On the Predictability of Large Transfer TCP Throughput*

Qi He
Georgia Tech
qhe@cc.gatech.edu

Constantine Dovrolis
Georgia Tech
dovrolis@cc.gatech.edu

Mostafa Ammar
Georgia Tech
ammar@cc.gatech.edu

ABSTRACT

Predicting the throughput of large TCP transfers is important for a broad class of applications. This paper focuses on the design, empirical evaluation, and analysis of TCP throughput predictors. We first classify TCP throughput prediction techniques into two categories: Formula-Based (FB) and History-Based (HB). Within each class, we develop representative prediction algorithms, which we then evaluate empirically over the RON testbed. FB prediction relies on mathematical models that express the TCP throughput as a function of the characteristics of the underlying network path. It does not rely on previous TCP transfers in the given path, and it can be performed with non-intrusive network measurements. We show, however, that the FB method is accurate only if the TCP transfer is window-limited to the point that it does not saturate the underlying path, and explain the main causes of the prediction errors. HB techniques predict the throughput of TCP flows from a time series of previous TCP throughput measurements on the same path, when such a history is available. We show that even simple HB predictors, such as Moving Average and Holt-Winters, using a history of few and sporadic samples, can be quite accurate. On the negative side, HB predictors are highly path-dependent. We explain the cause of such path dependencies based on two key factors: the load on the path and the degree of statistical multiplexing.

Categories and Subject Descriptors: C.2.5 [Computer Communication Networks]: Internet

General Terms: Experimentation, Measurement

Keywords: Network measurement, TCP modeling, time series forecasting, performance evaluation

1. INTRODUCTION

With the advent of overlay and peer-to-peer networks [3, 5], Grid computing, and CDNs, performance prediction of network paths becomes an essential task. To name just a few applications, such predictions can be used in path selection for overlay and multihomed networks [2], dynamic server selection [15], and peer-to-peer parallel downloads. Arguably, the most important performance metric of a path is the average throughput of TCP transfers. The reason is that most data-transfer applications, and about 90% of the Internet traffic, use the TCP protocol. When it comes to performance prediction, the focus is typically on bulk TCP transfers, lasting more than a few seconds. Short TCP flows are often limited by slow-start, and their performance is determined by the Round-Trip Time (RTT) and the presence of random losses [7].

In this work, we focus on predicting the throughput of a bulk TCP transfer in a given network path, *prior* to actually starting the transfer. For many applications, such as server selection and overlay route selection, a throughput prediction is needed before the flow starts. The reason is that rerouting an established TCP connection to a different network path or server can cause problems such as migration delays, packet reordering, and re-initialization of the congestion window. Note that TCP throughput prediction is different than TCP throughput *estimation*. The latter is performed while the flow *is in progress*. An example of a TCP throughput estimation scheme is TCP-Friendly Rate Control (TFRC) [6]. Unlike the prediction of RTT and loss rate, which can be based on direct and low-overhead measurements, predicting TCP throughput is significantly harder. First, TCP throughput depends on a large number of factors, including the transfer size, maximum sender/receiver windows, various path characteristics (RTT, loss rate, available bandwidth, nature of cross traffic, reordering, router/switch buffering, etc) and the exact implementation of TCP at the end-hosts. Second, direct measurement of TCP throughput using large "probing" transfers are highly intrusive because the latter can saturate the underlying paths for significant time periods. What is really desired is *a low-overhead TCP throughput prediction technique that either avoids probing transfers altogether, or requires only a limited amount of probing traffic.*

This paper focuses on the design, empirical evaluation, and analysis of TCP throughput predictors for a broad class of applications. The common requirement of such applications is that they rely on an accurate throughput prediction prior to the start of the TCP transfer. We first classify TCP throughput prediction techniques into two categories: *Formula-Based* (FB) and *History-Based* (HB). Within each class we develop representative prediction algorithms, which

*This work was supported by the NSF CAREER award ANIR-0347374.

we then evaluate empirically over the RON testbed [1]. Note that our objective is not to compare FB and HB predictors. In fact, the two schemes are complementary, as they require different types of measurements and previous information about the underlying path. Instead, our objective is to examine the key issues in each prediction scheme, evaluate their accuracy under different conditions, explain the major causes of prediction errors, and provide insight regarding the factors that affect the predictability of large transfer TCP throughput in a given path.

Specifically, FB prediction relies on mathematical models that express the TCP throughput as a function of the characteristics of the underlying network path (e.g., RTT, loss rate). For instance, the throughput-optimizing routing component of RON follows the FB approach [3], predicting TCP throughput based on the simple "square-root" formula of [11]. That formula expresses the average throughput of a congestion-limited bulk transfer as a function of the RTT and the loss rate that the connection experiences on a given path. Several similar models have been proposed in the literature [4, 12, 19], differing in terms of complexity and accuracy, modeling assumptions, and TCP flavor. In this paper, we prefer to use the main result of [12], referred to as the *PFTK formula*, because it is both simple and accurate. The main advantage of FB prediction is that it does not require any history of previous TCP transfers. In addition, FB prediction can be performed with relatively lightweight, non-intrusive network measurements of parameters such as RTT and loss rate. Unfortunately, however, our measurements show that FB schemes can lead to large prediction errors. The main reason is that throughput models require knowledge of the path characteristics *during* the TCP flow, whereas FB predictions measure the corresponding a priori characteristics *before* the flow starts. If the flow itself causes significant changes in those characteristics, the resulting prediction errors can be unacceptably large. Another reason is that the delays or losses that a TCP flow experiences are not necessarily the same as those observed by a periodic probing stream, such as *ping* [8]. On the positive side, we do observe that the prediction errors are much lower, and probably acceptable for many applications, if the TCP transfer is limited by the receiver's advertised window to the point that the transfer does not saturate its path.

On the other hand, HB approaches use standard time series forecasting techniques to predict TCP throughput based on a history of throughput measurements from previous TCP transfers on the same path. Obviously, HB prediction is applicable only when large TCP transfers are performed repeatedly on the same path. This is the case with several applications of TCP throughput prediction, including overlay routing, parallel downloading and Grid computing. Our measurements over the RON testbed show that even simple linear HB predictors, such as Moving Average and non-seasonal Holt-Winters, are quite accurate. Furthermore, in agreement with previous work on HB prediction [22, 23], we found no major differences among a few candidate HB predictors. We do find, however, that two simple heuristics can noticeably improve the accuracy of HB predictors. The first is to detect and ignore outliers, and the second is to detect level shifts and restart the HB predictors. We next show, perhaps surprisingly, that even with a short history of a few previous transfers performed sporadically in intervals up to 30-40 minutes, prediction errors are still fairly low. On the negative side, our measurements show that HB predictors are highly path-dependent, which begs for answers to the following two questions. What makes TCP throughput much more predictable on some paths than on others, and which are the fundamental factors that affect the throughput predictability on a path? We focus on two factors that we believe are the most important: the load on the path, and the degree of statistical multiplexing. Specifically, we show using simple queueing models that the prediction error increases with the load on the bottleneck link, and decreases with the number of competing flows under constant load. Consequently, paths that are heavily loaded with just a few big flows are expected to be most difficult to predict.

The structure of the paper is as follows. We summarize the related work in Section 2. Section 3 presents a representative FB predictor and Section 4 evaluates its accuracy. Section 5 presents some HB predictors and Section 6 evaluates their accuracy. Section 7 focuses on two major factors for throughput predictability. We conclude in Section 8.

