Stochastic Ordering for Internet Congestion Control and its Applications

Han Cai Do Young Eun*
Dept. of ECE,
NC State University
Email: {hcai2, dyeun}@ncsu.edu

Sangtae Ha Injong Rhee
Dept. of CS,
NC State University
Email: {sha2, rhee}@ncsu.edu

Lisong Xu
Dept. of CSE,
University of Nebraska
Email: xu@cse.unl.edu

***Abstract*— Window growth function for congestion control is a strong determinant of protocol behaviors, especially its second and higher-order behaviors associated with the distribution of transmission rates, its variances, and protocol stability. This paper presents a new stochastic tool, called *convex ordering*, that provides an ordering of any convex function of transmission rates of two protocols and valuable insights into high order behaviors of protocols. As the ordering determined by this tool is consistent with any convex function of rates, it can be applied to any unknown metric for protocol performance that consists of some high-order moments of transmission rates, as well as those already known such as rate variance. Using the tool, it is analyzed that a protocol with a growth function that starts off with a concave function and then switches to a convex function (e.g., an odd order function such as x^3 and x^5) around the maximum window size in the previous loss epoch, gives the smallest rate variation under a variety of network conditions. Among existing protocols, BIC and CUBIC have this window growth function. Experimental and simulation results confirm the analytical findings.**

I. Introduction

Congestion control is an integral component of a transport protocol in a packet-switched network; much of the Internet's success is attributable to TCP, a commonly used transport protocol of the Internet. As the Internet evolves in its capacity and characteristics, demands for new congestion control adapting to the new operating conditions and goals are constantly increasing. As a result, many new protocols whose behaviors significantly deviate from TCP have lately been proposed. An emerging class of congestion control, called *high-speed TCP variants* (e.g., [1], [2], [3], [4], [5], [6]) are designed specifically for high bandwidth-delay product networks.

One goal of these high speed TCP protocols is to increase the scalability of Reno-style TCP which uses an AIMD window adjustment algorithm, and many of these protocols differ mainly in their choices of window adjustment algorithms, in particular in the functions used in the growth phase of the congestion window – for simplicity, however, most of them use the same multiplicative decrease function with possibly different multiplicative factors to reduce the window during packet losses. The choices of growth functions are diverse from exponential to some polynomial functions. For instance, STCP [3] uses an exponential growth function, HSTCP [2] uses a polynomial function, HTCP [5] uses a square function, BIC [4] uses a combination of logarithmic and exponential functions, and CUBIC [7] uses a cubic function.

*This work was supported in part by NSF CAREER Award CNS-0545893

The goal of this paper is to compare these growth functions, especially in terms of the second or higher-order stochastic behaviors of the protocols that employ these functions. Stochastic behaviors of congestion control protocols beyond the first order are important because of the richness of information they provide. A higher-order stochastic analysis offers a rich set of information about protocols, including the distribution of transmission rates, its variance and protocol stability. These are important information about protocols. For instance, the variance of transmission rates is critical for a large class of Internet applications. Applications like scientific collaboration, telemedicine and real-time environment monitoring require access to high-volume real-time data, images and video captured from remote sensors such as satellite, radars and echocardiography. In addition, they require predictable, low-latency access to this data, in real time. High rate variability incurs delays or loss of quality which lessen the value of information received for these applications. So the requirements for a good transport protocol are ever more stringent as it has to be adaptive to fluctuations in available bandwidth and also exhibit smaller rate variations under steady-state.

Stability is also an important goal of congestion control as it can affect the general well-beings of the network including utilization, queue oscillations and packet loss characteristics. In practice, stability has been frequently associated with the variance of rate distributions. Protocol rate variations can influence fluctuations and oscillations in router queues and thus, queue overflows. The severity of these overflows may cause loss synchronization across many co-existing flows and therefore, under-utilization of link capacity, which are the general signs of network instability. Thus, measuring the rate variations of flows is commonly used in practice to quantify the practical sense of "protocol stability". For instance, [6], [7] use the CoV (coefficient of variance, defined by the standard deviation over its mean) of per-flow transmission rate to measure stability. Therefore, it is clear that in practice, a quantifiable degree of stability is closely related to some higher order behaviors of protocols.

Calculating the exact distribution of transmission rates* stochastically is non-trivial because of states involved in describing the behavior of protocols. However, there is a hope. The main contribution of this paper is to use an alternative tool,

*The transmission rates are obtained by dividing the congestion window size by RTT. Since we are assuming the same RTT for every protocol we compare, we use window sizes and rates interchangeably for convenience.

called *convex ordering*, that provides a powerful insight into the high-order behaviors of protocols. Although it cannot be used to compute the rate distribution itself, convex ordering is extremely useful in comparing any convex function of congestion window sizes of protocols. We find that convex ordering can be applied to many existing protocols that use multiplicative decrease (we call *MD-style* protocols) such as Scalable TCP, HSTCP, BIC, HTCP, etc. A salient feature of convex ordering is that the ordering can be applied to any (unknown) convex function. Thus, for any metric of protocol performance defined in terms of some high-order functions (or statistics) of transmission rates, our tool can be applied to define the ordering of metrics among protocols. At the minimum, we can use it to compare the rate variance or CoV of per-flow rates of protocols (note that the function is convex).

Our study of convex ordering on various existing growth functions has revealed the followings:

- Under stationary conditions, protocols with a more concave growth function has a lower convex ordering than those with a more convex function.
- Under non-stationary conditions, a protocol with a growth function that starts off with a concave function and then switches to a convex function at the origin (which we call a *concave-convex* function) has a lower convex ordering than those with just concave or convex functions. Concave-convex functions have an inflection point where growth becomes zero at the origin. For instance, an odd order function such as x^3, x^5, etc. has this profile.

Our results indicate that, under a variety of network conditions, a protocol with a concave-convex window growth function that uses the maximum window size in the last congestion epoch to be the inflection point, has mostly a concave window growth profile during steady state where available bandwidth remains stationary and a concave-convex window growth profile during non-stationary conditions where available bandwidth undergoes abrupt change. Thus according to our analysis, such a protocol has the lowest convex ordering. Among the existing protocols, BIC and CUBIC have this property. Our NS-2 simulation and Linux-based experimental results confirm these findings.

