Performance Evaluation of Multiple Time Scale TCP under Self-similar Traffic Conditions

Kihong Park
and
Tsunyi Tuan
Network Systems Lab
Department of Computer Sciences
Purdue University
West Lafayette, IN 47907, USA

K. Park was additionally supported by NSF grants ANI-9875789 (CAREER), ESS-9806741, EIA-9972883, and grants from the Purdue Research Foundation, Santa Fe Institute, and Sprint.

This work was supported in part by NSF grant ANI-9714707.
Name: Kihong Park
Affiliation: Department of Computer Sciences, Purdue University
Address: West Lafayette, IN 47907; e-mail: park@cs.purdue.edu, tel.: 765-494-7821

Name: Tsunyi Tuan
Affiliation: Purdue University
Address: West Lafayette, IN 47907

Measurements of network traffic have shown that self-similarity is an ubiquitous phenomenon spanning across diverse network environments. In previous work, we have explored the feasibility of exploiting long-range correlation structure in self-similar traffic for congestion control. We have advanced the framework of multiple time scale congestion control and shown its effectiveness at enhancing performance for rate-based feedback control. In this paper, 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. Our contribution is threefold. First, 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 performance. Second, 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 wide area networks. Third, we investigate the influence of three traffic control dimensions—tracking ability, connection duration, and fairness—on performance. Performance evaluation of multiple time scale TCP is facilitated by a simulation benchmark environment which is based on physical modeling of self-similar traffic. We explicate our methodology for discerning and evaluating the impact of changes in transport protocols in the protocol stack under self-similar traffic conditions. We discuss issues arising in comparative performance evaluation under heavy-tailed workloads.

Categories and Subject Descriptors: C.2 [**Computer Systems Organization**]: Computer-Communication Networks

General Terms: Algorithms, Performance

Additional Key Words and Phrases: Network protocols, TCP, self-similar traffic, simulation, multiple time scale, congestion control, performance evaluation

---

1. INTRODUCTION

1.1 Background

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, from Ethernet to ATM, VBR video, and WWW traffic [Crovella and Bestavros 1996; Garret and Willinger 1994; Huang et al. 1995; Leland et al. 1994; Paxson and Floyd 1994; Willinger et al. 1995]. A number of performance studies have shown that self-similarity can have a detrimental impact on network performance leading to amplified queueing delay and packet loss rate [Adas and Mukherjee 1995; Addie et al. 1995; Duffield and O'Connel 1993; Likhanov et al. 1995; Norros 1994]. 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 size which has led to a "small buffer capacity-large bandwidth" resource provisioning strategy [Grossglauser and Bolot 1996; Ryu and Elwalid 1996]. A more comprehensive discussion of performance issues is provided in [Park and

Willinger 2000a].

The problem of *controlling* self-similar network traffic is still in its infancy. By the control of self-similar traffic, we mean the problem of regulating traffic flow—possibly exploiting the properties associated with self-similarity and long-range dependence—such that network performance is optimized. 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, information to which current traffic control algorithms are impervious. 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, irrespective of buffer capacity and details in the protocol stack or network configuration [Feldmann et al. 1998; Park et al. 1996]. How to effectively utilize large time scale, probabilistic information afforded by traffic characteristics to improve performance is a nontrivial problem.

In previous work [Tuan and Park 1999], we have 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. We showed that by incorporating correlation structure at large time scales into a generic rate-based feedback congestion control, we are able to improve performance significantly. In [Tuan and Park 2000], we applied MTSC to the control of real-time multimedia traffic—in particular, MPEG video—using adaptive redundancy control, and we showed that end-to-end quality of service (QoS) is significantly enhanced by utilizing large time correlation structure in both the background and source traffic. The real-time traffic control framework is called Multiple Time Scale Redundancy Control which improves on earlier work in packet-level adaptive forward error correction for end-to-end QoS control [Park and Wang 1999; Park 1997a].

1.2 New Contributions

In this paper, we extend the multiple time scale traffic control framework to reliable transport and window-based congestion control based on 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 round-trip time (RTT). 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 [Arlitt and Williamson 1996; Barford and Crovella 1998; Crovella and Bestavros 1996]. 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.

Our contribution is threefold. First, we construct a robust, modular extension of TCP—a function call with a simple, well-defined interface which adjusts a single constant (now a variable) in TCP's congestion window update. The same extension applies to various flavours of TCP including Tahoe, Reno, Vegas, and rate-based extensions. We show that the resulting protocol—Multiple Time Scale TCP (TCP-MT)—significantly improves performance. Performance gain is measured by the ratio of reliable throughput of TCP-MT vs. the throughput of the corresponding TCP without the large time scale component. We show that performance gain is increased as long-range dependence is increased approaching that of measured network traffic.

Second, 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 of broadband wide area networks [Kim and Farber 1995; Lakshman and Madhow 1997; Pecelli and Kim 1995]. As RTT increases, the information conveyed by feedback becomes more outdated, and the effectiveness of reactive actions undertaken by a feedback control diminishes. TCP-MT, by exploiting large time scale information exceeding the scope of the feedback loop, can affect control actions that remain timely and accurate, thus offsetting the cost incurred by reactive control. It is somewhat of an "irony" that self-similar burstiness which, in addition to its first-order performance effects causes second-order effects in the form of concentrated periods of over- and under-utilization, can nonetheless help mitigate the Achilles' heel of feedback traffic controls which has been a dominant theme of congestion control research in the 1990s.

Third, we investigate the influence of three traffic control dimensions—tracking ability, connection duration, and fairness—on performance. Tracking ability refers to a feedback control's ability to track system state by its interaction with other flows at routers. It is relevant when performing on-line estimation of large time scale correlation structure using per-flow input/output behavior. TCP-MT yields the highest performance gain when connection duration is long. Since network measurements have shown that most connections are short-lived but *the bulk of traffic is contributed by the few long-lived ones* [Feldmann et al. 1998; Park et al. 1996], effectively managing the long-lived ones—by Amdahl's law—is important for system performance. We complement this basic focus by exploring ways of actively managing short connections using a priori and shared information across connections. With respect to fairness, we show that the bandwidth sharing behavior of TCP-MT is similar to that of TCP, neither improving nor deproving the well-known (un)fairness properties associated with TCP [Lakshman and Madhow 1997].

1.3 Simulation-based Protocol Evaluation under Self-similar Traffic

Our performance evaluation method is based on a simulation benchmark environment that is derived from physical modeling of self-similar network traffic [Park et al. 1996]. Setting up a framework where the impact of changes in transport protocols—under self-similar traffic conditions—can be effectively discerned and evaluated is a nontrivial problem. Feedback control induces a *closed system* where the very control actions that are subject to modification can affect the traffic properties and performance being measured. To yield meaningful experimental evaluations and facilitate a comparative benchmark environment where "other things being equal" holds, the meaning of *self-similar traffic conditions* needs to be made precise and well-defined. Physical models show that self-similarity in network systems is primarily caused by an application layer property—heavy-tailed objects on WWW servers, UNIX file servers [Arlitt and Williamson 1996; Crovella and Bestavros 1996; Park et al. 1996]—whose transport, as mediated by the protocol stack, induces self-similarity at multiplexing points in the network. Moreover, the degree of long-range dependence as measured by the Hurst parameter is directly determined by the tail index (i.e., heavy-tailedness) of heavy-tailed distributions. Thus by varying the tail index in the application layer, we can influence—and keep constant across different experimental set-ups—the intrinsic propensitity of the system to generate and experience self-similar burstiness in its network traffic while at the same time incorporating the modulating influence of transport protocols in the protocol stack. Related to the comparative performance evaluation issue, we discuss problems associated with sampling from heavy-tailed distributions, and the solution we employ to facilitate comparative evaluation.

The rest of the paper is organized as follows. In the next section, we give a brief overview of self-similar network traffic, its predictability properties, and the method employed to achieve on-line estimation of large time scale correlation structure. Section 3 describes the multiple time scale congestion control framework for TCP, the form of the large time scale module including its instantiation on top of Tahoe, Reno, Vegas, and rate-based extensions. Section 4 discusses simulation issues and describes the performance evaluation environment employed in the paper. In Section 5 we present performance results of TCP-MT and show its efficacy under varying resource configurations, couplings with different TCP flavours, round-trip times, long-range dependence, and resource sharing behavior as the number of TCP-MT connections competing for network resources is increased. We conclude with a discussion of our results and future work.