2. RELATED WORK

The motivation for some of the previous work on TCP throughput modeling has been to predict the throughput of a transfer as a function of the underlying network characteristics [6, 11, 12]. However, the accuracy of FB prediction depends on the accuracy with which these characteristics can be estimated or measured. Recently, Goyal et al. have shown that the end-to-end packet loss rate p on a path can be quite different from the "congestion event probability" p' required by the well-known PFTK model of Padhye et al. [12], and they have proposed a way to estimate p' from p [8]. Note that that work does not address the problem of estimating the required path characteristics during a flow from those observed prior to the flow.

HB throughput prediction, on the other hand, has received more attention. An operational system is the Network Weather Service (NWS) project [20]. In NWS, throughput prediction is based on small (64KB) TCP transfer probes with a limited socket buffer size (32KB). Vazhkudai et al. use bulk TCP transfers (1MB-1GB) and a large socket buffer (1MB), performed sporadically (1 minute-1 hour) [22]. They show that various linear predictors (including ARIMA models) perform similarly, and that the average prediction error on two paths ranges from 10% to 25%. Zhang et al. examine TCP throughput predictability based on a large set of paths and transfers [23]. Their TCP throughput measurements use 1MB transfers performed every minute, with 200KB socket buffers. Their main results are that 1) with several simple linear predictors, about 95% of the prediction errors are below 40%, and 2) predictions using a very long history (e.g., Moving Average with 128 samples) perform rather poorly. A study by Qiao et al. has shown that the predictability of network traffic is highly path dependent [14]. Some mathematical models (such as MMPP) have been previously used to analyze the predictability of aggregate network traffic [18].

3. FORMULA-BASED PREDICTION

The central component of an FB predictor is a mathematical formula that expresses the average TCP throughput as a function of the underlying path characteristics. Probably the most well-known such model is the "square-root" formula of [11]:

$$E[R] = \frac{M}{T\sqrt{\frac{2bp}{3}}} \quad (1)$$

where $E[R]$ is the *expected TCP throughput* (as opposed to R which denotes the *actual or measured throughput* and $\hat{R}$ which denotes the *predicted throughput*). In the previous formula, M is the flow's Maximum Segment Size, b is the number of TCP segments per new ACK, while T and p are the RTT and loss rate, respectively, as experienced by the TCP flow. This model is fairly accurate for bulk TCP transfers in which packet losses are recovered with Fast-Retransmit. In this section, we first present a more complete TCP throughput formula, as well as the corresponding FB predictor. We emphasize that our remarks regarding the accuracy and limitations of FB prediction are not specific to the particular formula we use, however.

3.1 An FB predictor

The TCP throughput formula that we use is the PFTK result of [12], which improves on the square-root formula especially in the presence of retransmission timeouts and/or a limited maximum window:

$$E[R] = \min\left(\frac{M}{T\sqrt{\frac{2bp}{3}} + T_o \min(1, \sqrt{\frac{3bp}{8}})p(1+32p^2)}, \frac{W}{T}\right) \quad (2)$$

where T_o is the TCP retransmission timeout period, and W is the maximum window size. We emphasize that p and T are the average loss rate and RTT that the *target flow* (i.e., the TCP flow whose throughput we try to predict) experiences (the main symbols we use are summarized in Table 1). Notice that the loss rate p may be zero, in which case the flow is *lossless* and $E[R]$ is given by the term W/T.

T	RTT experienced by flow†
$\hat{T}$	RTT measured with periodic probing before flow
$\tilde{T}$	RTT measured with periodic probing during flow
p	loss rate experienced by flow
$\hat{p}$	loss rate measured with periodic probing before flow
$\tilde{p}$	loss rate measured with periodic probing during flow
p'	congestion event probability experienced by flow
R	actual throughput of flow
$\hat{R}$	predicted throughput of flow
$\tilde{R}$	expected throughput of flow based on $\tilde{T}$ and $\tilde{p}$
$\hat{A}$	available bandwidth measured prior to flow
W	maximum window of flow

† The word "flow" in this table refers to the target flow.

Table 1: Table of symbols.

Suppose now that we want to apply (2) to TCP throughput prediction. The main problem is that we do not know the loss rate and RTT that the flow will experience during its lifetime. The obvious approach, which has been used in practice (e.g., in overlay routing [3]), is to measure the loss rate and RTT *before* the transfer with a utility such as *ping*, and then apply those estimates of p and T in (2). Suppose that $\hat{p}$ and $\hat{T}$ are the loss rate and RTT estimates based on *a priori* measurements. Then, if $\hat{p} \approx p$ and $\hat{T} \approx T$, the prediction accuracy will be only limited by the accuracy of these approximations and the accuracy of the mathematical model that led to (2). We can expect that $\hat{p} \approx p$ and $\hat{T} \approx T$ when the TCP flow imposes a minor load on the path's bottleneck, without affecting significantly the RTT and loss rate of the path.

A limitation of the previous approach is that it does not apply to *lossless paths*, i.e., when $\hat{p}$=0. In that case, $W/\hat{T}$ can be unrelated to the realized throughput, especially if W is much larger than the bandwidth-delay product of the path. One approach to deal with lossless paths is to predict the TCP throughput based on the *available bandwidth* (avail-bw) $\hat{A}$ of the path prior to the flow, when $\hat{A} < W/\hat{T}$. The avail-bw is the non-utilized part of the bottleneck capacity, and it can be measured non-intrusively with end-to-end probing techniques [16, 10]. Although the avail-bw and TCP throughput are not expected to be exactly equal, $\hat{A}$ can be used as a first-order approximation of R when the flow is not limited by its maximum window size W. On the other hand, if $W/\hat{T} < \hat{A}$, the flow cannot obtain all the avail-bw due to its limited maximum window, so $W/\hat{T}$ is a more reasonable predictor; we refer to such flows as *window-limited*.

To summarize, the FB predictor that we consider in the rest of this paper is given by the following equation:

$$\hat{R} = \begin{cases} \min\left(\frac{M}{\hat{T}\sqrt{\frac{2b\hat{p}}{3}} + \hat{T}_o \min(1, \sqrt{\frac{3b\hat{p}}{8}})\hat{p}(1+32\hat{p}^2)}, \frac{W}{\hat{T}}\right) & \text{if } \hat{p} > 0 \\ \min\left(\frac{W}{\hat{T}}, \hat{A}\right) & \text{if } \hat{p} = 0 \end{cases} \quad (3)$$

where $\hat{R}$ is the predicted throughput, while $\hat{T}$, $\hat{p}$, and $\hat{A}$, are the measured RTT, loss rate, and avail-bw prior to the TCP flow. We estimate the retransmission timeout period as: $\hat{T}_o = \max(1\text{sec}, 2\text{SRTT})$, where SRTT is set to the measured RTT $\hat{T}$ prior to the target flow. Note the differences between (2) and (3): the latter relies on the estimates $\hat{T}$, $\hat{p}$, $\hat{T}_o$, rather than the actual values T, p, T_o, and it also has a component that depends on the avail-bw estimate $\hat{A}$.

In the following, we discuss three potential limitations of the above predictor using some basic insight.

3.2 Errors due to load increase

An increase in the utilization of a queue (with non-periodic arrivals) typically increases the average queueing delay. Similarly, in a queue with finite buffering, an increase in the offered load can cause a higher loss probability. The increase in the queueing delays and/or the loss rate is more significant when the utilization becomes close to 100% after the load increase, or when the utilization was already that high even before the additional load. These basic facts can cause major errors in FB prediction. The reason is that the RTT $\hat{T}$ measured prior to the target flow may not reflect the increased queueing delay during that transfer. So, $\hat{T}$ can be lower than the RTT T that the target flow experiences. Similarly for the loss rate, it can be that $\hat{p} < p$. The net result of either effect is that the FB predictor can overestimate the TCP throughput, especially when the target flow saturates the bottleneck link or when the latter is already heavily loaded.

