Generalized TCP Congestion Avoidance and its Effect on Bandwidth Sharing and Variability

Archan Misra
archan@research.telcordia.com

John Baras
baras@isr.umd.edu

Teunis Ott
tjo@research.telcordia.com

Abstract—**To model possible suggested changes in TCP window adaptation in response to randomized feedback, such as ECN, we formulate a generalized version of the TCP congestion avoidance algorithm. We first consider multiple such generalized TCP flows sharing a bottleneck buffer under the Assured Service model and use a fixed point technique to obtain the mean window sizes and throughputs for the TCP flows. To further study how changes in the adaptation algorithm affect the variability in the throughput, we use an analytical-cum-numerical technique to derive the window distribution (and related statistics) of a single generalized flow under state-dependent randomized congestion feedback.**

I. Introduction

There has recently been a revival of interest in investigating how TCP's congestion avoidance algorithm can be modified to better utilize newer congestion control mechanisms, such as ECN (Explicit Congestion Notification) [3]. As a partial attempt to analyze possible changes to TCP's congestion avoidance algorithm, we consider the performance of *generalized TCP* flow control, as presented in [10], in a couple of scenarios. We first present a technique to determine the achieved throughput when multiple generalized TCP flows are buffered at a bottleneck buffer performing congestion notification via ECN. In particular, we present the analysis under the *Assured Service* [1] model, wherein each flow is associated with a minimum assured rate. By extending the fixed point analysis technique presented in [14], we obtain the mean TCP window sizes and the mean queue occupancy under this model. We then use these mean values to accurately compute the throughputs of the individual generalized TCP flows. By performing this analysis for different generalized adaptation parameters, we study how the different modifications to TCP congestion avoidance affect the relative sharing of the excess bandwidth.

To further understand how changes in TCP congestion avoidance affect the distribution of the window size (and hence the variability in the instantaneous rates), we also analyze a single generalized TCP flow subject to a state-dependent congestion notification probability. We use a generalization of the technique presented in [13] to determine the stationary window distribution and other window-related statistics in this case. By studying how such statistics vary as a function of the parameters of the generalized congestion avoidance algorithm, we deduce the relative importance of various suggested modifications on the behavior of the TCP source. In each case, simulations are used to verify the accuracy of our analytical technique. In particular, our analysis indicates that specifying a smaller reduction in the window size on receiving a congestion indicator can lead to a smaller relative fluctuation in the short-term throughput of a TCP flow.

The *generalized TCP* process increases its congestion window from the current value W by $c_1 W^\alpha$ on receiving an acknowledgement where the ECN feedback bit is not set and decreases its window by $c_2 W^\beta$ on receiving an acknowledgement where the ECN feedback bit is set. By disregarding the transients present in real TCP implementations (such as timeouts and fast recovery), we can model the window evolution of the idealized generalized TCP process $(W_n)_{n=1}^{\infty}$ by the equations:

$$P\{W_{n+1} = w + c_1 w^\alpha | W_n = w\} = 1 - p_m(w) \quad (1)$$

$$P\{W_{n+1} = w - c_2 w^\beta | W_n = w\} = p_m(w) \quad (2)$$

where α, β, c_1 and c_2 are the parameter constants (clearly $c_1, c_2 > 0$) and $p_m(w)$ denotes the probability of a packet being *marked* (ECN bit set) when the congestion window (expressed in MSSs) is w. The current TCP congestion avoidance algorithm has the parameter set $(\alpha = -1.0, c_1 = 1.0, \beta = 1.0, c_2 = 0.5)$; algorithms with $(\alpha = -1.0, \beta = 1.0)$ are referred to as subadditive-increase, multiplicative-decrease (SAIMD) in this paper. Also, the case of $(\alpha = 0, \beta = 1)$ $(c_2 < 1)$ has received significant attention in literature and is called additive-increase, multiplicative-decrease (AIMD) [1] here. Our analysis of generalized TCP behavior under the Assured Service model assumes that the bottleneck buffer behaves somewhat similar to the RIO mechanism presented in [1] except that it (randomly) *marks* only *out* packets (those that exceed the profiled rate) with an occupancy-dependent probability; *in* packets (those that stay within the assured profile) are never marked by the router and are dropped only due to buffer. Since this mechanism is similar to Random Early Detection (RED) [4] applied to only *out* packets, we shall call it ORED[2] (Out-RED) for convenience.

A. Motivation and Related Work

Under the current congestion avoidance scheme [2], a TCP flow increases its congestion window by 1 every round trip time in the absence of congestion and halves its congestion window on detecting congestion. This philosophy of conservative increase and rapid decrease was particularly appropriate for an Internet consisting of tail-drop queues where packet loss was

Archan Misra and Teunis Ott are with the Applied Research division of Telcordia Technologies, 445 South Street, Morristown, NJ 07960. John Baras is with the Center for Satellite and Hybrid Communication Networks at the University of Maryland, College Park, MD 20742.

[1] In literature, $(\alpha = -1, \beta = 1)$ is often referred to as AIMD, since the congestion avoidance algorithm results in a unit increase per round-trip time. However, we shall use the notation SAIMD to refer to the current congestion avoidance algorithm.

[2] Unlike classical RED, our router port provides randomized congestion notification exclusively through ECN marking. Strictly speaking, this is still within the purview of RED, which really stands for Random Early Detection (and not Random Early Drop).

0-7803-6451-1/00/$10.00 © 2000 IEEE

the sole indicator of congestion and where congestive losses occurred only when a link was subject to sustained overload, resulting in buffer overflow. However, such a drastic reduction leads to several problems with TCP traffic on the Internet:

- It makes the instantaneous rates of TCP traffic vary wildly, making it harder to stablize the queue occupancy in router buffers.
- The sharp drop in the transmission rate on detection of congestion leads to significant wastage of bandwidth, especially over high-speed large-latency routes, such as those involving satellite links.