2. TECHNICAL BACKGROUND AND SET-UP

2.1 Self-similarity and Long-range Dependence

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

$$X_i^{(m)} = \frac{1}{m}(X_{im-m+1} + \cdots + X_{im}).$$

That is, X_t is partitioned into blocks of size m, their values are averaged, and i is used to index these blocks. 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 $k \geq 1$,

$$r^{(m)}(k) \sim \frac{1}{2}\big((k+1)^{2H} - 2k^{2H} + (k-1)^{2H}\big), \qquad m \to \infty. \tag{1}$$

H is called the *Hurst parameter* and its range $1/2 < H < 1$ plays a crucial role. The significance of (1) stems from the following two properties being satisfied:

(i) $r^{(m)}(k) \sim r(k)$,
(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 that $1/2 < H < 1$ as $H = 1 - \beta/2$. 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.*[1] [Willinger et al. 1995], Likhanov *et al.*'s source model [Likhanov et al. 1995], and the M/G/∞ queueing model with heavy-tailed service times [Cox 1984]—among others—all lead to second-order self-similarity. In general, self-similarity and long-range dependence are not equivalent. For example, fractional Brownian motion with $H = 1/2$ is self-similar but it is not long-range dependent. For second-order self-similarity with $H > 1/2$, however, one implies the other and it is for this reason that we sometimes use the terms interchangably within the traffic modeling context. A more comprehensive discussion can be found in [Park and Willinger 2000b].

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 [Feldmann et al. 1998; Park et al. 1996; Willinger et al. 1995]. We say a random variable Z has a *heavy-tailed distribution* if

$$\Pr\{Z > x\} \sim c\,x^{-\alpha}, \qquad x \to \infty \tag{2}$$

where $0 < \alpha < 2$ is called the *tail index* or *shape parameter* and c is a positive constant. That is, the tail of the distribution, asymptotically, decays hyperbolically. This is in contrast to *light-tailed distributions*—e.g., exponential and Gaussian—which possess an exponentially decreasing tail. A distinguishing mark of heavy-tailed distributions is that they have infinite variance for $0 < \alpha < 2$, and if $0 < \alpha \le 1$, they also have an unbounded mean. In the networking context, we will be primarily interested in the case $1 < \alpha < 2$. This is due to the fact that when heavy-tailedness causes self-similarity, the Hurst parameter is related to the tail index by $H = (3 - \alpha)/2$. A frequently used heavy-tailed distribution is the *Pareto distribution* whose distribution function is given by

$$\Pr\{Z \le x\} = 1 - (b/x)^{\alpha}$$

where $1 < \alpha < 2$ is the shape parameter and $0 < b \le x$ is called the *location parameter*. Its mean is given by $\alpha b/(\alpha - 1)$. A random variable obeying a heavy-tailed distribution exhibits extreme variability. Practically speaking, a heavy-tailed distribution gives rise to very large values with nonnegligible probability so that sampling from such a distribution results in the bulk of values being "small" but a

[1] That is, via its relation to fractional Brownian motion and its increment process, fractional Gaussian noise.

few samples having "very" large values. Not surprisingly, heavy-tailedness impacts sampling by slowing down the convergence rate of the sample mean to the population mean, dilating it as the tail index α approaches 1. Sampling and convergence issues are discussed in Section 4.3.

2.2 Long-range Dependence and Predictability

Given X_t and $X_i^{(m)}$, we will be 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 $a, b > 0$ ($a \neq b$) of the "recent" past corresponding to time scale m,

$$\Pr\{X_{i+1}^{(m)} \mid X_i^{(m)} = b\} \neq \Pr\{X_{i+1}^{(m)} \mid X_i^{(m)} = a\}$$

and this information may be exploited to enhance congestion control actions undertaken at smaller time scales. We employ a simple, easy-to-implement—both on-line and off-line—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 [Beran 1994], 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 $L_k \approx s$ is understood as the traffic level being "high." The process $x_i^{(m)}$ is related to the *level process* used in [Duffield and Whitt 2000] for modeling LRD traffic.

Figure 1 shows the estimated conditional probability densities for $\alpha = 1.05$ (long-range dependent) and 1.95 (short-range dependent) traffic for absolute time scales[2] $T = 1$ second and 5 second. The quantization level is set to $h = 8$. We use L_1 and L_2 without reference to the specific time index i to denote consecutive quantized traffic levels $x_i^{(m)}$, $x_{i+1}^{(m)}$. Therefore, in a causal system, the pair (L_1,L_2) can be used to represent the current observed network traffic level and the predicted traffic level based on the current observation, respectively. For the aggregate throughput traces with $\alpha = 1.05$—Figure 1 (top row)—the 3-D conditional probability densities can be seen to be skewed diagonally from the lower left side toward the upper right side. This indicates that if the current traffic level L_1 is low, say $L_1 = 1$, chances are that L_2 will be low as well. That is, the probability mass of $\Pr\{L_2 \mid L_1 = 1\}$ is concentrated *toward* 1. Conversely, the plots show that $\Pr\{L_2 \mid L_1 = 8\}$ is concentrated *toward* 8. Thus for $\alpha = 1.05$ traffic, conditioning at time scales $t =$ 1s and 5s does help predict the future. The corresponding probability densities for $\alpha = 1.95$ traffic are shown in Figure 1 (bottom row). We observe that the shape of the distribution is insensitive to conditioning (i.e., $\Pr\{L_2 \mid L_1\} \approx \Pr\{L_2\}$) which

[2]The corresponding aggregation levels, expressed with respect to $X_i^{(m)}$, are $m = 100$ and 500.

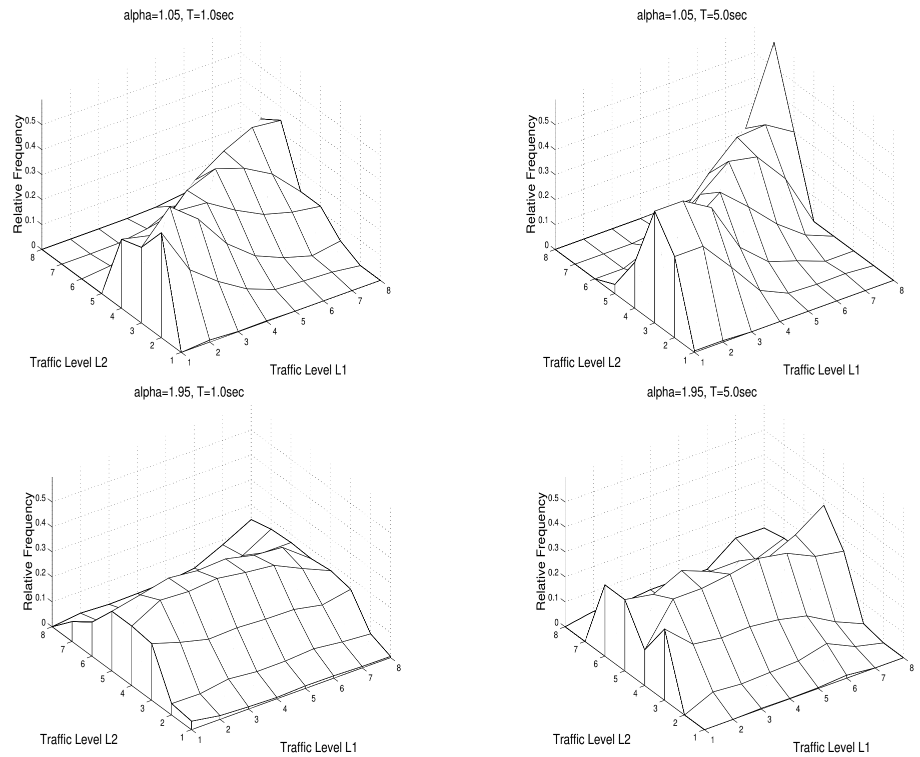

Fig. 1. Top Row: Probability densities with L_2 conditioned on L_1 for $\alpha = 1.05$ with time scales of 1 second (left) and 5 second (right). Bottom Row: Corresponding probability densities with L_2 conditioned on L_1 for $\alpha = 1.95$.

implies a lack of predictability structure at large time scales. At short time scales, both $\alpha = 1.05$ and 1.95 traffic contain predictability, structure toward which current protocols—feedback or otherwise—are geared to. The large time scale correlation structure is empirically observed to stay invariant in the 1–10 second range (cf. the distributions for 1 second and 10 second time scales). Due to this robustness, as far as 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

The framework of multiple time scale congestion control [Tuan and Park 1999], in general, allows for n-level time scale congestion control for $n \geq 1$ where information extracted at n separate time scales is cooperatively engaged to modulate the output behavior of the feedback congestion control residing at the lowest time scale (i.e., $n = 1$). The ultimate goal of MTSC is to improve performance vis-à-vis the congestion control consisting of feedback congestion control alone. Thus even when

$n > 1$, if the large time scale modules are deactivated, then the congestion control degenerates to the original feedback congestion control.

