Multiple Time Scale TCP: Harnessing Proactivity in Self-similar Burstiness*

Kihong Park[a†] and Tsunyi Tuan[a]

[a]Network Systems Lab, Department of Computer Sciences, Purdue University,
West Lafayette, IN 47907, U.S.A.

Measurements of network traffic have shown that self-similarity is an ubiquitous phenomenon present across diverse network environments. In previous work, we studied the feasibility of exploiting long-range correlation structure in self-similar traffic in the context of rate-based congestion control. In this work, we extend the multiple time scale control framework to window-based congestion control, in particular, TCP. This is performed by interfacing TCP with a large time scale control module which adjusts the aggressiveness of bandwidth consumption behavior exhibited by TCP as a function of "large time scale" network state, i.e., information that exceeds the time horizon of the feedback loop as determined by RTT. How to effectively utilize such information—due to its probabilistic nature, dispersion over multiple time scales, and affection on top of existing window-based congestion controls—is a nontrivial problem. We define a modular extension of TCP—a function call with a simple interface—that applies to various flavours of TCP (e.g., Tahoe, Reno, Vegas) and show that it significantly improves throughput performance. We show that multiple time scale TCP endows the underlying feedback control with proactivity by bridging the "uncertainty gap" associated with reactive controls which is exacerbated by the high delay-bandwidth product in broadband WANs. By harnessing large time scale correlation structure in network traffic, we are able to mitigate the reactive cost of TCP's feedback congestion control.

1. Introduction

Measurements of local area and wide area traffic have shown that network traffic exhibits variability at a wide range of time scales and that this is an ubiquitous phenomenon which has been observed across diverse networking contexts [6,8,11,17,21]. A number of performance studies have shown that self-similarity can exert a detrimental impact on network performance leading to amplified queueing delay and packet loss rate [1, 12,13]. From a queueing perspective, a principal distinguishing characteristic of long-range dependent traffic is that queue length distribution decays much more slowly—i.e., polynomially—vis-à-vis short-range-dependent traffic sources which exhibit exponential decay. These performance effects, to some extent, can be curtailed by delimiting the buffer

*This work was supported in part by NSF grants ANI-9714707 and EIA-9972883.
†Contact author. E-mail: park@cs.purdue.edu; tel.: +1-765-494-7821, fax.: +1-765-494-0739.
Additionally supported by NSF grants ANI-9875789 (CAREER), ESS-9806741, ANI-0082861 (ITR), and grants from the Purdue Research Foundation, Santa Fe Institute, Sprint, and Xerox.

size which has led to a "small buffer-large bandwidth" resource provisioning strategy [18]. A more comprehensive discussion of performance issues can be found in [15].

The "good news"—within the "bad news" with respect to performance effects—is long-range dependence which, by definition, implies the existence of nontrivial correlation structure at larger time scales, which may be exploitable for traffic control purposes[3]. Long-range dependence and self-similarity of aggregate traffic can be shown to persist at multiplexing points in the network as long as connection durations or object sizes being transported are heavy-tailed, insensitive to details in the protocol stack or network configuration [14].

In previous work [19], we explored the feasibility of exploiting long-range correlation structure in self-similar network traffic for congestion control. We introduced the framework of Multiple Time Scale Congestion Control (MTSC) and showed its effectiveness at enhancing performance for rate-based feedback control. In this work, we extend the multiple time scale traffic control framework to reliable transport and window-based congestion control based on TCP. The adaptation of MTSC to TCP is relevant due to the fact that the bulk of current Internet traffic is governed by TCP, and this is expected to persist due to the growth and dominance of HTTP-based World Wide Web traffic [2,6]. The effective realization of MTSC for TCP is nontrivial due to the following constraints: (a) large time scale correlation structure of network state is inferred by observing the output behavior of a single TCP connection as it shares network resources with other flows at bottleneck routers; (b) we engage probabilistic, large time scale information while instituting minimal changes confined to the sender side; (c) we construct a uniform mechanism in the form of a function call with a simple, well-defined interface that is applicable to a range of TCP flavours; (d) performance of multiple time scale TCP should degenerate to that of TCP when network traffic is short-range dependent. Feedback congestion controls—including TCP—due to their reactive nature suffer under the delay-bandwidth problem of broadband WANs which characterize today's high-speed network environment. The reactive penalty is an intrinsic problem stemming from the outdatedness of reactive actions when subject to long feedback time lags. This also holds for satellite networks which possess a half second RTT latency limitation, over which TCP is increasingly being run as a transport protocol when providing Internet services. Perhaps the most important feature of our work is that multiple time scale TCP—by explicitly engaging correlation structure at large time scales an order of magnitude above and beyond that of typical RTT values—is able to inject timeliness to otherwise outdated feedback information, thus imparting performance-critical proactivity.