Note that the experimental validation of the PFTK result, reported in [12], was based on the "posthumous" estimation of p and T, i.e., from *tcpdump* packet traces collected at the sender/receiver while the target flow was in progress. Of course the same approach is not possible for prediction prior to the target transfer .

3.3 Errors due to TCP sampling behavior

Even when the target flow does not affect significantly the path's RTT and loss rate, it is still hard to estimate the RTT and loss rate that the TCP target flow experiences. TCP reduces its packet transmission rate when it experiences losses, which means that it tends to "sample" the RTT and packet loss processes less frequently when the path is congested. This is a very different sampling behavior than that of a utility such as ping, which typically sends periodic probing packets. Also, TCP tends to send bursts of data packets when self-clocking fails (e.g., due to ACK compression), which also leads to a different sampling behavior than periodic probing.

To make things more complex, mathematical models for TCP throughput are typically based on certain assumptions that affect the interpretation of parameters such as T or p. For instance, the PFTK model assumes that T is constant and independent of the transfer's window, and that when a packet is dropped all the remaining packets in that "flight" are also dropped (referred to as a "congestion event"). As a result, the parameter p in (2) should not be the unconditional loss probability among all packets of the target flow, but the congestion event probability. The discrepancy between these two parameters was one of the main focus points in [8]. Our $ns2$ simulations suggest that a loss rate estimate based on a periodic ping-based measurement can be an order of magnitude different than the congestion event probability. The differences between the unconditional loss probability and the congestion event probability are also noticeable, although not so major.

3.4 Errors due to avail-bw

As previously mentioned, when $\hat{p}$=0 and $\hat{A} < W/\hat{T}$, we predict the throughput of the target flow based on the path's avail-bw $\hat{A}$ prior to that flow. These two metrics, however, can be significantly different in certain cases [10]. First, whether a TCP flow can saturate the avail-bw of a path depends on the buffer space B at the bottleneck. If B is not sufficiently large, packet losses can cause significant underutilization and the resulting TCP throughput can be lower than $\hat{A}$. Second, if the competing cross traffic at the bottleneck is made of elastic flows (e.g., persistent TCP flows), the target flow can capture more than $\hat{A}$, receiving some of the bandwidth previously occupied by cross traffic flows. The actual difference between avail-bw and TCP throughput in that case depends on the number and the RTTs of the competing TCP flows.

Consequently, the avail-bw $\hat{A}$ prior to the target flow can be either an overestimation or an underestimation of the flow's throughput, depending on the amount of buffering and the "congestion responsiveness" of the cross traffic in the path. Given that it is hard to infer network buffering and cross traffic elasticity in practice, it is unclear whether we can design a better FB predictor than $\hat{A}$ for the case of lossless paths.

4. FB PREDICTION ACCURACY

The previous section argued that FB prediction can be inaccurate under certain conditions. In this section, we present measurement results from Internet paths that quantify the inaccuracy of FB prediction, and further analyze the prediction errors. First, we describe the measurement methodology and datasets we use throughout this paper.

4.1 Overview of measurement methodology

Our measurements were collected on 35 Internet paths that interconnect nodes of the RON testbed [1]. The RON nodes that we used are located mostly in US universities, but there are also two nodes in Europe and one in Korea. Out of the 35 paths, five are transatlantic paths, one between Korea and New York-NY, and the rest within the US. Based on the information provided by the RON team, seven of the paths had a DSL bottleneck at the time of our measurements. The capacities for the rest of the paths are at least 10Mbps.

We collected seven measurement "traces" on each path, with a total of 245 traces across all paths. Each trace consists of 150 back-to-back measurement "epochs". An epoch starts with an avail-bw measurement using pathload, followed by a 60-sec measurement of $\hat{p}$ and $\hat{T}$ using a home-spun ping utility that generates a 41-byte probing packet every 100ms, followed by a 50-sec TCP transfer (target flow) generated by IPerf [9] (see Figure 1). RTT and loss rate estimates are also measured during the TCP transfer. A 50-second transfer on these paths is long enough to ensure that the flow spends a negligible fraction of its lifetime in the initial slow-start. A total of 36750 TCP transfers (in the same number of epochs) were performed. The duration of each epoch (and also the time interval between successive TCP transfers) was about 2-3 minutes, while the duration of each trace was about 6 hours. The measurements were collected during a week in May 2004.

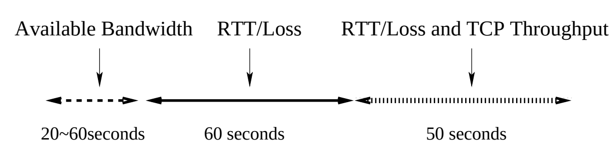


Figure 1: A measurement epoch. 150 such epochs were recorded during each trace, with 7 traces collected per path.

IPerf allows us to control the maximum TCP window size W by limiting the socket buffer size. Unless otherwise noted, we used W=1MB, which is large enough to saturate all the paths we experimented with and cause congestion. To examine the effect of W, we also performed the same measurements with W=20KB, which limits the transfer to only a fraction of the avail-bw on most paths.

Each epoch provides the following measurements: the pre-transfer estimates $\hat{p}$, $\hat{T}$, $\hat{A}$, the actual TCP throughput R, and the estimates of the loss rate $\tilde{p}$ and RTT $\tilde{T}$ *during* the transfer. The first three estimates are used in (3) to predict the TCP throughput $\hat{R}$, which is then compared with the actual throughput R. We collected $\tilde{p}$ and $\tilde{T}$ in order to evaluate how the corresponding metrics change due to the target flow, and also to quantify the prediction error if it was possible to estimate $\tilde{p}$ and $\tilde{T}$ before the target flow.

We define the *relative prediction error* E of an individual measurement epoch as

$$E = \frac{\hat{R} - R}{\min(\hat{R}, R)} \tag{4}$$

Notice that the denominator $\min(\hat{R}, R)$ gives E the property that overestimation or underestimation by the same factor $w > 1$, i.e., $\hat{R}$=wR for the former and $\hat{R}$=R/w for the latter, yields the same relative error $w - 1$ (in absolute value).

To report a single accuracy figure for n measurements in a time series (specifically, for all 150 epochs of a trace), we use the *Root Mean Square Relative Error (RMSRE)* statistic, defined as

$$\text{RMSRE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} E_i^2} \tag{5}$$

where E_i is the relative error of measurement i.

4.2 Results

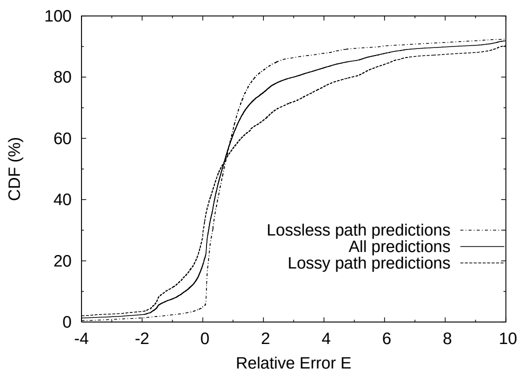


Figure 2: CDF of E for all predictions, for predictions in lossy paths, and for predictions in lossless paths.

