A Case for Exploiting Self-Similarity of Network Traffic in TCP Congestion Control

Guanghui He†, Yuan Gao†, Jennifer C. Hou†, Kihong Park‡

†Dept. of Computer Science
Univ. of Illinois at Urbana Champaign
Urbana, IL 61801

‡Dept. of Computer Science
Purdue University
West Lafayette, IN 47907

Abstract—

Analytical and empirical studies have shown that self-similar traffic can have detrimental impact on network performance including amplified queuing delay and packet loss ratio. On the flip side, the ubiquity of scale-invariant burstiness observed across diverse networking contexts can be exploited to better design resource control algorithms. In this paper, we explore the issue of exploiting the self-similar characteristics of network traffic in TCP congestion control. We show that the correlation structure present in long-range dependent traffic can be detected on-line and used to predict the future traffic. We then devise an novel scheme, called TCP with traffic prediction (TCP-TP), that exploits the prediction result to infer, in the context of AIMD steady-state dynamics, the optimal operational point at which a TCP connection should operate. Through analytical reasoning, we show that the impact of prediction errors on fairness is minimal. We also conduct ns-2 simulation and FreeBSD 4.1-based implementation studies to validate the design and to demonstrate the performance improvement in terms of packet loss ratio and throughput attained by connections.

Keywords: — **Long range dependence, traffic prediction, congestion control, TCP**

I. Introduction

A number of recent empirical studies of traffic measurements from a variety of working packet networks have convincingly demonstrated that network traffic is self-similar or long-range dependent (LRD) in nature [11], [5], [20]. This implies the existence of concentrated periods of high activity and low activity (i.e., burstiness) at a wide range of time scales. Scale-invariant burstiness introduces new complexities into resource control and QoS provisioning. On the one hand, burstiness at coarser time scales induces extended periods of either over-utilization or under-utilization. Packets that arrive in the periods of over-utilization experience long delays and are even dropped due to buffer overflow, while packets that arrive in the periods of under-utilization experience the opposite. The large variation in the end-to-end delay packets experience has an adverse impact on transport of QoS-sensitive traffic such as multimedia traffic. On the other hand, if resource reservation is deployed for QoS provisioning, resources have to be reserved with respect to the *peak* rates over a *wide* range of time scales. This adversely affects resource control and degrade the overall performance.

While the LRD characteristic of network traffic introduces difficulty and complexity into traffic and resource management, it also opens up a new direction for research — the existence of nontrivial correlation structure at larger time scales can be judiciously exploited for better congestion and resource control. How to exploit the abundant correlation structure to improve the TCP performance is the subject matter of this paper.

In this paper, we propose a novel scheme, *TCP with traffic prediction* (or *TCP-TP*). We show that the correlation structure present in LRD traffic can be detected on-line and used to predict the future traffic at least one round-trip time (RTT) ahead. This is realized with the use of a traffic predictor that minimizes the linear mean square errors (LMMSE). Through *ns-2* simulation, we show that the predicted traffic agrees extremely well with the actual traffic. The prediction results are then used to infer the optimal operational point at which a TCP connection should operate. By optimal operational point, we mean the point that achieves fairness among the connections that traverse the same bottleneck link. Specifically, consider Fig. 1. The vertical and horizontal axes represent the throughput, f, attained by the TCP connection of interest and that, B, by the background traffic, respectively. Let the number of connections that traverse the bottleneck link be denoted as N and the bandwidth of the bottleneck link as C. Then the optimal operational point is the joint point of line $f + B = C$ (which we call the capacity line) and line $\frac{f}{B} = \frac{1}{N-1}$ (which we call the fairness line). Without traffic prediction, a TCP connection usually reaches the optimal point through several additive increase (AI) and multiplicative decrease (MD) phases (Fig. 1). With accurate traffic prediction, we show that if all TCP connections are synchronized in making their congestion control decisions and prediction results are accurate, a TCP connection can reach the optimal point in one RTT without commencing MD phases (and hence without incurring packet losses), if all TCP connections are synchronized in making their congestion control decisions. This leads to significant performance improvement in terms of fast convergence to the optimal operational point, packet loss ratio, and attainable throughput. The above results are corroborated by *ns-2* simulation and empirical experimentation over the Internet.

In the case of existence of prediction errors, we show via phase plots that with the use of the MD phase, TCP-TP can still retain the stability established in the AIMD al-

The work reported in this paper was supported in part by NSF under Grant No. ANI-0082861.

Proceedings of the 17th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT'02)
1063-6722/02 $17.00 © 2002 IEEE

gorithm. Moreover, we analyze rigorously the impact of prediction errors on fairness, and show when the prediction error is 100% (i.e., the predicted value is twice as large as the original value), the index of fairness only degrades 2.5%. Finally we demonstrate the change needed to incorporate the traffic prediction extension into TCP is minimal (tens of lines of code changes) by implementing TCP-TP both in *ns-2* and in FreeBSD 4.1 and conducting experiments. In the case that TCP connections do not synchronize in window adjustment and are subject to different RTTs, we show via *ns-2* simulation and FreeBSD implementation and experimentation that both TCP and TCP-TP cannot achieve fairness, but a TCP-TP connection still performs better than TCP (67% improvement in terms of packet loss ratio).

TCP-TP is especially well-suited for improving the performance of long-lived TCP connections, as short-lived TCP connections may not reach the optimal operational points before they terminate. Although the number of short-lived TCP connections are much larger than that of long-lived TCP connections, as reported in [4], the majority of Internet traffic is still dominated by long-lived TCP connections and long-lived TCP connections play much more important roles in network utilization. As TCP-TP improves the attainable throughput of long-lived TCP connections, it helps to increase the overall network utilization. On the other hand, it has been suggested in the work of *Congestion Manager* (CM) [7] that (short-lived) TCP connections destined for the same destination (e.g., web downloads from a web server) should be bundled together and subject to the same congestion control (so as to avoid blind competition among concurrent connections). TCP-TP can be used in conjunction with CM.

The rest of the paper is organized as follows: In Section II, we give the definition of long range dependency and an overview of TCP-TP. Then, we delve into the technical details of TCP-TP in Section III. In particular, we elaborate on how a TCP-TP sender predicts its future attainable throughput and adjusts its congestion window. In Section IV, we analyze how prediction errors impact the performance in terms of fairness. In Section V, we present results from *ns-2* simulation and FreeBSD 4.1 implementation based experiments. The paper concludes with Section VI.

II. Exploitation of LRD Characteristics in TCP Congestion Control

A. Definition of Self-Similar Traffic

Let $\{f(t), t \in Z+\}$ be a time series which represents the traffic traces measured at some fixed time granularity. We define the aggregated series $f^{(m)}(k)$ as

$$f^{(m)}(k) = \frac{1}{m} \sum_{i=(k-1)m+1}^{km} f(i), \qquad (1)$$

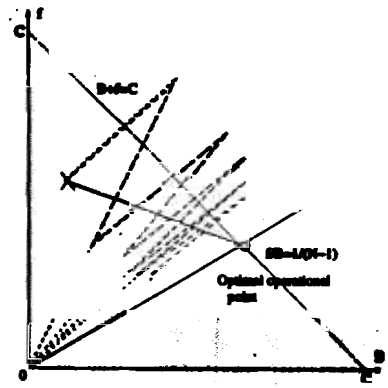


Fig. 1. The phase plot that illustrates how fairness is achieved in the case that N connections traverse a bottleneck link. A TCP connection usually goes through several AI and MD phases, and follows the dashed line to reach the optimal operational point.

where m is the number of time series samples in a measurement interval and k is used to index the measurement intervals.