2. Technical Set-up

2.1. Basic Definitions

Let $\{X_t; t \in \mathbb{Z}_+\}$ be a time series which represents the trace of data traffic measured at some fixed time granularity. We define the aggregated series $X_i^{(m)}$ as $X_i^{(m)} = \frac{1}{m}(X_{im-m+1}+$

[3]It is of secondary importance whether the *traffic model* of network traffic is assumed mathematically long-range or short-range dependent. The critical feature that matters is that empirical network traffic exhibits correlation structure at the several second time scale (vs. millisecond time scale), information to which current feedback congestion controls acting at the time scale of RTT are impervious.

$\cdots + X_{im})$. Let $r(k)$ and $r^{(m)}(k)$ denote the autocorrelation functions of X_t and $X_i^{(m)}$, respectively, where k is the time lag. Assume X_t has finite mean and variance. X_t is *asymptotically second-order self-similar with parameter* H $(1/2 < H < 1)$ if for all $k \geq 1$, $r^{(m)}(k) \sim \frac{1}{2}\left((k+1)^{2H} - 2k^{2H} + (k-1)^{2H}\right)$ as $m \to \infty$. H is called the *Hurst parameter*. The following two properties are satisfied: (i) $r^{(m)}(k) \sim r(k)$, and (ii) $r(k) \sim c\,k^{-\beta}$, as $k \to \infty$, where $0 < \beta < 1$ and $c > 0$ is a constant. Property (i) states that the correlation structure is preserved with respect to time aggregation, and it is in this second-order sense that X_t is "self-similar." Property (ii) says that $r(k)$ decays hyperbolically which implies $\sum_{k=0}^{\infty} r(k) = \infty$. This is referred to as *long-range dependence* (LRD). The second property hinges on the assumption $1/2 < H < 1$. The relevance of asymptotic second-order self-similarity for network traffic derives from the fact that it plays the role of a "canonical" model where the on/off model of Willinger *et al.* [21], Likhanov *et al.*'s source model [12], and the M/G/∞ queueing model with heavy-tailed service times [5]—among others—all lead to second-order self-similarity.

There is an intimate relationship between heavy-tailed distributions and long-range dependence in the networking context in that the former can be viewed as causing the latter [14,21]. A random variable Z has a *heavy-tailed distribution* if $\Pr\{Z > x\} \sim c\,x^{-\alpha}$, $x \to \infty$, where $0 < \alpha < 2$ is called the *tail index* or *shape parameter* and c is a positive constant. A distinguishing mark of heavy-tailed distributions is that they have infinite variance for $0 < \alpha < 2$, and if $0 < \alpha \leq 1$, they also have an unbounded mean. Further background on self-similar network traffic can be found in [15].

2.2. Long-range Dependence and Predictability

Given X_t and $X_i^{(m)}$, we are interested in estimating $\Pr\{X_{i+1}^{(m)} \mid X_i^{(m)}\}$ for some suitable aggregation level $m > 1$. If X_t is short-range dependent, we have $\Pr\{X_{i+1}^{(m)} \mid X_i^{(m)}\} \sim \Pr\{X_{i+1}^{(m)}\}$ for large m whereas for long-range dependent traffic, correlation provided by conditioning is preserved. Thus given traffic observations $x, y > 0$ $(x \neq y)$ of the "recent" past corresponding to time scale m, $\Pr\{X_{i+1}^{(m)} \mid X_i^{(m)} = y\} \neq \Pr\{X_{i+1}^{(m)} \mid X_i^{(m)} = x\}$ and this information may be exploited to enhance congestion control actions undertaken at smaller time scales. We employ a simple, easy-to-implement prediction scheme to estimate $\Pr\{X_{i+1}^{(m)} \mid X_i^{(m)}\}$ based on observed empirical distribution. We note that *optimum estimation* is a difficult problem for LRD traffic [3], and its solution is outside the scope of this paper. Our estimation scheme provides sufficient accuracy with respect to extracting predictability and is computationally efficient, however, it can be substituted by any other scheme if the latter is deemed "superior" without affecting the conclusions of our results. To facilitate normalized contention levels, we define a map $L : \mathbb{R}_+ \to [1, h]$, monotone in its argument, and let $x_i^{(m)} = L(X_i^{(m)})$. Thus $x_i^{(m)} \approx 1$ is interpreted as the aggregate traffic level at time scale m being "low" and $L_k \approx h$ is understood as the traffic level being "high." The process $x_i^{(m)}$ is related to the *level process* used in [7] for modeling LRD traffic. The large time scale correlation structure is empirically observed to stay invariant in the 1–10 second range. Due to this robustness, as far as large time scale predictability is concerned, picking the exact time is not a critical component. On the other hand, to achieve reasonable responsiveness to changes in large time scale network state, we choose a time scale closer to 1 second than 10 second. We use a 2 second time scale for this reason in the rest of the paper.