Prediction error in lossy and lossless paths: Figure 2 shows the CDF of E for all measurements, across all traces and paths. It also shows separately the CDFs of E for the subset of lossy path predictions (based on the PFTK model) and for the subset of lossless path predictions (based on the avail-bw estimate $\hat{A}$)[1]. Let us first focus on the "all predictions" curve. Notice that *for roughly 40% of all measurements, the prediction is an overestimation by more than a factor of two ($E \geq 1$)*. In fact, the overestimation errors are larger than an order of magnitude ($E \geq 9$) for almost 10% of the measurements. The underestimation errors are much less dramatic and common, with only 10% of the measurements suffering from an underestimation by more than a factor of two ($E < -1$).

In the case of lossless paths, underestimation errors occur very rarely, while the overestimation errors are considerably lower and less common than in lossy paths. The reason is that in lossless paths, our FB predictor does not rely on the erroneous RTT and loss rate estimates prior to the target flow. The remaining errors can be attributed to the differences between TCP throughput and avail-bw, discussed in § 3.4. The fact that, in lossless paths, overestimation is the only major type of prediction error implies that either pathload overestimates the path's avail-bw, or that TCP cannot saturate the avail-bw in its path due to random losses or insufficient buffering at the bottleneck link.

[1]For W=1MB, we have that $\hat{A} < W/T$ in all paths.

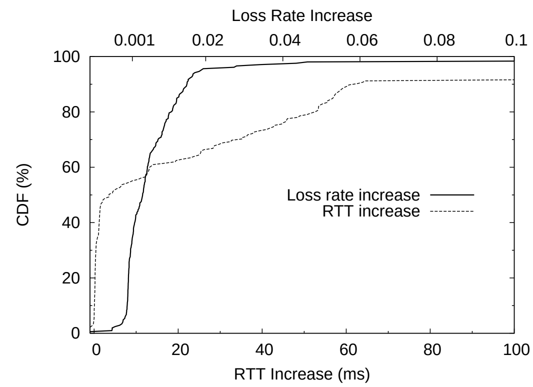


Figure 3: CDF of RTT and loss rate increase due to target flow.

RTT and loss rate increases due to target flow: Returning to the case of lossy paths, the fact that overestimation is much more dramatic than underestimation illustrates the dominance of the issue discussed in § 3.2, namely $\hat{T} < T$ and $\hat{p} < p$. Figure 3 shows the distributions of the increases in RTT and in loss rate after the start of the target flow. The increases were measured as $\tilde{T} - \hat{T}$ and $\tilde{p} - \hat{p}$ respectively (recall that $\tilde{T}$ and $\tilde{p}$ are estimates of T and p during the target flow). Notice that in about 50% of the measurements, the RTT did not increase significantly. In 40% of the measurements, however, the target flow caused an RTT increase between 5ms and 60ms. In 10% of the measurements the RTT increase was higher than 100ms, probably due to congested low-capacity links. The loss rate, on the other hand, increased by 0.1% to 2% in almost all measurements. Although this loss rate increase may appear small in magnitude, recall that TCP throughput is inversely proportional to the square-root of the loss rate. For example, an increase of the loss rate from 0.1% to 1% can cause a throughput overestimation by a factor of about 3.2.

Errors due to periodic RTT and loss rate sampling: An interesting hypothetical question is the following: *how accurate would FB prediction be, if we had ping-based estimates of the path's RTT $\tilde{T}$ and of loss rate $\tilde{p}$ during the target flow?* The answer to this question would allow us to examine the magnitude of the prediction errors due to the differences between periodic probing and TCP sampling (discussed in § 3.3). Figure 4 shows the CDF of the FB prediction error when we feed in (3) the ping-based RTT $\tilde{T}$ and loss-rate $\tilde{p}$ during the target flow. The CDF refers only to lossy paths. Note that using $\tilde{T}$ and $\tilde{p}$ makes the relative error significantly lower than using $\hat{T}$ and $\hat{p}$ ($-3 < E < 3$ for about 80% of the predictions). Also, overestimation and underestimation become equally likely (the CDF of E is practically symmetric). Despite the benefits of knowing $\tilde{T}$ and $\tilde{p}$, the prediction errors are still significant, however: more than half of the prediction errors are still larger than a factor of two. The remaining prediction errors can be attributed to the sampling differences discussed in § 3.3. In terms of

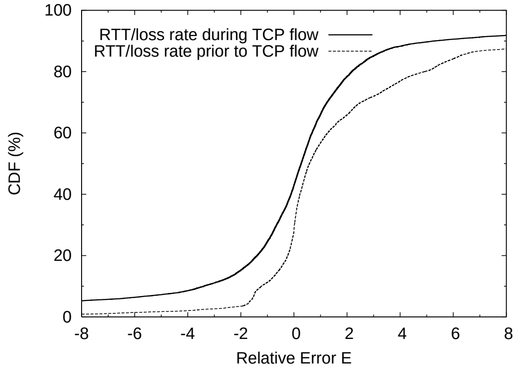


Figure 4: Prediction errors using $\tilde{T}$ and $\tilde{p}$ (RTT and loss-rate during the target flow) and using $\hat{T}$ and $\hat{p}$ (RTT and loss-rate prior to the target flow).

our notation, they are due to the differences between $\tilde{T}$ and T, and between $\tilde{p}$ and p.

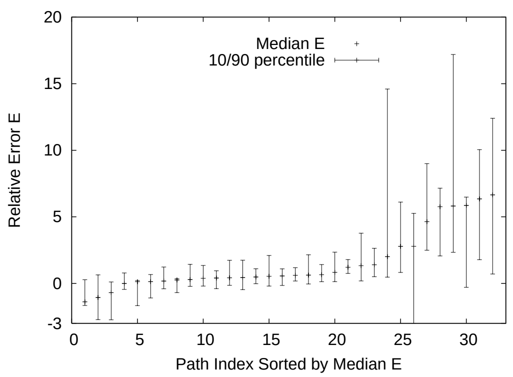


Figure 5: Variation of the prediction error across different paths.

Variation of prediction error across different paths and traces: Figure 5 shows the median, as well as the 10/90-th percentiles, of the relative prediction error on a per path basis (recall that we have 7×150 measurements from each path). There are three paths that we did not include in this graph because they have excessive prediction errors. With the exception of 4-5 paths that mostly give smaller underestimation errors, the rest of the paths give mainly overestimation errors. Another interesting point is that *different paths exhibit widely different predictability.* About 10 out of the 35 paths have much larger prediction errors as well as wider error ranges than the rest of the paths, extending up to E=10 or higher. This implies that, not only is it hard to predict TCP throughput with an FB method, but also it is hard to bound the prediction error that should be anticipated.

Figure 5 raises the following question: which paths have the largest prediction errors? Figure 6 is a scatter plot that shows the relation between the actual throughput R of each transfer and the corresponding prediction error E. Clearly, most of the large overestimation errors occur in transfers that have very small throughput. Specifically, 42% of the samples with $R \leq 0.5$Mbps have $E > 10$, compared to 0.2% for samples with $R \geq 0.5$Mbps.[2]

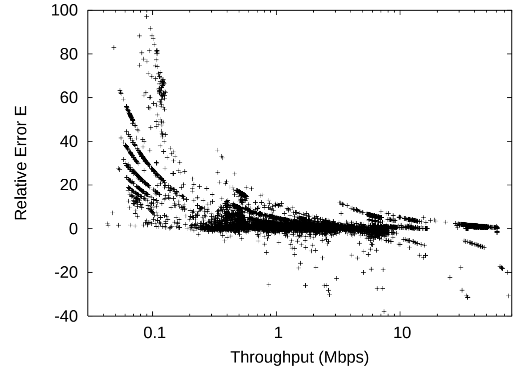


Figure 6: Actual throughput vs. prediction error.