II. Preliminaries

A. Stochastic vs. Fluid method

Most congestion control algorithms can be written in the form of the following stochastic recursion:

$$X_{t+1} = F(X_t, U_t), \tag{1}$$

where X_t is in general a random vector in a suitably chosen state-space and $\{U_t\}$ is a stationary "driving" sequence, independent of X_t. Mapping F defines how the system evolves in an appropriate time scale. On the other hand, the so-called fluid model system dynamics take the following form:

$$x_{t+1} = \mathbb{E}\{X_{t+1} \mid X_t = x_t\} = \mathbb{E}\{F(x_t, U_t)\} = G(x_t), \tag{2}$$

i.e., (2) captures the average behavior of (1).[†]

As the first-order behavior is under discussion, the fluid method is much simpler and convenient than the stochastic counterpart. For instance, the stability refers to the convergence of the recursion $x_{t+1} = G(x_t)$ to its fixed point $\hat{x}$ where $\hat{x} = G(\hat{x})$. Other performance metrics including fairness and responsiveness can also be analyzed through the relationship among the average throughput (the fixed point $\hat{x}$) and other system parameters. In contrast, under the stochastic setting in (1), suppose it is 'stable' and thus X_t is stationary. Then, in principle, one should be able to solve the corresponding distributional equation $X_t \stackrel{d}{=} F(X_t, U_t)$ to obtain the distribution of X_t. Unfortunately, however, this kind of equation is extremely hard to solve and its solutions are known only for a very small set of functions F [8]. Thus this approach seems to be severely limited in computing and comparing any meaningful performance metric given by some function of X_t. Although there exist other stochastic approaches providing a feasible way to give accurate steady-state value, the fluid method is typically favored because of its simplicity.

As the *second or higher-order* behaviors of protocols, such as rate variations, are under discussion, the stochastic method becomes more attractive. The main reason lies in that (2) loses too many details originally contained in (1). For example, suppose U_t, $t = 1, 2, \ldots$ is $i.i.d.$ and X_t takes only a discrete value in $\Omega = \{1, 2, \ldots, M\}$. Then, (1) becomes a homogeneous discrete-time Markov chain with its transition probability $p_{ij} = \mathbb{P}\{X_{t+1} = j \mid X_t = i\}$ $(i, j \in \Omega)$ and the equation (2) becomes $G(i) = \sum_{k=1}^{M} k p_{ki}$. If this equation and $\sum_{j=1}^{M} p_{ij} = 1$ hold true for all $i \in \Omega$, (2) remains invariant. That is, we only have $2M$ equations for $M \times M$ unknown variables p_{ij} $(i, j \in \Omega)$. Obviously, for large M, many different Markov chains will be mapped into the same deterministic recursion, and we lose all the detailed stochastic information of X_t when we simply focus on the fluid model represented as (2). In other words, two different "protocols" may have totally different stochastic behaviors and variability even though they use the same fluid model and thus have the same first-order behavior. This observation leads us to believe that we need a stochastic tool to compare two different protocols in terms of its variability and high order behaviors.

B. Related Work

In the literature there have been numerous results on the stability and the first-order behaviors of congestion control protocols based on fluid models [9], [10]. While all these fluid-based studies provide clear-cut conditions on system parameters for stability, they do not tell us how to compare two "stable" protocols in terms of more practically meaningful high order behaviors such as the degree of rate fluctuations. On the other hand, most results via stochastic models have focused

[†]For example, a fluid model for AIMD with rate-based AQM can be written as $x_{t+1} = (x_t + 1)(1 - p(x_t)) + x_t p(x_t)/2$, while the stochastic one X_{t+1} always takes either $X_t + 1$ with probability $1 - p(X_t)$ or $X_t/2$ with probability $p(X_t)$. Here, $p(x)$ is the probability of receiving congestion when the current rate or the window size is x.

on the average values of stochastic quantities [11], [12], [13] or have been obtained under some limiting conditions to make the analysis more tractable [14], [13]. Still, these studies do not provide a means to compare the high order stochastic behaviors of different protocols. The only comparison result we can find in the literature based on some stochastic model is in [15] showing that steady-state window sizes with a larger upper bound is stochastically larger than with a smaller bound, which is then used for proving the stochastic stability of their model and obtaining its stationary distribution solution. Yet, it does not show how to provide any ordering of high-order protocol performance.

III. Convex ordering for congestion control

In this section we show there exists a convex ordering between two congestion control protocols. We first consider stationary inter-loss processes, and then discuss non-stationary loss processes later in Section III-D.

A. Model Description

A congestion event (or loss event) is defined to be a chunk of time during which Reno-style TCPs make one window reduction using fast recovery. Thus, one loss event includes multiple packet losses and consists of at least one RTT. Let $T_1, T_2, \ldots$ be a stationary sequence of intervals between two consecutive congestion events, and $\tau_n = \sum_{i=1}^{n} T_i$ $(n = 1, 2, \ldots)$ the time instant at which the n^{th} congestion occurs (the n^{th} congestion epoch). We denote by $W(t)$ the window size at time t and define $X_n = W(\tau_n)$, the window size at the n^{th} congestion epoch. When congestion occurs at τ_n, the window size first decreases by some amount, and then keeps increasing according to some profile f until the next congestion epoch τ_{n+1}. Thus, we can write $X_{n+1} = f(T_n, X_n)$, where the function $f = f(t, x)$ is increasing in t and x and represents the profile for X_n.

For a given $\{T_n\}$, we consider the following recursive equations for X_n and Y_n with profiles f and g, respectively.

$$X_{n+1} = f(T_n, X_n), \quad \text{and} \quad Y_{n+1} = g(T_n, Y_n) \tag{3}$$

Our goal is to compare the stochastic properties of X_n and Y_n in (3). As T_n is stationary in n (its distribution does not depend on n), we use a random variable T to denote a generic inter-loss interval. Similarly, we will use X and Y when X_n and Y_n are stationary (which is indeed the case as shown later). Then, *for a given inter-loss interval random variable* T, we consider f and g satisfying the followings:

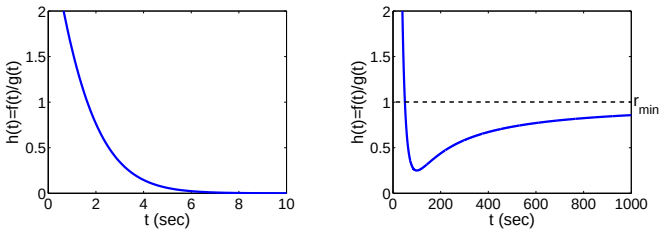
(C1): The functions $f(t, w)$ and $g(t, w)$ are of the following form (with a little abuse of notation):

$$f(t, w) = f(t) + (1-\beta)w, \quad g(t, w) = g(t) + (1-\beta)w \tag{4}$$

where $f(t)$ and $g(t)$ are non-decreasing, $f(0) = g(0) = 0$, and $0 < \beta < 1$.

(C2): There exists the only one root $t_0 > 0$ for the following:

$$h(t) := \frac{f(t)}{g(t)} = \frac{\mathbb{E}\{X\}}{\mathbb{E}\{Y\}}. \tag{5}$$



(a) monotone $h(t)$ (b) non-monotone $h(t)$

Fig. 1. (a): Condition (C2) is always satisfied regardless of the value of $\mathbb{E}\{X\}/\mathbb{E}\{Y\}$. (b): (C2) is satisfied if we know that $\mathbb{E}\{X\}/\mathbb{E}\{Y\} \in (1, \infty)$.