3. Multiple Time Scale TCP

3.1. Multiple Time Scale Congestion Control

We use Selective Slope Control (SSC) [19] to engage large time scale correlation structure when modulating the traffic control behavior of a feedback congestion control. SSC adjusts the slope of linear increase during the linear increase phase of AIMD congestion controls based on the predicted large time scale network state. If network contention is low, then slope is increased, and vice versa when network contention is high. Selective slope control is motivated by TCP performance evaluation work [9] which shows that the conservativeness due to asymmetry of TCP's congestion control—necessitated by stability considerations—leads to inefficient utilization of bandwidth that is especially severe in large delay-bandwidth product networks. By varying the slope across persistent network states, SSC is able to modulate the aggressiveness of the feedback congestion control's bandwidth consumption behavior without triggering instability—the slope is held *constant* over a sufficiently large time interval exceeding the RTT or feedback loop by an order of magnitude or more. Due to the large gap in time scale, the feedback congestion control has ample time to converge, and it perceives the slope shifts as stemming from a *quasi-stationary* system for which it is stable. The conjoint evolution of large and small time scale control actions and events is depicted in Figure 1 (left).

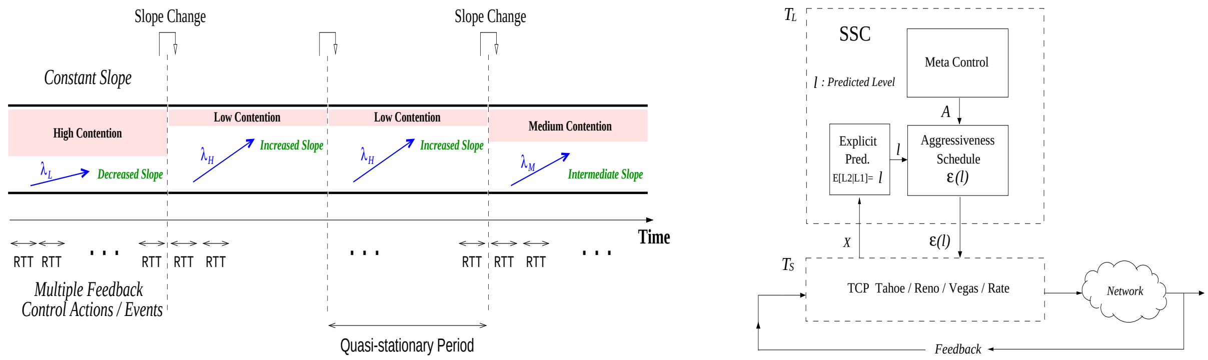

Figure 1. Left: Multiple short-time scale congestion control actions and events (e.g., ACK arrivals and congestion window updates) at the time scale of RTT within a single large time scale window. The slope is constant within a large time scale window. Right: Structure of TCP-MT. Information extracted at large time scale T_L in the Selective Slope Control module is used to modulate the bandwidth consumption behavior of TCP acting at time scale T_S of the feedback loop ($T_L \gg T_S$).

3.2. Structure of TCP-MT

TCP-MT consists of two components: the underlying feedback control—i.e., particular flavour of TCP—and the large time scale module implementing SSC. The large time scale module, in turn, is composed of three parts: (1) an explicit prediction module that extracts large time scale correlation structure on-line, (2) an aggressiveness schedule that determines the final magnitude of slope that is passed to TCP, and (3) a meta control which adjusts the range of slope values to be used by the aggressiveness schedule. The overall structure of TCP-MT is shown in Figure 1 (right).

3.3. Explicit Prediction

We estimate the conditional probability densities $\Pr\{L_2 \mid L_1 = \ell\}$, $\ell \in [1, h]$, on-line using per-connection information without active probing. In TCP, there are a number of approaches—e.g., timeout and ACK arrival pattern, congestion window update, throughput behavior—that can be employed to estimate network state. We use a *uniform* approach to inferring persistent network state where $X_i^{(m)}$ (aggregation m corresponds to the time scale T_L) is defined to be the *number of bits transmitted by TCP over a* T_L *time interval*, which is a simple, observable quantity at the sender side. Although timeouts and ACK arrivals can be used directly to estimate network state, a drawback of this method lies in its dependence on the idiosyncracies of the underlying TCP congestion control—different versions of TCP, principally, diverge in the mechanism they employ to estimate network state—which would require nontrivial customization to couple SSC on top of each TCP. Our approach is predicated on the fact that, whatever the underlying TCP's private estimation and control method, ultimately its impact and effectiveness is captured and conveyed by the sender's *throughput behavior* which is the quantity we employ. On-line estimation can be accomplished using $O(1)$ operations at every update interval, i.e., SSC's time scale T_L. We omit the details due to space constraints.