Further analysis of the 10 paths with the largest median prediction errors in Figure 5 reveals that 2 of them are Europe-to-US paths, while the rest are within the US. 77% of the predictions for these paths are based on the PFTK model. This percentage is larger than that among all paths (56%). This implies that very large prediction errors are more likely in lossy paths. For most of the PFTK-based predictions in these 10 paths, the loss rate increases significantly after the target flow starts, while the RTT does not show a large increase. These observations agree with the hypothesis that the bottleneck link was already congested before the target transfer. For most of the predictions based on avail-bw, on the other hand, the loss rate remains negligible during the flow, while the RTT increases slightly after the flow starts. We do not know whether the errors in those cases are due to avail-bw overestimation, or due to bursty losses experienced by TCP but not by our periodic probes.

We also found that, with the exception of a single path, the variation of E across different traces of the same path is not significantly different. Consequently, the "time-of-day", which differentiates traces of the same path, does not seem to be a major factor in determining the FB prediction error.

Predictability of window-limited flows: Another interesting question is whether the FB predictor would be more accurate for window-limited flows (i.e., $W/\hat{T} < \hat{A}$), given that those flows do not attempt to saturate the network path. To answer this question, we extended each epoch with another IPerf TCP transfer with W=20KB. We verified that this transfer was window-limited on 18 of the 35 paths, and the ratio $W/(\hat{T}\hat{A})$ varied between 0.02 to 0.81. Figure 7 compares the RMSRE between the transfers with a large maximum window (W=1MB) and a small maximum window (W=20KB). Note the log-scale of the Y-axis. In all paths, the prediction error of window-limited flows was lower, often by a large factor. In particular, 14 out of the 18 paths have an RMSRE that is less than 1.0 for window-limited flows. We anticipate that for many applications a TCP throughput prediction that is accurate within a factor

[2]The exponentially decreasing patterns shown in Figure 6 are caused by sets of almost equal predictions (due to similar RTT and loss rate estimates) for individual paths, while the corresponding throughput measurements varied uniformly in a certain range.

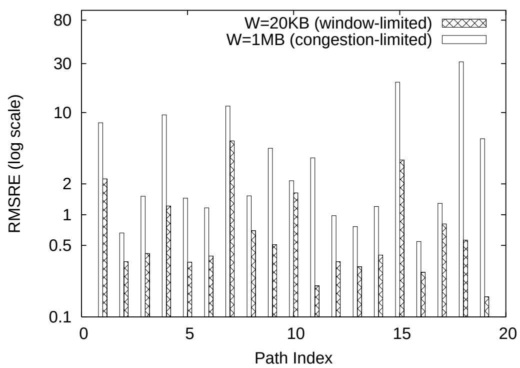


Figure 7: Prediction accuracy for window-limited vs. congestion-limited flows.

of two would be acceptable. Consequently, if predictability is more important than throughput maximization, the TCP flow should have a limited advertised window such that it does not attempt to saturate the underlying avail-bw.

4.3 Summary

The results of this section showed that FB prediction can be significantly inaccurate, mostly in congested paths and when the target flow attempts to saturate the underlying avail-bw. The major cause of prediction errors is that the RTT and/or loss rate before the transfer are significantly different than while the transfer is in progress. We note again that this cause of prediction errors is not specific to the PFTK formula. So, it is unlikely that other TCP throughput models would have produced more accurate FB predictions. Other important causes of prediction errors are the difference between periodic and TCP sampling of the RTT and loss rate processes, and the difference between TCP throughput and avail-bw.

5. HISTORY-BASED PREDICTION

A fundamentally different approach to predicting the TCP throughput of a large transfer is to use throughput measurements of previous transfers in the same path. This *History-Based* (HB) prediction method is similar to traditional time series forecasting, where past samples of an unknown random process are used to predict the value of the process in the future. The HB approach is possible in applications where large TCP transfers are performed repeatedly over the same path.

In this section, we first introduce three families of simple linear predictors (Moving Average, Exponential Weighted Moving Average, and non-seasonal Holt-Winters). We do not examine more complex linear predictors such as ARMA or ARIMA because selecting their order and linear coefficients requires a large number of past measurements [13]; instead, we expect that applications will have to perform TCP throughput HB prediction based on a limited number of past transfers (say 10-20). We then show that two distinct time series "pathologies", namely *outliers* and *level shifts*, can have a major impact on the prediction error, and propose simple heuristics that can deal with these pathologies effectively.

5.1 Linear Predictors

Moving Average (MA): Given a time series X, the one-step n-order MA (n-*MA*) predictor is:

$$\hat{X}_{i+1} = \frac{1}{n} \sum_{k=i-n+1}^{i} X_k$$

where $\hat{X}_i$ is the predicted value and X_i is the actual (observed) value at time i. If n is too small, the predictor cannot smooth out the noise in the underlying measurements. On the other hand, if n is too large the predictor cannot aptly adapt to non-stationarities (e.g., level shifts due to load variations or routing changes).

Exponentially Weighted Moving Average (EWMA): The one-step EWMA predictor is

$$\hat{X}_{i+1} = \alpha X_i + (1-\alpha)\hat{X}_i$$

where α is the weight of the last measurement (0< α <1). Similar to the MA predictor, a higher α cannot smooth out the measurement noise, while a lower α is slow in adapting to changes in the time series.

Holt-Winters (HW): The non-seasonal Holt-Winters predictor is a variation of EWMA that attempts to capture the *trend* in the underlying time series, if such a trend exists.This predictor is more appropriate than EWMA for non-stationary processes, especially if the latter exhibit a linear trend. A non-seasonal HW predictor maintains a separate smoothing component $\hat{X}_i^s$ and a trend component $\hat{X}_i^t$, and it depends on two parameters α and β, both in $(0, 1)$. Specifically, the predicted value at time i is

$$\hat{X}_i^f = \hat{X}_i^s + \hat{X}_i^t$$

where

$$\hat{X}_{i+1}^s = \alpha X_i + (1-\alpha)\hat{X}_i^f$$

and

$$\hat{X}_{i+1}^t = \beta(\hat{X}_i^s - \hat{X}_{i-1}^s) + (1-\beta)\hat{X}_{i-1}^t.$$

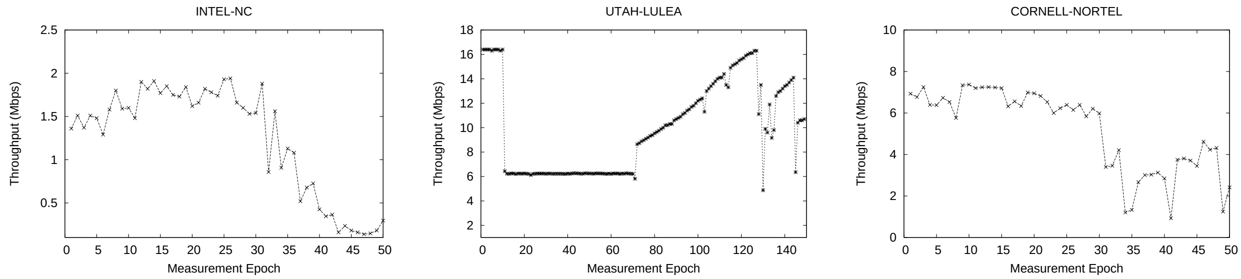
The initial values of $\hat{X}^s$ and $\hat{X}^t$ are X_0 and $X_1 - X_0$, respectively, assuming that the time series starts at i=0.