Without loss of generality, we assume $h(t) > h(t_0)$ for $t < t_0$, and $h(t) < h(t_0)$ for $t > t_0$.

(C1) says that the window size is first reduced by βw at each congestion epoch (MD-style), and then increases according to $f(t)$ (or $g(t)$) as the inter-loss interval t increases until the next congestion epoch. (C2) puts some condition on the shape of two increasing profiles f and g of protocols under comparison in relation to the ratio between their average window sizes or throughput. We note that (5) always has *at least* one root. This follows from (i) $\mathbb{E}\{X\}/\mathbb{E}\{Y\} = \mathbb{E}\{f(T)\}/\mathbb{E}\{g(T)\}$ as we will show later, and (ii) $b(t) \triangleq f(t)\mathbb{E}\{g(T)\} - g(t)\mathbb{E}\{f(T)\} = 0$ must have at least one root since $E\{b(T)\} = 0$. Since $f(t)/g(t)$ is larger than a certain threshold for $0 < t < t_0$ and smaller than that threshold for $t > t_0$ and since f and g are both non-decreasing, intuitively, (C2) implies that $f(t)$ tends to increase faster than $g(t)$ initially but slower afterwards. In other words, we say that $f(t)$ is *more concave* than $g(t)$.

In practice, the value of $\mathbb{E}\{X\}/\mathbb{E}\{Y\}$ may be difficult to compute *a priori* as it's a function of f and g. Suppose we choose f and g such that $h(t)$ is monotone (or, without loss of generality, decreasing), then *regardless of* $\mathbb{E}\{X\}/\mathbb{E}\{Y\}$, we see that (C2) is always satisfied since we already know that (5) has at least one root. In addition, if we have some information about $\mathbb{E}\{X\}/\mathbb{E}\{Y\}$ such as its range (e.g., from knowing the distribution of T), then even for non-monotone $h(t)$, (C2) may be still satisfied. For example, in Figure 1(b), (C2) is satisfied if $\mathbb{E}\{X\}/\mathbb{E}\{Y\}$ lies in $(1, \infty)$.

There exists a large set of profiles f, g for which the function $h(t) = f(t)/g(t)$ is monotone, e.g., the first two examples in the following. In the last example, $h(t)$ is not monotone, but (C2) may still be satisfied if some knowledge of $\mathbb{E}\{X\}/\mathbb{E}\{Y\}$ is available. (Here, f' means the derivative of $f(t)$ (similarly for others) and $a_i's$ $(i = 1, 2, 3)$ are all positive constants.)

(i) $f(t)$ and $g(t)$ are strictly concave and convex, respectively. In this case, $h' = (f'g - fg')/g^2 < 0$ because $f(0)g'(0) - f'(0)g(0) = 0$ from (C1) and $(f'g - fg')' = f''g - fg'' < 0$ from $f'' < 0,\ g'' > 0$.

(ii) $f(t) = a_1 t^p,\ g(t) = a_2 t^q$ where $p \neq q$. Obviously, $h(t) = (a_1/a_2)t^{p-q}$ is monotone.

(iii) $f(t) = a_1\Big((t - a_2)^3 + a_2^3\Big),\ g(t) = a_3 t^3$, where a_i's are chosen such that $\mathbb{E}\{X\}/\mathbb{E}\{Y\} > a_1/a_3$. This can be seen from $h' \leq 0$ when $t \leq 2a_2$, $h' > 0$ when $t > 2a_2$, and $h(0^+) > a_1/a_3,\ h(a_2) = a_1/a_3,\ \lim_{t\to\infty} h(t) = a_1/a_3$. ($h(t)$ is similar to the one in Figure 1(b).)

In general, window growth functions can be divided into three classes according to their shapes: (a) concave ([6], [16]); (b) convex ([2], [3], [5]); (c) concave-convex ([4], [7]). We can then use condition (C2) to investigate how these shapes of window growth functions affect the second and higher order behaviors of a protocol and its rate fluctuation and to compare the stochastic properties of these classes.

To proceed, we impose the following assumption:

(A1): The inter-loss intervals T_n $(n = 1, 2, \ldots)$ are independent and identically distributed $(i.i.d.)$.

Assumption (A1) is well supported. An *i.i.d* process of congestion epochs (not packet losses) is commonly observed in Internet measurement studies (e.g., [17], [18]) and thus, commonly assumed in the stochastic analysis of TCP (e.g., [19]). For example, large-scale Internet measurement studies in [18] show that the loss process is very close to $i.i.d.$ (using autocorrelation-based Box-Ljung test), and in fact is well modeled by a Poisson process. Also, the $i.i.d.$ inter-loss interval (i.e., loss event) allows dependency among congestion events over different RTTs. To see this, suppose that congestion occurs with probability p in each RTT, independently from other RTTs, then the distribution of the inter-loss interval T_n *automatically* becomes $\mathbb{P}\{T_n = k\} = (1-p)^{k-1}p$ (memoryless). Since we allow any arbitrary distribution for T_n, a congestion event in the current RTT may depend on whether there was a congestion event in the previous RTT.

When T_n's are $i.i.d.$, the sequence of window sizes at congestion epochs defined in (3) now becomes a homogeneous Markov chain. Without loss of generality, we can assume that X_n and Y_n are both irreducible and aperiodic. For the positive recurrence, in view of Pake's Lemma [20], we have

$$\lim_{w\to\infty} \mathbb{E}\{X(n+1) \mid X(n) = w\} - w = \lim_{w\to\infty} \mathbb{E}\{f(T)\} - \beta w < 0.$$

and similarly for Y_n. Thus, they are both positive-recurrent, and hence ergodic. Since an ergodic chain always enters steady-state in which the distribution does not depend on time, we will use X and Y to denote X_n and Y_n, respectively, whenever there is no ambiguity.

B. Convex Ordering for Congestion Control

In this section we show that there exists a convex ordering between two congestion control protocols. Before presenting our main result, we need the following definition.

Definition 1: Let X and Y be random variables with finite means. Then we say that X is less than Y in a *convex order* (written $X \leq_{cx} Y$), if $\mathbb{E}\{\phi(X)\} \leq \mathbb{E}\{\phi(Y)\}$ for all convex functions ϕ for which the expectations exist. □

Similarly, we write $X \leq_{icx} Y$ if $\mathbb{E}\{\varphi(X)\} \leq \mathbb{E}\{\varphi(Y)\}$ for all increasing convex functions φ.