We distinguish two strategies for engaging large time scale correlation structure to modulate the traffic control behavior of a feedback congestion control. The first method—Selective Slope Control (SSC)—adjusts the slope of linear increase during the linear increase phase of linear increase/exponential decrease 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. This is depicted in Figure 2 (left). Selective slope control is motivated by TCP performance evaluation work [Kim 1995; Kim and Farber 1995] which shows that the conservativeness or 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 provably stable. We have shown the effectiveness of SSC in the context of rate-based feedback congestion control [Tuan and Park 1999], and we adopt it as the basic strategy for realizing multiple time scale TCP.

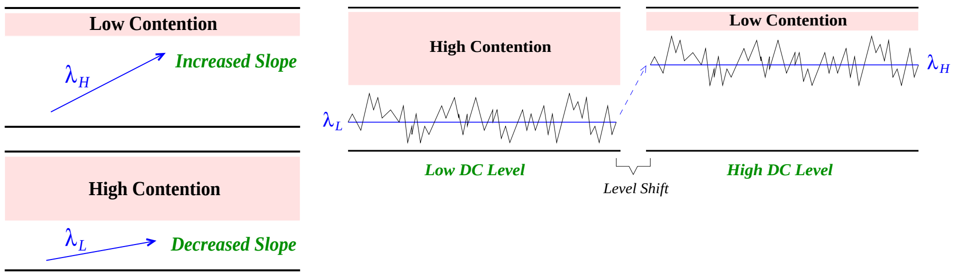

Fig. 2. Left: Selective slope adjustment—i.e., slope shift—during linear increase phase for high- and low-contention periods. Right: Selective "DC" level adjustment—i.e., level shift—between high- and low-contention periods.

The second method for utilizing large time scale correlation structure in feedback traffic controls is called Selective Level Control (SLC), and it *additively* adjusts output rate as a function of large time scale network state, increasing the "DC" level when network contention is low and decreasing it when the opposite is true. This is depicted in Figure 2 (right). SLC is a more general scheme not necessarily customized toward congestion control. For example, we have employed SLC successfully for real-time multimedia traffic control where adaptive packet-level forward error correction is applied to facilitate timely arrival and decoding of MPEG I video frames when retransmission is infeasible [Tuan and Park 2000]. It is a UDP based videoconferencing implementation running over UNIX and Windows

NT where SLC is built on top of AFEC, an adaptive redundancy control protocol for achieving user-specified end-to-end QoS [Park and Wang 1999; Park 1997a].

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: an explicit prediction module that extracts large time scale correlation structure on-line, an aggressiveness schedule that determines the final magnitude of slope that is passed to TCP, and a meta control which adjusts the range of slope values to be used by the aggressiveness schedule. SSC bases its computation on the underlying feedback congestion control's per-flow, observable input-output behavior—number of TCP segments transmitted as well as incoming ACKs. Only the sender-side is augmented by the large time scale module; the receiver-side stays untouched. The overall structure of TCP-MT is depicted in Figure 3. The next sections describe the various components of TCP in more detail including the specific instantiations on top of Tahoe, Reno, Vegas, and a rate-based extension of TCP.

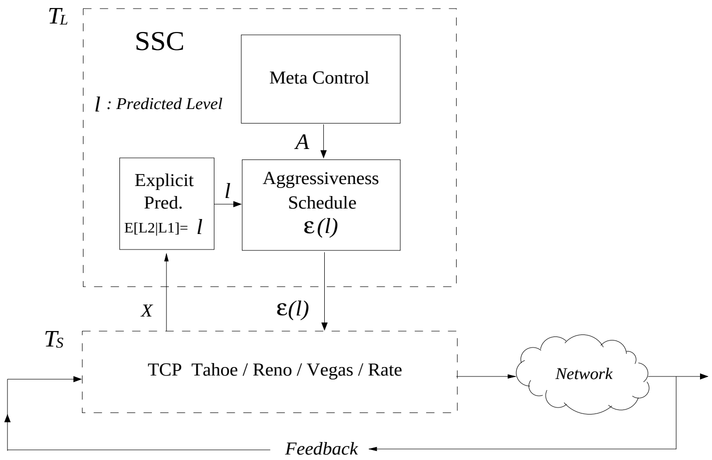

Fig. 3. Structure of TCP-MT. Information extracted at large time scale T_L in the Selective Slope Control module is used to modulate the bandwidth consumption behavior of TCP acting at time scale T_S of the feedback loop ($T_L \gg T_S$).

3.3 Explicit Prediction

Per-connection, on-line estimation of conditional probability densities $\Pr\{L_2 \mid L_1 = \ell\}$, $\ell \in [1, h]$, is achieved via Bayesian estimation. 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 that 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. The same approach was successfully used in the rate-based congestion control context [Tuan and Park 1999]. We note that the meaning of the quantization $x_i^{(m)} = L(X_i^{(m)})$ is reversed—high $X_i^{(m)}$ implies large available bandwidth, and small $X_i^{(m)}$ implies that available bandwidth is small.

On-line estimation can be accomplished using $O(1)$ operations at every update interval, i.e., SSC's time scale T_L. On the sender side, the explicit prediction module of SSC maintains a 2-dimensional array `CondProb`$[\cdot][\cdot]$ of size $h \times (h+1)$, one row for each $\ell \in [1, h]$. The last column of `CondProb`, `CondProb`$[\ell]$[h+1], is used to keep track of h_ℓ, the number of blocks observed thus far whose traffic level map to ℓ. For each $\ell' \in [1, h]$, `CondProb`$[\ell][\ell']$ maintains the count $h_{\ell'}$. Since

$$\Pr\{L_2 = \ell' \mid L_1 = \ell\} \;=\; h_{\ell'}/h_\ell$$

in the long run, having the table `CondProb` is tantamount to knowing the conditional probability densities. Given a current observed traffic level $x > 0$ at time scale T_L, we compute the conditional expectation $\ell = E[L_2 \,|\, L_1 = x]$ which is then used to index the aggressiveness schedule. A discussion of the conditional mean as a predictor for long-range dependent traffic can be found in [Beran 1994].

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. That is, it is turned into a control variable. 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

$$\varepsilon(\hat{L}_2) = \frac{A-a}{h}(\hat{L}_2 - h) + a, \qquad \hat{L}_2 \in [1, h]$$

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 [Kim 1995; Kim and Farber 1995]—that we call selective slope control a form of selective aggressiveness control[3]. 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 [Tuan and Park 2000]—can can yield slightly improved performance, however, at the cost of more overhead for estimation. Moreover, the individually estimated $\varepsilon(\ell)$ values are approximated by a linear aggressiveness schedule [Tuan and Park 2000]. It is for these reasons that we use the inverse linear schedule in the paper. The effect of using a more conservative inverse schedule, $\varepsilon(\hat{L}_2) = a + A/\hat{L}_2$, is studied in [Tuan and Park 1999]. The *threshold schedule*,

$$\varepsilon(\ell) = \begin{cases} A, & \text{if } \ell \leq \theta, \qquad \theta \in [1, h], \\ a, & \text{otherwise.} \end{cases}$$

is a performance evaluation tool which is used to discern the impact of *statically* varying aggressiveness. As θ is increased, the underlying congestion control is made more aggressive.