5.2 Detection of Level Shifts and Outliers

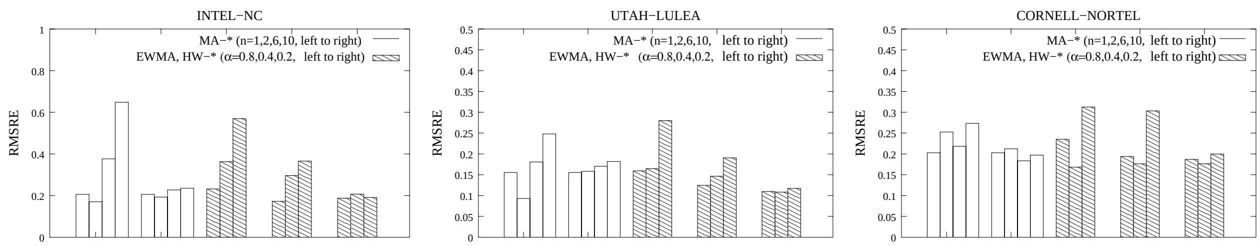
While experimenting with various predictors, we found that the largest prediction errors are often caused by level shifts and outliers in the observed time series. Furthermore, if we manage to avoid these two characteristics in the throughput time series, the exact choice of the predictor, or of its parameters, does not make a significant difference.

A level shift is a type of non-stationarity, and it causes a significant and typically sudden change in the mean of the observed time series. An outlier is a measurement that is significantly different, beyond the typical level of statistical variations, relative to nearby measurements. Both outliers and level shifts have been studied extensively in the theory of forecasting [17]. In Figures 8(a), 8(b) and 8(c) we show examples of traces that exhibit both outliers and level shifts, observed in our TCP throughput measurements. One way to deal with level shifts, after they are detected, is to restart the predictor, ignoring all previous history. Outliers, on the other hand, can be just ignored.

We next describe simple heuristics to detect level shifts and outliers. Suppose that $\{X_1, ...X_n\}$ is the sequence of



(a) Level Shift (b) Trend, Level Shift, & Outliers (c) Level Shift & Outliers



(d) Prediction error for Trace (a) (e) Prediction error for Trace (b) (f) Prediction error for Trace (c)

Figure 8: Examples of TCP throughput traces and the prediction errors (RMSRE) with various predictors.

past measurements, ignoring outliers, where X_1 is the first measurement after the last detected level shift. We determine that the measurement X_k is an increasing (decreasing) level shift if it satisfies the following three conditions:
1. The measurements $\{X_1, ...X_{k-1}\}$ are all lower (higher) than the measurements $\{X_k, ...X_n\}$,
2. The median of $\{X_1, ...X_{k-1}\}$ is lower (higher) than the median of $\{X_k, ...X_n\}$ by more than a relative difference χ,
3. $k + 2 \leq n$.

The last condition aims to avoid misinterpreting an outlier as a level shift. Upon the detection of a level shift, we ignore all measurements prior to X_k and restart the predictor from X_k. On the other hand, a measurement X_k (with $k < n$) is considered an outlier if it differs from the median of the measurements in $\{X_1, ...X_n\}$ by more than a relative difference of ψ. Outliers are discarded from the history of previous measurements.

Figures 8(d), 8(e) and 8(f) show the RMSRE for the three sample traces with five different predictors: MA, MA-LSO, EWMA, HW, and HW-LSO. The *LSO* acronym is used when we use the previous heuristics for the detection of Level Shifts and Outliers. For the MA and MA-LSO predictors, we show results for four different values of n (note the Figure's legend). For the EWMA and HW predictors, we show results for three values of α. We observed that, at least for our datasets, the RMSRE does not strongly depend on β, χ and ψ. We found empirically that the following values perform reasonably well, in terms of minimizing the RMSRE, at least in our datasets: β=0.2, χ=0.3, and ψ=0.4. On the other hand, the parameters n and α could play a major role in the prediction accuracy when the LSO heuristic is *not* used. The LSO heuristic decreases the prediction error significantly, and makes the predictors more robust to the selection of n or α. The difference between the accuracy of MA-LSO and HW-LSO is not major, although the latter tends to perform slightly better. More results for the HB prediction accuracy is given in the next section.

6. HB PREDICTION ACCURACY

In this section, we apply the HB predictors of the previous section to the measurements described in § 4. Our objective is to investigate the overall HB prediction accuracy, compare the most promising HB predictors that we experimented with, and to examine how this prediction accuracy varies in different paths and with different transfer frequencies.

6.1 Results

Accuracy of HB predictors: Figure 9 summarizes the prediction error (in terms of RMSRE) for some MA and HW predictors. The EWMA predictor performs similarly to HW. Without LSO, the n-MA predictors perform very similarly when n <20 (we do not show all of them), except the trivial case of n=1 that performs worse. With LSO, there is a significant reduction in the RMSRE of MA predictors. For HW predictors, α=0.8 (0.8-HW) performs close to the optimal for our dataset, and we use this value hereafter. The HW predictor is also significantly improved with LSO. A comparison of MA-LSO (with n=10) and HW-LSO shows that the accuracy of the latter is only slightly better. This is an indication that not many of our traces exhibit linear trends.

Comparison of FB and HB predictors: Even though these two classes of predictors are complementary, in some

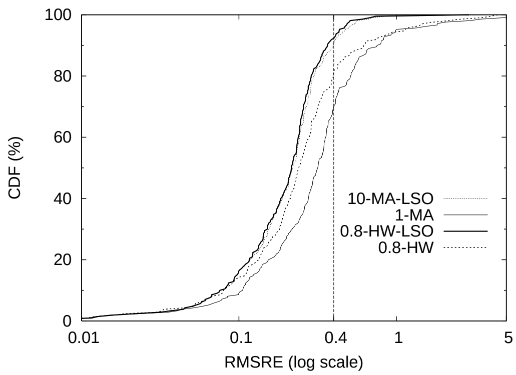


Figure 9: MA and HW prediction errors.

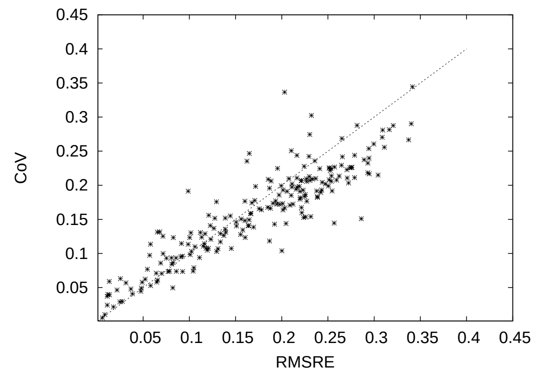


Figure 10: Prediction error vs. CoV.

cases it may be possible to use either FB or HB predictor. Compared to FB predictors, HB predictors give dramatically better accuracy. Specifically, HB predictors give RMSRE less than 0.4 for about 90% of the traces. The same RMSRE percentile for the FB predictor is 20, while the median RMSRE is about 2 (figure for the RMSRE of FB is not shown due to space constraint). One may argue that this comparison is not fair for FB prediction, since the latter is applicable without any knowledge of previous TCP transfer throughput measurements. If it is possible to collect and use such historical data, however, this comparison shows that HB prediction should be preferred to FB prediction.

RMSRE vs. CoV of throughput measurements: We are interested in the relation between the prediction RMSRE for a given trace and the Coefficient of Variation (CoV) of the corresponding TCP throughput time series.[3] The reason for this comparison will become clear in the following section, where we use the CoV of TCP throughput measurements as an indirect measure of the *predictability* of a path. To calculate the CoV of a trace, we isolate stationary periods based on the detected level shifts and exclude outliers. We then calculate the weighted average of the CoVs for different periods (with the weight of each period being the number of corresponding measurements). In the RMSRE calculations, we also exclude measurements that were identified as outliers. Figure 10 shows a scatter plot for the CoV and RMSRE for each trace, using the HW-LSO predictor. Note the strong correlation between the two metrics. Their correlation coefficient is 0.91. We can thus assume, at least as a first-order approximation, that *the RMSRE prediction error with HW-LSO is approximately equal to the CoV of the corresponding time series*, at least in the datasets we experimented with.