Let $R(k)$ and $R^{(m)}(k)$ denote the autocorrelation functions of $f(t)$ and $f^{(m)}(i)$, respectively. $f(t)$ is (asymptotically second-order) self-similar, if the following conditions hold:

$$R(k) \sim \text{const} \cdot k^{-\beta}, \qquad (2)$$

$$R^{(m)}(k) \sim R(k), \qquad (3)$$

for large values of k and m where $0 < \beta < 1$. That is, $f(t)$ is self-similar in the sense that the correlation structure is preserved with respect to time aggregation (Eq. (3)) and $R(k)$ behaves hyperbolically with $\sum_{k=0}^{\infty} R(k) = \infty$ (Eq. (2)). The latter property is also referred to as long range dependency (LRD).

An important parameter that characterizes the LRD is the Hurst parameter, $H = 1 - \beta/2$. By the range of β, $1/2 < H < 1$. It can be seen from Eq. (2) that the larger H is, the more long-range dependent $f(t)$ is. A test for LRD can be obtained by checking if H significantly deviates from 1/2 or not. A comprehensive discussion of LRD can be found in [2].

Several fractional models and their corresponding predictors have been proposed for time series with the LRD characteristic, among which the *Fractional Brownian Motion* (*FBM*) [12] model and the *fractal ARIMA* model [8] have perhaps received the most attention. The interested reader is referred to the extended version of this paper [6] for a brief summary of the two fractional model-based predictors.

B. An Overview of TCP-TP

The rationale behind exploitation of traffic prediction in TCP congestion control is to enable TCP to predict, with exploitation of the correlation structure across multiple time scales, its attainable throughput at least one

Proceedings of the 17th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT'02)
1063-6722/02 $17.00 © 2002 **IEEE**

round trip time (RTT) ahead. With the predicted information, the TCP connection then infers the optimal operational point, and adjusts the window increase/decrease in its congestion control (in particular, the AI phase of the *AIMD* algorithm). The window adjustment scheme is devised with the following objectives:

(1) Packet loss: With consideration of future available bandwidth, a TCP connection needs not explore the available bandwidth blindly and rely on packet loss as an indication of congestion. The packet loss incurred should be reduced.

(2) Fairness: The bandwidth of the bottleneck link should be as fairly shared as possible by all the connections that traverse the link. Note that as indicated in [3] the fairness criterion is 100% met only when all TCP connections are subject to the same RTT.

(3) Stability: The stability of the AIMD algorithm (which has been established in [3] under the assumption of equal RTTs) should not be compromised.

Consider a TCP connection that traverses a path, P, from the source to the destination. We characterize the background traffic on the bottleneck link of P as a time series, $\{r(t), t \in Z+\}$, that exhibits LRD with the Hurst parameter H. We also assume that

(A1) $r(t)$, as a superposition of numerous component connections, does not adapt to the TCP connection of interest.

(A2) The capacity (bandwidth), C of the bottleneck link is known. This is not an unreasonable assumption, as several strategies, e.g., the one-packet techniques [9], [16], the packet-pair techniques [13], a combination thereof [10], and the packet chirp probe train technique [14], have been proposed to measure, without router support, the bandwidth (or cross traffic) of a bottleneck link.

With the above setting, it is clear that the bandwidth available to the TCP connection of interest is a time series, $\{f(t) = C - r(t), t \in Z+\}$. It can be analytically shown (after simple derivation) that if $r(t)$ exhibit LRD, so does $f(t)$ with the same Hurst parameter. The fact that $f(t)$ is also self-similar serves the theoretical base of TCP-TP.

TCP-TP follows TCP, and uses non-duplicate ACKs, duplicate ACKs (which serve as NACKs), and timeouts as means of inferring whether or not congestion occurs and the level of congestion. In addition, the sender of a TCP-TP connection keeps track of the amount of data acknowledged and samples the attainable throughput periodically or aperiodically (e.g., whenever a fixed amount of data is acknowledged since the last measurement) using a low-pass filter with an exponentially weighted moving average. That is, if t_0 and t denote the previous and current sampling instants, then

$$f(t) = (1 - \alpha) \cdot f(t_0) + \alpha \cdot \frac{\#\text{ bytes acked since } t_0}{t - t_0}.$$

The sender then keeps track of the time series, predicts the attainable throughput at least one RTT ahead using a *linear minimum mean square error* (*LMMSE*) predictor (to be discussed in Section III-A), and adjusts its congestion window, with the objective of reaching the optimal operational point without incurring MD phases (to be discussed in Section III-B).

III. Detailed Description of TCP-TP

A. Prediction of Attainable Throughput

Given the time series of the attainable throughput, $\{f(t), t \in Z+\}$, a TCP-TP sender keeps track of the aggregated series, $f^{(m)}(k-n+1), f^{(m)}(k-n+2), \ldots, f^{(m)}(k)$, measured in the past n measurement intervals (Eq. (1)). Based on these aggregated series samples, the sender predicts the attainable throughput, $f^{(m)}(k+1)$, in the next measurement interval. The reasons for using aggregated time series in the prediction are two fold: first, *LRD* implies that the autocorrelation function of the aggregated series obeys the same law as that of the original time series (Eq. (3)). Hence the aggregated series is still a good representative of the original series. Second, the averaging operation in Eq. (1) can be viewed as sampling and smoothing operations. As reported in [15], smoothed and sampled traffic exhibits more predictable behavior, and hence prediction based on the aggregated series is more accurate.

We propose to use a *LMMSE* predictor to predict the attainable throughput in the next measurement interval. Specifically, given the series $f^{(m)}(k), k = 1, \ldots, n$, $f^{(m)}(n+1)$ can be expressed as a weighted sum of the past n samples. That is, the estimate (written as $\hat{f}^{(m)}(n+1)$) of $f^{(m)}(n+1)$ is expressed as

$$\hat{f}^{(m)}(n+1) = \begin{bmatrix} a_1 & a_2 & \cdots & a_n \end{bmatrix} \begin{bmatrix} f^{(m)}(1) \\ f^{(m)}(2) \\ \cdots \\ f^{(m)}(n) \end{bmatrix}, \quad (4)$$

where $a_1, a_2, \ldots, a_n$ are the *LMMSE* coefficients and can be expressed as

