

[10] T. Ott, J. Kemperman, M. Mathis. The Stationary Behavior of Idea TCP Congestion Avoidance, *Preprint.*

[11] S. Sahu, D. Towsley, J. Kurose. A Quantitative Study of Differentiated Services for the Internet, *Proc. IEEE Global Internet, Globecom'99,* pp. 1808-1817.

[12] S. Sahu, P. Nain, D. Towsley, C. Diot, V. Firoiu. On Achievable Service Differentiation with Token Bucket Marking for TCP, CMPSCI TR-99-72, Univ. of Massachusetts, November 1999.

[13] I. Stoica, H.Zhang. LIRA: A Model for Service Differentiation in the Internet, *Proceedings of NOSSDAV,* July 1998.

[14] S. Floyd, V. Jacobson. Random Early Detection for Congestion Avoidance, *IEEE/ACM Transaction on Networking,* Vol. 1(4), pp. 397-413, July 1993.

[15] W. Stevens. *TCP/IP Illustrated, Vol. 1 The Protocols,* Addison-Wesley, 1994.

[16] W. Stevens. TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms. RFC2001, Jan 1997.

[17] R. W. Wolff. Stochastic Modeling and the Theory of Queues, Prentice-Hall.

[18] I. Yeom and A.L.N. Reddy. Modeling TCP Behavior in a Differentiated-Services Network, Texas A&M Technical Report, May 1999.

[19] I. Yeom and A.L.N. Reddy. Realizing Throughput Guarantees in a Differentiated-services Network, *Intl. Conference on Multimedia and Computing Systems,* June 1999.