3.4. Aggressiveness Schedule

In the application of selective slope adjustment, SSC makes the following assumptions about the underlying TCP. The magnitude of congestion window changes in TCP is parameterized by an *aggressiveness constant* $a > 0$—typically $a = 1$ for the TCP flavours considered—and a is replaced by an *aggressiveness variable* ξ. We will use $\texttt{TCP}_\xi$ to denote the parameterized version of TCP. $\texttt{TCP}_\xi$ degenerates to TCP if $\xi = a$. $\texttt{TCP}'_\xi$ is *more aggressive* than $\texttt{TCP}_\xi$ if $\xi' > \xi$ since the slope of increase in the linear increase phase is strictly greater in one over the other. Coupling of the large time scale module with TCP is completed by setting $\xi = \varepsilon(\hat{L}_2)$ where $\hat{L}_2$ is the predicted contention level at time scale T_L, computed by the explicit prediction module. $\varepsilon(\cdot)$ is called the *aggressiveness schedule* and is a decreasing function of $\hat{L}_2 = E[L_2 \,|\, L_1]$. A specific schedule of interest is the *inverse linear schedule* given by $\varepsilon(\hat{L}_2) = (A - a)(h - \hat{L}_2)/(h - 1)$, $\hat{L}_2 \in [1, h]$, where A represents the maximum aggressiveness level. $\hat{L}_2 = 1$ yields the largest slope, and thus, affects the most aggressive action, while $L_2 = h$ yields the least aggressive action reducing to the default slope $\xi = a$. In the latter, TCP-MT degenerates to the default action of its underlying TCP congestion control. It is due to this asymmetry—motivated by [9]—that we call selective slope control a form of selective aggressiveness control. The meta control is responsible for setting the maximum slope value A which, then, in the inverse linear schedule, determines the rest of the values.

More generally, the aggressiveness schedule is made to satisfy $\ell \leq \ell' \Rightarrow \varepsilon(\ell) \geq \varepsilon(\ell')$, and each value $\varepsilon(\ell)$ can be computed separately, i.e., independently of the other values of $\varepsilon(\cdot)$ by the meta control. For TCP-MT, we have used the inverse linear schedule as the default aggressiveness schedule. The generalized schedule—used in multiple time scale redundancy control for real-time data transport [20]—can can yield slightly improved

	Feedback Congestion Control	Coupling with SSC
TCP Reno & Tahoe	$cwnd \leftarrow cwnd{+}\frac{1}{cwnd}$, $ssthresh \leftarrow cwnd/2$	$cwnd \leftarrow cwnd{+}\frac{\varepsilon(\hat{L}_2)}{cwnd}$, $ssthresh \leftarrow cwnd(\hat{L}_2)$
TCP Vegas	$cwnd \leftarrow \begin{cases} cwnd{+}\frac{1}{cwnd}, & \text{if } Diff{<}\alpha, \\ cwnd{-}1, & \text{if } Diff{>}\beta \end{cases}$	$cwnd \leftarrow \begin{cases} cwnd{+}\frac{\varepsilon(\hat{L}_2)}{cwnd}, & \text{if } Diff{<}\alpha, \\ cwnd{-}1, & \text{if } Diff{>}\beta \end{cases}$
TCP Rate	$cwnd \leftarrow \begin{cases} cwnd{+}\frac{a}{cwnd}, & \text{if } \Delta RTT{<}0, \\ cwnd{-}\frac{\tau}{RTT}\, b, & \text{if } \Delta RTT{\geq}0 \end{cases}$	$cwnd \leftarrow \begin{cases} cwnd{+}\frac{\varepsilon(\hat{L}_2)}{cwnd}, & \text{if } \Delta RTT{<}0, \\ cwnd{-}\frac{\tau}{RTT}\, b, & \text{if } \Delta RTT{\geq}0 \end{cases}$

Table 1
Coupling of Selective Slope Control (SSC) with different flavours of TCP.

performance, however, at the cost of increased overhead for estimation.

3.5. Meta Control

The maximum aggressiveness parameter A can be set to a fixed a priori value or, more generally, it can be adjusted dynamically as a function of network state. Since A itself governs the feedback control behavior of the small time scale congestion control (i.e., TCP), dynamic adjustment of A is a form of *meta control*. For a stationary or quasi-stationary (i.e., piece-wise stationary) network environment, A is well-defined and the problem becomes one of designing a control that converges to the equilibrium value of A, call it A^*. A symmetric control law that converges to A^* under stationary and quasi-stationary conditions is given by

$$\frac{dA}{dt} = \begin{cases} \nu, & \text{if } d\gamma_\ell/dA_\ell > 0, \\ -\nu, & \text{if } d\gamma_\ell/dA_\ell < 0, \end{cases}$$