Explicit Congestion Notification [3] has been proposed and standardized as a mechanism for faster and clearer congestion indication to adaptive flows. In this scheme, routers set a bit (*mark* the packet) in the packet header on the forward path on detecting congestion. The receiver echoes this bit in the acknowledgement packet; on receipt of an acknowledgement with the congestion bit set, the sender reduces its transmission rate appropriately. [3] required the TCP sender to treat an ECN indicator in the same manner as a lost packet. Given the significantly enhanced congestion signaling capacity of ECN, this requirement may indeed be called into question. Since TCP performance degrades rapidly when the packet loss rate exceeds $\sim 5\%$, feedback mechanisms based purely on packet drops cannot increase the maximum probability for random drops beyond this value. In contrast, since ECN does not cause packet drops, the associated marking probabilities can be much larger. This flexibility permits ECN to operate over a much wider range of randomized congestion indication; this, in turn, provides us an opportunity to reduce TCP's current drastic response to congestion signaled via ECN. It should be clear that prospective modification to TCP behavior needs to be closely coupled with the design of ECN mechanisms in router buffers. Several studies have considered the advantages of using the AIMD congestion control mechanism; [5] showed the optimality properties associated with a rate-based AIMD mechanism. [10] recently studied the behavior of a generalized TCP window (governed by equations (1) & (2)) as a function of the router marking probability and suggested reasons why AIMD ($\alpha = 0, \beta = -1$) might be a better model for TCP response to ECN feedback.

The Assured Service model [1] describes a framework for preferential bandwidth sharing, where each flow (user) is guaranteed a minimum or *assured* rate as part of their service profile. Adequate capacity provisioning is assumed to ensure that packets from a flow experience minimal congestive losses as long as its transmission rate lies within this assured rate. Flows are allowed to inject additional (opportunistic) packets beyond this assured rate; such packets are treated as best-effort and have lower priority. To enable network buffers to differentiate between such packets, [1] proposes a tagging mechanism at the network edge. Packets which stay within the profiled rate are tagged as *in* packets while packets that violate the profile are tagged as *out* packets; mechanisms such as a leaky bucket [18] or modifications thereof [1] may be used to implement the tagging operation. *In* packets are provided preferential treatment in network buffers via the RIO (RED with In/Out) discard algorithm; RIO is similar to RED except that it uses different thresholds for *in* and *out* packets to ensure that *out* (opportunistic) packets were dropped before *in* packets. Limitations on the practical implementation of this service model using current TCP implementations have been reported. For example, accurate accurate differentiation based on tagging requires the tagging function to be embedded in the source (host node). Also, the practice of dropping *out* packets via RIO has been shown to cause some unfairness towards flows with larger rate profiles, primarily because of TCP's drastic rate reduction in response to packet drops. Note that the Assured Service model can be practically implemented in the Different Services paradigm [6], through the appropriate use of the Assured Forwarding [7] per-hop behavior.

The analytical approach for estimating the mean window sizes and throughputs of individual TCPs is based on the modification of a mean value-based technique presented in [14], which considered the case of multiple TCP flows, implementing the current congestion avoidance algorithm, interacting with a random drop queue. To derive the probability distribution (and hence, other statistics such as the variance) of the congestion window of a generalized TCP flow (for $\beta = 1.0$), we modified the approach described in [13], which presented an analytical-cum-numerical technique to compute the TCP window distribution when congestion was signaled via packets dropped with a *state-dependent* probability. A purely analytical computation of the window distribution of an idealized TCP flow subject to packet drops with a *constant* drop probability was presented in [12].

II. Mathematical Models

Let N be the number of TCP flows which are sharing the router buffer. We assume that each TCP source is persistent (has infinite data to send) and transmits packets in equal sized segments (different flows can have different segment sizes). The i^{th} TCP flow is assumed to have a nominal round-trip time (excluding the queuing delay in the bottleneck buffer) of RTT_i secs and a segment size (MSS) of M_i bytes. We shall let W_i denote the window size of the i^{th} flow in MSSs; $W_i * M_i$ will then provide the window size of the i^{th} flow in bytes. The i^{th} flow has an *assured* rate of R_i bytes/sec and can consequently expect to receive no congestion feedback as long as its transmission rate is less than R_i.

We consider the generalized TCP window adjustment paradigm. As presented in [10], a process acting in this paradigm can be thought of as increasing its window by a function $incr(W)$ on receiving an acknowledgement in the absence of congestion and decreasing its window by $decr(W)$ on receiving an acknowledgement indicating congestion. For the discussion at hand, we restrict these functions such that:

$$\begin{aligned} incr(W) &= c_1 W^{\alpha} \\ decr(W) &= c_2 W^{\beta}, \end{aligned}$$

where α, β, c_1 and c_2 are constants that parametrize the flow control algorithm. Although our analytical technique holds even when different TCP flows have different parametric values, we restrict ourselves in this paper to the case where all flows use identical values of α, β, c_1 and c_2. As stated earlier, our router buffer implements an algorithm which we call ORED. The bandwidth of the bottleneck link serving the buffer is denoted by C bytes/sec. Our analysis assumes that [3]

$$C > \sum_{i=1}^{N} R_i. \tag{3}$$

The queue randomly sets the ECN bit on *out* packets with a probability based on the buffer occupancy. Since *in* packets are never marked, the only possible loss of *in* packets occurs due to buffer overflow. The model thus essentially assumes that marking *out* packets with a sufficiently aggressive probability is adequate to ensure that the window sizes of the connections do not grow without limit. Mathematically speaking, this assumes that $lim\ w \uparrow \infty\ \frac{incr(W)}{decr(W)} \rightarrow 0$. i.e., while $\alpha < \beta$, which is true in all practical cases of interest. Again, although our analysis holds for any queue where the the marking probability is a non-decreasing function of the buffer occupancy, we use the standard RED linear marking model for concreteness. Hence, the marking probability f_{mark} for *out* packets is given by:

$$\begin{aligned} f_{mark}(Q) &= 0 \quad for\ Q < min_{th} \\ &= p_{max} * \frac{Q - min_{th}}{max_{th} - min_{th}} \quad for\ min_{th} \le Q < max_{th} \\ &= p_{max} \quad for\ Q > max_{th} \end{aligned}$$

where min_{th} and max_{th} are the minimum and maximum marking thresholds and p_{max} is the maximum marking probability. Of course, p_{max} can now be much larger than conventional RED queues, since packets are only marked and not dropped.