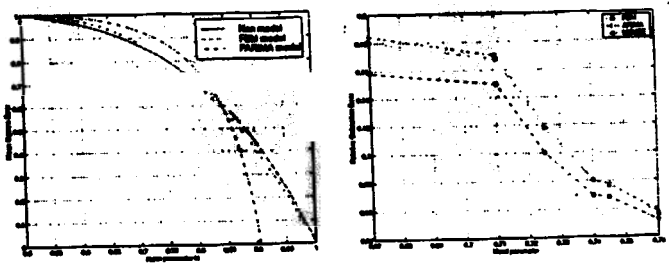
$$\begin{bmatrix} a_1 & a_2 & \cdots & a_n \end{bmatrix} = \begin{bmatrix} R(n) & R(n-1) & \cdots & R(1) \end{bmatrix} \begin{bmatrix} R(0) & R(1) & \cdots & R(n-1) \\ R(1) & R(0) & \cdots & R(n-2) \\ \cdots & \cdots & \cdots & \cdots \\ R(n-1) & R(n-2) & \cdots & R(0) \end{bmatrix}^{-1} \quad (5)$$

where $R(n)$ is the covariance function of the time series, and can be estimated in practice as

$$R(i) \cong R^{(m)}(i) = \frac{1}{n} \sum_{t=i+1}^{n} f^{(m)}(t) f^{(m)}(t-i), \quad (6)$$

for $0 \le i \le n-1$, where n is the number of aggregated series samples kept and is a tunable parameter. (In both the simulation and empirical studies, we use $n = 20$, as in all the experiments conducted, the performance improvement in prediction accuracy beyond $n = 20$ is marginal.) Note that the parameter H that characterizes LRD has been implicitly calculated in the covariance function $R(i)$.

Proceedings of the 17th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT'02)
1063-6722/02 $17.00 © 2002 **IEEE**



(a) prediction induced error (b) total prediction error

Fig. 2. Comparison of estimation errors among the *FBM*, *FARIMA*, and *LMMSE* predictors.

The mean square error of the *LMMSE* predictor can be calculated (after a few algebraic operations) as

$$\sigma^2 = \sigma_x^2 - \begin{bmatrix} R(n) & R(n-1) & \dots & R(1) \end{bmatrix} \times \begin{bmatrix} R(0) & \dots & R(n-1) \\ R(1) & \dots & R(n-2) \\ R(n-1) & & R(0) \end{bmatrix}^{-1} \begin{bmatrix} R(n) \\ R(n-1) \\ \dots \\ R(0) \end{bmatrix}. \quad (7)$$

Implementation issues:. To practically implement the *LMMSE* predictor, we consider the following three issues:

1. The *LMMSE* predictor is derived under the assumption of zero mean stationary stochastic process. As the time series on-line measured is not of zero mean, we subtract the mean value from the original time series, apply the *LMMSE* predictor to estimate the aggregated series in the next interval, and then add back the mean value.
2. We have to determine how far ahead traffic prediction is made. This translates into the problem of determining an appropriate value, τ, for the measurement interval. Fortunately, the *LRD* characteristic of the network traffic implies the relatively low decay of the autocorrelation function, and hence the value of τ is not very critical to the performance. In the simulation study, we set τ to be in the order of one to several RTTs.
3. The operations involved in the calculation of LMMSE coefficients (Eqs. (5) and (6)) are multiplication of time series samples and matrix manipulation, for which fast algorithms exist [1] and can be readily implemented. In particular, the author of [1] gave an adaptive algorithm in linear prediction in which matrix inverse in Eq. (6) can be avoided. Specifically, the algorithm starts with an initial estimate of the coefficient a_0. Each time a new data point, $f^m(n)$, is obtained, the algorithm updates a_{n+1} using the recursive equation:

$$a_{n+1} = a_n + \mu \cdot e(n) \cdot f^m(n), \quad (8)$$

where $e(n)$ is the prediction error and μ is a constant. If $f^m(n)$ is stationary, a_i is shown to converge in the mean to the optimal solution [1].

Why we use a LMMSE predictor:. As mentioned in Section II-A, several fractional models and their corresponding predictors have been proposed for time series with LRD. The reasons for designing a LMMSE predictor, rather than using FBM and fractal ARIMA, are two fold:
Accuracy: The most important criterion in choosing a predictor is the accuracy. Specifically, let $f(t)$, $\overline{f(t)}$, and $\hat{f(t)}$ represent, respectively, the real traffic, the result of fitting $f(t)$ into a model (*FARIMA* or *FBM*), and the estimated traffic. Then the total predictor error is

$$\begin{aligned} |\hat{f}(t+\tau) - f(t+\tau)| &= |\hat{f}(t+\tau) - \overline{f}(t+\tau) + \overline{f}(t+\tau) - f(t+\tau)| \\ &\le |\hat{f}(t+\tau) - \overline{f}(t+\tau)| + |\overline{f}(t+\tau) - f(t+\tau)| \\ &= err_{model} + err_{predictor}. \end{aligned} \quad (9)$$

That is, the total prediction error consists of the model fitting error, err_{model}, introduced by fitting real traffic into a model and the prediction error, $err_{predictor}$, introduced by the predictor itself. The second term can usually be analytically derived, while the first term has to be empirically measured. We depict the second term versus H under the LMMSE predictor (Eq. (7)/σ_x^2), the FBM predictor (Eq.(34) in [6]), and the fractal ARIMA predictor (Eq. (39) in [6]) in Fig. 2 (*a*). When $H \to 1$ the relative error converges to 0 under all three models. Moreover, the three curves are close to one another when $H \le 0.85$ (beyond which the curve corresponding to the fractal ARIMA model differs notably from the other curves). Since analysis of real traffic traces indicates that the H parameter of the Internet traffic rarely exceeds 0.85 [20], from the theoretic perspective, all three predictors are equally well-suited for Internet traffic prediction.

We have also conducted simulation to evaluate these three predictors in terms of the total prediction errors. The three predictors are used to predict the composite traffic that traverse a bottleneck link of capacity 20 Mbps and buffer size 100 packets. Totally 60 connections are established, with each generating packets (of size 1000 bytes) using the on-off traffic model in *ns-2*. As shown in Fig. 2 (*b*), the *LMMSE* makes better prediction than the other two predictors. This is perhaps because *LMMSE* is non-model based and hence does not introduce err_{model}.

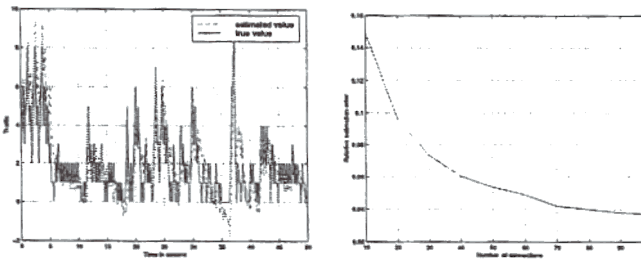
Implementability: To implement the two fractional model based predictors, one has to on-line estimate the H parameter and engage in complicated calculation of weight coefficients. Specifically, one has to estimate the value of H in the *FBM* model and calculate the weight coefficient in the form of

$$\frac{sin(\pi(H-\frac{1}{2}))}{\pi}(-t(T+t))^{-H+\frac{1}{2}} \int_0^a \frac{(\tau(\tau+T))^{H-\frac{1}{2}}}{\tau - t} d\tau.$$

where T is the interval during which the history information is gathered to predict the value at time $T+t$. Similarly, one has to estimate the value of $d = H - 1/2$ in the fractal *ARIMA* model and calculate the weight coefficient in the form of

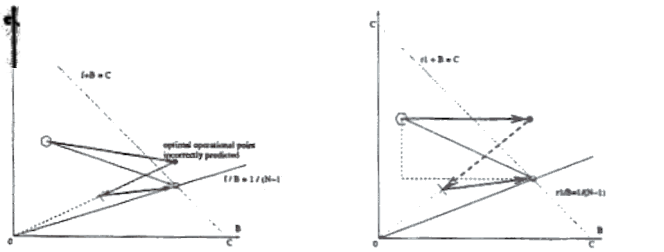
$$\beta_{kj} = -\binom{k}{j} \frac{\Gamma(j-d)\Gamma(k-d-j+1)}{\Gamma(-d)\Gamma(k-d+1)},$$

Proceedings of the 17th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT'02)
1063-6722/02 $17.00 © 2002 **IEEE**



(a) actual vs. estimate (b) estimation error

Fig. 3. Actual and estimated throughput attained by a TCP connection.



