Stochastic Ordering for Internet Congestion Control

Han Cai† Do Young Eun† Sangtae Ha‡ Injong Rhee‡ Lisong Xu*

***Abstract*— 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

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.

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. 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.

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. 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.

Further, stability is 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. Thus, measuring the rate variations of flows is commonly used in practice to quantify the practical sense of "protocol stability". For instance, [6], [7], [8] 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, 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. 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.

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. 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], [11]. 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

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

†Department of Electrical and Computer Engineering, North Carolina State University, Raleigh NC 27695

‡Department of Computer Science, North Carolina State University, Raleigh NC 27695

*Department of Computer Science and Engineering, University of Nebraska, Lincoln NE 68588

*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 them interchangeably for convenience.

high order behaviors such as the degree of rate fluctuations. On the other hand, most results via stochastic models have focused on the average values of stochastic quantities [12], [13] or have been obtained under some limiting conditions to make the analysis more tractable [14], [15]. 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 [16] 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

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.

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

Our goal is to compare the stochastic properties of X_n and Y_n in (1). 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):

$$f(t, w) = f(t) + (1-\beta)w, \quad g(t, w) = g(t) + (1-\beta)w \qquad (2)$$

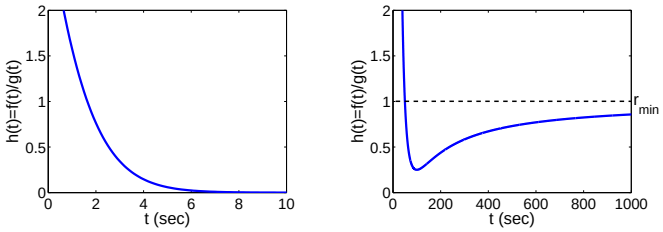
where $f(t), g(t)$ are non-decreasing, $f(0) = g(0) = 0$, $0 < \beta < 1$.

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

$$h(t) := f(t)/g(t) = \mathbb{E}\{X\}/\mathbb{E}\{Y\}. \qquad (3)$$

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 (3) always has *at least* one root. 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 $g(t)$.



(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 $\mathbb{E}\{X\}/\mathbb{E}\{Y\} \in (1, \infty)$.

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 (3) 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], [17]); (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., [18], [19]) and thus, commonly assumed in the stochastic analysis of TCP (e.g.,

[20]). For example, large-scale Internet measurement studies in [19] 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.

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 means that the distributions of all loss intervals do not change over time, rather than that they are the same.

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.

Remark: In Theorem 1, we have considered the process of window sizes only at congestion epochs, i.e., $X_n = W(\tau_n)$ for $n = 1, 2, \ldots$. If we assume that the loss process is Poisson, i.e., $\{T_n\}$ is a sequence of $i.i.d.$ exponential random variables, then we can show that there exists a convex ordering between normalized window size processes *at any time* t [21].

C. Protocols with the Same Mean Behavior

This section shows the importance of stochastic method. When the first-order behavior is under discussion, the fluid method which captures the average behavior is much simpler and convenient than the stochastic counterpart. However, in this section, we show that two protocols with the same fluid model, i.e., the same average behavior, may be different in stochastic sense. From the viewpoint of protocol design, this implies that it is possible to achieve better stochastic property while preserving the same average 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

$$\begin{aligned}\mathbb{E}\{X_{n+1} \mid X_n = w\} &= \mathbb{E}\{f(T)\} + (1-\beta)w \\ &= \mathbb{E}\{g(T)\} + (1-\beta)w = \mathbb{E}\left\{Y_{n+1} \mid Y_n = w\right\}. \quad (4)\end{aligned}$$

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

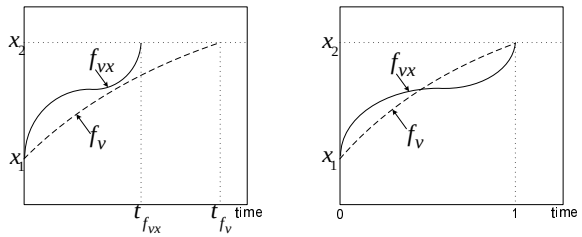
Note that there exists a large set of profiles that satisfy (4). For instance, consider $f(t) = c_1 t^{\alpha_1}$ and $g(t) = c_2 t^{\alpha_2}$. Then, for a given exogenous loss process (i.e., given T), (C1), (C2) and $\mathbb{E}\{f(T)\} = \mathbb{E}\{g(T)\}$ are satisfied if c_i and α_i are chosen in 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.