III. Mean Window Sizes and Throughputs for Multiple Generalized TCPs

To estimate the mean TCP window sizes and their achieved throughputs when N generalized flows interact with an ORED queue, we use drift analysis techniques to define a set of fixed point equations. We first formulate the set of equations defining the fixed point and then solve them using a gradient and binary search-based technique. We finally provide comparisons of our numerical predictions with simulated values to validate our analysis. It should be clear that under condition (3), each flow will obtain at least its profiled rate, as otherwise it would never have any packet tagged as *out* and hence, would have its congestion window increase without bound.

A. Characterizing the Fixed Point

Following the approach in [14] and [10], we define the *drift* in the congestion window of the i^{th} flow by the expected change, ΔW_i, in its window size as a function of its window size W_i. The window size increases by $c_1 W_i^{\alpha}$ with a probability $1 - p_i(W)$ and decreases by $c_2 W_i^{\beta}$ with a probability $p_i(W)$, where $p_i(W)$ is the probability of a packet being marked (ECN bit set). Thus, the drift is 0 (corresponding to the 'mean' or center of the window) when W_i satisfies the condition:

$$c_1 W_i^{\alpha} * (1 - p_i(W_i)) = c_2 W_i^{\beta} * p_i(W_i). \tag{4}$$

Accordingly, given a specific function $p_i(.)$, we can obtain the mean value of the congestion window by solving:

$$\frac{c_2}{c_1} W_i^{\beta-\alpha} = \frac{1 - p_i(W_i)}{p_i(W_i)}. \tag{5}$$

Clearly, relation (5) defines a set of N equations for $i = 1, \ldots, N$.

If the mean ORED buffer occupancy is Q (bytes), we can determine the corresponding function $p_i(.)$. In this case, the marking probability for *out* packets is given by $f_{mark}(Q)$[4]. Now, if a fraction γ_i of the packets from flow i are marked as *out*, the unconditional marking probability for packets of flow i is $\gamma_i * f_{mark}(Q)$. Unfortunately, when more than 1 TCP flow is present, γ_i is itself a function of both W_i and Q. To see this, note that, when the queue occupancy is Q, the total round-trip time for flow i is given by $RTT_i + \frac{Q}{C}$. Since the flow control algorithm transmits $W_i * M_i$ bytes every round-trip time, the achieved throughput ρ_i is given by

$$\rho_i = \frac{W_i * M_i}{RTT_i + \frac{Q}{C}} \tag{6}$$

The probability of a packet being tagged as *out* is assumed to be equal to the fraction by which the achieved throughput exceeds the assured rate R_i. γ_i is thus given by $\gamma_i = \frac{\rho_i - R_i}{\rho_i}$ or, upon using equation (6):

$$\gamma_i = 1 - \frac{R_i * (RTT_i + \frac{Q}{C})}{W_i * M_i}. \tag{7}$$

Accordingly, the marking probability $p_i(W_i)$ is given by $p_i(W_i) = (1 - \frac{R_i * (RTT_i + \frac{Q}{C})}{W_i * M_i}) * f_{mark}(Q)$, which on substituting into equation (4) yields the following relationship (one for each $i = (1, \ldots, N)$)

$$\frac{c_2}{c_1} W_i^{\beta-\alpha} = (1 - \frac{R_i * (RTT_i + \frac{Q}{C})}{W_i M_i} * f_{mark}(Q))^{-1} - 1 \tag{8}$$

We denote the solution for W_i of the above equation as $h_i(Q)$ to explicitly indicate that the above equation is really a function of the queue occupancy Q. We shall elaborate on a technique for solving the above equation (to obtain $h_i(Q)$) in the next subsection.

Given a value for Q, we can then (at least in principle) solve the set of N equations (equation (8) for $i = 1, \ldots, N$) to obtain the N values, $h_i(Q)$, $i = 1, \ldots, N$. However, our solution must satisfy another constraint: assuming that no queue underflow occurs (after all, this is a bottleneck queue), the sum of the throughputs of the N flows must equal to the link capacity C, i.e., $\sum_{i=1}^{N} \rho_i = C$. For a specific value of Q, we note that $\rho_i = \frac{h_i(Q) * M_i}{RTT_i + \frac{Q}{C}}$ and hence, after trivial algebraic manipulations arrive at the other constraint:

$$\sum_{i=1}^{N} \frac{h_i(Q) * M_i}{Q + RTT_i * C} = 1 \tag{9}$$

[3] If $C < \sum_{i=1}^{N} R_i$, then packet drops (or ECN marks) will occur even before the TCPs flows obtain their assured rate. This case can be analyzed using the approach in [14].

[4] As in [13], our formulation can also be used when different flows have marking probabilities that are scalar multiples of each other, i.e., $f^{i}_{mark}(Q) = \kappa_i f_{mark}(Q)$ where κ_i are arbitrary constants. We do not explore this scenario in this paper.

The basis of our fixed-point theory should now be clear. As we vary Q and solve for the $h_i(Q)$ according to expression (8), there will be one value for which the constraint (9) is satisfied. This value of the queue occupancy is denoted by Q^*. The corresponding solutions for $h_i(Q^*)$ provides the theoretical mean window sizes W_i^*; the corresponding throughput for connection i is then computed by $\frac{W_i^* * M_i}{RTT_i + \frac{Q}{C}}$. The existence of a unique solution can be verified by varying Q from min_{th} to ∞. At values close to min_{th}, $f_{mark}(Q) \approx 0$ and hence, from equation (8), we see that $h_i(Q)$ will be very large. Accordingly, the LHS of equation (9) will be much larger than 1. On the other hand, as $Q \uparrow \infty$, the value of $h_i(Q)$ also increases (since it is clearly always larger than $R_i * (RTT_i + \frac{Q}{C})$). In that case, if we neglect the constant term of 1 in the RHS of equation (8), we can easily see, after elementary manipulation, that the expression (8) reduces to

$$\frac{c_2 M_i}{c_i} * W^{\beta-\alpha} = \frac{c_2}{c_1} * R_i * (RTT_i + \frac{Q}{C}) W^{\beta-\alpha-1} + M_i \quad (10)$$

which, for large values of Q and W_i, yields

$$W_i * M_i = h_i(Q) * M_i \approx R_i * (RTT_i + \frac{Q}{C}) \quad (11)$$

By plugging expression (11) into the LHS of constraint (9), we can see that the LHS turns out to be equal to $\frac{\sum_{i=1}^{N} R_i}{C}$. But by our assumption (3), this is clearly less than 1. We can further show that as Q increases from min_{th} to ∞, the LHS of (9) decreases monotonically and crosses 1 at some point. Such a value of Q accordingly defines the unique solution of the fixed point.

