June 2004

Congestion Controllers for High Bandwidth Connections with Fiber Error Rates

Egemen Kavak
University of Pennsylvania

Srisankar S. Kunniyur
University of Pennsylvania, kunniyur@seas.upenn.edu

Congestion Controllers for High Bandwidth Connections with Fiber Error Rates

Abstract
The inefficiency of a TCP connection in the presence of high bandwidth links due to the constant multiplicative decrease factor has been well documented in recent literature. In this paper we look at the effect of fiber error rates on the throughput of a TCP connection. We propose a congestion controller that removes the ill-effects of fiber error rates on TCP throughput by lower bounding the marking probability. We show that this congestion controller can achieve extremely high utilizations in high bandwidth links. We also discuss the TCP friendliness of this congestion controller and present simulation results that validate our analysis.

Congestion Controllers for High Bandwidth Connections with Fiber Error Rates

Egemen Kavak
Department of Electrical and Systems Engineering
University of Pennsylvania
Email: kavak@seas.upenn.edu

Srisankar S. Kunniyur
Department of Electrical and Systems Engineering
University of Pennsylvania, Philadelphia, PA 19104
Email: kunniyur@seas.upenn.edu; Tel: (215) 898 3559

***Abstract*— The inefficiency of a TCP connection in the presence of high bandwidth links due to the constant multiplicative decrease factor has been well documented in recent literature. In this paper we look at the effect of fiber error rates on the throughput of a TCP connection. We propose a congestion controller that removes the ill-effects of fiber error rates on TCP throughput by lower bounding the marking probability. We show that this congestion controller can achieve extremely high utilizations in high bandwidth links. We also discuss the TCP friendliness of this congestion controller and present simulation results that validate our analysis.**

I. Introduction

The huge success of the Internet has encouraged the installation of high bandwidth networks across the globe. Such networks are designed to foster new substantial collaborative functions among scientists around the world. As a result, file transfers of more than 10 petabytes through links of 100 Gbps or more are common in such networks. However most of the applications still employ TCP as the transport protocol for such transfers.

The throughput of a TCP connection in the presence of high bandwidth links has been well documented in recent literature [1]. Throughput of a TCP connection traversing high bandwidth connections is limited by its own congestion window (assuming that the receiver window is set to a high value). This problem can be decomposed into three distinct subproblems: (a) the slow linear increase of the congestion window during the congestion avoidance phase, (b) the halving of the congestion window when a packet is dropped or marked and (c) the effect of fiber error rates on the congestion window of a TCP connection. The first subproblem deals with the speed at which the congestion window size ramps to the full capacity of the link and has been discussed in [5]. The authors in [1] propose a modification that adjusts the multiplicative decrease parameter to combat the throughput inefficiency due to the halving of the window when a packet is marked or dropped. However the proposed scheme in [1] does not consider the effect of fiber error rates on the throughput. In this paper we will discuss a modification for TCP that will jointly address the throughput degradation imposed by subproblems (b) and (c). Note that the approach in this paper can be adopted for any form of congestion-controller and is not restricted to the specific congestion-controller that is discussed in this paper.

Current fiber error rates are on the order of 10^{-9} packets [9]. By design, the TCP congestion control algorithm assumes that all packet losses are a result of congestion in the network and hence decreases its sending rate (by halving its congestion window) to alleviate the congestion. As a result, TCP interprets a packet loss due to fiber errors as a congestion notification and reduces its rate. Therefore corruption of a packet due to fiber errors constitutes a false feedback for TCP. With low capacity links (say, 10Mbps) the packet loss/marking rate due to congestion is high in comparison to the packet corruption rate. Hence the false feedbacks due to fiber errors are negligible and hence can be ignored. However for high capacity networks, the packet loss rate or marking rate is comparable or smaller than the fiber error rates. For example, a throughput of 100 Gbps with a round-trip delay of 100 ms and packet size of 1000 bytes would require a marking/dropping rate of 10^{-12}. As a result, the effect of false congestion notification due to fiber error rates is significant and hence it decreases the utilization of the network dramatically.