(a) prediction error (b) different RTT among connections

Fig. 4. The phase plots that illustrate why the operational point occasionally goes beyond the capacity line.

where $\Gamma()$ is the gamma function. (The interested reader is referred to [6] for a detailed account.) The *LMMSE* predictor, on the other hand, does not require estimation of such parameters, but instead calculates the parameter needed (i.e., $R(i)$'s in Eq. (6)) directly from the collected series samples. Moreover, as mentioned above, there exist fast algorithms that can be readily implemented to perform operations involved in the calculation of LMMSE coefficients (Eqs. (5), (6), and (8)) [1].

Validation of the LMMSE predictor:. To validate the design of the *LMMSE* predictor, we have implemented it at the TCP sender in *ns-2* and tested its prediction capability in both the single-bottleneck and multiple-bottleneck networks. We found that the throughput actually attained and the corresponding estimate agree very well. To illustrate this, we depict in Fig. 3 (a) the simulation results of the actual/estimated throughput of a TCP connection in the single bottleneck network. Totally 60 TCP connections are established over the bottleneck link, and generate packets using the on-off model. The H parameter of the attainable throughput series is 0.75, confirming the LRD characteristic. The estimated attainable throughput agrees very well with the actual values, and the ratio of the mean estimation error to the actual value is 0.05.

B. Congestion Control with the Use of Prediction Results

To study the effect of the number of background connections on the prediction accuracy, we repeat the above experiment, but vary N and depict the result in Fig. 3 (b). The estimation error ($|\frac{\hat{f}(t)-f(t)}{f(t)}|$) of the *LMMSE* predictor decreases as N increases and is less than 0.15 when $N \geq 10$.

We use the phase plot in Fig. 1 to describe how the prediction result, $\hat{f}^{(m)}(n+1)$, can be utilized to adjust the window increase/decrease in TCP congestion control. As discussed in Section I, the optimal operational point of N TCP connections sharing the capacity, C, of a bottleneck link is the interaction of line $f + B = C$ (the capacity line) and line $f/B = 1/(N-1)$ (the fairness line).

Suppose the initial operational point is the cross point. Without the knowledge of N, a TCP connection employs the AIMD algorithm, and follows the dashed line to reach the optimal operational point in several round trip times [3]. Moreover, in the course of reaching the optimal point, the dashed line crosses the capacity line multiple times, implying that multiple packet losses occur (and the MD phase takes effect multiple times). If the TCP connection can infer the value of N (and hence the optimal operational point), it can adjust its congestion window to directly move to the optimal operational point (as shown by the solid line in Fig. 1). Moreover, this can be achieved, in the best case, in one RTT without crossing the capacity line (and hence incurring packet losses).

The key issue now is how to infer the number of connections, N. Let $W_i(n)$ and $W_i(n+1)$ denote, respectively, the current congestion window size and the congestion window size in the next RTT for connection i, and $\eta_i(n)$ denote the ratio of $W_i(n+1)/W_i(n)$. With the estimate of the attainable throughput, $\hat{f}^{(m)}(n+1)$, in the next RTT, and under the fairness criterion, the TCP-TP sender sets

$$N = \left\lceil \frac{C}{\hat{f}^{(m)}(n+1)} \right\rceil. \tag{10}$$

TCP-TP then operates as follows. TCP-TP does not change the operations in the slow start phase or the MD portion in the congestion avoidance phase. It only changes the AI portion in the congestion avoidance phase. That is, when positive acknowledgment is received, a TCP-TP sender sets the congestion window, $W_i(n+1)$, in the next RTT as

$$W_i(n+1) \leftarrow \frac{C \cdot RTT}{N}. \tag{11}$$

Note that the new adjustment in the AI phase may be (multiplicative) window increase or decrease, depending on the ratio $\eta_i(n)$ is greater than or less than 1. In what follows, we give the correctness claim of TCP-TP under the "ideal" case in Theorem 1.

Theorem 1: If (i) all TCP connections are subject to the same RTTs, (ii) the initial operational point is below the capacity line, and (iii) the throughput prediction can be accurately made, then the operational point of a TCP-TP connection will not go beyond the capacity line and the TCP-TP connection will reach the optimal operational point in one RTT. *Proof:* Refer to [6]. □

Now we discuss how TCP-TP operates if one or more of the conditions in Theorem 1 do not hold. If the initial

Proceedings of the 17th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT'02)
1063-6722/02 $17.00 © 2002 IEEE

operational point is above the capacity line, the MD phase eventually takes effect and drags the operational point below the capacity line. In the case that prediction errors occur and/or RTTs differ among connections, the operational point may occasionally go beyond the capacity line and packet loss may occur (Fig. 4). For example, as shown in Fig. 4 (b), if the TCP-TP connection of interest responds to the system significantly faster (i.e., with a smaller RTT), then the system follows the thin dotted line to reach the optimal operational point. On the other hand, if the "composite" background traffic responds faster, then the system follows the boldfaced line, and may go beyond the capacity line and incur packet loss (at which point the MD phase takes effect). Since we do not modify the MD phase, once packet loss occurs, the congestion window is halved and the operational point is dragged back to below the capacity line (along the dashed line from the filled circle point to the cross point in Fig. 4). After that, the revised AI phase takes effect and attempts to drag the operational point toward the optimal point in the next RTT (along the solid line from the cross point to the hollow circle point in Fig. 4). As indicated in [3], the MD phase is necessary for TCP to reach equilibrium. By leaving the MD phase unchanged, we ensure that the stability is not impaired by the modification made in the AI phase.

In the case that prediction errors persist, the operational point may never reach the optimal operational point, but instead stagger in the neighboring area of the optimal point. Although the stability is ensured with the use of the MD phase, the long-term fairness may be impaired. We will analytically study the impact of inaccurate prediction on the fairness in Section IV.