Variations in path predictability: Figure 11 provides close-up views of the accuracy of several predictors in 12 sample paths. We classify these paths into four representative classes (described in the figure's caption), based on the average prediction error as well as the variation of the error across different traces in the same path. Each subfigure represents a specific path, with the X-axis numbers indicating different traces. For each trace, successive bars show the RMSRE with 1-MA, 10-MA, HW, and HW-LSO, from left to right. As previously noted, *the HW-LSO predictor is almost always the best in terms of RMSRE*. A more important observation from these graphs, however, is that *there are major differences in the prediction error between different paths*. Some paths have quite low RMSRE and they are fairly predictable, others have larger RMSRE but the RMSRE is quite stable (predictable errors), while others have either large RMSRE variations (unpredictable errors), or high RMSRE (unpredictable throughput). Unlike FB predictions (see Figure 6), we do not observe a significant correlation between the actual throughput and the HB prediction accuracy. What causes different paths to behave so differently in terms of their throughput predictability? We focus on this question in the next section.

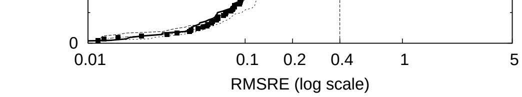


Figure 12: Prediction error with different TCP transfer periods.

The effect of the target flow frequency: All previous results are based on periodic TCP transfers, performed approximately every 3 minutes. We expect the prediction accuracy to depend on this "transfer period". A time series with a larger period spans a wider history horizon, and so route changes or major load variations become more likely. To see how the measurement period affects the prediction error, we down-sample the original traces at different frequencies. We then apply the HW-LSO predictor to the down-sampled traces, and calculate the RMSRE for transfer periods of 6, 24, and 45 minutes. Figure 12 shows the results.

As we would expect, *the prediction accuracy degrades as*

[3] CoV is the ratio of the standard deviation to the mean.

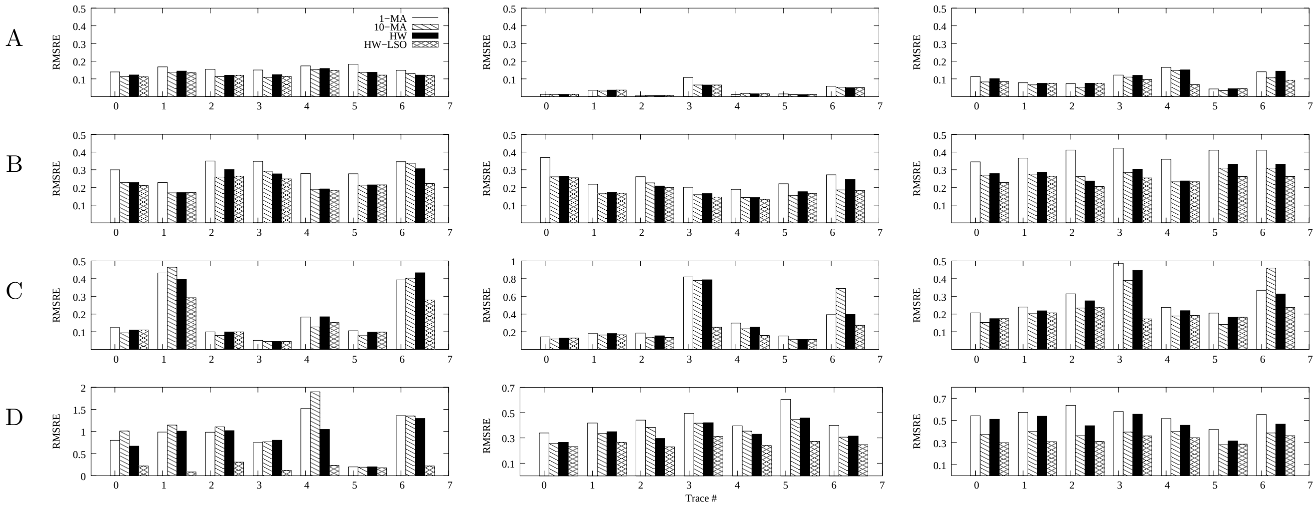


Figure 11: A: Predictable paths (low RMSRE), B: Paths with small and predictable errors (stable RMSRE), C: Paths with small but unpredictable errors (varying RMSRE), D: Unpredictable paths (high RMSRE, notice the different Y-axis ranges) .

we increase the measurement period. Fortunately, though, the prediction errors remain reasonable even with the largest measurement period. Specifically, with the 45-min period, 65% of the traces have an RMSRE below 0.4. At the 90-th percentile of the traces the RMSRE is less than 0.4 with the 3-min period, and less than 1.0 with the 45-min period. This is an encouraging result, as it implies that *HB prediction is fairly accurate even when it relies on sporadic previous TCP transfers, every few minutes, on the given paths.* Of course we emphasize that this conclusion is based on our datasets; it is possible that other Internet paths have significantly different stationarity characteristics.

6.2 Summary

This section has evaluated the accuracy of HB prediction with respect to several factors, some of which have not been examined before. Specifically, we have shown that:

1. Even a limited history of sporadic TCP transfers is often sufficient to achieve a fairly good prediction accuracy.
2. Simple heuristics to detect outliers and level shifts can significantly reduce the number of large prediction errors.
3. If HB prediction is feasible, i.e., if there is a short history of recent TCP transfers in the same path, HB prediction is much more accurate than FB prediction.
4. Different paths can exhibit distinct patterns of prediction accuracy. Consequently, even with the same prediction algorithm and available history, the resulting accuracy can be significantly different from path to path.
5. Similarly with FB prediction, the HB prediction errors are lower for window-limited transfers. Those results are not shown here due to space constraints.

7. TWO PREDICTABILITY FACTORS

The empirical results of the previous section raise the following question: *what makes TCP throughput much less predictable in some paths than in others?* In this section, we identify two major factors that affect the accuracy of HB prediction in a path: load and the degree of multiplexing. We rely on simple queueing models that provide a framework for reasoning about the relationship between TCP throughput predictability and these two factors.

First, we focus on the connection between the relative prediction error and the Coefficient of Variation (CoV) of a given time series. Consider a second-order stationary time series X with mean μ_X, variance σ_X^2, and covariance $\gamma_X(k)$. According to the Yule-Walker forecasting model [13], an autoregressive one-step predictor based on the n most recent samples of X has the following prediction error variance:

$$\text{Var}[e_n] = \text{Var}[X_{n+1} - \hat{X}_{n+1}] = \sigma_X^2 - \sum_{k=1}^{n} a_{X,n}(k)\gamma_X(k)$$

where X_i and $\hat{X}_i$ are the actual and predicted values of X, respectively, at time i. $\{a_{X,n}(i), i = 1, \ldots n\}$ are the autoregressive coefficients of X that minimize the mean square prediction error. The corresponding relative prediction error, in terms of the Normalized Root Mean Square Error (NRMSE) [4] is given by:

$$\frac{\sqrt{\text{Var}[e_n]}}{\mu_X} = \frac{\sqrt{E[e_n^2]}}{\mu_X} = \sqrt{\text{CoV}_X^2 - \frac{\sum_{k=1}^{n} a_{X,n}(k)\gamma_X(k)}{\mu_X^2}}, \tag{6}$$

where $\text{CoV}_X{=}\sigma_X/\mu_X$. The key point here is that *the relative prediction error increases with the CoV of the underlying time series.* Also, if the time series has a weak correlation structure then the relative prediction error is approximately equal to the time series CoV,

$$\text{if } \gamma_X(k) \approx 0, \text{ then NRMSE} = \frac{\sqrt{E[e_n^2]}}{\mu_X} \approx \text{CoV}_X \tag{7}$$

Now, recall the observation from Figure 10: the RMSRE with the HW-LSO predictor and the CoV of the corresponding time series are approximately equal. Consequently, in

[4]Notice that although NRMSE is not exactly the same as RMSRE, they are reasonably close as long as μ_x does not vary significantly. This is true for most of our traces.

the following we are interested in the effects of load and degree of multiplexing on the CoV of the TCP throughput time series, rather than in the effect of these factors on the RMSRE or the NRMSE.