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 $f(t_f) = x_2$.

[†]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.

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.,

$$\mathbb{E}\{W_f\} = \mathbb{E}\{f(U_{t_f})\} = \mathbb{E}\{g(U_{t_g})\} = \mathbb{E}\{W_g\}, \qquad (5)$$

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 (5) is necessary to avoid trivialities. For instance, for given x_1 and x_2, if we choose a profile f with $f(0) = x_1$ 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 (5) .

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 (5) . 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 differentiating 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.

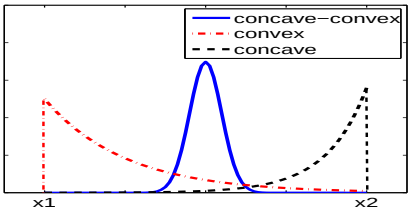


Fig. 3. The probability density functions of concave-convex, convex, and concave profiles. Under 'fair' comparison with the same throughput, a concave-convex profile is least variable, as its probability mass is more concentrated around the mean.

IV. Simulation

In this section, we verify our theoretical results via NS-2 simulation. Packet losses are generated by using various cross traffic. This 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 V.

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: $f(t) = 8t^2 e^{0.02t}$. 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. 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.

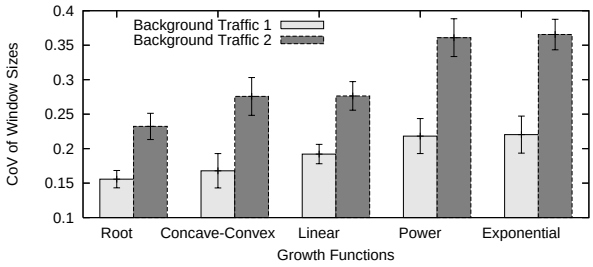


Fig. 4. The CoVs of window sizes of the five pseudo-protocols when competing with two types of background traffic. This simulation result closely follows our analytic result.

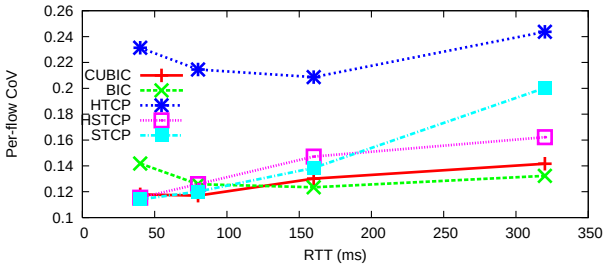


Fig. 5. Impact of RTT on CoV. The buffer size is fixed to 1MB and the number of high-speed flows is four. The CoV of window sizes increases as RTT increases, and under 320ms where they show the worst case performance, we can clearly see that concave-convex protocols have lower variation.

We measure the CoV of congestion windows of all five pseudo protocols. Figure 4 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 background traffic. 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.

V. Experimental Evaluation

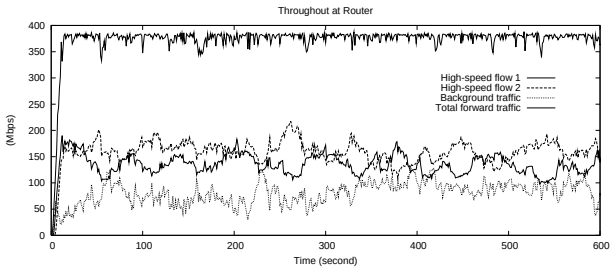
In this section, we verify the relationship between the window growth function and the second-order behavior of existing high-speed TCP protocols using a Linux/FreeBsd based dummynet testbed. We claim that the profiles of their window growth functions strongly influence their second-order behaviors and CoVs. All the experimental results and their details can be found from the following web site: **http://netsrv.csc.ncsu.edu/convex-ordering/**.