Consider a single link of capacity 100 Gbps, a round-trip delay of with 100ms, packet size of 1500 bytes and a packet corruption rate of 10^{-9} and a single TCP user utilizing this link. The throughput of the TCP connection can be approximated using [10]:

$$x^* \approx \frac{1}{d\sqrt{\beta p}}, \tag{1}$$

where x^* is the steady state throughput of the TCP connection, β is the multiplicative decrease factor (approximately equal to $2/3$), p is the loss rate on the connection path and d is the round-trip delay. Using this relationship, the average sending rate of the TCP connection can be calculated to be roughly 4.6 Gbps. That is, the utilization on the link is 4.6 percent. *Note that this rate is derived assuming no congestion on the link and is also independent of the capacity of the link.* It solely depends on the fiber error rates. If the link was corruption free, the average sending rate of the TCP connection would be 75Gbps. As a result we can see that fiber error rates impose a fundamental limit on the achievable throughput using a TCP connection.

In [1], the authors propose a modification on TCP to solve this problem. When the window size of a user exceeds some threshold, the additive-increase and multiplicative-decrease

parameters become functions of the size of the congestion window. With this modification, the steady-state throughput of the TCP connection changes to

$$x^* = \frac{0.15}{dp^{0.82}} \tag{2}$$

in the presence of high bandwidth links. Although this relationship increases the achievable window sizes, it does not remove the adverse effect of packet corruption rates on TCP throughput. In [3], a new protocol called XCP is proposed which uses explicit rate information from the routers to set its rate. The XCP protocol requires changes to the router and the source design and the effect of fiber errors on the protocol is unknown. A new protocol based on TCP Vegas has been proposed in [8] to improve transfer speeds in high bandwidth connections. However, the effect of fiber error rates have not been incorporated in the protocol.

In this paper we propose a complementary approach to modify the multiplicative-decrease parameter of TCP algorithm in such a way that the loss rate experienced by a TCP user is lower bounded by a constant $\alpha < 1$. In essence, we wish to have the following relation between the throughput and loss rate p:

$$x^* \propto \frac{1}{d\sqrt{(p-\alpha)}}, \tag{3}$$

where $\alpha < 1$ is the design parameter. Note that as $p \to \alpha$, the throughput $x^* \to \infty$. With this relationship a TCP connection will be able to achieve a high sending rate without requiring very low loss rates. Our approach also gives us the control of the loss rate such that the minimum loss rate that can be experienced by the user is lower bounded by the parameter α. When α is chosen to be greater than the corruption rate, the amount of false feedback remains negligible and the packet corruption rate does not affect the throughput of the TCP algorithm.

While it is essential to modify TCP to achieve higher throughputs in high capacity links, it is equally important for the new protocol to maintain TCP friendliness at low capacities. In the proposed congestion controller, we can control the capacity below which the protocol is TCP friendly using the parameter α.

The rest of the paper is organized as follows: in Section II we use the utility maximization approach proposed in [4] to set up the fluid-model representation of a TCP connection. In Section III we define the new controller and discuss its properties. We discuss the choice of parameters in Section III-A to maintain TCP friendliness and show some preliminary simulations in Section IV. We finally conclude with discussions and future work in Section V.

II. System Model

We adopt a fluid model description of the congestion-controllers in this paper. Consider a network with a set $\mathcal{L}$ of links and a set $\mathcal{R}$ of users. Let C_l be the capacity of link l. Associate a route r with each user where r is a non-empty subset of $\mathcal{L}$. The terms user, flow and route will be used interchangeably throughout the paper. Assume User r generates traffic at a rate x_r. The rate x_r is assumed to have an utility $U_r(x_r)$ to flow r. We will assume that the utility functions are strictly concave functions and that $U_r(x) \to \infty$ as $x \to 0$ for all $r \in \mathcal{R}$. The TCP congestion controller can be approximated by the utility function $\frac{-1}{x_r}$ as shown in [6]. We ignore the slow-start and the timeout behavior of TCP in this analysis since we are interested in the steady-state behavior of the congestion controller. Henceforth we will assume that all users employ the utility function $\frac{-1}{x}$.