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, $\gamma \geq 0$ is 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 [Park 1993]. 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 [Park 1993]. The reason that the multi-level feedback control system—feedback control of TCP coupled with the control law

[3]The generalization to $\xi < a$ is of interest and a task for future work.

	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	$cwnd \leftarrow \begin{cases} cwnd + \frac{1}{cwnd}, & \text{if } Diff < \alpha, \\ cwnd, & \text{if } \alpha < Diff < \beta, \\ cwnd - 1, & \text{if } Diff > \beta \end{cases}$	$cwnd \leftarrow \begin{cases} cwnd + \frac{\varepsilon(\hat{L}_2)}{cwnd}, & \text{if } Diff < \alpha, \\ cwnd, & \text{if } \alpha < Diff < \beta, \\ cwnd - 1, & \text{if } Diff > \beta \end{cases}$
TCP Rate	$cwnd \leftarrow \begin{cases} cwnd + \frac{a}{cwnd}, & \text{if } \Delta RTT < 0, \\ cwnd - \frac{\tau}{RTT}\, b, & \text{if } \Delta RTT \geq 0 \end{cases}$	$cwnd \leftarrow \begin{cases} cwnd + \frac{\varepsilon(\hat{L}_2)}{cwnd}, & \text{if } \Delta RTT < 0, \\ cwnd - \frac{\tau}{RTT}\, b, & \text{if } \Delta RTT \geq 0 \end{cases}$
Rate-based	$\frac{d\lambda}{dt} = \begin{cases} \delta, & \text{if } d\gamma/d\lambda > 0, \\ -b\lambda, & \text{if } d\gamma/d\lambda < 0 \end{cases}$	$\frac{d\lambda}{dt} = \begin{cases} \varepsilon(\hat{L}_2), & \text{if } d\gamma/d\lambda > 0, \\ -b\lambda, & \text{if } d\gamma/d\lambda < 0 \end{cases}$

Table 1. Coupling of SSC with different flavours of TCP. Bottom row shows coupling with rate-based control for ATM.

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. The parameter A *influences* the output rate of the overall system but it does not *determine* it—the small time scale feedback congestion control acting at the time scale of T_S remains the dominant factor.

At the start, A is set to the default aggressiveness of TCP (i.e., $A(0) = a$). For each non-overlapping time block of size T_L, the maximum aggressiveness A is dynamically adjusted such that the reliable throughput at each level L_1 is maximized based on the sign of throughput changes with respect to A conditioned on the h levels. A is always kept positive and larger than a, $A \geq a$. With A evolving in time, individual levels of aggressiveness are set in accordance with the inverse linear schedule taking on values in the range $[a, A]$.

3.6 Instantiations of Couplings with TCP

This section describes the various instantiations of couplings with SSC based on different flavours of TCP—Tahoe, Reno, Vegas, and a rate-based extension called TCP Rate. We also show a rate-based congestion control for ATM as a reference which points toward the broad applicability of our scheme. The couplings are summarized in Table 1.

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 [Brakmo and Peterson 1995] 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, Δ`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.

3.6.4 *Rate-based Linear Increase/Exponential Decrease Control.* The last row of Table 1 shows a rate-based linear increase/exponential decrease feedback congestion control in the context of ATM where λ denotes data rate, γ represents throughput, and $\delta, b > 0$ are positive constants. If increasing the data rate results in increased throughput (i.e., $d\gamma/d\lambda > 0$) then a linear increase in the data rate is affected. Conversely, if increasing the data rate results in a decrease in throughput (i.e., $d\gamma/d\lambda < 0$) then the data rate is exponentially decreased. In general, condition $d\gamma/d\lambda < 0$ can be replaced by various measures of *congestion*. In the coupling, we replace the constant δ by $\varepsilon(\hat{L}_2)$. The qualitative performance results when running on top of UDP are analogous to that of TCP, and are omitted in the paper.

4. SIMULATION ISSUES FOR SELF-SIMILAR TRAFFIC CONTROL

4.1 Protocol Stack Influence

Setting up a framework where the impact of changes in transport protocols under self-similar traffic conditions can be effectively evaluated is a nontrivial problem. In traditional queueing oriented performance evaluation for self-similar traffic [Erramilli et al. 1996; Grossglauser and Bolot 1996; Heyman and Lakshman 1996], a queue is fed with self-similar input—either from analytic source models or traffic traces—and the resulting queueing behavior is observed and analyzed. Simulation based evaluation closely follows the analytical framework comprised of an open-loop queueing system where the input is independent of network (i.e., queue) state. It is for this reason that simulation is frequently used to validate analysis which, for self-similar traffic, has thus far been successful only in the asymptotic case where buffer capacity is taken to infinity. It is difficult to generalize this set-up to performance evaluation of congestion control since self-similar network traffic—either in trace form or as analytic source models—is produced by the very protocols being studied (the "horse before the cart" problem), and furthermore, congestion controls typically are feedback controls whose behavior is a function of network state leading to a closed-loop system.

4.2 Physical Models

Physical models [Feldmann et al. 1998; Gilbert et al. 1999; Park et al. 1996] address this problem by pushing the causality of self-similarity and burstiness to the application layer which is supported by empirical evidence of file systems and WWW servers possessing heavy-tailed object size distributions [Crovella and Bestavros 1996; Park et al. 1996]. A comprehensive discussion of traffic modeling issues, including physical models, can be found in [Riedi and Willinger 2000]. The on-off model of Willinger *et al.* [Willinger et al. 1995], Likhanov *et al.*'s source model [Likhanov et al. 1995], and the $M/G/\infty$ based input model [Cox 1984], provide the theoretical underpinning for why heavy-tailed traffic sources—multiplexed or singular—lead to self-similarity and long-range dependence assuming source behavior is *independent* of other sources and network state. Park *et al.*'s application layer model [Park et al. 1996] addresses *dependency issues* arising from feedback congestion control in closed-loop network systems. They show that aggregate traffic self-similarity is an intrinsic property of networked client/server systems mediated by TCP/UDP/IP protocol stacks where the size of the objects being accessed is heavy-tailed. In particular, there exists a linear relationship between the heavy-tailedness measure of file size distributions as captured by α—the shape parameter of the Pareto distribution—and the Hurst parameter of the resulting multiplexed traffic. This is shown in Figure 4 (left). This relationship holds under the fact

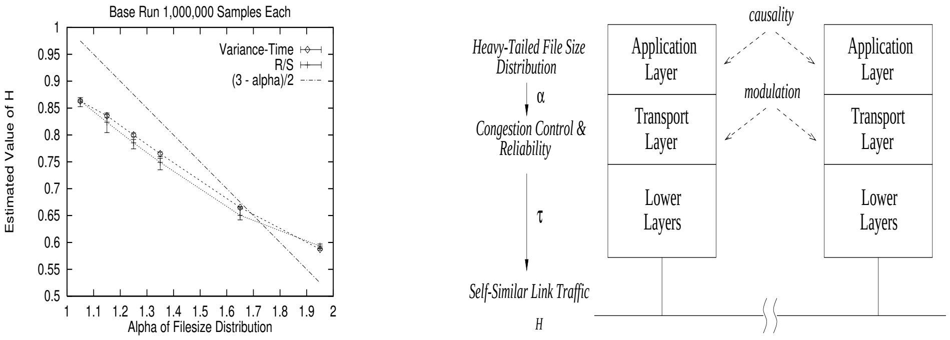

Fig. 4. Left: Hurst Parameter estimates for tail index α varying from 1.05 to 1.95 when file transport is mediated by TCP. Right: Application layer causality.

that dependencies arising from inter-connection coupling at bottleneck routers—which affect the behavior of transport layer feedback congestion controls which, in turn, affect measured traffic and performance—are incorporated. The induced self-similar network traffic, in terms of its traffic characteristics, is insensitive to details in the transport layer protocols—TCP Tahoe, Reno, Vegas, flow-controlled UDP—although extremeties in control actions and resource configurations can affect the property of induced network traffic, in some instances, diminishing self-similar burstiness significantly[4] [Park et al. 1996]. Thus by controlling the tail index parameter α at the application layer, it is possible to induce self-similarity at the link layer while incorporating the influence of transport protocols in the protocol stack. Furthermore, by fixing the application layer access pattern in conjunction with α, we are able to facilitate a comparative performance evaluation environment where two different transport protocols—e.g., one stemming from modifications to the other—can be evaluated under the same *network conditions* with respect to the propensity of generating self-similar burstiness in network traffic.