In what follows, we prove that the rescaled window size $X/\mathbb{E}\{X\}$ for profile f is always less than $Y/\mathbb{E}\{Y\}$ with profile g in convex ordering. Note that these rescaled variables have the same mean, and the choice of $\varphi(x) = x^2$ leads to $\mathrm{Var}\{X/\mathbb{E}\{X\}\} \leq \mathrm{Var}\{Y/\mathbb{E}\{Y\}\}$, i.e., $CoV(X) \leq CoV(Y)$ (by taking square root in both sides). This kind of ordering holds true for *any other* convex function ϕ, so in general we can say that the normalized steady-state window size for profile f is *less variable* than that for g. In addition, it implies that the system with f is "more predictable" than with g in the sense that the window size fluctuations (rate fluctuations) are more concentrated around its mean, thus requiring a smaller buffer to absorb temporal fluctuations. Our theorem below provides a theoretical support in that, for stationary loss-interval processes, it would be better from the second and higher order behavior point of view to increase the window size initially faster and then to slow down later on (i.e., more concave), rather than the other way around as typically used in many current TCP protocols (e.g., [5], [2], [3]). Note that a stationary loss interval process does not mean that all loss intervals are the same, and it means that their distributions do not change over time.

Throughout the paper, every proof is omitted due to space constraints and the readers are referred to our technical report [21] for the details on the proof. We now present our main theorem.

Theorem 1: Consider two different profiles f and g satisfying (C1) and (C2). Then, under Assumption (A1), we have $X/\mathbb{E}\{X\} \leq_{cx} Y/\mathbb{E}\{Y\}$. □

Theorem 1 shows that convex ordering can compare the high order behavior of congestion control protocols simply by comparing the shapes of their increasing profiles.

More important, our result also gives guidance for designing a more 'predictable' protocol with less fluctuations. Suppose there is an existing protocol with increasing profile $g(t)$. Then, by shaping it in a more concave way into $f(t)$ such that (C2) is satisfied, we can obtain *a set of protocols* whose normalized rate fluctuations are all smaller than that of g. Among these set of protocols, we can then choose a protocol satisfying the other required properties such as high throughput, etc. In summary, our main theorem indicates that: (i) it is possible to design such a protocol by simply reshaping the increasing profile of the original one; (ii) a concave profile will be an essential part of a more predictable protocol given that the loss process in the Internet is either stationary over a short time scale or a concatenation of stationary processes with different distributions over a long time scale.

C. Protocols with the Same Mean Behavior

In addition to (C1) and (C2), suppose that two protocols satisfy $\mathbb{E}\{f(T)\} = \mathbb{E}\{g(T)\}$, i.e., they have the same mean throughput. Then, it follows that

$$\begin{aligned}\mathbb{E}\{X_{n+1} \mid X_n = w\} &= \mathbb{E}\{f(T)\} + (1-\beta)w \\ &= \mathbb{E}\{g(T)\} + (1-\beta)w = \mathbb{E}\{Y_{n+1} \mid Y_n = w\}. \quad (6)\end{aligned}$$

for all w, i.e., it says, "For any given window size at the current congestion epoch, the expected window size at the next congestion epoch is the same for both profiles." In other words, two protocols with profiles f and g are *indistinguishable* from an average point of view: they have the same fluid recursion

as in (2) and thus have the same fixed point and Lyapunov stability property (i.e., convergence).

Note that there exists a large set of profiles that satisfy (6). For instance, consider $f(t) = c_1 t^{\alpha_1}$ and $g(t) = c_2 t^{\alpha_2}$. Then, for a given exogenous loss process (i.e., given T), (C1), (C2) and $\mathbb{E}\{f(T)\} = \mathbb{E}\{g(T)\}$ are satisfied if c_i and α_i are chosen in such a way that $c_1\mathbb{E}\{T^{\alpha_1}\} = c_2\mathbb{E}\{T^{\alpha_2}\}$. Theorem 1 asserts that we can still define a convex ordering between X and Y despite $\mathbb{E}\{X\} = \mathbb{E}\{Y\}$. This confirms the importance of the stochastic approach toward any second and higher order behaviors of protocols. It also shows that, although these two protocols share the same fluid model, their high order stochastic properties can be widely different, as mentioned in Section II.

D. Convex Ordering under Non-stationary Loss: A Closer Look at Single Loss Interval

As the loss interval process is more like stationary over a certain time period, we already know from Theorem 1 that concave-like profiles work very well. When it dramatically changes so that its distribution may change, however, Theorem 1 may provide little information about how to 'shape' the profile toward the next unpredictable target. Further, when the target process is non-stationary, the inter-loss intervals T_n become also non-stationary, and it is impossible to show any stochastic ordering, invariant with respect to time, between two protocols. For this reason, we consider only a single loss interval where the new target is *arbitrary*.[‡]

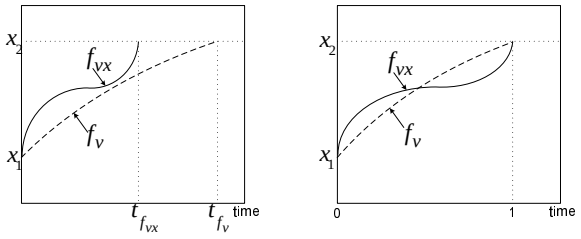
Specifically, let x_1 denote the window size immediately after the current congestion epoch, and x_2 the window size just before the next congestion epoch. Assume that x_1 and x_2 ($x_1 < x_2$) are arbitrary given (fixed). We do not consider the case of consecutive reductions in window size (i.e., $x_1 > x_2$). Clearly, the amount of time to hit the new target x_2 from x_1 depends on our choice of increasing profile f (and of course on x_1 and x_2). Set $x = (x_1, x_2)$ and let t_f be the resulting inter-loss interval for the profile $f = f^x$. The superscript in f^x represents the dependency of f upon the given $x = (x_1, x_2)$. As x is fixed (arbitrary) in this section, to make the notation simple, we will use f instead of f^x. Note that f is increasing, and we have $f(0) = x_1$ and $f(t_f) = x_2$.

We now consider the window size sampled at any arbitrary *random* time over $[0, t_f]$. If we define by U_t the uniform random variable distributed over $[0, t]$, then the window size at any arbitrary random time is given by $W_f = f(U_{t_f})$. Note that different choices of f give different distributions for $f(U_{t_f})$. We consider two increasing profiles f and g whose average throughput over their inter-loss intervals remain the same, i.e.,

$$\mathbb{E}\{W_f\} = \mathbb{E}\{f(U_{t_f})\} = \mathbb{E}\{g(U_{t_g})\} = \mathbb{E}\{W_g\}, \qquad (7)$$

where $\mathbb{E}\{f(U_{t_f})\} = \int_0^{t_f} f(s)ds/t_f$ (similarly for $\mathbb{E}\{g(U_{t_g})\}$). The requirement of (7) is necessary to avoid trivialities. For instance, for given x_1 and x_2, if we choose a profile f with

[‡]This should be distinguished from the stationary case, where the 'actual' value of the next target is also unknown but its average remains the same.