where $\nu > 0$ is an adjustment factor, $\gamma \geq 0$ denotes throughput, and $\ell \in [1, h]$. The control actions are conditioned on the current observed contention level $L_1 = \ell \in [1, h]$, and $d\gamma_\ell/dA_\ell$ is computed with respect to the latest time block classified into the same level ℓ, $\ell \in [1, h]$. Stability analysis of symmetric congestion controls of the above form can be found in [16]. When the network system is "congestion susceptible" in the sense of having a unimodal load-throughput curve, then asymmetry is needed to assure stability. Otherwise, a sufficiently small $\nu > 0$ suffices to achieve asymptotic stability. The reason that the multi-layered feedback control system—feedback control of TCP coupled with the control law governing SSC's meta control—remains stable in spite of a symmetric meta control lies in the large gap between the time scales T_L and T_S. Since A is held constant over time intervals of duration T_L while TCP's congestion control is active, by the stability property of linear increase/exponential decrease control and $T_S \ll T_L$, we have a quasi-stationary system that achieves stability during each T_L interval.

3.6. Instantiations of Couplings with TCP

3.6.1. TCP Reno and Tahoe

Multiple time scale coupling for TCP Reno is constructed in two separate forms, one for its *Congestion Avoidance* component and another for *Slow Start*. The latter is used as a further optimization. By straightforward extension, the same couplings also hold for TCP Tahoe.

Congestion Avoidance. During TCP Reno's congestion avoidance phase, the *aggressiveness constant* a as mentioned in Section 3.4 can be understood as the slope of the congestion window change, i.e., $cwnd \leftarrow cwnd + \frac{a}{cwnd}$ with $a = 1$. The coupling replaces a with $\varepsilon(\hat{L}_2)$ and affects the slope of the linear increase phase such that a more aggressive—but still linear—climb is affected during the next T_L interval if the overall network state is deemed beneficial to do so.

Slow Start. Whenever a timeout occurs, we make an association between the size of the congestion window *cwnd* and the current traffic level L_1, i.e., $cwnd = cwnd(L_1)$. Based on the empirical association, we set the slow-start threshold to $ssthresh \leftarrow cwnd(\hat{L}_2)$ where *cwnd* is indexed by the predicted traffic level $\hat{L}_2$. Similar ways of coupling can be constructed for Reno's Fast Recovery mechanism for further optimization. The dominant performance gain, however, is affected by congestion avoidance.

3.6.2. TCP Vegas

TCP Vegas [4] tries to keep an amount of extra data in the network by maintaining the estimated difference between the actual and expected rate, *Diff*, within pre-specified target bounds $\alpha < Diff < \beta$. If successful, this induces a measure of proactivity by preventing, and thus reducing, timeouts and retransmissions leading to a more continuous, efficient transmission. The coupling with TCP Vegas is achieved through its modified *Congestion Avoidance* mechanism by adjusting the slope of linear increase when $Diff < \alpha$. Thus, except for the triggering event, the coupling instantiation is the same as for Reno and Tahoe.

3.6.3. TCP Rate

TCP Rate is a rate-based extension of TCP Reno which modifies Reno's Congestion Avoidance procedure based on delay variation as shown in Table 1. In the control law, $\Delta\mathtt{RTT}$ is the difference between two consecutive `RTT` values, τ is the packet spacing of the corresponding ACK packets, and $0 < a < b$. Coupling replaces the constant a of the increase part with $\varepsilon(\hat{L}_2)$. We use TCP Rate, in part, to study the influence of the feedback congestion control's "tracking ability" on the effectiveness of the large time scale module SSC. The better the tracking ability of the underlying feedback congestion control with respect to network state, the greater the performance gain due to coupling with SSC.

4. Performance Results

4.1. Network Configuration and Simulation Set-up

We use the LBNL Network Simulator—*ns* (version 2)—as the basis of our simulation environment. TCP-MT was realized by coupling SSC with the various versions of TCP under *ns*. Figure 2 shows a 2-server, n-client ($n \geq 33$) network configuration with a bottleneck link connecting gateways G_1 and G_2. The link bandwidths were set at 10Mbps and the latency on each link was set to 5ms. The maximum segment size was fixed at 1kB. Some of the clients—i.e., 32 connections—act as background traffic for other connections by engaging in interactive transport of files with heavy-tailed sizes across the bottleneck link to the servers, sleeping for an exponential time between successive transfers. The connections whose performance we measure are *infinite sources*—they always have data to send—executing the various flavours of TCP and their corresponding multiple time scale extensions TCP-MT with the objective of maximizing reliable throughput. We study fairness issues by increasing the number of TCP-MT connections while keeping the background traffic flows the same and observing the resulting bandwidth sharing behavior.

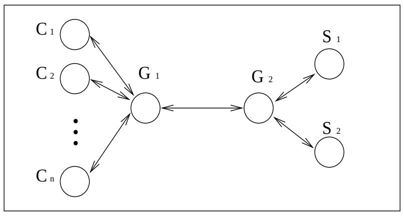

Figure 2. Network configuration with bottleneck link (G_1, G_2).