B. Solving the Fixed Point

Our algorithm for solving the fixed point essentially consists of varying Q and solving for $h_i(Q)$ until the condition (9) is satisfied.

An iterative gradient scheme (based on the Newton method) can be used to solve for $h_i(Q)$. A value of W_i that satisfies equation (8) is essentially the unique zero of the function $g(W)$ defined by

$$(1 - \frac{R_i * (RTT_i + \frac{Q}{C})}{W_i M_i} * f_{mark}(Q))^{-1} - 1 - \frac{c_2}{c_1} W_i^{\beta-\alpha} \quad (12)$$

Define $g_1(W_i) = (1 - \frac{R_i * (RTT_i + \frac{Q}{C})}{W_i M_i} * f_{mark}(Q))^{-1} - 1$ and $g_2(W) = \frac{c_2}{c_1} W_i^{\beta-\alpha}$. By taking derivatives, we can see that $g_1(W_i)$ is convex and decreasing in W_i while $g_2(W_i)$ is increasing in W_i (since $\beta > \alpha$). Furthermore, if $\beta - \alpha < 1$, then $g_2(W_i)$ is also concave. Accordingly, we start with a value of W_i slightly larger than $R_i * (RTT_i + \frac{Q}{C})$ and repeat the iterations until we converge. In particular, if $\beta - \alpha \leq 1$, $g(W_i)$ is convex and hence, we can guarantee convergence without any overshoot. When $\beta - \alpha > 1$, we have the possibility of overshoot and hence, need to take special care in our numerical procedure. However, in all our numerical calculations, we were able to attain convergence using the Newton iterative method using the iteration

$$W_i^{j+1} = W_i^j + \frac{g(W_i^j)}{g'(W_i^j)}. \quad (13)$$

The appropriate value for Q i.e., Q^*, on the other hand, can be obtained by a binary search procedure, since we have established that $\sum_{i=1}^{N} \frac{h_i(Q) * M_i}{RTT_i + \frac{Q}{C}}$ is monotonically decreasing and smaller than C when $Q > Q^*$ and larger than C when $Q < Q^*$. Thus, the entire algorithm consists of two loops: an outer loop varying Q via a binary search method and an inner loop evaluating $h_i(Q)$ via the Newton gradient method.

C. Simulations and Comparative Results

We performed fairly extensive tests, using a modified version of the $ns - 2$ simulator [17] to compare the accuracy of our analytical/numerical results with those obtained via simulations. The modifications included incorporation of the generalized $incr(W)$ and $decr(W)$ functions in the TCP code and augmentation of the RED code to implement the ORED mechanism.

To better illustrate our results, and also to understand how changes in the adaptation parameters affect the sharing of the excess capacity, we concentrate on the case of only 2 generalized flows. (We have however used between $2-20$ TCP flows in additional simulations to verify the accuracy of our technique.) Both flows had the same segment size of 512 bytes. To provide illustrative results, we use four parameter sets, two belonging to the SAIMD paradigm and two belonging to the AIMD paradigm:

1. Parameter Set 1: ($\alpha = -1, \beta = 1, c_1 = 1, c_2 = 0.5$), i.e., the current TCP window adaptation procedure.
2. Parameter Set 2: ($\alpha = 0, \beta = -1, c_1 = 0.2, c_2 = 0.1$), i.e., an interesting choice of AIMD parameters.
3. Parameter Set 3: ($\alpha = -1, \beta = 1, c_1 = 0.5, c_2 = 0.1$), i.e., SAIMD with a reduction in the coefficients for window increase and decrease.
4. Parameter Set 4: ($\alpha = 0, \beta = 1$,i $c_1 = 0.4$ and $c_2 = 0.2$), i.e., AIMD with larger coefficients for window increase and decrease than parameter set 2.

The link capacity was varied between $4.5 - 12$ Mbps. While max_{th} and min_{th} was maintained at 20 and 100 respectively for both parameter sets, p_{max} was kept at 0.01 for parameter set 1 and 3, and at 0.1 for parameter sets 2 and 4. This was done to ensure reasonable mean window sizes: for identical marking probabilities, parameter sets 2 and 4 would have much larger mean window sizes than parameter sets 1 and 3. We present here the results of two different experiments, designed to study two different performance characteristics of generalized TCP flows.

In the first set of experiments, which we shall call **Experiment** A, we kept the round-trip times identical for both flows but provided them different profiled rates. TCP flow 1 had a profile of 1.5 Mbps and TCP 2 had a profile of 3 Mbps. Both flows were tagged by a leaky bucket-based conditioner with a moderate bucket size of 20 packets. Figure 1 shows the theoretical and simulated TCP mean window sizes/ throughputs for parameter set 1 as the link capacity C is varied. Figure 2 shows the corresponding plots for parameter set 2 (we do not provide plots for the other parameter sets due to space limitations). The figures show remarkably close agreement between our analytical predictions and the simulated results. We conducted similar experiments where N varied from $2 - 20$; our predictions were always within 5% of the values obtained via simulations.

In the second set of experiments, which we shall call **Experiment B**, the two TCP flows had identical profiled rates (1.5 Mbps) but different round-trip times. Flow 1 had an RTT of 20 msec while flow 2 had an RTT of 100 msec. Figure 3 shows the theoretical and simulated TCP mean window sizes/ throughputs for parameter set 1 as the link capacity C is varied; we see the close agreement between the analytical predictions and the simulated values. Similar agreement is obtained with the other parameter sets; we omit the figures due to space constraints.