(a) Original profiles (b) Re-scaled profiles

Fig. 2. Comparison of concave-convex profile f_{vx} vs. concave profile f_v. $t_{f_{vx}}$ and t_{f_v} represent the time to reach the arbitrary chosen target x_2 for different profiles. After rescaling, all the profiles start and end at the same points. Similar plots can be drawn for concave-convex vs. convex profile.

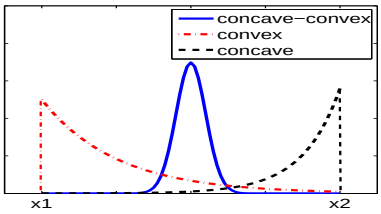


Fig. 3. The probability density functions of concave-convex, convex, concave profiles. Under 'fair' comparison with the same throughput, a concave-convex profile is least variable, as its probability mass is more concentrated around the mean.

$f(0) = x_1$ and $f(t) = x_2$ for all $t > 0$ (i.e., it instantaneously jumps to x_2 and stays there), it would be "optimal" giving the maximum throughput with the smallest variation. But, such a choice is meaningless because of its dependency on the value of x_2. Instead, by enforcing constant $\mathbb{E}\{W_f\}$ for different choices of f, we can find a better shape of profiles toward a fixed, yet randomly chosen x_2 satisfying (7) .

We next show that for any given $f(t)$, the distribution of $f(U_{t_f})$ remains the same if we rescale $f(t)$ to $f(at)$ for any arbitrary positive constant a.

Lemma 1: For any given increasing function f, we define a collection of profiles $\Omega_f = \{f(at),\ a > 0\}$. Then, the distribution of W_f for $f \in \Omega_f$ does not depend on a. □

Without loss of generality, we can assume $t_f = 1$ for any given profile f by suitably rescaling $f(t)$ if necessary. In this case, $\mathbb{P}\{W_f \le y\} = f^{-1}(y)$, i.e., the cumulative distribution function of W_f is simply the inverse of the 'rescaled' increasing profile. We then obtain the following:

Proposition 1: For any given $x_1 < x_2$ and two increasing profiles f and g such that $\mathbb{E}\{W_f\} = \mathbb{E}\{W_g\}$, let $\tilde{f} = f(a_1 t)$ and $\tilde{g} = g(a_2 t)$ where a_1 and a_2 are chosen in such a way that $\tilde{f}(1) = \tilde{g}(1) = x_2$. If there exists t_0 such that $\tilde{f}(t) \ge \tilde{g}(t)$ for $t < t_0$ and $\tilde{f}(t) \le \tilde{g}(t)$ for $t > t_0$, then $W_f \le_{cx} W_g$. □

Proposition 1 gives us a tool to compare any two different profiles f and g satisfying (7) . To get more intuition, consider the following three different sets of profiles: concave-convex, convex, and concave profiles denoted by f_{vx}, f_x and f_v, respectively. After suitably rescaling each profile, we can assume that the inter-loss interval for (x_1, x_2) is always set to $[0, 1]$. See Figure 2 for illustration.

From $\mathbb{P}\{W_f \le y\} = \tilde{f}^{-1}(y)$, we can easily obtain the probability density function (pdf) of window sizes by differ-

entiating the inverse of the rescaled profiles in Figure 2(b). As shown in Figure 3, the concave-convex type profile makes the pdf more concentrated around the mean than the others. This is expected as the concave-convex profile spends more time in the middle between x_1 and x_2 while the pure concave or convex makes the pdf lopsided.

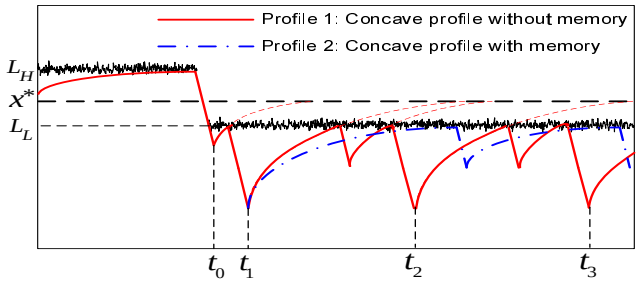
IV. How to Shape an Increasing Profile under Changing Environments

Theorem 1 implies that, when a stationary loss process $\{T_n\}$ is given, it would be better to increase the window size fast at the beginning and then slow down later on. As the chain X_n is stationary with $x^* = \mathbb{E}\{X_n\}$, it is likely that the new "target" should be somewhere around the maximum window size at the previous congestion epoch. Thus, intuitively, to increase fast at first and then slow down around the probable target x^* seems to be a good strategy. (It is in fact closer to the "optimal" in the sense that it is *less variable* than any others.) Internet measurement studies (e.g., [18]) show some evidence that this stationary period is more than several minutes or longer.

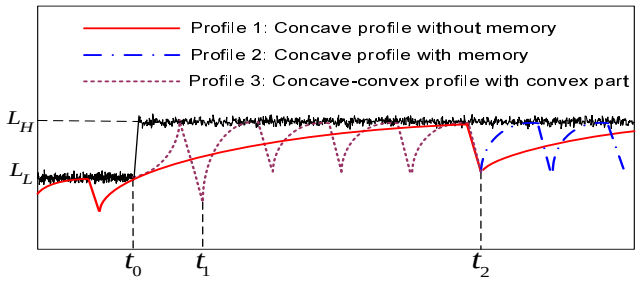
In reality, however, the current stationary process may change to another, i.e., X_n may become non-stationary over the timescale of the flow's lifetime. When a large group of flows arrives (departs), the available bandwidth drops (jumps) from one stationary process to another, stays at the new process for a relatively short period over which the number of flows remains about the same, and is "pushed" to another process with different mean later on. Then, the question is, "how do we shape the increasing profile under this changing network environments?"

We deal with this changing network environments by considering the following three typical cases: (i) the available bandwidth drops from a higher level (mean) process to a lower one; (ii) it jumps from a lower level process to a higher one; (iii) it is stationary in between. Theorem 1 has already covered case (iii) and shows the importance of concave-like profile to the predictability of the protocol. Results in Section III-D show that, for cases (i) and (ii), a concave-convex profile or any other profile that spends more time in between can be a good candidate. However, the following questions still remain: first, the role of the convex part is not clear enough, except that it makes the profile 'flatter' in the middle, thus giving smaller variations. Second, we do not know where the transition from being concave to convex (or the inflection point) should take place. In what follows, we will show that the convex part is responsible for fast responsiveness to abrupt changes in the available bandwidth, and that the inflection point should occur around the most likely 'target' – the maximum window size in the previous congestion epoch.

For case (i), a fixed concave profile always aims at the same target x^* regardless of the actual available bandwidth at present. In Figure 4(a), when the average number of flows competing over the bottleneck so abruptly increaes, many flows arrive at t_0 and subsequently the available bandwidth drops from somewhere around L_H to a much smaller value L_L, profile 1 is still in its way aggressively increasing to x^*,



(a) Comparison of profiles when 'target' drops.