IV. Impact of Prediction Errors on Fairness

In this section, we analyze how prediction errors affect the performance of TCP-TP in terms of fairness. Specifically, let $\hat{f}_i^{(m)}(n+1)$ and $f_i^{(m)}(n+1)$ denote, respectively, the estimate and actual values of the throughput attained by connection i. Then the prediction error is written as

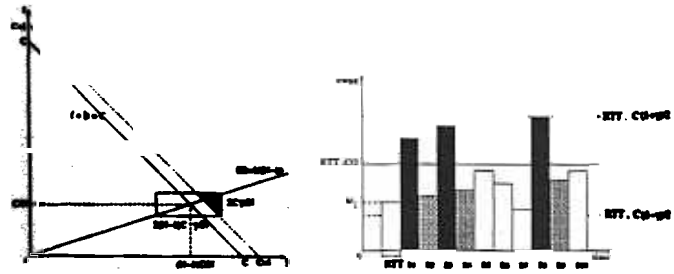
$$\tau_i(n) = \frac{\hat{f}_i^{(m)}(n+1) - f_i^{(m)}(n+1)}{f_i^{(m)}(n+1)}, \tag{12}$$

and the fairness index among N connections (defined in [3]) as

$$F(\tau) = \frac{(\sum_{i=1}^{N} f_i^{(m)}(n))^2}{N(\sum_{i=1}^{N}(f_i^{(m)}(n))^2)}. \tag{13}$$

Note that $F(\tau) = 1$ if the bandwidth of the bottleneck link is fairly shared among the N connections ($f_i^{(m)}(n) = C/N, \forall i$). We will analyze the impact of $\tau_i(n)$ on $F(\tau)$ under the cases of $N = 2$ and $N > 2$. We will only focus on the case of $N > 2$, interested readers please refer to [6] for detailed derivation of case $N = 2$.

Let $\gamma \triangleq \max(|\tau_1(n)|, ..., |\tau_N(n)|)$. Then, as shown in Fig. 5 (a), the predicted optimal operational point falls



(a) phase plot (b) congestion window

Fig. 5. The phase plot and the time diagram of the congestion window in the case that the prediction error is bounded by γ. W_i is the initial window size.

in a rectangle with edges equal to $\frac{2C\gamma}{N}$ and $\frac{2(N-1)C\gamma}{N}$. If the buffer size at the bottleneck link is L, then under the assumption that all connections are synchronous, packet loss occurs when the operational point goes beyond line $f + B = C + \frac{L}{RTT}$ (which we call the augmented capacity line, Fig. 5 (a)). For clarity of notation, let $\ell \triangleq L/RTT$.

For ease of analysis, we assume that the prediction is independently made by each individual connection and that the prediction made in disjoint time intervals is independent of one another. Then the probability that the predicted optimal point falls in any point in the square is uniform over the square. The probability that the predicted optimal point goes beyond the capacity line is then

$$p = \frac{\text{shaded area in Fig. 5 (a)}}{\text{rectangular area}} \tag{14}$$

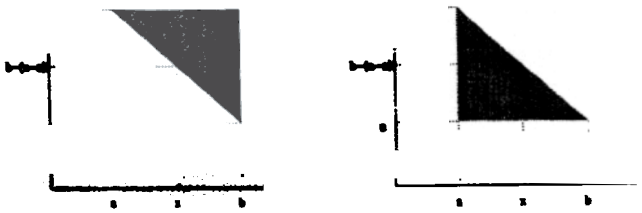
As shown in Fig. 5 (b), all the congestion windows under TCP-TP are constrained between the two dashed lines $RTT \cdot C(1+\gamma)/2$ and $RTT \cdot C(1-\gamma)/2$. Furthermore, in the time intervals labeled as $R1$, $R3$, $R8$, congestion occurs, and the congestion window is reduced in half in the following interval. The long-term attainable throughput T can be calculated as

$$\begin{aligned} T &= \lim_{M\to\infty} \frac{\sum_{n=1}^{M} W_i(n)}{M \times RTT} \\ &= \lim_{M\to\infty} \frac{M_c \cdot E(W_c) + M_d \cdot E(W_d) + M_u \cdot E(W_u)}{M \times RTT}, \end{aligned} \tag{15}$$

where M_c, M_d, and M_u are, respectively, the number of intervals in which congestion occurs, the number of intervals immediately following an interval in which congestion occurs, and the number of the other intervals, and $E(W_c)$, $E(W_d)$, and $E(W_u)$ are, respectively, the average window sizes in the M_c, M_d, and M_u intervals.

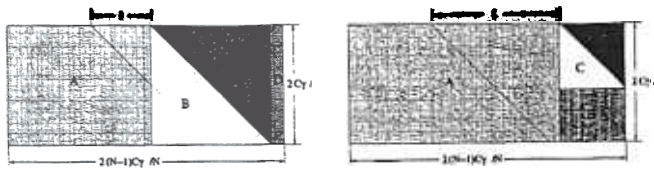
Derivation of M_c, M_d, and M_u:. Based on the uniform distribution assumption, with probability p (Eq. (14)) the predicted optimal point goes beyond the augmented capacity line and congestion occurs. Following the interval in which congestion occurs, the MD phase takes effect, the congestion window is halved, and the operational point falls below the augmented capacity line. Then, in the next interval a new prediction is made and the cycle repeats.

Proceedings of the 17th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT'02)
1063-6722/02 $17.00 © 2002 IEEE



(a) p.d.f. for Lemma 1 (b) p.d.f. for Lemma 2

Fig. 6. Probability density functions of two random variables, X and Y.



(a) Sub-case 1 **(b) Sub-case 2**

Fig. 7. The phase plot in the case that the prediction error is bounded by γ and $N > 2$.

Consequently,

$$M_c = M_d = \frac{p}{1+p} \cdot M; \quad M_u = \frac{1-p}{1+p} \cdot M. \tag{16}$$

Derivation of $E(W_c)$, $E(W_d)$, and $E(W_u)$:. To facilitate derivation of $E(W_c)$, $E(W_d)$, and $E(W_u)$, we first give the following two lemmas:

Lemma 1: Given two random variables, X and Y, with the joint probability density function

$$f(x,y) = \begin{cases} \left(\frac{(b-a)^2}{2}\right)^{-1} & \text{if } (x,y) \in \text{shaded triangle in Fig. 6(a)}, \\ 0, & \text{otherwise.} \end{cases}$$

Then $E(X) = E(Y) = \frac{a+2b}{3}$.

Lemma 2: Given two random variables, X and Y, with the joint probability density function

$$f(x,y) = \begin{cases} \left(\frac{(b-a)^2}{2}\right)^{-1}, & \text{if } (x,y) \in \text{shaded triangle in Fig. 6(b)}, \\ 0, & \text{otherwise.} \end{cases}$$

Then $E(X) = E(Y) = \frac{2a+b}{3}$.

As shown in the enlarged version of the rectangle that characterizes the prediction error in Fig. 7 (a)–(b), we have to consider the following three sub-cases.

Sub-Case 1:. In this case, $0 \le \ell \le \frac{N-2}{2N} \cdot C \cdot \gamma$, and the rectangle area can be divided into four areas, labeled as A, B, C, and D in Fig. 7 (a). When the operational point falls in area C or D, packet loss occurs, the probability of which can be derived, with a little bit geometrical reasoning, as $p = \frac{(N-1)C\gamma - N\ell}{2(N-1)C\gamma}$.