4.2. Proactivity and RTT

An important—perhaps *the* most important—property of multiple time scale TCP is its ability to mitigate some of the cost of reactive congestion control when subject to long round-trip times (RTT). As the RTT associated with the feedback loop increases, the state information conveyed by feedback becomes more outdated, and the effectiveness of reactive actions undertaken by TCP diminishes. The penalty is especially severe in broadband wide area networks where the delay-bandwidth product increases proportionally with delay or bandwidth. TCP-MT—by exercising explicit prediction at time scale T_L which exceeds the time scale T_S of the feedback loop by an order of magnitude or more—is able to bridge the "uncertainty gap" and affect actions that remain timely and accurate thus offsetting the cost incurred by reactive control.

Figure 3 (left) shows performance gain as a function of RTT where *performance gain* ν is defined as $\nu = (\Lambda_{\text{TCP-MT}} - \Lambda_{\text{TCP}})/\Lambda_{\text{TCP}}$. Λ_{TCP} is the reliable throughput of TCP—for any fixed particular flavour—and $\Lambda_{\text{TCP-MT}}$ is the reliable throughput of the corresponding multiple time scale extension. Thus, assuming $\Lambda_{\text{TCP-MT}} \geq \Lambda_{\text{TCP}}$, $\nu \geq 0$ represents the percentage of improvement achieved by TCP-MT vis-à-vis its underlying TCP. We observe that performance gain amplifies as RTT is increased, reaching up to 45% in the case of TCP Rate for RTT = 450ms. Thus SSC endows the underlying feedback congestion control with proactivity which increases as the feedback loop is increased. In fact, in spite of the diminished room for improvement when going from TCP Reno to Vegas to Rate—the better a feedback congestion control is able to utilize available bandwidth, the less unused bandwidth there is for TCP-MT to further exploit—we observe a robust, even increasing, performance gain when SSC is coupled on top of improved feedback congestion controls.

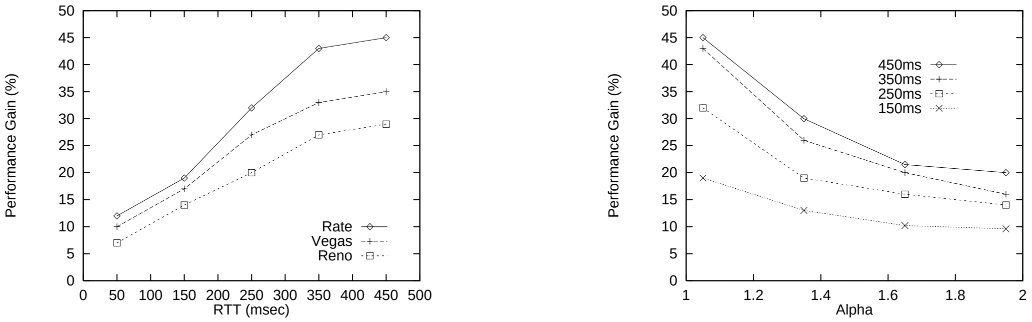

Figure 3. Left: Performance gain as a function of RTT when coupling SSC on top of TCP Reno, Vegas, and Rate. The increasing gain with RTT show the proactivity property of TCP-MT. Right: Impact of long-range dependence as captured by $\alpha = 1.05, 1.35, 1.65, 1.95$ on TCP-MT's performance gain.

4.3. Impact of Long-range Dependence

Another dimension of interest is the impact of long-range correlation structure on performance. As $\alpha \searrow 1$, $H \nearrow 1$ (empirical network traffic has Hurst parameter $H \approx 1$), and the strength of large time scale correlation structure increases. Figure 3 (right) shows performance gain for $\alpha = 1.05, 1.35, 1.65$, and 1.95 background traffic. First, the throughput level for the feedback congestion control (not shown here) is higher for $\alpha = 1.95$ traffic than $\alpha = 1.05$ traffic. This is as expected since self-similar burstiness is known to lead to degraded performance unless resources are overextended, at which point the burstiness associated with short-range dependent traffic can dominate queueing behavior. More importantly, we observe that performance gain increases by a factor of two or more for $\alpha = 1.05$ background traffic when compared with the corresponding gain for $\alpha = 1.95$ traffic. This indicates that self-similar burstiness—although, in general, detrimental to network performance—possesses structure that can be exploited to reduce its negative performance impact. Figure 3 (right) shows that the more long-range dependent the network traffic, the more structure there is to exploit.

4.4. Tracking Ability

The tracking ability of the underlying feedback congestion control can exert a nonnegligible influence on performance and impact the effectiveness of selective slope modulation. The better the feedback congestion control at tracking network state, the more accurate the large time scale correlation structure extracted, resulting in more effective control actions. This dependence of TCP-MT on the underlying TCP congestion control stems from SSC using TCP's *per-connection output behavior* to estimate network contention at large time scales. This is more efficient—in terms of overhead—than constructing a separate state observation module that sends probe packets into the network to estimate state, or assume otherwise cooperation by the network. "Zero overhead probing" using correlation information contained in per-connection TCP output behavior is illustrated in Figure 4 (left).