(b) Comparison of profiles when 'target' jumps.

Fig. 4. Comparison of concave profiles with or without memory and concave-convex with memory. In (a), when the available bandwidth drops, a concave profile with 'memory' yields higher throughput than the one without memory because the latter leads to multiple consecutive losses at $t_1, t_2, t_3, \ldots$, while the former only causes one at t_1. In (b), when the available bandwidth jumps up, a concave-convex profile with memory responds fast to this change and grabs the available bandwidth quicker by having one consecutive loss only at t_1 in (b). Note that profile 1 and 2 overlap between t_0 and t_2 in (b).

resulting in consecutive losses and reduction in throughput. To solve this problem, a memory of the previous target can be incorporated into the concave profile. This makes it possible for a flow with a concave profile to discover such a change in the available bandwidth and respond very quickly by adjusting its target (changing the curvature of its concave profile). The improvement in throughput by introducing the memory can be seen from the difference between profiles 1 and 2 in Figure 4(a). Note that profile 2 can slow down its increasing rate (still being concave) after its first loss at t_1, leading to reduced packet losses and higher throughput.

On the other hand, as many flows depart to reduce the average number of competing flows and the available bandwidth increases dramatically at t_0 (case (ii)), Figure 4(b) shows the difference in the window size evolution of a concave profile with and without memory after their first loss at t_2. Note that, at this time, unlike in Figure 4(a), consecutive losses do not happen and hence, there will not be much difference in throughput. In fact, as we show in Lemma 1, if profile 2 after t_2 is just a time-compressed version of profile 1, they possess the same long-term mean throughput and variance. However, after t_2, profile 2 can grab more bandwidth than profile 1 in a shorter time as the figure shows, and thus it is better in terms of *responsiveness* and can be used by short-lived flows to finish their transmission earlier. Similarly, profile 2 can be further improved as it still loses its chance to grab more bandwidth in time when the change happens, i.e., profile 2 (and also profile 1) spends too much time in between

t_0 and t_2 to recognize the sudden increase in the available bandwidth. This is done by introducing a fast increasing curve (a convex-like curve) after the flow has slowed down around the previous target for a while (see profile 3), which shows the convex part, first proposed in Section III-D to provide smaller variations combined with the concave part, is indeed a very good candidate for applications requiring fast responsiveness.

V. Simulation

In this section, we verify our theoretical results via NS-2 simulation. Packet losses are generated by using two methods: (i) some pre-defined loss models, and (ii) various cross traffic. The first method enables us to precisely control properties of the loss process and the second method allows us to test the protocols under more realistic Internet-like scenarios. In this section we consider only a stationary loss process. We examine a non-stationary process in Section VI.

A. Protocols to be Simulated

In order to numerically verify our analytic results, we consider several pseudo-protocols. Within a loss interval, a pseudo-protocol sets its congestion window to $f(t)+(1-\beta)w$, where t is the elapsed time since the last congestion epoch, w is the window size just before the last congestion epoch, and β is a decrease factor. We fix β to various values, but in this paper, we report the results from $\beta = 0.3$. The other values do not change our conclusion. We choose the following five functions to represent the typical growth functions of TCP variants : 1) Root function: $f(t) = 300t^{0.5}$, 2) Concave-Convex function: $f(t) = 0.77((t-8.87)^3 + 8.87^3)$, 3) Linear function: $f(t) = 100t$, 4) Power function: $f(t) = 10t^2$, and 5) Exponential function: $f(t) = 8t^2e^{0.02t}$. The coefficients of these functions are chosen such that they achieve similar average window sizes around 1500–1900 packets. We chose these average window sizes because it is simpler to find coefficients giving similar window sizes for all these functions.

B. Case 1: Packet Loss Generated by Loss Models

We first generate packet losses according to a pre-defined loss model in order to measure the impact of different loss distributions on the window size fluctuations. Existing Internet measurement studies show that in the timescale of a few tens of minutes, most of the CoVs of inter-loss intervals are close to 1 (i.e., Poisson) [18], [22] while some could be as high as 2.5 [22]. So we consider the CoV of inter-loss intervals from 1 to 2. We tried both Coxian and Lognormal distributions of loss epochs. Both results show similar results, so we report only Lognormal results.

Figure 5 confirms that the five protocols have approximately the same ordering as predicted by our analytical result: Root $\leq_{cx}$ Linear $\leq_{cx}$ Power $\leq_{cx}$ Exponential, and Root $\leq_{cx}$ Concave-Convex $\leq_{cx}$ Exponential. Also the ordering among the protocols is not changed even with more variations of inter-loss intervals. This results confirm our analytical results.

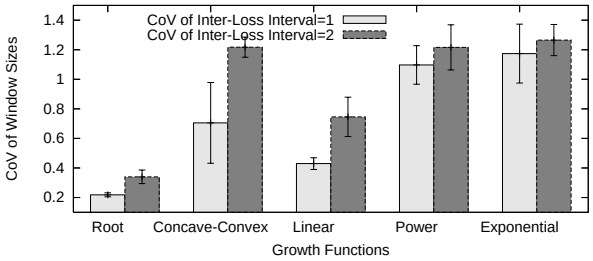


Fig. 5. The CoVs of window sizes of the five pseudo-protocols when we vary the CoV of inter-loss intervals. Even with the variation of inter-loss intervals, the ordering of the CoVs of their window sizes still closely follows our analytical result.

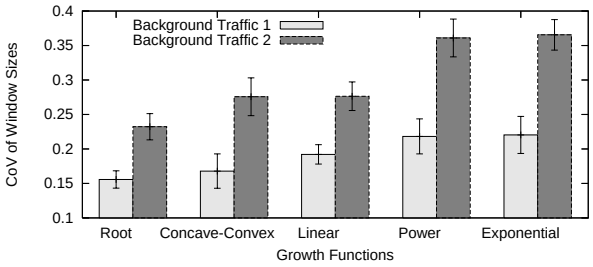


Fig. 6. The CoVs of window sizes of the five pseudo-protocols when competing with two types of background traffic. This simulation result closely follows our analytic result.

C. Case 2: Packet Losses Generated by Background Traffic

We now consider a packet loss process induced by cross traffic. We simulate a dumbbell network, where the bandwidth and one-way delay of the bottleneck link are set to 250Mbps and 50ms, respectively. The bottleneck router implements a DropTail queue discipline and the router buffer size is set to the bandwidth-delay product. To generate different background traffic patterns, we consider two types of background traffic with a different mix of web traffic, medium-size and long-lived TCP traffic: 1) five long-lived forward TCP flows, two forward web sessions, and some backward traffic; 2) 300 forward web sessions, and some backward traffic. In both cases, the total amount of forward background traffic is chosen to consume about 20% of the total link bandwidth.