To derive $E(W_c)$ (and hence $E(W_d)$), note that when packet loss occurs, (i) with probability $P_C = \frac{C\gamma}{(N-1)C\gamma - N\ell}$, the operational point falls in area C, and the average window size is $RTT \cdot \frac{2(\frac{C}{N}+\frac{C\gamma}{N})+(\frac{C}{N}-\frac{C\gamma}{N})}{3} = RTT \cdot \left(\frac{C}{N} + \frac{C\gamma}{3N}\right)$ (Lemma 1); and (ii) with probability $1 - P_C$, the operational point falls in area D, and the average window size is $RTT \cdot \frac{C}{N}$. Thus, $E(W_c)$ can be written as

$$E(W_c) = P_C \cdot RTT \cdot (\frac{C}{N} + \frac{C\gamma}{3N}) + (1 - P_C) \cdot RTT \cdot \frac{C}{N} \tag{17}$$

and

$$E(W_d) = \frac{E(W_c)}{2} = RTT \cdot \frac{C}{2N} \cdot (1 + \frac{1}{3}\frac{C\gamma^2}{(N-1)C\gamma - 2N\ell}). \tag{18}$$

To derive $E(W_u)$, note that when packet loss does not occur, (i) with probability $P_A = \frac{C\gamma}{(N-1)C\gamma + N\ell}$, the operational point falls in area A, and the average window size is $RTT \cdot \frac{C}{N}$; and (ii) with probability $1 - P_A$, the average window size is $RTT \cdot \frac{\frac{C}{N}+\frac{C\gamma}{N}+2(\frac{C}{N}-\frac{C\gamma}{N})}{3} = RTT \cdot (\frac{C}{N} - \frac{C\gamma}{3N})$ (Lemma 2). Thus $E(W_u)$ can be written as

$$E(W_u) = P_A \cdot RTT \cdot \frac{C}{N} + (1 - P_A) \cdot RTT \cdot (\frac{C}{N} - \frac{C\gamma}{3N}) \tag{19}$$

Finally, the long-term attainable throughput T can be expressed as:

$$T = \frac{1}{2(1+p)} \cdot \frac{C}{N} \cdot (2 + p + \frac{C\gamma^2}{3}(\frac{3p}{(N-1)C\gamma - N\ell} - \frac{2(1-p)}{(N-1)C\gamma + N\ell})). \tag{20}$$

Sub-Case 2:. In this case, $\frac{(N-2)C\gamma}{N} \le \ell < C\gamma$, and the rectangle area can also be divided into four areas (Fig. 7 (b)). When the operational point falls in area D, packet loss occurs, the probability of which can be derived, following the same line of reasoning in sub-case 1 to be $p = \frac{N^2(C\gamma-\ell)^2}{8(N-1)C^2\gamma^2}$. Following the same line of reasoning in sub-case 1, we also have $M_c = M_d = \frac{p}{1+p} \cdot M$, and $M_u = \frac{1-p}{1+p} \cdot M$.

Using Lemma 1, $E(W_c)$ can be written as:

$$E(W_c) = RTT \cdot \frac{2(\frac{C}{N}+\frac{C\gamma}{N})+(\frac{C}{N}+\ell-\frac{(N-1)C\gamma}{N})}{3} = RTT \cdot \frac{3C + N\ell + (3-N)C\gamma}{3N}, \tag{21}$$

and hence $E(W_d) = \frac{E(W_c)}{2}$ can be expressed as

$$E(W_d) = RTT \cdot \frac{3C + N\ell + (3-N)C\gamma}{6N}. \tag{22}$$

To derive $E(W_u)$, note that (i) with probability $P_C = \frac{N^2(C\gamma-\ell)^2}{8C^2\gamma^2(N-1)-N^2(C\gamma-\ell)^2}$ the operational point falls into area C, and the average window size is $RTT \cdot \frac{(\frac{C}{N}+\frac{C\gamma}{N})+2(\frac{C}{N}+\ell-\frac{(N-1)C\gamma}{N})}{3}$ (Lemma 2); (ii) with probability $P_B = \frac{2N^2(\ell-\frac{(N-2)C\gamma}{N})(C\gamma-\ell)}{8C^2\gamma^2(N-1)-N^2(C\gamma-\ell)^2}$, the operational point falls in area B, and the average window size is $RTT(\frac{C}{N} + \frac{\ell}{2} - \frac{C\gamma}{2})$; and (iii) with probability $1 - P_C - P_B$, the operational point

Proceedings of the 17th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT'02)
1063-6722/02 $17.00 © 2002 **IEEE**

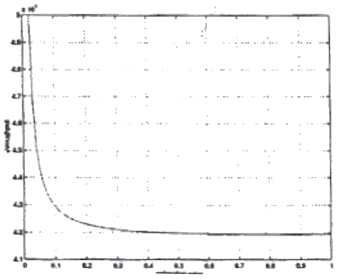


Fig. 8. The plot of the long-term attainable throughput, T, versus the maximum percentage of prediction errors, γ.

falls in area A, and the average window size is $RTT \cdot \frac{C}{N}$ Thus $E(W_u)$ can be written as:

$$E(W_u) = (1-P_B-P_C)\cdot RTT\cdot\frac{C}{N} + P_B\cdot RTT\cdot(\frac{C}{N}+\frac{\ell}{2}-\frac{C\gamma}{2}) + P_C\cdot RTT\cdot\frac{\frac{C}{N}+\frac{C\gamma}{N}+2(\frac{C}{N}+\ell-\frac{(N-1)C\gamma}{N})}{3} \quad (23)$$

Finally the long-term attainable throughput T can be expressed as

$$T = \frac{C}{(1+p)2N}\cdot(2+p+\frac{Np\ell}{C}+(3-N)p\gamma+\frac{(1-p)\ell N}{2C} (3P_B+4P_C)+\frac{(1-p)\gamma}{3}((4-2N)P_C-3NP_B)) \quad (24)$$

Sub-Case 3:. In this case, $\ell \le C\gamma$, the operational point always falls within the augmented capacity line, no packet loss occurs ($p = 0$), and hence $T = \frac{C}{N}$ and $F = 1$.

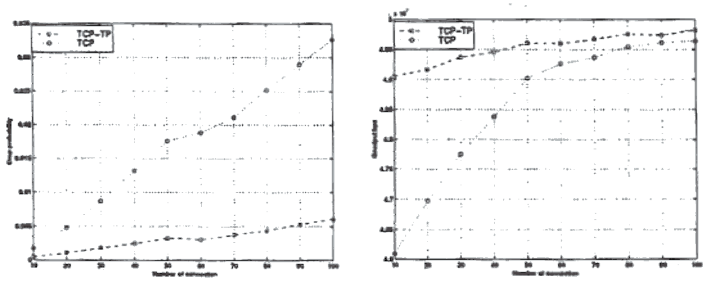
Fig. 8 gives an example plot of T versus γ, with $C = 10$ Mbps, $N = 20$, $L = 40$ packets with the average packet size of 1000 bytes, and RTT = 20 ms. As γ increases from 0 to 1, T decreases from $C/N = 0.5$ Mbps to 0.42 Mbps and F decreases from 1.0 to 0.975 (approximately 2.5% degradation). This conclusion also holds true for other combinations of C, N and L. This demonstrates that the impact of prediction errors on fairness is minimal.

V. Performance Evaluation

We have implemented TCP-TP both in ns-2 and in the FreeBSD 4.1. kernel, and conducted simulation and empirical studies to validate the proposed design and compared the performance of TCP-TP against TCP-new Reno.