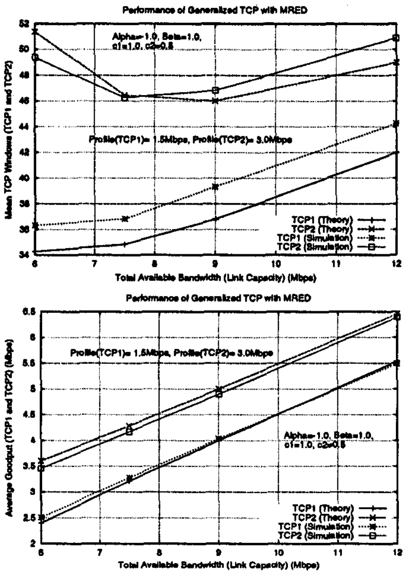

Figure 1: Mean Window Sizes and Throughputs for Parameter Set 1 (Different Rate Profiles)

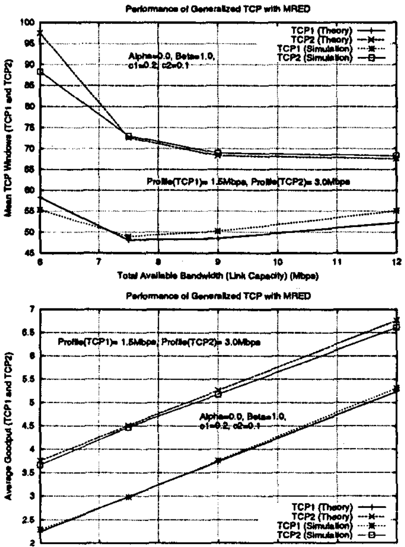

Figure 2: Mean Window Sizes and Throughputs for Parameter Set 2 (Different Rate Profiles)

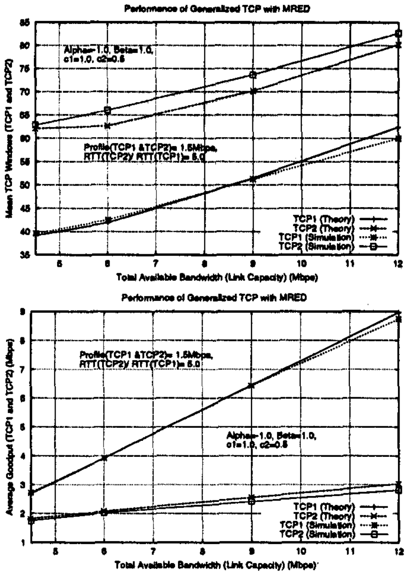

Figure 3: Mean Window Sizes and Throughputs for Parameter Set 1 (Different RTT values)

We can also use this analytical technique to study another interesting question: *how do the TCP flows share the excess capacity (i.e., $C - \sum_{i=1}^{N} R_i$) in this service model and how do changes to the parameters affect this relative sharing?* The relative sharing of the excess capacity is certainly of secondary importance in the Assured Service model, which merely seeks to provide a minimum rate guarantee to each flow. Proportional sharing and differentiation is, however, an interesting alternative for service differentiation; for example, the User Service Differentiation (USD) [8] model advocates a framework where the available bandwidth is simply apportioned among the constituent flows in the ratio of the assigned weights. We now use experiments A and B outlined earlier to study whether certain choices of parameters in the generalized congestion avoidance procedure are more effective in dividing the excess bandwidth among the flows in the ratio of their assured rates.

In Experiment A, the assured rate of TCP flow 2 is twice the assured rate of TCP flow 1. We use both our mean value-based analysis technique as well as simulations to study how the ratio of the achieved TCP throughputs varies as a function of the window adjustment parameter sets and the amount of the excess bandwidth. Figure 4 show the simulation and theoretical results separately. We can see that as the excess bandwidth increases, the excess is never shared in the ratio of the profiled rates. Rather, as the excess capacity (the sum of the profiles is 4.5 Mbps) is increased, this excess is increasingly evenly distributed among the two competing flows, as a result of which the ratio of the attained throughputs decreases from 2 towards 1. Also, more importantly, we see that parameter sets 2 and 4 ($\alpha = 0$, AIMD) provide a closer conformance to the proportional sharing model than parameter sets 1 and 3 ($\alpha = -1$, SAIMD). Furthermore, although our theory indicates that the ratio of the throughputs depends only on the ratio c_1 to c_2, we see that, in practice, a lower value of c_2 (a less drastic reduction in the window size on receiving congestion indication) provides for larger differentiation.

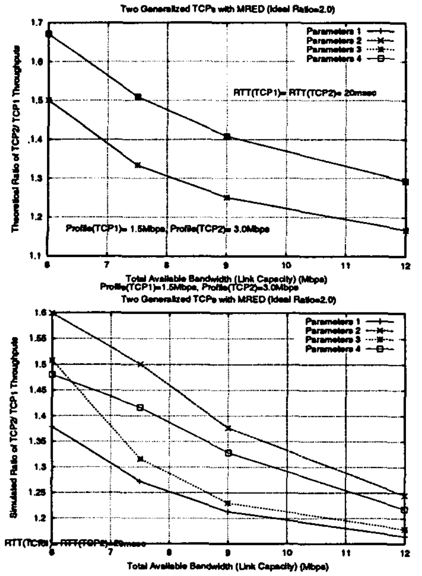

Figure 4: Ratio of Attained Throughput for Different Parameter Sets (Different Rate Profiles)

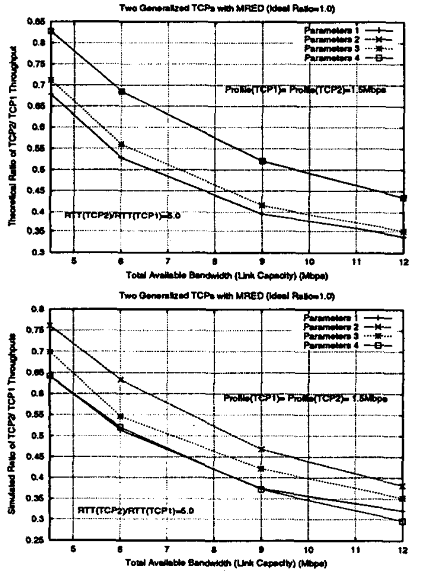

Figure 5: Ratio of Attained Throughput for Different Parameter Sets (Different RTT)

Figure 5 shows the ratio of the obtained throughputs (both theoretical and practical) for Experiment B. As stated earlier, since the two TCP flows had identical assured rates (1.5 Mbps each), the ratio of their throughputs would ideally be 1. Given the inherent bias of window-based algorithms against longer RTT connections, we can expect the lower RTT TCP connection (TCP1) to obtain the greater share of the excess bandwidth. The graphs in figure 5 do indeed confirm this phenomenon. More importantly, as with experiment A, they illustrate that an AIMD algorithm ($\alpha = 0$) provides for a closer fit to the proportional model of capacity sharing than an equivalent $SAIMD$ adjustment procedure. Also, as in Experiment A, specifying a smaller value of c_2 leads to a closer approximation to the proportional model.