We measure the *tracking ability* of TCP Reno, Vegas, and Rate by computing the

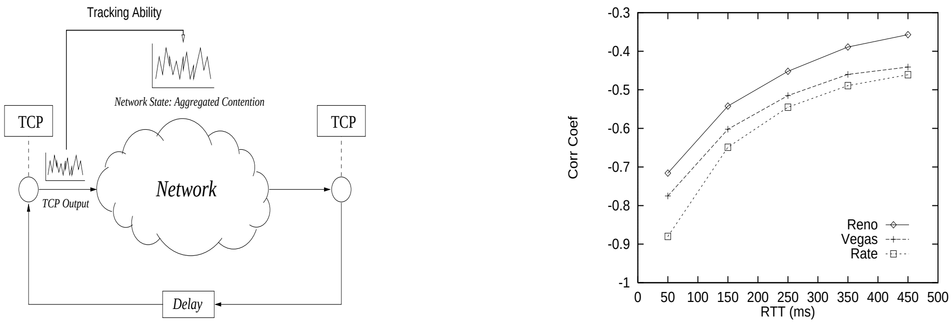

Figure 4. TCP tracking ability. Left: Correlation between per-connection TCP output behavior and aggregated network contention ("zero overhead probing"). Right: Correlation coefficient of per-connection TCP output with bottleneck network contention for Reno, Vegas, and Rate.

correlation coefficients of their reliable throughput with the aggregate background traffic at the bottleneck link (G_1,G_2). Effective tracking implies that when background traffic level is low—i.e., available bandwidth is high—reliable throughput should be high, and vice versa. Hence, under perfect tracking, the correlation coefficient computed should equal -1. The correlation coefficient values for Reno, Vegas, and Rate are shown in Figure 4 (right). We observe that TCP Rate exhibits the best tracking ability followed by Vegas and Reno. Reno's reduced tracking ability can be understood in terms of Reno's linear increase phase during which speedy and accurate discerning of available bandwidth is impeded. Another feature we observe is that as round-trip time increases, tracking ability decreases due to the outdatedness of feedback information which is characteristic of reactive controls.

4.5. Fairness

TCP-MT is designed to run in shared network environments where multiple connections compete for available resources. We investigate the behavior of TCP-MT with respect to fairness when multiple connections engage in SSC. We compare the bandwidth sharing behavior of TCP-MT connections with that of multiple TCP Reno connections. We show that fairness is well preserved when SSC is applied on top of TCP in the sense that bandwidth sharing behavior—and the resultant fairness property—is qualitatively the same as TCP. This also implies that SSC suffers under the same fairness problems as TCP such as those associated with long- and short-latency connections, packet sizes, and window sizes [10]. The results are based on the set-up shown in Figure 2 except for an increase in the bottleneck link bandwidth to 20Mbps to accommodate up to 18 TCP-MT connections for a total of 50. The mean traffic rate of the first 32 connections—i.e., non-SSC background traffic sources—is held constant at 5Mbps. Figure 5 (right) shows that as we increase the number of TCP-MT connections from 2 to 18 (i.e., 33rd connection and beyond), bandwidth continues to be shared fairly in the max-min sense. The spread in individual throughput—even for 18 connections—stays within a narrow range with the individual share decreasing as the number of TCP-MT connections is increased. Figure 5

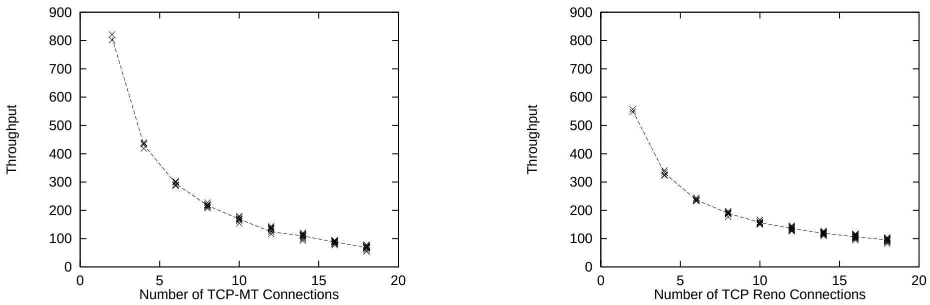

Figure 5. Bandwidth sharing behavior. Left: Dashed line denotes mean throughput of multiple TCP-MT connections; dark marks show the spread of individual throughput values. Right: Corresponding plot for TCP Reno connections.

(left) shows the corresponding performance figures when TCP-MT is replaced by TCP Reno. We observe a qualitatively similar behavior as before.