A. Experimental Setup

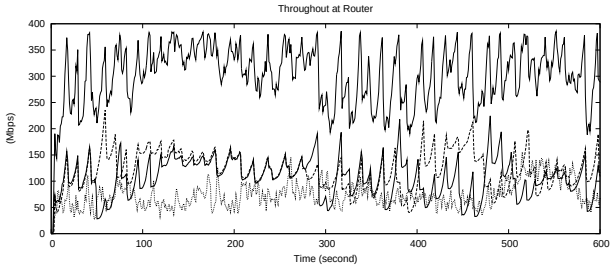
We use a dumbbell topology of dummynet routers where each end-point consists of a set of Dell Linux servers dedicated to high-speed TCP variant flows and background traffic. Background traffic is generated by using a modification of a web-traffic generator, called Surge [22] and Iperf. The RTT of each background flow is set based on an exponential distribution [23]. The same amount of background traffic is pushed into forward and backward directions of the dumbbell. The maximum bandwidth of the bottleneck router is set to 400 Mbps and a drop-tail queue discipline is used.

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. 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) and buffer sizes (1MB to 8MB) 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.

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



(a) BIC (Utilization 94%)



(b) HTCP (Utilization 82%)

Fig. 6. The snapshots of BIC and HTCP over 160 ms RTT shown in Figure 5. The darkline at the top means the total forward traffic rates. HTCP with the lager CoV (0.21) shows severe loss synchronization and under-utilization, while BIC shows a fairly good utilization with the smaller CoV (0.13).

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 5 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 larger window sizes and small router buffers (1MB), we have more variations in transmission rates. Unfortunately, however, these rate-variations (higher CoVs) degrade the network utilization. Figure 6 shows the snapshots of BIC and HTCP under 160 ms RTT. We clearly see HTCP with the higher CoV (0.21) shows severe loss synchronization and under-utilization in the bottleneck, while BIC shows a fairly good utilization with the smaller CoV (0.13).

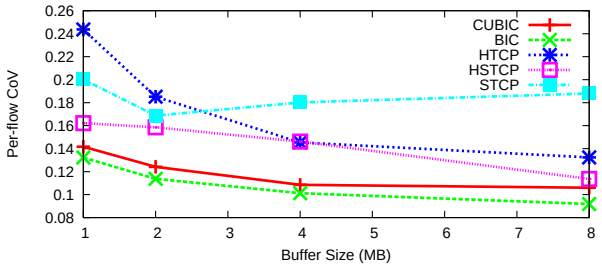


Fig. 7. Impact of buffer size on CoV. As the buffer size increases, the protocols become 'less variable'. A clear separation between convex protocols and concave-convex protocols is visible, independently of buffer sizes.

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 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 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.

C. Impact of Buffer Sizes

In this experiment, we fix the number of high-speed flows to four and their RTTs to 320ms. Figure 7 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.

VI. 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] S. Floyd, M. Handley, J. Padhye, and J. Widmer, "Equation-based congestion control for unicast control algorithms," in *Proceedings of ACM SIGCOMM*, 2000.
[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] R. Johari and D. Tan, "End-to-end congestion control for the Internet: delays and stability," *IEEE/ACM Transactions on Networking*, vol. 9, pp. 818–832, Dec. 2001.
[12] 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.
[13] E. Altman, K. Avrachenkov, and C. Barakat, "A Stochastic Model of TCP/IP with Stationary Random Loss," in *Proceedings of ACM SIGCOMM*, 2000.
[14] T. Ott, J. Kemperman, and M. Mathis, "The stationary behavior of ideal TCP congestion avoidance," 1996.
[15] V. Dumas, F. Guillemin, and P. Robert, "Limit results for Markovian models of TCP," in *Proceedings of IEEE GLOBECOM*, 2001.
[16] 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.
[17] D. Bansal and H. Balakrishnan, "Binomial congestion control algorithms," in *Proceedings of IEEE INFOCOM*, Anchorage, Alaska, April 2001, pp. 631–640.
[18] V. Paxson, "End-to-end Internet packet dynamics," *IEEE/ACM Transactions on Networking*, vol. 7, no. 3, pp. 277–192, June 1999.
[19] 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.
[20] 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.
[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] 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.
[23] 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.