D. Salient Features of Analysis

We have extended the mean value analysis presented in [14] to obtain the individual throughputs when multiple generalized TCP flows interact with an ORED queue under the Assured Service framework. A variety of simulation experiments validate the accuracy of our analysis.

We have also evaluated how modifying the parameters of generalized congestion avoidance affects the proportional sharing of excess bandwidth. While it is hard to draw generic conclusions, it does appear that an AIMD adjustment procedure leads to a closer and more robust approximation to the proportional sharing model than a comparable SAIMD algorithm. Our simulations also demonstrate that a smaller value of c_2 results in a closer approximation to the proportional model; bandwidth sharing with a smaller c_2 is also more robust to variations in RTT. In the next section, we shall see that this is really the result of smaller variance in the congestion window size.

IV. Window Distribution and Analysis of a Generalized TCP Process ($\beta = 1$)

To further study the implications of changing the window adjustment parameters in ECN-enabled TCP, we now consider the special case of a *single* TCP flow, being regulated by an ORED buffer under the Assured Service model. [10] presented an analysis of the window distribution for a generalized TCP flow subject to a *constant* marking probability. While our analytical technique can be applied only when $\beta = 1$, we believe that this is not a significant restriction as almost all popular flow control algorithms use multiplicative-decrease ($\beta = 1$). We show how to characterize the window evolution of a single generalized TCP flow and provide the mathematical technique to obtain the window *distribution* in this case. We then compare the accuracy of our analytical predictions with simulation results and use such studies to further understand the implications of changing TCP's window adjustment procedure.

A. Formulating the Window Evolution Model

As before, consider a TCP flow with a round-trip time of RTT secs and a segment size of M (the sub-scripts being dropped since only one flow is considered here). It interacts with an ORED buffer serving a link of capacity C (which, for notational efficiency, is now expressed in segments/sec) and has an assured bandwidth of R (also in segments/sec). Also, let Q, the buffer occupancy, and min_{th} and max_{th} (the ORED thresholds) be similarly similarly expressed in segments. Our aim is to find the stationary distribution of the stochastic process $(W_n)_{n=1}^{\infty}$.

If, as before, we assume that buffer underflow never occurs, it is clear that the TCP average transmission rate will be equal to the link capacity C. The packet tagging probability, γ, is then independent of W and Q, and is simply given by the fraction by which the capacity exceeds the profiled rate

$$\gamma = \frac{C-R}{C} \tag{14}$$

Also, since we assume that the buffer never underflows, 'the pipe is always full' and hence, the window size and the queue occupancy are related according to

$$W = Q + C * RTT \tag{15}$$

Now consider the evolution of the TCP generalized window. It is easy to see that although packets will be tagged as *out* as soon as the TCP throughput exceeds R, they will not be marked (ECN bit set) until the window has expanded to ensure that the queue occupancy exceeds min_{th}; this, of course, occurs only after the throughput has reached the bottleneck bandwidth C and the window size has exceeded $C * RTT + min_{th}$. Accordingly, a reasonably accurate model of the marking probability $p(W)$, as a function of the window size W, is given by the equations

$$\begin{aligned} p(W) &= 0 \quad for\ W < min_{th} + C.RTT, \\ &= \gamma * f(W - C.RTT)\ for\ W < max_{th} + C.RTT \\ &= \gamma * p_{m\bar{a}x} \quad for\ W > max_{th} + C.RTT, \end{aligned} \tag{16}$$

where $\gamma = \frac{C-R}{C}$. The conditional transition probability of the generalized TCP process is thus as follows:

$$\begin{aligned} Prob(W_{n+1} = W_n + c_1 W_n^{\alpha} | W_n \leq W^*) &= 1 \\ Prob(W_{n=1} = W_n + c_1 W_n^{\alpha} | W_n > W^*) &= 1 - p(W) \\ Prob(W_{n=1} = W_n - c_2 W_n^{\beta} | W_n > W^*) &= p(W) \end{aligned} \tag{17}$$

where $W^* = min_{th} + C * RTT$.

B. Solving the Stochastic Process

The window evolution process characterized by the equations (17) is clearly a state-dependent model. We accordingly use the technique presented in [13] (which considered the special case of current TCP congestion avoidance). This approach uses a set of state-dependent mappings to define an associated process $X(t)$, which can be characterized by a differential equation between Poisson points of failure. Although space limitations prevent us from furnishing all the steps, we provide the space-and-time rescalings which are necessary in this generalized case.

The analysis consists of deriving a process $X(t)$ through the following state and time mappings:

$$X(t) = p_{max}^{\frac{1}{1-\alpha}} W_n \tag{18}$$

$$\Delta t = p(W_n)\Delta n \tag{19}$$

While the space-rescaling is a constant, the time-rescaling is state-dependent; the resulting time-frame t is referred to as *subjective time*. Subjective time is a non-linear, invertible contraction of the TCP time index n.