We measure the CoV of congestion windows of all five pseudo protocols as shown in Figure 6. We see that all the simulation results are consistent with our theoretical result. One interesting finding is that the CoV of Power is almost comparable to that of Exponential. Otherwise it should have a smaller CoV than Exponential. These two functions are very close to each other until Exponential exceeds Power. So the packet losses induced by cross traffic leave these two functions operate in an area where the convexity of their growth functions are similar.

VI. Experimental Evaluation

In this section, we verify the relationship between the window growth function and the second-order behavior of existing high-speed TCP protocols in realistic scenarios using a Linux/FreeBsd based dummynet testbed. All protocols are implemented in Linux kernel 2.6.13. We claim that the profiles of their window growth functions strongly influence their

second-order behaviors and CoVs. All the experimental results can be found from: http://netsrv.csc.ncsu.edu/convex-ordering.

A. Experimental Setup

We use a dumbbell topology of dummynet routers where each end-point consists of a set of Dell Linux servers dedicated to high-speed TCP variant flows and background traffic. Background traffic is generated by using a modification of a web-traffic generator, called Surge [23] and Iperf. The RTT of each background flow is set based on an exponential distribution [24]. The maximum bandwidth of the bottleneck router is set to 400 Mbps. The same amount of background traffic is pushed into forward and backward directions of the dumbbell. Our dummnet router emulates a drop-tail router at the bottleneck.

We test the following MD-style protocols: HSTCP [2], HTCP [5][§], STCP [3], CUBIC [7] and BIC [4]. All are implemented in Linux 2.6.13. Other protocols are not tested because their implementations in the same platform are not available and using the same OS platform is important to reduce OS-dependent issues. These protocols employ different window growth functions with varying convexity. HSTCP (Linear), HTCP (Power), and STCP (Exponential) use convex functions and CUBIC and BIC use concave-convex functions. Depending on the operating range of windows, protocols have different degrees of convexity. CUBIC is much more concave than BIC in our operating range and its behavior is close to a concave protocol. The experimental parameters we control are RTT (40ms to 320ms), buffer sizes (1MB to 8MB), and the degree of congestion in the bottleneck link. The running time of each experiment is from 10 to 20 minutes. We repeat each run at least five times and report only average data from these runs. For each run, we reboot the entire network testbed to remove any system-related dependencies and artifacts. Totally, we have accumulated more than 1500 experimental runs which constitute more than 500 hours worth of experimental data.

B. Impact of RTTs

In this experiment, we fix the number of high-speed flows to four and the buffer size to 1 Mbytes. In each experiment, all the high-speed flows have the same RTT and we vary RTT from 40ms to 320ms for different experiments. Figure 7 shows the CoV of transmission rates of various protocols measured at the bottleneck link for different RTT settings. Clearly, as RTT increases, the transmission rates of protocols become more variable. With 320ms RTT, protocols show the largest variance. This is because as RTT increases, the bandwidth-delay product and congestion window sizes increase. With larger window sizes and small router buffers (1MB), we have more variations in transmission rates. With 320ms RTT, we observe a clear separation between convex protocols and concave-convex protocols. The convex ordering among the protocols is still observed except for HTCP. We can explain it as follows. HTCP adapts its window size by using its quadratic growth function as well as its estimation of buffer size. The quadratic growth function dominates the window size for large buffers. However, when the buffer size is small, we find that HTCP increases and drops its window size very steeply even more than STCP which employs an exponential growth function. We also find that CUBIC performs slightly worse than BIC. Our analysis in Section III-D can be applied to explain this behavior where concave-convex protocols are shown to have smaller variance than pure concave protocols under abrupt target changes. Since CUBIC uses a more concave growth function than BIC (i.e., it stays longer at the flat region than BIC), this argument makes sense.

[§]We applied the latest bug patch from the HTCP author.

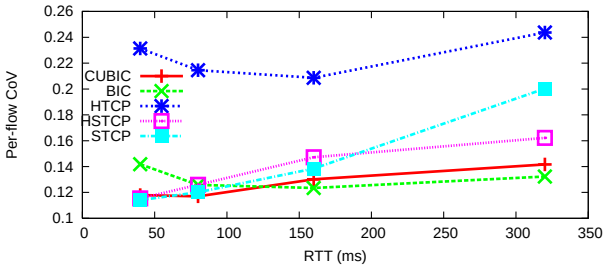


Fig. 7. Impact of RTT on CoV. The buffer size is fixed to 1MB and the number of high-speed flows is four. The CoV of window sizes increases as RTT increases, and under 320ms where they show the worst case performance, we can clearly see that concave-convex protocols have lower variation.

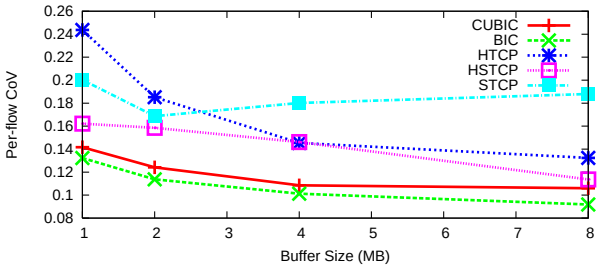


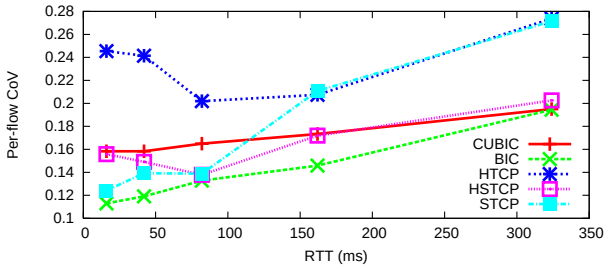
Fig. 8. Impact of buffer size on CoV. As the buffer size increases, the protocols become 'less variable'. A clear separation between convex protocols and concave-convex protocols is visible, independently of buffer sizes.

C. Impact of Buffer Sizes

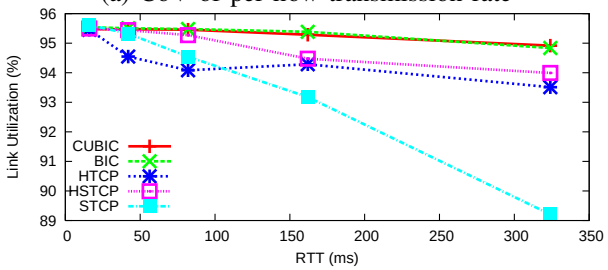
In this experimental scenario, we fix the number of high-speed flows to four and their RTTs to 320ms. Figure 8 shows the average CoV of per-flow rates as we vary the router buffer size. As the router buffer size increases, the CoV for all protocols decreases because the buffer can provide 'cushion' for high rate variation. BIC and CUBIC show the least difference while HTCP gets improved the most. As we observed in the RTT experiment, the performance of HTCP is strongly tied to the router buffer size. When the buffer size increases, we observe that the window growth tends to follow a quadratic function. With large buffers (from 4MB to 8MB), the convex ordering among protocols exactly follows our analytical result. Also, we find clear separation between convex protocols and concave-convex protocols, independently of buffer sizes.