5. Conclusion

In this paper, we have shown that the multiple time scale congestion control framework [19] can be successfully applied to TCP yielding its multiple time scale extension TCP-MT. The large time scale unit—Selective Slope Control—is modular with a simple, well-defined interface which allows the same module to be coupled on top of various flavours of TCP including Tahoe, Reno, Vegas, and a rate-based extension. The relevance of this work derives from the fact that network traffic has been shown to exhibit self-similarity and long-range dependence, and TCP—being a dominant protocol governing the bulk of current Internet traffic—can benefit from performance improvement stemming from a novel traffic control dimension: self-similar burstiness of network traffic. An important property of TCP-MT is its ability to mitigate the performance cost of reactive congestion controls, which is especially severe in broadband wide area networks where the delay-bandwidth product is high. By engaging predictability structure resident at time scales exceeding typical RTT values, TCP-MT is able to offset the outdatedness of feedback information and, thus, inject a measure of proactivity.

REFERENCES

1. R. Addie, M. Zukerman, and T. Neame. Fractal traffic: measurements, modelling and performance evaluation. In *Proc. IEEE INFOCOM '95*, pages 977–984, 1995.
2. M. F. Arlitt and C. L. Williamson. Web server workload characterization: The search for invariants. In *Proceedings of SIGMETRICS '96*, pages 126–137, May 1996.
3. Jan Beran. *Statistics for Long-Memory Processes.* Monographs on Statistics and Applied Probability. Chapman and Hall, New York, NY, 1994.
4. L. Brakmo and L. Peterson. TCP Vegas: end to end congestion avoidance on a global internet. *IEEE J. Select. Areas Commun.*, 13(8):1465–1480, 1995.

5. D. R. Cox. Long-range dependence: a review. In H. A. David and H. T. David, editors, *Statistics: An Appraisal*, pages 55–74. Iowa State Univ. Press, 1984.
6. M. Crovella and A. Bestavros. Self-similarity in world wide web traffic: Evidence and possible causes. *IEEE/ACM Trans. Networking*, 5:835–846, 1997.
7. N. Duffield and W. Whitt. Network design and control using on-off and multi-level source traffic models with heavy-tailed distributions. In K. Park and W. Willinger, editors, *Self-Similar Network Traffic and Performance Evaluation.* Wiley-Interscience, 2000.
8. C. Huang, M. Devetsikiotis, I. Lambadaris, and A. Kaye. Modeling and simulation of self-similar variable bit rate compressed video: a unified approach. In *Proc. ACM SIGCOMM '95*, pages 114–125, 1995.
9. Hyogon Kim. *A Non-Feedback Congestion Control Framework for High-Speed Data Networks.* PhD thesis, University of Pennsylvania, 1995.
10. T. V. Lakshman and U. Madhow. The performance of TCP/IP for networks with high bandwidth-delay products and random loss. *IEEE/ACM Trans. Networking*, 5(3):336–350, 1997.
11. W.E. Leland, M.S. Taqqu, W. Willinger, and D.V. Wilson. On the self-similar nature of Ethernet traffic (extended version). *IEEE/ACM Transactions on Networking*, 2:1–15, 1994.
12. N. Likhanov, B. Tsybakov, and N. Georganas. Analysis of an ATM buffer with self-similar ("fractal") input traffic. In *Proc. IEEE INFOCOM '95*, pages 985–992, 1995.
13. I. Norros. A storage model with self-similar input. *Queueing Systems*, 16:387–396, 1994.
14. K. Park, G. Kim, and M. Crovella. On the relationship between file sizes, transport protocols, and self-similar network traffic. In *Proc. IEEE International Conference on Network Protocols*, pages 171–180, 1996.
15. K. Park and W. Willinger, editors. *Self-Similar Network Traffic and Performance Evaluation.* Wiley-Interscience, 2000.
16. Kihong Park. Warp control: a dynamically stable congestion protocol and its analysis. In *Proc. ACM SIGCOMM '93*, pages 137–147, 1993.
17. V. Paxson and S. Floyd. Wide-area traffic: the failure of Poisson modeling. In *Proc. ACM SIGCOMM '94*, pages 257–268, 1994.
18. J. W. Roberts. Engineering for quality of service. In K. Park and W. Willinger, editors, *Self-Similar Network Traffic and Performance Evaluation.* Wiley-Interscience, 2000.
19. T. Tuan and K. Park. Multiple time scale congestion control for self-similar network traffic. *Performance Evaluation*, 36:359–386, 1999.
20. T. Tuan and K. Park. Multiple time scale redundancy control for QoS-sensitive transport of real-time traffic. In *Proc. IEEE INFOCOM '00*, pages 1693–1692, 2000.
21. W. Willinger, M. Taqqu, R. Sherman, and D. Wilson. Self-similarity through high-variability: statistical analysis of Ethernet LAN traffic at the source level. In *Proc. ACM SIGCOMM '95*, pages 100–113, 1995.