4.3 Sampling from Heavy-tailed Distributions

A core component of our comparative performance evaluation framework is sampling from heavy-tailed distributions to generate file sizes at the application layer which, then, drive the rest of the system. A random variable obeying a heavy-tailed distribution exhibits extreme variability. Practically speaking, a heavy-tailed distribution gives rise to very large values with nonnegligible probability so that sampling from such a distribution results in the bulk of values being "small" but a few samples having "very" large values. Not surprisingly, heavy-tailedness impacts sampling by slowing down the convergence rate of the sample mean to the population mean, dilating it as the tail index α approaches 1. For example, pending on the sample size m, the sample mean $\bar{Z}_m$ of a Pareto distributed random variable Z may significantly deviate from the population mean $\alpha b/(\alpha - 1)$, oftentimes underestimating it. In fact, the absolute estimation error $|\bar{Z}_m - E(Z)|$ asymptotically behaves as $m^{(1/\alpha)-1}$ (see, e.g., [Crovella and Lipsky 1997]), and thus for $\alpha \approx 1$, care must be given when sampling from heavy-tailed distributions such that conclusions about network behavior and performance attributable to sampling error are not advanced.

Sampling variations and errors have a ripple effect in that they influence the average traffic intensity at the network layer which, in turn, affects performance measures such as packet loss rate and mean delay. Of practical relevance is the case where a number of connections is used as "background" traffic for other connections whose throughput behavior we observe as we make changes to their control protocol. To ascertain the impact of long-range dependence on performance, we would like to vary the tail index α while generating the same average traffic intensity at the link layer so that observed performance differences are due to

[4]Refined structure in the form of multiplicative scaling in *short-range* correlation structure—first considered in [Levy-Vehel and Riedi 1997] for network traffic—has been recently observed in empirical IP traffic measurements [Feldmann et al. 1998]; it is conjectured to be attributable to TCP's feedback congestion control mechanisms. Traffic modeling using cascades has been carried out in [Riedi et al. 1999], and the performance impact of multiplicative scaling has been investigated in [Ribeiro et al. 2000].

burstiness characteristics, and not sampling variations. For example, in the case of the Pareto distribution with population mean $\alpha b/(\alpha - 1)$, to compare performance of the *same* protocol under $\alpha_1 = 1.05$ and $\alpha_2 = 1.95$ traffic conditions, we would solve $\alpha_1 b_1/(\alpha_1 - 1) = \alpha_2 b_2/(\alpha_2 - 1)$ for a pair of values (b_1, b_2) to keep the population mean invariant while allowing the burstiness structure to differ. For light-tailed distributions—e.g., exponential, Gaussian—this method works fine. For heavy-tailed distributions, however, even with "large" sample sizes [Crovella and Lipsky 1997; Park et al. 1996], the sample means of the respective distributions can significantly differ which has direct bearing on the traffic intensities, rendering the performance results inconclusive. Our approach is a form of sample path normalization where by varying (b_1, b_2)—while keeping (α_1, α_2) *fixed*—we reach a regime where the measured traffic intensities, on average, are constant for $\alpha = \alpha_1$ and α_2. Since b_1, b_2 do not significantly impact the burstiness property of underlying traffic as captured by the Hurst parameter—recall that $H = (3 - \alpha)/2$ in the analytic models—we are able to achieve comparability by normalizing traffic intensities while holding invariant the traffics' long-range dependence properties.

5. PERFORMANCE RESULTS

5.1 Network Configuration and Simulation Set-up

We use the LBNL Network Simulator—*ns* (version 2)—as the basis of our simulation environment. *ns* is an event-driven simulator derived from Keshav's REAL network simulator supporting several flavors of TCP and router packet scheduling algorithms. We have modified *ns* in order to model a bottleneck network environment where several concurrent connections are multiplexed over a shared bottleneck link. A rate-based extension of TCP—TCP Rate—was added to the existing protocol suite as were a number of UDP-based unreliable transport protocols. TCP-MT was realized by coupling SSC with the various versions of TCP under *ns*. Figure 5 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 (the nomenclature for "client" and "server" is reversed here), 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.

For any assignment of bandwidth, buffer size, mean file request size, and other system parameters, by either adjusting the number of clients or the mean idle time between successive file transfers, we were able to produce a target contention level. In a typical configuration, the first 32 connections serve as background traffic transferring files from clients to servers (or sinks) where the file sizes were drawn from Pareto distributions with shape parameter $\alpha = 1.05, 1.35, 1.65$, and 1.95. As

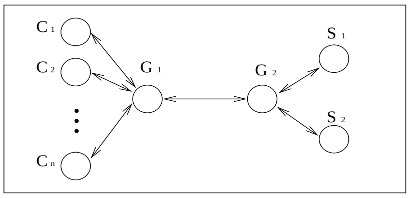

Fig. 5. Network configuration with bottleneck link (G_1, G_2). Traffic flows from left-to-right.

shown in [Park et al. 1996], there is a linear relationship between α and the Hurst parameter H of aggregate traffic measured at the bottleneck link (G_1, G_2). H was close to 1 when α was near 1, and H was close to 1/2 when α was near 2. A typical run lasted for 10000 seconds—simulated time—with traces collected at 10ms granularity. This yields 1 million data points for a single run which helps offset some of the variability associated with heavy-tailed sampling in addition to the sample path normalization method described in Section 4.3. The basic performance evaluation set-up, with variations, has been employed in previous studies [Park et al. 1996; Park et al. 1997; Park 1997b] where the focus has been on causality and performance impact issues of self-similar network traffic.

5.2 Basic Performance Characteristics of Selective Slope Control

5.2.1 *Unimodal Throughput Curve.* We measure the *incremental* benefit gained by applying aggressiveness in the form of slope control *selectively*, first, by applying it only when the chances for benefit are highest (i.e., $\hat{L}_2 = 1$), then second highest ($\hat{L}_2 = 2$), and so on. Eventually, we expect to reach a point when the cost of aggressiveness outweighs its gain, thus leading to a net decrease in throughput as the stringency of selectivity is further relaxed. We use the threshold schedule—aggressive action is taken if, and only if, $\hat{L}_2 \leq \theta$ where θ is the aggressiveness threshold—to demonstrate this phenomenon. Figure 6 (left) shows the reliable throughput vs. aggressiveness threshold curve for threshold values in the range $1 \leq \theta \leq 8$ for $\alpha = 1.05$ traffic. We observe that the curve is unimodal with peak at $\theta = 4$. If $\theta = 8$, this corresponds to the case where aggressiveness is applied at all times, i.e., there is no selectivity.

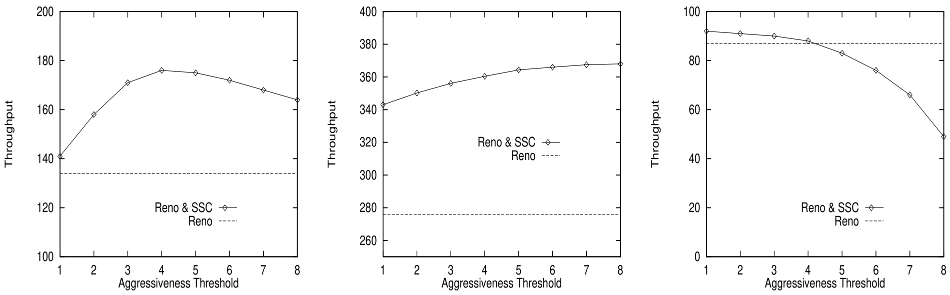

Fig. 6. TCP Reno. Shape of reliable throughput curve as a function of aggressiveness threshold for three levels of background traffic 5Mbps (left), 2.5Mbps (middle), and 7.5Mbps (right).