For ease of exposition we will ignore feedback delays in this section. The fluid model for the TCP congestion control algorithm for each user $r \in \mathcal{R}$ can be written as:

$$\dot{x}_r = \kappa_r \Big(1 - \beta x_r^2 \sum_{l \in r} p_l(\sum_{j: l \in j} x_j)\Big), \tag{4}$$

where the parameter κ_r determines the adaptation speed of the congestion controller, $p_l(.)$ is the fraction of packets marked at link l and β denotes the multiplicative decrease parameter. For a TCP congestion-controller, β is approximately equal to 0.5. It is shown in [4] that the above congestion control scheme converges to the unique solution of the following utility maximization problem:

$$\max_{\{x_r\} \geq 0} \sum_{r \in \mathcal{R}} \frac{-1}{\beta x_r} - \sum_{l \in \mathcal{L}} \int_0^{\sum_{j: l \in j} x_j} p_l(z). \tag{5}$$

The term $\frac{-1}{\beta x_r}$ can be thought of as the utility of user r which can be interpreted as the potential delay for transmitting one unit of data at the rate x_r. The optimization problem in (5) then maximizes the total utility of the system with a penalty on exceeding the link capacities. The above optimization problem is the penalty function formulation (with suitable marking functions $p_l(.)$) of the following optimization problem:

$$\begin{aligned} \max_{\{x_r\} \geq 0} \quad & \sum_{r \in \mathcal{R}} \frac{-1}{\beta x_r} \\ \text{subject to} \quad & \sum_{j: l \in j} x_j \leq C_l \quad \forall l \in \mathcal{L}. \end{aligned} \tag{6}$$

It is shown in [6], [7] that the congestion controller in (4) solves the system problem in (6) exactly when the Adaptive Virtual Queue algorithm is employed at the routers to provide marks. Denote the equilibrium rate of user r by $\hat{x}_r$, and the equilibrium value of the marking probability seen by the user by $\tilde{p}_r (= \sum_{l \in r} p_l(\sum_{j: l \in j} \hat{x}_j)$. From (4), the value of $\tilde{p}_r$ can be evaluated as

$$\tilde{p}_r = \frac{1}{\beta \hat{x}_r^2}. \tag{7}$$

Note that as the equilibrium rates of the users increase, the marking probabilities go to zero. With the algorithm defined in (4), it is inevitable to experience very low marking rates in high speed networks. As a result, at very high capacities, the fiber error rates dominates the marking probability leading to very low utilizations. We describe a modified congestion controller in the next section.

III. Modified TCP for high speed connections

Assume each user $r \in \mathcal{R}$ employs the following modified congestion controller:

$$\dot{x}_r = \kappa_r \left(1 - \beta_r(x_r) x_r^2 \sum_{l \in r} p_l \Big(\sum_{j: l \in j} x_j \Big) \right), \tag{8}$$

where

$$\beta_r(x_r) = \frac{1}{\alpha_r x_r^2 + c_r}, \tag{9}$$

and α_r and c_r are design parameters. The parameter c_r determines the decrease parameter of the user when the user rate is close to zero. Therefore, to maintain TCP friendliness, we can assume $c_r = \frac{1}{\beta}$. Without loss of generality, we assume $\alpha_r = \alpha$ and $c_r = c$ for all users. Then from the equilibrium condition for (8), the value of equilibrium sending rate $\{\hat{x}_r\}$ and the loss rate $\{\tilde{p}_r\}$ is given by

$$\hat{x}_r = \sqrt{\frac{c}{\tilde{p}_r - \alpha}}, \tag{10}$$

$$\text{(or) } \tilde{p}_r = \frac{\alpha \hat{x}_r^2 + c}{\hat{x}_r^2} \qquad \forall r \in \mathcal{R}. \tag{11}$$

Note that the design parameter $\alpha < 1$ is the marking probability experienced when the equilibrium rate of the user is ∞, in other words, the equilibrium marking probability of the user cannot be lower than α.

Theorem 3.1: The system of congestion controllers in (8) converges to a unique equilibrium point. Moreover the equilibrium point solves the system problem

$$\max_{\{x_r\} \geq 0} \sum_{r \in \mathcal{R}} \left(\frac{-1}{\beta x_r} + \alpha x_r \right) - \sum_{l \in \mathcal{L}} \int_0^{\sum_{j: l \in j} x_j} p_l(z). \tag{12}$$

The above theorem shows that by adding a linear term to the utility function of TCP, one can lower bound the loss rate of a connection. In the next section, we will discuss the choice of α in the congestion controller.

A. Choice of α

While it is essential to modify TCP to achieve high throughput in high capacity links, it is necessary for the new protocol to maintain TCP friendliness at low bandwidths. The goal in this paper is to make sure that the new controller is TCP friendly at low (or normal) bandwidths while retaining the high throughput properties at high bandwidths. In the proposed congestion controller, we can control the capacity below which the protocol is TCP friendly using the parameter α. Because of the distributed nature of TCP, we assume α to be a global constant in this paper. We first examine the effect of α on the congestion control algorithm in more detail.