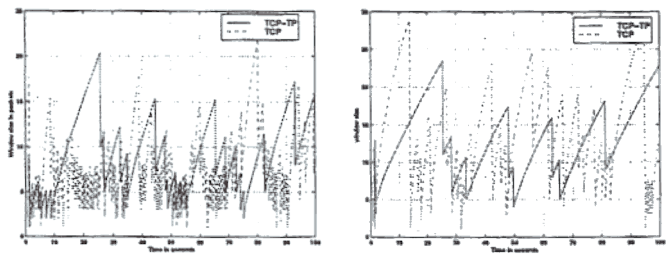
A. Simulation Results

In the simulation study, we examine the behavior of TCP-Reno and TCP-TP under a variety of network topologies, e.g., the single bottleneck topology, the multiple bottleneck link topology (e.g., Fig. 13), and arbitrary network topology, and under different scenarios. Each data point is the result averaged over 10 simulation runs. Due to the space limitation, in what follows we only report on a small set of simulations which we believe is most representative.



(a) Packet loss ratio (b) Attained throughput

Fig. 9. Performance of TCP-new Reno and TCP-TP w.r.t. packet loss rate and attained throughput in the case of equal RTTs.



(a) single-bottleneck (b) multiple-bottleneck

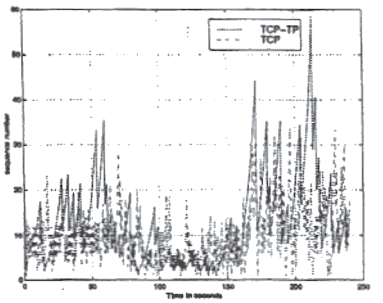
Fig. 10. The instantaneous window size of a connection in Experiment 1.

A.1 Results in the Single-Bottleneck Topology

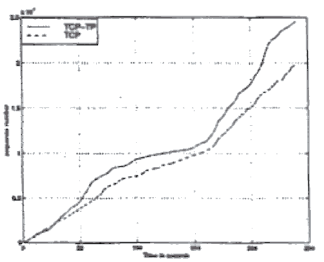
We establish N TCP connections that generate packets (of size 1000 bytes) according to the on-off traffic model, where N varies from 10 to 100. We then measure the packet loss ratio, and the throughput attained by all TCP receivers, and the congestion window of each TCP connection.

Performance under the case of equal RTTs:. In the first experiment, we assume all the connections are subject to the same RTT = 50 ms. Fig. 9 gives the performance of TCP-new Reno and TCP-TP. TCP-TP outperforms TCP-new Reno both in terms of packet loss ratio and throughput attained by all receivers. In particular, the performance improvement in packet loss ratio can be as high as 75%. The performance improvement in the total attained throughput is not as significant. This is due to the fact that the attained throughput is constrained by the bandwidth of the bottleneck link (evidenced by the decreasing performance gap between TCP-TP and TCP as N increases). To further characterize the cause of the performance gain, we give in Fig. 10 (a) the instantaneous window size of one of the connections. It is obvious to see that TCP-TP incurs much less window reductions as compared to TCP-new Reno. This shows that with the prediction results, TCP-TP can greatly reduce the likelihood that the operational point goes beyond the capacity line. We also calculate the fairness, F, under both TCP-TP and TCP-new Reno. The results are both in the range of 0.98–0.99, and are very close to each other.

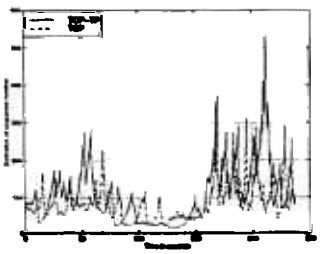
Proceedings of the 17th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT'02)
1063-6722/02 $17.00 © 2002 IEEE



(a) Instantaneous window size

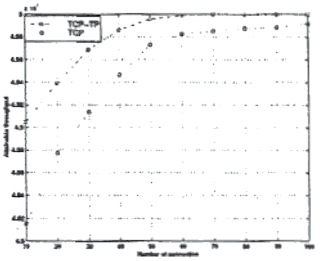


(b) Sequence number

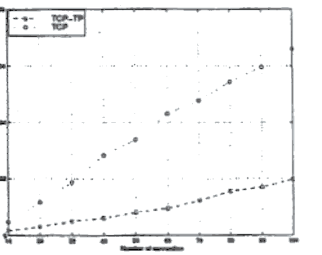


(c) Derivative of SN

Fig. 11. Performance of TCP-new Reno and TCP-TP in the case of dynamic connection establishment and termination.



(a) Attainable throughput



(b) Packet loss ratio

Fig. 12. Performance of TCP-new Reno and TCP-TP in the case of different RTTs.

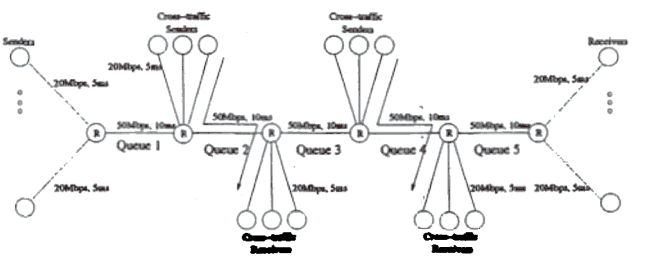


Fig. 13. The multiple bottleneck simulation topology.

Performance under dynamic changes of TCP connections:. In the second experiment, we use the same topology as in Experiment 1, but dynamically establish/terminate connections. Initially 30 connections commerce at time 0s. At time 100s, additional 30 connections are established. At time 160s, 40 connections are terminated, while the others continue until the end of the simulation (240 s). We measure the instantaneous window size and the sequence number increment of a connection that commences at time 0s and runs until 240s.

Fig. 11 gives the instantaneous window size, the increment in sequence number, and the derivative of the sequence number increment under both TCP-TP and TCP-new Reno. As shown in Fig. 11 (a), TCP-TP responds more quickly to the change of N at time 100s and 160s, meaning that it reaches the new optimal operational point faster. This is corroborated by the observation in Fig. 11 (c): around time 100s, TCP-TP has a smaller value of derivative of sequence number, implying that the sequence number increases slower in response to the establishment of new connections. A similar conclusion can be made at time 160s at which TCP-TP has a larger value of derivative of sequence number, implying that the sequence number increases faster. Also, as shown in Fig. 11 (b), the sequence number under TCP-TP is always higher than that under TCP, suggesting a better goodput under TCP-TP.

Performance in the case of different RTTs:. In the third experiment, we use the same network topology as in Experiment 1, but vary the RTT experienced by different connections. The RTT of a TCP connection is drawn from an uniform distribution $[20, 100]$ ms. Fig. 12 gives the performance of TCP-new Reno and TCP-TP. The performance shows the similar trend to that of the equal RTT case (Experiment 1).

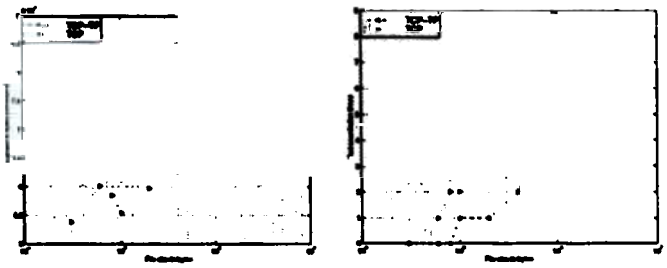
A.2 Results in the Multiple-Bottleneck Topology

The multiple-bottleneck topology used is shown in Fig. 13, in which there exist five queues in the topology, and cross traffic is generated between the second and third routers (queue 2), and between the forth and the fifth routers (queue 4). We establish N TCP connections with the on-off traffic model, where N varies from 10 to 100. The cross traffic is composed of TCP connections as well, and the number of TCP connections in each cross traffic bundle is set to $0.5N$. The performance is very similar to that of the single bottleneck case. Hence we depict in Fig. 10 (b) only the curve of the instantaneous window size under TCP-New Reno and TCP-TP.