5.2.2 *Monotone Throughput Curve.* Although the unimodal throughput curve is a representative shape, two other shapes—monotonically increasing or decreasing—are possible depending on the network configuration. The shape of the curve is dependent upon the relative magnitude of available resources vs. the magnitude of aggressiveness. If resources are "plentiful" then aggressiveness is least penalized, and it can lead to a monotonically increasing throughput curve. On the other hand, if resources are "scarce" then aggressiveness is penalized most heavily and this can result in a monotonically decreasing throughput curve. These effects are shown in Figure 6 (middle) and (right), respectively. TCP-MT is designed to operate under all three network conditions finding a near-optimum throughput in each case. The most challenging task arises when the network configuration leads to a unimodal throughput curve for which finding the maximum throughput is least trivial. That is, neither blindly applying aggressiveness nor abstaining from it are optimal strategies. SSC's adaptability is also useful in nonstationary situations where network state can shift from one quasi-static regime to another.

Figure 7 shows the throughput vs. aggressiveness threshold curves for the previous set-up except that TCP Reno was replaced by TCP Rate. We observe that both performance as well as curvature of the throughput curves have increased which is, in part, due to TCP Rate's superior tracking ability (cf. Section 5.2.3) which allows SSC to extract large-time scale correlation structure more effectively. Figure 7 also shows the individual effect of employing SSC in Congestion Avoidance, Slow Start, and both.

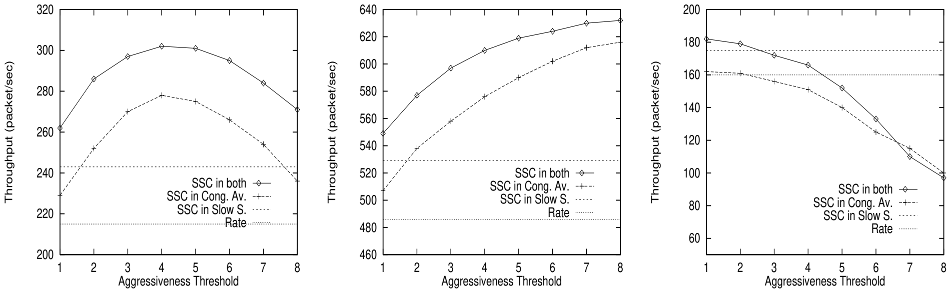

Fig. 7. TCP Rate. Shape of reliable throughput curve as a function of aggressiveness threshold for three levels of background traffic: 5Mbps (left), 2.5Mbps (middle), and 7.5Mbps (right).

5.2.3 *Tracking Ability.* The tracking ability of the underlying feedback congestion control can exert a nonnegligible influence on performance and thus 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, hence 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. We measure the *tracking ability* of TCP Reno, Vegas, and Rate by computing the 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 8 (left). 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. Figure 8 (right) shows the correlation coefficients for the same set-up with the difference that SSC was coupled with TCP Reno, Vegas, and Rate. We observe that all curves have shifted downward toward -1 indicating a synergy effect stemming from coupling which enhances the tracking ability of TCP-MT vis-à-vis TCP due to improved timeliness of its actions.

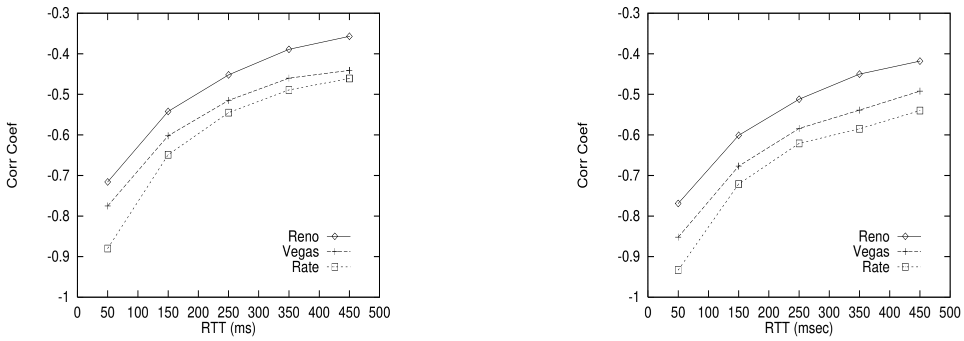

Fig. 8. Left: Tracking ability in terms of correlation coefficient for TCP Reno, Vegas, and Rate. Right: Synergy effect of TCP-MT which increases tracking ability when SSC is coupled with TCP Reno, Vegas, and Rate.

5.3 RTT and Proactivity

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 9 shows performance gain as a function of RTT where *performance gain* ν is defined as

$$\nu = \frac{\Lambda_{\text{TCP-MT}} - \Lambda_{\text{TCP}}}{\Lambda_{\text{TCP}}}$$

where 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.

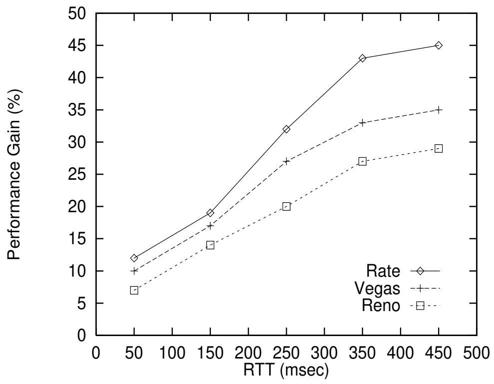

Fig. 9. Performance gain as a function of RTT when coupling SSC on top of TCP Reno, Vegas, and Rate. The increasing gain with RTT show the proactivity property of TCP-MT.

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. We can also relate the performance gain in Figure 9 with the tracking ability shown in Figure 8, both of which are obtained from the same set-up. We observe that the tracking ability of the underlying feedback congestion control influences performance. 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 ever "better" feedback congestion controls.

5.4 Impact of Long-range Dependence

Another dimension of interest is the impact of long-range dependence 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 10 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 10 shows that the more long-range dependent the network traffic, the more structure there is to exploit.

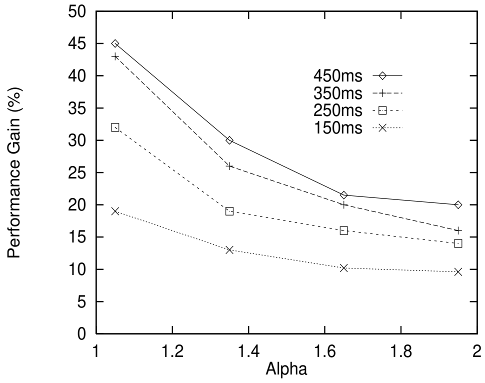

Fig. 10. Impact of long-range dependence as capture by $\alpha = 1.05, 1.35, 1.65, 1.95$ on TCP-MT's performance gain.