The parameter α serves as a lower bound to the marking probability. As previously discussed, α must be also be greater than the fiber error rate in order to overcome the throughput limitation imposed on TCP by fiber errors. Hence, one of the design requirements is to choose the value of α at least a magnitude or two greater than the fiber error rates.

In addition to acting as the lower bound to the marking probability, α also controls the TCP friendliness of the congestion controller. In the proposed congestion control algorithm, the multiplicative decrease parameter is a decreasing function of the TCP rate unlike the constant multiplicative decrease factor of TCP. As a result, when a congestion event occurs, the reduction in rate is smaller in the proposed controller. However, the difference in reduction between the proposed controller and the original TCP protocol is governed by the choice of α.

We wish to make the new congestion controller TCP friendly when the connection passes through links with capacity less than $\hat{C}$ Mbps. Let $\hat{p}$ be the marking rate that a TCP connection experiences when the TCP connection passes through this link. From (8), the steady state throughput of the new congestion controller is given by:

$$x^* \propto \frac{1}{\sqrt{p - \alpha}}, \tag{13}$$

where p is the marking rate experienced by the congestion controller. If α is chosen a magnitude smaller than $\hat{p}$, then $(\hat{p} - \alpha) \approx \hat{p}$. Therefore, the steady state throughput of the new congestion controller is very close to the steady state throughput of TCP.

For example, assume that we wish to make sure that the new congestion controller is TCP friendly for capacities till 10 Mbps. A flow rate of 10 Mbps corresponds to a loss rate of 2×10^{-4} for a packet length of 1500 bytes and round-trip delay of 100 msec. As a result, a choice of $\alpha = 2 \times 10^{-6}$ guarantees that the new congestion controller is TCP friendly for capacities less than 10 Mbps.

IV. Simulations

In this section, we present simulation results for the fluid model and the packet level implementation of the congestion controller algorithm proposed in Section III. We use the software package MATLAB for the simulations and the software package ns-2 for the packet model simulations. In all simulations we consider a single link topology. We first investigate the performance of the proposed algorithm in terms of achieved sending rate by a TCP user in high capacity links. Then we present the TCP friendliness of the algorithm for low capacity links.

For the fluid model simulations, we first consider a single flow traversing a link of capacity 1000 units with fiber loss rate of 10^{-3}. Note that the capacity values and the fiber loss rates are scaled appropriately to facilitate simulations. The parameter α is chosen to be 10^{-2}. We also choose κ_r as 1. In the first experiment we compare the throughput of a TCP connection with the throughput achieved by the modified congestion controller. The evolution of flow rates are shown in Fig. 1. From Fig. 1, we can see that due to the fiber loss rate, the rate of the TCP user cannot exceed 50 units and the link is severely under-utilized. Since the fiber loss rates have negligible effect on the proposed modified TCP algorithm,

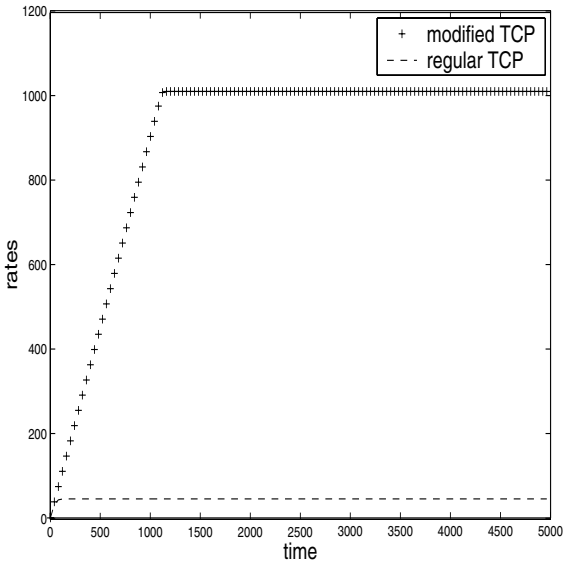

Fig. 1. Throughput comparison between a TCP congestion controller and the modified TCP algorithm.