D. Impact of Congestion

So far we have tested under the environments where high-speed flows dominate the bottleneck traffic. To see how the



(a) CoV of per-flow transmission rate



(b) Link utilization

Fig. 9. We add more TCP traffic so that the dominating traffic is not from high-speed flows. Link utilization improves as cross traffic fills in any "gap" left by high-speed flow fluctuations. Even in this environment, we find that protocols with convex profiles have more rate variation and significantly lower link utilization.

ordering would change under more congested environments where regular TCP (SACK) flows dominate, we add a dozen of long-lived TCP flows that start at random time with random RTTs drawn from an exponential distribution. We run the same experiment as in Figure 7 except that this time we have only two flows of high-speed protocols and the router buffer size is increased to 2MB. Figure 9 shows the average CoV of transmission rates and link utilization. In this experiment, we also observe that the same ordering exists between convex protocols and concave-convex protocols (convex protocols having larger rate variations). We also find that the performance of STCP has worsen significantly under 320ms RTT. As STCP is being very aggressive (exponential), even under high congestion, the STCP flow tends to have relatively large window sizes, thus leading to higher variations and the degraded performance.

VII. Conclusion

In this paper, we have examined the high-order behaviors of MD-style protocols via the shape of window growth functions using a powerful stochastic tool called convex ordering. It shows that a protocol employing a window growth function that starts off with a concave growth function and then later switches to a convex growth function around the maximum window size of the last congestion epoch, tends to give the smallest rate variation. BIC and CUBIC are the congestion control protocols that have this property. Our work is significant because it provides a way to compare stochastically any high-order properties of MD-style protocols. The comparison is general enough so that it can be applied to any MD-protocols that might have the same or different first-order behaviors (e.g., different average throughput). In this paper, we study the per-flow dynamics as it directly affects each user's perceived performance and possibly the degree of stability, but a more in-depth study would involve the dynamics of aggregate flows and their impact on the general health of the networks. We leave that study as future work.

References

[1] D. Katabi, M. Handley, and C. Rohrs, "Internet congestion control for high bandwidth-delay product networks," in *Proceedings of ACM SIGCOMM*, Pittsburgh, August 2002.

[2] S. Floyd, "HighSpeed TCP for large congestion windows," *RFC 3649*, December 2003.

[3] T. Kelly, "Scalable TCP: Improving performance in highspeed wide area networks," *ACM SIGCOMM Computer Communication Review*, vol. 33, no. 2, pp. 83–91, April 2003.

[4] L. Xu, K. Harfoush, and I. Rhee, "Binary increase congestion control for fast long-distance networks," in *Proceedings of IEEE INFOCOM*, Hong Kong, March 2004.

[5] R. N. Shorten and D. J. Leith, "H-TCP: TCP for high-speed and long-distance networks," in *Proceedings of the Second PFLDNet Workshop*, Argonne, Illinois, February 2004.

[6] C. Jin, D. X. Wei, and S. H. Low, "FAST TCP: motivation, architecture, algorithms, performance," in *Proceedings of IEEE INFOCOM*, Hong Kong, March 2004.

[7] I. Rhee and L. Xu, "CUBIC: A new TCP-friendly high-speed TCP variant," in *Proceedings of the third PFLDNet Workshop*, France, February 2005.

[8] D. Aldous and A. Bandyopadhyay, "A survey of max-type recursive distributional equations," *The Annals of Applied Probability*, vol. 15, no. 2, pp. 1047–1110, 2005.

[9] S. Deb, S. Shakkottai, and R. Srikant, "Stability and Convergence of TCP-like Congestion Controllers in a Many-Flows Regime," in *Proceedings of IEEE INFOCOM*, San Francisco, CA, April 2003.

[10] F. P. Kelly, "Fairness and stability of end-to-end congestion control," *European Journal of Control*, vol. 9, pp. 159–176, 2003.

[11] J. Padhye, V. Firoiu, and D. Towsley, "A Stochastic Model of TCP Reno Congestion Avoidance and Control," Dept. of Computer Science, University of Massachusetts, Amherst, Tech. Rep., 1999.

[12] E. Altman, K. Avrachenkov, and C. Barakat, "A Stochastic Model of TCP/IP with Stationary Random Loss," in *Proceedings of ACM SIGCOMM*, 2000.

[13] V. Dumas, F. Guillemin, and P. Robert, "Limit results for Markovian models of TCP," in *Proceedings of IEEE GLOBECOM*, 2001.

[14] T. Ott, J. Kemperman, and M. Mathis, "The stationary behavior of ideal TCP congestion avoidance," 1996.

[15] E. Altman, A. A. Kherani, K. Avratchenkov, and B. J. Prabhu, "Performance analysis and stochasitc stability of congestion control protocols," in *Proceedings of IEEE INFOCOM*, Miami, FL, March 2005.

[16] D. Bansal and H. Balakrishnan, "Binomial congestion control algorithms," in *Proceedings of IEEE INFOCOM*, Anchorage, Alaska, April 2001, pp. 631–640.

[17] V. Paxson, "End-to-end Internet packet dynamics," *IEEE/ACM Transactions on Networking*, vol. 7, no. 3, pp. 277–292, June 1999.

[18] Y. Zhang, N. Duffield, V. Paxson, and S. Shenker, "On the constancy of Internet path properties," in *Proceedings of ACM SIGCOMM Internet Measurement Workshop*, November 2001.

[19] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP Throughput: a Simple Model and its Empirical Validation," in *Proceedings of ACM SIGCOMM*, 1998.

[20] P. Brémaud, *Markov Chains: Gibbs Fields, Monte Carlo Simulation, and Queues*. Springer-Verlag, 1999.

[21] H. Cai, D. Y. Eun, S. Ha, I. Rhee, and L. Xu, "Stochastic ordering for internet congestion control and its applications," North Carolina State University, Raleigh, NC, Tech. Rep., Aug. 2006. [Online]. Available: "http://www4.ncsu.edu/~dyeun/pub/Techrep-TCPordering.pdf"

[22] L. Xu and J. Helzer, "Media streaming via TFRC: An analytical study of the impact of TFRC on user-perceived media quality," in *Proceedings of IEEE INFOCOM*, Barcelona, Spain, April 2006.

[23] P. Barford and M. Crovella, "Generating representative web workloads for network and server performance evaluation," in *Measurement and Modeling of Computer Systems*, 1998, pp. 151–160.

[24] J. Aikat, J. Kaur, F. Smith, and K. Jeffay, "Variability in TCP round-trip times," in *Proceedings of the ACM SIGCOMM Internet Measurement Conference*, Miami, FL, October 2003.