5.5 Short Duration Connection Management

Network measurements have shown that most connections are short-lived but the bulk of traffic is contributed by the few long-lived ones [Feldmann et al. 1998; Park et al. 1996]. Thus, by Amdahl's law, effectively managing long-lived connections is of disproportionate importance. In fact, since about 80% of current Internet traffic is governed by TCP, a trend which is expected to persist due to the growth and dominance of HTTP-based World Wide Web traffic [Arlitt and Williamson 1996; Barford and Crovella 1998; Crovella and Bestavros 1996], managing long-lived TCP flows takes on special relevance. Nonetheless, since most connections are short-lived (on the order of a few TCP segments), improving service to short-lived flows—to the extent possible—is a desirable objective. Two constraints that are intrinsically difficult to overcome are: (a) it is infeasible to consider performing per-connection, on-line estimation with any degree of accuracy when connection duration is short; (b) when a transmission consists of a few segments, even feedback control is of limited utility [Kim 1995]. We consider three cases with successively decreasing connection duration times, and the effectiveness of open-loop and closed-loop control. In Case I, an accurate, *a priori* conditional probability table is assumed given, and a connection accesses this table to engage SSC, by-passing its explicit prediction module which is disabled. In Case II, on-line prediction is engaged for 300 seconds before turning on the aggressiveness schedule of SSC. In Case III, after affecting on-line prediction for 30 seconds, SSC is activated full-fledged. Table 2 gives performance results showing the performance gain for the three cases when a connection is run for 100, 500, 1000, and 2000 seconds after estimation. We observe

Short Conn.	100sec	500sec	1000sec	2000sec
Case I	25.4%	23.2%	31.6%	29.7%
Case II	4.5%	13.75%	20.23%	25.39%
Case III	6.3%	9.2%	19.2%	27.2%

Table 2. Performance gain for short-lived SSC connections. Case I: with an a priori conditional probability table. Case II: SSC is on after on-line training for 300sec. Case III: SSC is on after on-line training for 30sec.

that the performance gain is highest for Case I which assumes access to an a priori information base. Case III possesses the least accurate table and thus yields the smallest performance gain among the three cases. Case II lies in-between. As connection duration increases, the performance impact of SSC for Case III eventually catches up with that of Case II and I. These results indicate that although SSC is optimally suited for long-lived connections, it can yield performance gains even for short-lived connections depending on the exact duration and availability of a priori information. The approach of using a priori information—e.g., by inter-connection sharing and statefulness—also holds promise from an estimation perspective due to the fact that under long-range dependent traffic conditions, the conditional expectation estimator $\hat{L}_2 = E[L_2 \,|\, L_1]$ can be shown to degenerate to $E[L_1]$ under certain simplifying assumptions [Beran 1994]. That is, extrapolate the current traffic level as the traffic level for the next T_L interval.

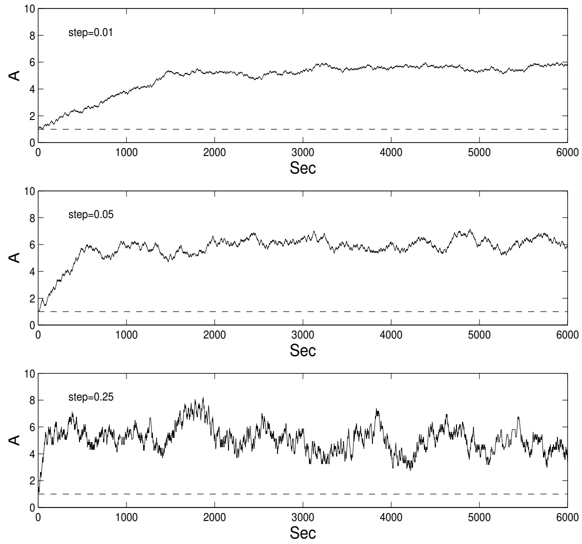

Fig. 11. Dynamics of symmetric meta control as a function of adjustment factor ν and the resultant evolution of A. Top: $\nu = 0.01$. Middle: $\nu = 0.05$. Bottom: $\nu = 0.25$.

5.6 Symmetric Meta Control

Section 3.5 showed the role of meta control for dynamically adjusting the maximum slope level A within SSC. Assuming sufficient time scale separation ($T_L \gg T_S$)

between the long and short time scale modules, stability of the symmetric meta control depends on the adjustment factor ν where ν sufficiently small leads to asymptotic stability, and bigger ν values can lead to oscillatory behavior. Figure 11 shows the dynamics of the symmetric meta control for different adjustment factors ν where the value is successively increased by a factor of five. As expected, we observe that the larger ν, the more pronounced the resulting oscillation. What is most interesting is that the traces show that in all three cases, the symmetric meta control "settles" to a common equilibrium $A^* \approx 6$ with the magnitude of oscillation around A^* determined by ν.

Figure 12 shows throughput performance for static vs. dynamic setting of maximum aggressiveness. The unimodal curve shows reliable throughput for the static case where A is set to a fixed a priori value in the range 1–10. The throughput corresponding to the dynamic meta control is shown by the upper dashed line. It closely approximates the performance of the optimal static maximum aggressiveness value $A^* = 6$. In general, it is difficult to know a priori what A should be for a given network configuration, and dynamic meta control is needed to address this problem. The lower dashed line shows the throughput of TCP Rate as a reference.

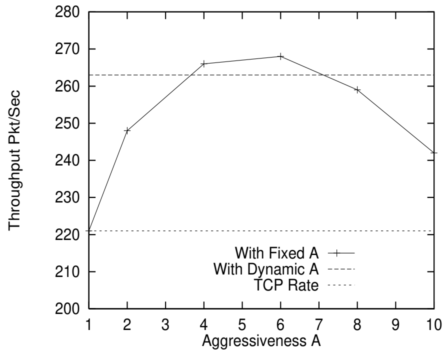

Fig. 12. Throughput performance with different maximum slope levels A.

5.7 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. The results are based on the set-up shown in Figure 5 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 13 (left) 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 13 (right) shows the corresponding performance figures when TCP-MT is replaced by TCP Reno. We observe a qualitatively similar behavior as before. Table 3 gives more detailed information with respect to total through-

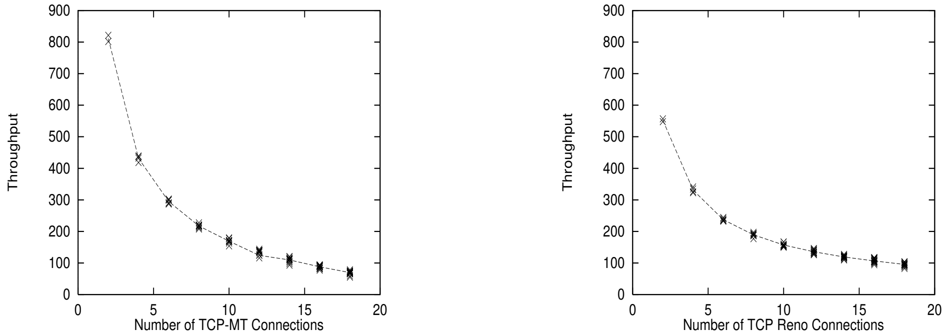

Fig. 13. Bandwidth sharing behavior. Left: Dashed line denotes mean throughput of multiple TCP-MT connections; dark marks show the spread of individual throughput values. Right: Corresponding plot for TCP Reno connections.

put and range of throughput values for individual connections. The first row of Table 3 shows that the *total* throughput of TCP-MT increases with the number of connections up until $n = 6$ after which it begins to decline. That is, as the number of TCP-MT connections is further increased, the amplification of the overall aggressiveness—due to its additive nature—asserts a negative impact on throughput, eventually yielding a net decrease. A similar result holds for TCP Reno due to the amplification in overall aggressiveness as the number of concurrent feedback congestion control connections is increased.

SSC	n=2	n=4	n=6	n=8	n=10	n=12	n=14	n=16	n=18
total	1623.2	1725.0	1764.0	1738.0	1692.0	1609.9	1537.9	1405.9	1251.0
avrg.	811.6	431.2	294.0	217.2	169.2	134.2	109.8	87.9	69.5
max.	821.6	439.1	302.2	227.3	179.4	143.7	120.3	94.2	78.4
min.	801.6	418.6	286.7	207.4	154.3	116.2	93.2	77.2	54.9

Table 3. Multiple TCP-MT connections. The first row shows total throughput achieved across all connections; the remaining three rows show the mean, maximum, and minimum of individual throughput.

We remark that the generalization of TCP-MT to *conservativeness control* where the slope of increase is allowed to be smaller than the default value in TCP (i.e., $\xi < a$, see Section 3.4) may facilitate further improvement in performance when the number of TCP connections is large by counter-acting the impact of simultaneous control actions. A study of integrated aggressiveness and conservativeness control is an item for future work.

6. CONCLUSION

In this paper, we have shown that the multiple time scale congestion control framework [Tuan and Park 1999] 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, injecting a measure of proactivity. The relative performance gain of TCP-MT vis-à-vis its underlying feedback congestion control was shown to increase as the RTT of the feedback loop is increased. Current work is directed at implementing TCP-MT over TCP Reno in the Linux and Solaris kernels, and carrying out performance measurements over wide area network environments. We are also extending the short duration connection management work by employing a priori state information to improve performance—e.g., average completion time—when transmissions comprise of only a few segments.