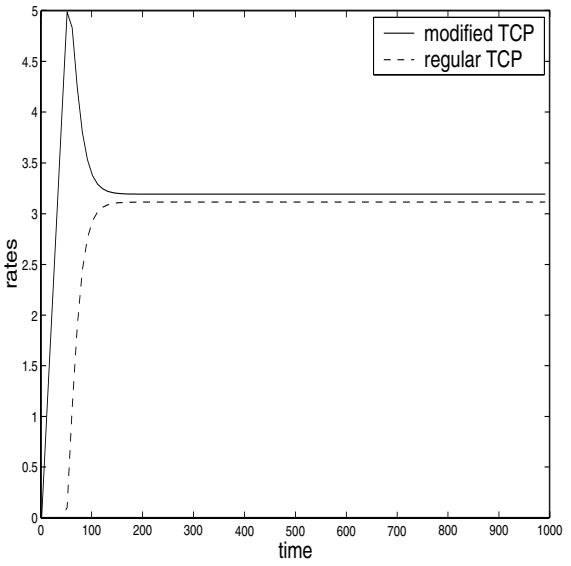

Fig. 2. TCP Friendliness of the modified TCP congestion controller.

we see that the user is able to grab the available capacity efficiently.

In the second experiment in fluid model, we demonstrate the TCP friendliness of the modified algorithm on a link of capacity 5 units. The other parameters remain the same. Note that the capacity is one order smaller than the maximum rate that a regular TCP can achieve with a fiber loss rate of 10^{-3} (see Fig. 1). The regular TCP flow is introduced to the link after 50 units of time the user with the modified TCP starts its transmission. The evolution of flow rates are shown in Fig. 2. We can see from Fig. 2 that the rates of the two users converge to almost the same value which validates the discussion in Section III-A. Note that the modified congestion controller has a slightly higher rate due to the fact that the multiplicative decrease parameter is slightly smaller than 0.5.

For the packet level implementation of our algorithm, we assume that the source knows its round-trip delay, so that the window size information can be used to infer the rate of the connection which is used to adapt the decrease parameter. The round-trip delay estimation is beyond the scope of this paper. We consider a single user on a single link of capacity 100 Mbps. Fiber error rate on the link is assumed to be 10^{-5} packets/sec and α is chosen to be 10^{-4}. The evolution of the window size of the user for different congestion controllers are shown in Fig. 3. We see that the fiber losses prevent the regular TCP user to increase its window size further. As a result, the regular TCP user fails to utilize the link capacity as seen from Fig. 4, where the utilization of the link is measured on intervals of 10 seconds.

From Fig. 5, we see that as the capacity of the link is decreased to 10 Mbps, the regular TCP congestion controller performs as good as the modified TCP congestion controller. Note that the fiber loss rate is small enough for regular TCP

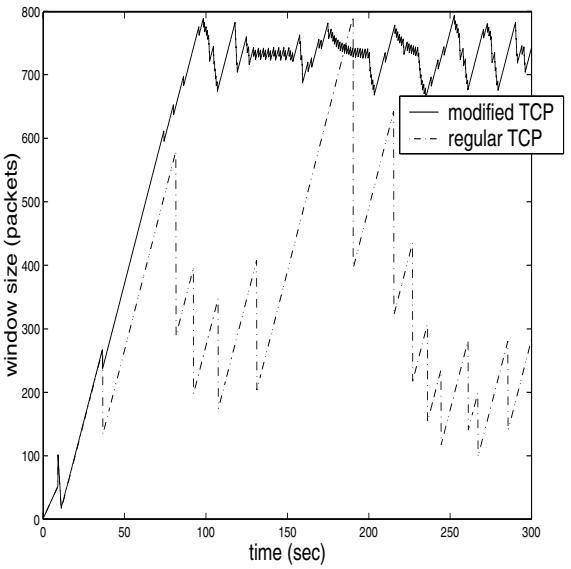

Fig. 3. Evolution of the window size of the user.

to increase beyond window sizes of 100 as can be seen from Fig. 3. However, the window size is now limited by the link capacity instead of fiber losses.