Proposition 1: It can be shown (using arguments similar to [13]), that as $p_{max} \downarrow 0$, the process $X(t)$ has the following description:
There is a Poisson process with intensity 1, with points denoted by $(\tau_n)_{n=1}^{\infty}$. In between the points of this Poisson process, X evolves according to the equation

$$\frac{dX}{dt} = \frac{c_1 * p_{max} * X^{\alpha}}{\gamma * f_{mark}(\frac{X}{c_1^{\frac{1}{1-\alpha}}} - C * RTT)}. \tag{20}$$

Let $q(X)$ denote the inverse of the RHS of equation (20). At the points of the realization of the Poisson process[5], we have

$$X(\tau^+) = X(\tau^-) * (1 - c_2)$$

◇

Once we have obtained a process $X(t)$ as above, we can then apply the numerical techniques presented in [13] for solving for the stationary distribution of $X(t)$. Briefly, the technique consists of showing that the cumulative distribution function for $X(.)$, denoted by $F_s(x)$, satisfies the differential equation

$$\frac{dF_s(x)}{dx} = q(x) * \{F_s(\frac{x}{1-c_2}) - F_s(x)\}.$$

The above relation on $F_s(x)$ is transformed into an equivalent equation for a function $H(x)$, defined by the relation $H(x) = (1 - F_s(x)) * e^{-\int_0^x q(x)dx}$. $H(x)$ is then solved using an iterative technique that was shown to be stable and rapidly convergent. Once $H(x)$ (and thereby $F_s(x)$) has been computed, the distribution for W_n is computed by reversing the space and time rescalings employed. Of course, one has to use caution to account for the state-dependent nature of the time scaling used. The interested reader is referred to [13] for further details.

C. Results

To illustrate the accuracy of our analysis, we take TCP's current window adjustment algorithm ($\alpha = -1, \beta = 1, c_1 = 1$ and $c_2 = 0.5$) as a baseline parameter set and vary each of the three parameters α, c_1 and c_2 in turn. A set of typical results are provided here, for the following network parameters: an MSS of 512 bytes, nominal RTT of 13.66 msec, an assured rate of 0.75 Mbps and an ORED queue with a service rate of 3 Mbps (the bandwidth-delay product is thus 5 segments), $min_{th} = 15$, $max_{th} = 95$ and $p_{max} = 0.02$.

Figure 6 shows the simulated and theoretical mean and variance of the window size of the TCP flow as a function of α and attests to the accuracy of our analysis. To further demonstrate the accuracy of our numerical technique, we also include a plot comparing the predicted and simulated window *distribution* for $\alpha = -1.0$. We see that increasing α from the current value of -1, i.e., SAIMD, to a larger value (say 0, i.e., AIMD) not only increases the mean window size but also the the coefficient of variation (defined as $\frac{Std.Deviation(W)}{Mean(W)}$). A larger coefficient of variation implies a larger relative variation in the short-term transmission rate; thus, making TCP more aggressive in its increment process can lead to higher fluctuation in the short-term

[5] It is at a point τ of the Poisson process that the condition $\beta = 1$ is required. If $\beta \neq 1$, then $X(\tau^+)$ becomes ill-defined as p_{max} (and by implication, $f_{mark}(.)$) tends to 0.

throughput. Note also that our technique becomes less accurate as α increases. A larger α implies a larger mean queue occupancy and hence a larger average marking probability; accordingly, our mathematical approximation, which is clearly based on the limiting process as $p_{max} \downarrow 0$, will be progressively less applicable.

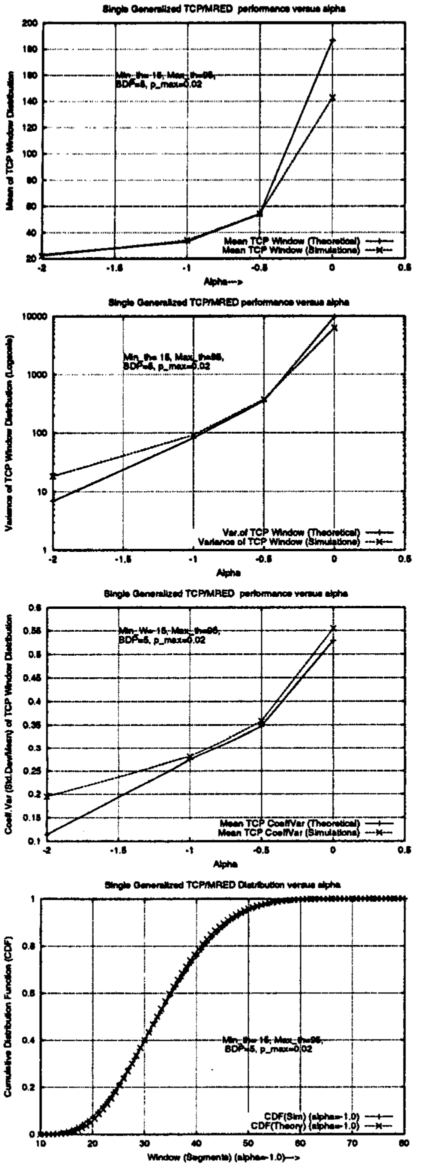

Figure 6: Variation of TCP Window Statistics (and Distribution) with α

We have similar studied the window statistics and distribution by varying c_1 and verified the accuracy of our technique. The figures do not provide any great insight and are thus omitted here. In general, we find that increasing c_1 increases not just the mean but the coefficient of variation as well. ([10] showed that the coefficient of variation would ideally be independent of c_1 if the marking probability was constant.) While decreasing c_1 might thus appear attractive, such an action retards the rate of window growth and consequently slows TCP's ability to utilize any unused capacity. Changes to c_1 should thus be considered only in conjunction with changes to the other parameters.