REFERENCES

Adas, A. and Mukherjee, A. 1995. On resource management and QoS guarantees for long range dependent traffic. In *Proc. IEEE INFOCOM '95* (1995), pp. 779–787.

Addie, R., Zukerman, M., and Neame, T. 1995. Fractal traffic: measurements, modelling and performance evaluation. In *Proc. IEEE INFOCOM '95* (1995), pp. 977–984.

Arlitt, M. F. and Williamson, C. L. 1996. Web server workload characterization: The search for invariants. In *Proceedings of SIGMETRICS '96* (May 1996), pp. 126–137.

Barford, P. and Crovella, M. 1998. Generating representative workloads for network and server performance evaluation. In *Proc. ACM SIGMETRICS '98* (1998), pp. 151–160.

Beran, J. 1994. *Statistics for Long-Memory Processes.* Monographs on Statistics and Applied Probability. Chapman and Hall, New York, NY.

Brakmo, L. and Peterson, L. 1995. TCP Vegas: end to end congestion avoidance on a global internet. *IEEE J. Select. Areas Commun. 13*, 8, 1465–1480.

Cox, D. R. 1984. Long-range dependence: a review. In H. A. David and H. T. David Eds., *Statistics: An Appraisal*, pp. 55–74. Iowa State Univ. Press.

Crovella, M. and Bestavros, A. 1996. Self-similarity in world wide web traffic: Evidence and possible causes. In *Proceedings of the 1996 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems* (May 1996).

Crovella, M. and Lipsky, L. 1997. Long-lasting transient conditions in simulations with heavy-tailed workloads. In *Proc. 1997 Winter Simulation Conference* (1997).

Duffield, N. and Whitt, W. 2000. Network design and control using on-off and multi-level source traffic models with heavy-tailed distributions. In K. Park and W. Willinger Eds., *Self-Similar Network Traffic and Performance Evaluation.* Wiley-Interscience.

Duffield, N. G. and O'Connel, N. 1993. Large deviations and overflow probabilities for the general single server queue, with applications. Technical Report DIAS-STP-93-30, DIAS Technical Report.

Erramilli, A., Narayan, O., and Willinger, W. 1996. Experimental queueing analysis with long-range dependent packet traffic. *IEEE/ACM Trans. Networking 4*, 209–223.

Feldmann, A., Gilbert, A. C., and Willinger, W. 1998. Data networks as cascades: Investigating the multifractal nature of Internet WAN traffic. In *Proc. ACM SIGCOMM '98* (1998), pp. 42–55.

Garret, M. and Willinger, W. 1994. Analysis, modeling and generation of self-similar VBR video traffic. In *Proc. ACM SIGCOMM '94* (1994), pp. 269–280.

Gilbert, A. C., Willinger, W., and Feldmann, A. 1999. Scaling analysis of conservative cascades, with applications to network traffic. *IEEE Trans. Information Theory 45*, 3, 971–991.

Grossglauser, M. and Bolot, J.-C. 1996. On the relevance of long-range dependence in network traffic. In *Proc. ACM SIGCOMM '96* (1996), pp. 15–24.

Heyman, D. and Lakshman, T. 1996. What are the implications of long-range dependence for VBR-video traffic engineering? *IEEE/ACM Transactions on Networking 4*, 3 (June), 301–317.

Huang, C., Devetsikiotis, M., Lambadaris, I., and Kaye, A. 1995. Modeling and simulation of self-similar variable bit rate compressed video: a unified approach. In *Proc. ACM SIGCOMM '95* (1995), pp. 114–125.

Kim, H. 1995. *A Non-Feedback Congestion Control Framework for High-Speed Data Networks*. Ph. D. thesis, University of Pennsylvania.

Kim, H. and Farber, D. 1995. The failure of conservative congestion control in large bandwidth-delay product networks. In *Proc. INET '95* (1995).

Lakshman, T. V. and Madhow, U. 1997. The performance of tcp/ip for networks with high bandwidth-delay products and random loss. *IEEE/ACM Trans. Networking 5*, 3, 336–350.

Leland, W., Taqqu, M., Willinger, W., and Wilson, D. 1994. On the self-similar nature of Ethernet traffic (extended version). *IEEE/ACM Transactions on Networking 2*, 1–15.

Levy-Vehel, J. and Riedi, R. 1997. Fractional Brownian motion and data traffic modeling: The other end of the spectrum. In *Fractals in Engineering* (1997), pp. 185–202. Springer-Verlag.

Likhanov, N., Tsybakov, B., and Georganas, N. 1995. Analysis of an ATM buffer with self-similar ("fractal") input traffic. In *Proc. IEEE INFOCOM '95* (1995), pp. 985–992.

Norros, I. 1994. A storage model with self-similar input. *Queueing Systems 16*, 387–396.

Park, K. 1993. Warp control: a dynamically stable congestion protocol and its analysis. In *Proc. ACM SIGCOMM '93* (1993), pp. 137–147.

Park, K. 1997a. AFEC: an adaptive forward error-correction protocol and its analysis. Technical Report CSD-TR-97-038, Department of Computer Sciences, Purdue University.

Park, K. 1997b. On the effect and control of self-similar network traffic: a simulation perspective. In *Proc. 1997 Winter Simulation Conference* (December 1997).

Park, K., Kim, G., and Crovella, M. 1996. On the relationship between file sizes, transport protocols, and self-similar network traffic. In *Proc. IEEE International Conference on Network Protocols* (1996), pp. 171–180.

Park, K., Kim, G., and Crovella, M. 1997. On the effect of traffic self-similarity on network performance. In *Proc. SPIE International Conference on Performance and Control of Network Systems* (1997), pp. 296–310.

Park, K. and Wang, W. 1999. QoS-sensitive transport of real-time MPEG video using adaptive forward error correction. In *Proc. IEEE Multimedia Systems '99* (1999), pp. 426–432.

Park, K. and Willinger, W. 2000b. Self-similar network traffic: An overview. In K. Park and W. Willinger Eds., *Self-Similar Network Traffic and Performance Evaluation*. Wiley-Interscience.

Park, K. and Willinger, W. Eds. 2000a. *Self-Similar Network Traffic and Performance Evaluation*. Wiley-Interscience.

Paxson, V. and Floyd, S. 1994. Wide-area traffic: the failure of Poisson modeling. In *Proc. ACM SIGCOMM '94* (1994), pp. 257–268.

Pecelli, G. and Kim, B. G. 1995. Dynamic behavior of feedback congestion control schemes. In *Proc. IEEE INFOCOM '95* (1995).

Ribeiro, V., Riedi, R., Crouse, M., and Baraniuk, R. 2000. Simulation and queuing analysis of non-Gaussian long-gange-dependent traffic using wavelets. In *Proc. IEEE INFOCOM '00* (2000).

Riedi, R., Crouse, M., Ribeiro, V., and Baraniuk, R. 1999. A multifractal wavelet model with application to network traffic. *IEEE Transactions on Information Theory 45*, 3.

Riedi, R. and Willinger, W. 2000. Toward an improved understanding of network traffic dynamics. In K. Park and W. Willinger Eds., *Self-Similar Network Traffic and Performance Evaluation*. Wiley-Interscience.

Ryu, B. and Elwalid, A. 1996. The importance of long-range dependence of VBR video traffic in ATM traffic engineering: myths and realities. In *Proc. ACM SIGCOMM '96* (1996), pp. 3–14.

Tuan, T. and Park, K. 1999. Multiple time scale congestion control for self-similar network traffic. *Performance Evaluation 36*, 359–386.

Tuan, T. and Park, K. 2000. Multiple time scale redundancy control for QoS-sensitive transport of real-time traffic. In *Proc. IEEE INFOCOM '00* (2000).

Willinger, W., Taqqu, M., Sherman, R., and Wilson, D. 1995. Self-similarity through high-variability: statistical analysis of Ethernet LAN traffic at the source level. In *Proc. ACM SIGCOMM '95* (1995), pp. 100–113.