B. Empirical Study

We have implemented TCP-TP in the FreeBSD 4.1. kernel. The interested reader is referred to [6] for a detailed account of the new data structures/functions created and the changes made in the existing kernel functions. The major changes we made are that we include a LMMSE predictor in the *TCP-Input* module. In particular, we added several data structures into the TCP protocol control block to keep track of the recent history of the amount of successfully transmitted (acknowledged) data. This data is updated by the *TCP-input* function whenever an non-duplicate acknowledgment is received. Finally, the window

Proceedings of the 17th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT'02)
1063-6722/02 $17.00 © 2002 **IEEE**



(a) Average throughput (b) Retransmission timeouts

Fig. 14. Empirical results of experiments between OSU and UW.

increase/decrease operations in the AI phase of the congestion avoidance phase is modified. The implementation requires an addition/change of approximately 100 lines of C code into the kernel. We also include an implementation of *Pathcar* [9] for measuring the bandwidth of the bottleneck link. Although this version of implementation is on the FreeBSD kernel, it is straightforward to port it to other UNIX-based operating systems.

We carried out experiments over the Internet, with receivers located at UCSB (alpha.ece.ucsb.edu), UCI (rodan.ics.uci.edu), UMD (dsp7.eng.umd.edu), and UW-Madison (hertz.ece.wisc.edu) and the sender located at OSU (eepc118.eng.ohio-state.edu). The experiments were carried out in 3 different time intervals (morning, afternoon, and night) on a daily basis for a period of 2 weeks. In each set of experiments, we establish a HTTP connection between the sender and a receiver, and either TCP or TCP-TP is used as the underlying transport protocol. The size of the file to be transferred varies from 30 Kbytes (which is the average size of a webpage) to 8Mbytes (which represents extremely large file transfer). We measure the average throughput for each connection and the total number of retransmission timeouts occurred (the later is an indication of multiple, consecutive packet losses). Fig. 14 gives the empirical results for experiments performed between OSU and Wisconsin in the afternoon intervals. (The other results exhibit similar trends and hence are not shown.) Again TCP-TP outperforms TCP with respect to the average throughput and the number of retransmission timeouts.

VI. Concluding Remarks

We have demonstrated in this paper that the self-similar characteristics of network traffic can be exploited to affect throughput gains and reduce packet losses in TCP congestion control. In particular, we show that with the use of a simple LMMSE predictor, one can accurately (with estimation error $\leq 15\%$) estimate the future traffic at least one RTT ahead. A TCP connection can then use the prediction result to infer the optimal operational point at which a TCP connection should operate. Although the analysis is pinpointed in the context of AIMD steady-state dynamics, the resulting scheme (called *TCP-TP*) is light weight, requires modification (tens of lines of code change) only at the TCP sender side, and achieves performance gains (in some simulation cases, up to 75%) in terms of packet loss ratio, attainable throughput, and responsiveness to dynamic network traffic changes.

Exploitation of LRD characteristics for better resource control is certainly not restricted to TCP congestion control. We are currently exploring alternative forms of exploiting correlation structures in different network components across the protocol stack.

References

[1] A. M. Adas. Using Adaptive Linear Prediction to Support Real-Time VBR Video Under RCBR Network Service Model. *IEEE/ACM Transactions on Networking*, Vol.6, No.5, October 1998.

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

[3] D. Chiu and R. Jain. Analysis of the Increase/Decrease Algorithms for Congestion Avoidance in Computer Networks. *Journal of Computer Networks and ISDN*, June 1989.

[4] K.Claffy, G.Miller and K.Thompson. The Nature of the Beast: Recent Traffic Measurements from An Internet Backbone. IN *Proc. of INET'98*

[5] A. Erramilli, O. Narayan, and W. Willinger. Experimental Queuing Analysis with Long-Range Dependent Traffic. *IEEE/ACM Transactions on Networking*, April 1996.

[6] G. He, Y. Gao, J. Hou, and K. Park. A Case for Exploiting Self-Similarity of Network Traffic in TCP Congestion Control. Extended version, July 2001. Available at http://eewww.eng.ohio-state.edu/~gaoy.

[7] B. Hari, R. Hariharan, S. Srinivasan. An Integrated Congestion Management Architecture for Internet Hosts. *Porc. ACM SIGCOMM'99*, September 1999.

[8] J. R. M. Hosking. Fractional differencing. Biometrika, 68, pp. 165-176, 1981.

[9] V. Jacobson. Pathchar — A Tool to Infer Characteristics of Internet Paths. Available at ftp://ftp.ee.lbl.gov/pathchar/, 1997.

[10] K. Lai and M. Baker. Measuring Link Bandwidths Using a Deterministic Model of Packet Delay. In *Proc. of ACM SIGCOMM'2000*, August 2000.

[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*, February 1994.

[12] I. Norros. On the Use of Fractional Brownian Motion in the Theory of Connectionless Networks. *IEEE Journal on selected areas in communications*, Vol. 13, No. 6, August 1995.

[13] V. Paxson. End-to-End Internet Packet Dynamics. In *Proc. of ACM SIGCOMM*, September 1997.

[14] V. Ribeiro, M. Coates, R. Riedi, S. Sarvotham, B. Hendricks, and R. Baraniuk. Multifractal Cross-Traffic Estimation. In *Proc. of the ITC Specialist Seminar on IP Traffic Measurement, Modeling and Management*, September 2000.

[15] A. Sang, and S. Li. A Predictability Analysis of Network Traffic. In *Proc. of IEEE INFOCOM'2000*, Tel-Aviv, Israel, March 2000.

[16] S. Savage. Sting: A TCP-Based Network Measurement Tool. In *Proc. of the USENIX Symp. on Internet Technologies and Systems*, 1999.

[17] T. Tuan and K. Park. Multiple Time Scale Congestion Control for Self-Similar Network Traffic. Performance Evaluation, 36:359–386, 1999.

[18] T. Tuan and K. Park. Multiple Time Scale Redundancy Control for QoS-Sensitive Transport of Real-Time Traffic. *Proc. IEEE INFOCOM '00*, April 2000.

[19] A. Veres, Z. Kenesi, S. Molnar, and G. Vattay. On the Propagation of Long-Range Dependence in the Internet. In *Proc. of ACM SIGCOMM'2000*, Stockholm, Sweden, August 2000.

[20] W. Willinger, M. S. Taqqu, R. Sherman, and D. V. Wilson. Self-Similarity Through High-Variability: Statistical Analysis of Ethernet LAN Traffic at the Source Level. In *Proc. ACM SIGCOMM'91*, pp. 149–157, 1991.

Proceedings of the 17th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT'02)
1063-6722/02 $17.00 © 2002 **IEEE**