V. Conclusions

Due to incapability of TCP to distinguish between congestion loss and packet corruption, the rate of a regular TCP connection is limited by the fiber error rate in high bandwidth links. In this paper, we proposed a modification on TCP that removes this limitation by presenting a designable lower bound on the loss rate. With a proper choice of this bound, the loss due to fiber errors is forced to remain negligible with respect to congestion losses. The new algorithm does not require any

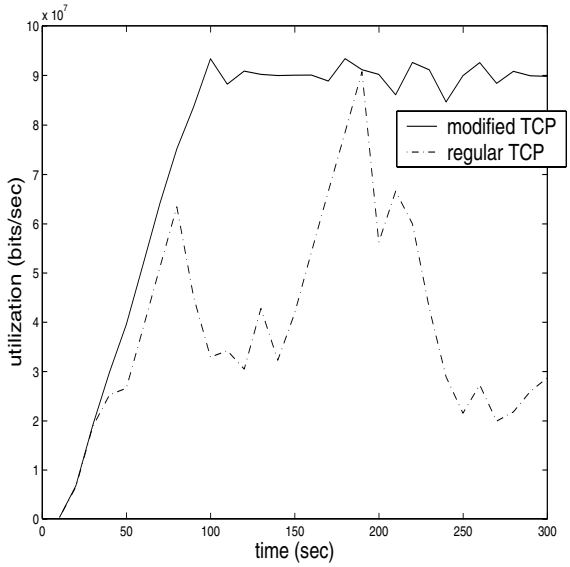

Fig. 4. Link Utilization averaged over 10 sec. intervals.

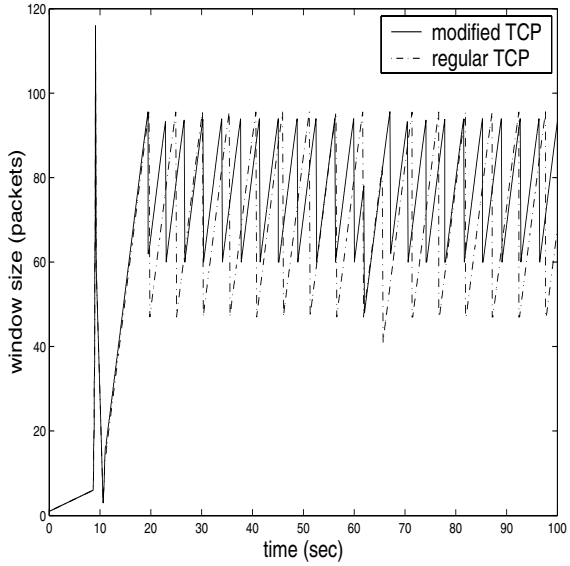

Fig. 5. Evolution of the window size of the user.

additional functionality at the routers and it is TCP friendly enough to provide reasonable rates to regular TCP flows. The simulation results validate our analysis.

REFERENCES

[1] S. Floyd, "Highspeed TCP for large congestion windows," Internet draft draft-floyd-tcp-highspeed-00.txt, Preprint, June 2002.
[2] R. Gibbens and F. Kelly, "Resource pricing and the evolution of congestion control," 1998, preprint.
[3] D. Katabi, M. Handley, and C. Rohrs, "Internet congestion control for future high bandwidth-delay product environments," in *Proceedings of ACM Sigcomm*, 2002.
[4] F. P. Kelly, A. Maulloo, and D. Tan, "Rate control in communication networks: shadow prices, proportional fairness and stability," *Journal of the Operational Research Society*, vol. 49, pp. 237–252, 1998.
[5] S. Kunniyur, "AntiECN Marking: A Marking Scheme for High Bandwidth Delay Connections," in *Proceedings of ICC 2002*, Anchorage, Alaska, May 2002.
[6] S. Kunniyur and R. Srikant, "End-to-end congestion control: utility functions, random losses and ECN marks," in *Proceedings of INFOCOM 2000*, Tel Aviv, Israel, March 2000, Also to appear in IEEE/ACM Transactions on Networking, 2003.
[7] S. Kunniyur and R. Srikant, "A time-scale decomposition approach to adaptive ECN marking," *IEEE Transactions on Automatic Control*, vol. 47, pp. 882–894, June 2002.
[8] C. Jin and D. Wei and S. H. Low, "Fast TCP: Motivation, Architecture, ALgorithms and Performance," To appear in *Proceedings of IEEE Infocom*, March 2004.
[9] Vern E. Paxson, "Measurements and Analysis of End-to-End Internet Dynamics," PHD thesis, University of California, Berkeley, 1997.
[10] J. Padhye and V. Firoiu and D. Towsley and J. Krusoe, "Modeling TCP Throughput: A Simple Model and its Empirical Validation ," in *Proceedings of ACM SIGCOMM 1998*, Vancouver, CA, September 1998.