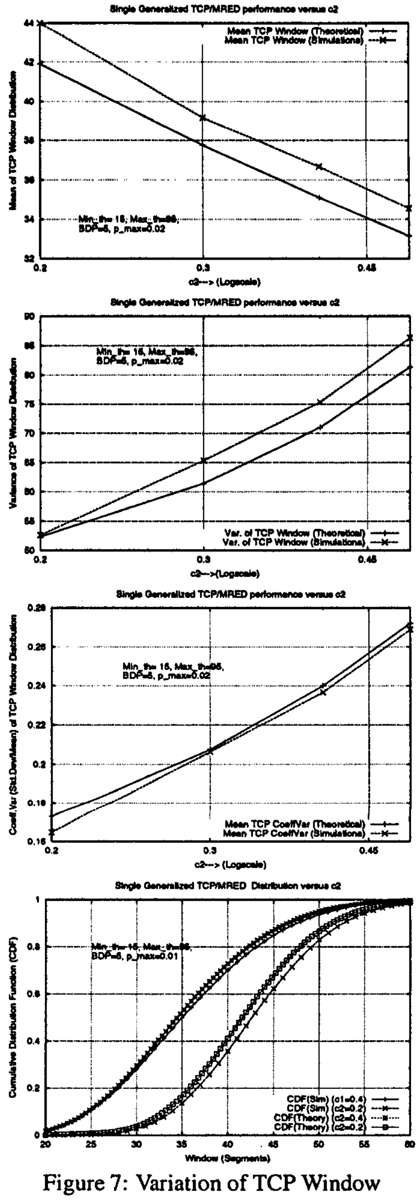

Figure 7: Variation of TCP Window Statistics (and Distribution) with c_2

Figure 7 shows the plots of the TCP window statistics (as well as the simulated and theoretical distributions for $c_2 = 0.2$ and $c_2 = 0.4$) when the decrease coefficient c_2 is varied. (Note that [10] showed that the coefficient of variation is proportional to $\sqrt{\frac{1}{1-c_2}}$, when the marking probability is constant.) It is most interesting to note that as c_2 is decreased from its current value of 0.5, the mean window size increases but the variance decreases, i.e., *the coefficient of variation decreases rapidly*. Thus, decreasing the multiplicative decrease coefficient c_2 appears to provide a much tighter control on the TCP window. While decreasing this factor does imply a less drastic reduction in the window size on receiving a single congestion indicator, routers can affect the same overall amount of window decrease by simply adopting a larger marking probability. As stated earlier, since ECN does not cause packet losses, the packet marking probability can be arbitrarily large. In fact, this decrease in the coefficient of variation explains our earlier observations on how a smaller c_2 helped to achieve throughput ratios closer to the proportional sharing model.

V. Conclusion

In this paper, we investigated how possible changes in TCP's current response to ECN-based congestion feedback might affect the distribution of bandwidth among multiple flows and the variation in the throughput of a single flow. To investigate this issue, we considered a generalized TCP window adjustment procedure, where the window is increased by $c_1 W^{\alpha}$ in the absence of congestion and decreased by $c_2 W^{\beta}$ in the presence of congestion.

We first analyzed the Assured Service model, when multiple generalized TCP flows interact with a queue that marks out-of-profile packets with an occupancy-dependent probability. Using a mean value-based fixed point iterative technique, we computed the mean TCP window sizes and TCP throughputs; simulations were used to verify the accuracy of our analysis. Our analysis indicates that the use of an additive-increase, multiplicative-decrease window adjustment paradigm results in a closer approximation to the proportional sharing of excess bandwidth than an equivalent sub-additive-increase, multiplicative-decrease window adjustment algorithm.

We then considered the case of a single generalized TCP flow (with $\beta = 1$) under the Assured Service model and provided an analytical-cum-numerical technique to evaluate the window distribution in this case. We used this technique to study the dependence of the window statistics on α, c_1 and c_2. In particular, we showed that decreasing the multiplicative decrease coefficient (from the current TCP value of 0.5) leads to a sharp decrease in the coefficient of variation and is probably the most important recommended modification to the current TCP algorithm. Although such a decrease reduces the effect that marking a single packet has on the buffer occupancy, buffers can achieve the same level of congestion control by simply increasing the marking probability. This observation illustrates the importance of designing marking functions in buffers in tandem with modified window adaptation algorithms at the TCP sources. Accordingly, in the near future, we intend to relate the marking function in an ECN queue to various studies on the window adjustment parameters.

References

[1] D Clark and W. Fang, 'Explicit Allocation of Best Effort packet Delivery Service', IEEE/ACM Transactions on Networking, August, 1998,

[2] V Jacobson, 'Congestion Avoidance and Control', Proceedings of SIGCOMM 1988.

[3] K K Ramakrishnan and S Floyd, 'A Proposal to add Explicit Congestion Notification (ECN) to IP', RFC 2481, January 1999

[4] S Floyd and V Jacobson, 'Random Early Detection Gateways for Congestion Avoidance', IEEE/ACM Transactions on Networking, August 1993.

[5] D M Chiu and R Jain, 'Analysis of the Increase and Decrease Algorithms for Congestion Avoidance in Computer Networks". Computer Networks and ISDN Systems 17, 1989.

[6] S Blake, D Black, et al, 'An Architecture for Differentiated Services', RFC 2475, December 1998.

[7] J Heinanen, F Baker, W Weiss and J Wroclawski, 'Assured Forwarding PHB', RFC 2597, June 1999.

[8] Z Wang, 'USD: Scalable Bandwidth Allocation for the Internet', Proceedings of HPN'98, Vienna, 1998.

[9] B Braden, D Clark et al, 'Recommendations on Queue Management and Congestion Avoidance on the Internet', RFC 2309.

[10] T Ott, 'ECN Protocols and the TCP Paradigm', ftp://ftp.telcordia.com/pub/tjo/ECN.ps.

[11] S Floyd, 'TCP and Explicit Congestion Notification', ACM Computer Communication Review, October, 1994.

[12] T Ott, M Matthis and J Kemperman, 'The Stationary Behavior of Idealized Congestion Avoidance', ftp://ftp.bellcore.com/pub/tjo/TCPwindow.ps, August 1996.

[13] A Misra and T Ott, 'The Window Distribution of Idealized TCP Congestion Avoidance with Variable Packet Loss', Proceedings of Infocom '99, March 1999.

[14] A Misra, T Ott and J Baras, 'The Window Distribution of Multiple TCPs with Random Loss Queues', Proceedings of Globecom '99, December 1999.

[15] S Floyd, 'Connections with Multiple Congested Gateways in Packet-Switched Networks Part 1: One-way Traffic', Computer Communication Review, Vol.21, No. 5, October 1991.

[16] V Jacobson, 'Modified TCP congestion avoidance algorithm', April 30, 1990, end2end-interest mailing list.

[17] The ns-2 network simulator, http://www-mash.CS.Berkeley.EDU/ns.

[18] M Schwartz, 'Broadband Integrated Networks', Prentice Hall, 1997.