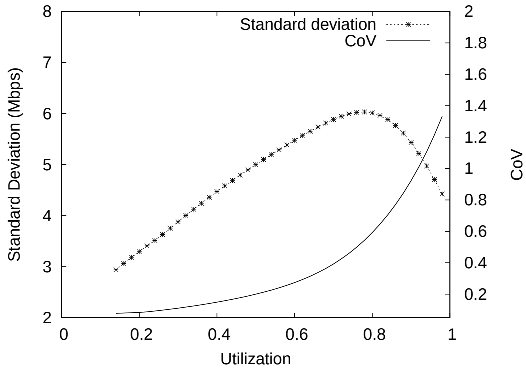


Figure 13: Gaussian process.

7.1 Effect of Load

Consider a link of capacity C, modeling the bottleneck of a path. We next examine the effect of that link's load conditions through two different models: first, an Independent and Identically Distributed (IID) process for the aggregate traffic at a bufferless server, and second, a Poisson process of IID session arrivals at a Processor Sharing server.

7.1.1 IID arrival process at bufferless server

Suppose that the *arriving* traffic rate at a given time scale T can be modeled as an IID process Y. Without loss of generality, T=1 time unit. Let Z be the *observed* traffic rate at the output of the link at the same time scale. For a bufferless link, the observed rate process is given by

$$Z = \begin{cases} Y & \text{if } Y < C \\ C & \text{if } Y \geq C \end{cases} \tag{8}$$

and so the probability distribution function of Z can be obtained from that of Y. The avail-bw is given by $A{=}C-Z$, and its CoV is

$$\text{CoV}(A) = \frac{\sqrt{\text{Var}[Z]}}{C - E[Z]}$$

If we assume that the TCP throughput is, as a first-order approximation, equal to the avail-bw, then the previous expression also gives the TCP throughput CoV.

We used Mathematica to derive CoV(A) for two offered load processes Y: a Gaussian process and a Poisson process. The resulting CoV(A), as well as the std-deviation of A, are shown for the Gaussian process in Figure 13, as a function of the link utilization $\rho{=}(C-A)/C$. The key observation is that *the CoV of the avail-bw increases with the link utilization*. If the TCP throughput follows the variations of the avail-bw, then based on (7) we should expect *a higher relative prediction error under heavier load conditions*.

As an interesting side-note, note that the standard deviation reaches a maximum as ρ increases, and then it drops. The reason for that drop is that, in heavy-load conditions, the link is almost always utilized and so there is little *absolute* variation in the avail-bw. This point has been studied in more depth by Tian et al. in [21]. In relative terms, however, the variability of A increases monotonically with ρ, as shown by the CoV curve.

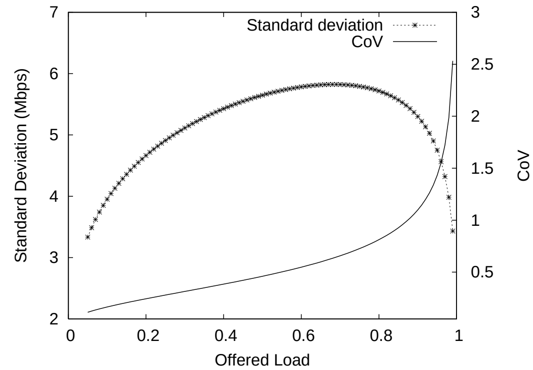


Figure 14: Processor Sharing model.

7.1.2 Processor Sharing model with Poisson session arrivals

The previous model does not capture what happens at a congested link, in which the avail-bw is zero. In this paragraph, we model the traffic as a stream of IID sessions arriving at a link, based on a Poisson process with average rate λ. The mean size of the sessions is θ. The normalized offered load is $\rho = (\lambda\theta)/C$. Furthermore, we model the link as a Processor Sharing server, meaning that if there are N sessions in the link then their instantaneous service rate is $r(N){=}C/N$. Since the avail-bw is zero when the link is not idle, this a more appropriate model for a congested link [7]. An arriving session, modeling the target flow, will obtain the same throughput $r(N)$ as any other active flow. So, in this model, we are not interested in the CoV of the avail-bw, but in the CoV of the per-flow throughput $r(N)$.

The probability distribution for the number of active flows N in the above Processor Sharing model is given by

$$\pi(N) = \rho^N(1-\rho)$$

We again use Mathematica to derive the CoV of the target flow's throughput $r(N)$:

$$\text{CoV}[r(N)] = \frac{(1-\rho)\log(1-\rho)^2 + \rho \cdot L(2,\rho)}{(\rho-1)\log(1-\rho)^2}$$

where $L(n,x) = \sum_{k=1}^{\infty} \frac{x^k}{k^n}$. Figure 14 shows the standard deviation and CoV of $r(N)$ as a function of the offered load ρ. The main observation is the same as in the IID traffic model: *the CoV of a flow's throughput increases with the offered load ρ, implying that we should expect a higher relative prediction error under heavier load conditions.*

7.2 Effect of Degree of Multiplexing

The conventional wisdom is that network traffic is smoother in links with a higher degree of multiplexing, i.e., with a larger number of simultaneously active flows. Using a simple queuing model, we aim to better understand this insight, and the conditions under which it is valid.

Consider again a model of Poisson session arrivals. Instead of the Processor Sharing model (which leaves no avail-bw), suppose that sessions are rate limited, and for simplicity, the rate for each session is constant and equal to r. The number of sessions N on the link follows a Poisson distribution with mean and variance $E[N] = \text{Var}[N] = (\lambda\theta)/r$ [7].

The utilized link capacity at any point in time is $Y = Nr$, with mean $E[Y] = rE[N] = \lambda\theta = \rho C$, and variance Var[$Y$] = $r^2\text{Var}[N]$. So, the CoV of the avail-bw is

$$\text{CoV}[A] = \text{CoV}[C - Y] = \frac{1}{\sqrt{E[N]}} \frac{\rho C}{C(1-\rho)} \quad (9)$$

Suppose that we keep the utilization ρ constant, but decrease the session service rate r so that the average number of sessions $E[N]$ increases. Equation (9) shows that the CoV of A decreases with the square root of $E[N]$. This confirms that *we should expect a lower relative prediction error as the number of competing flows on the link increases, but only when the utilization remains constant*.

7.3 Summary

This section relied on simple queueing models to confirm the following insights: (1) the relative prediction error increases with the CoV of the underlying time series, (2) the CoV of the avail-bw process in a non-congested link, or the CoV of a flow's throughput in a congested link, increases with the offered load on that link, (3) the CoV of the avail-bw process decreases with the number of competing flows on the link, if the utilization remains constant.

Obviously, our models are based on quite restrictive assumptions and they do not consider the idiosyncrasies of TCP. In particular, the previous analysis assumed that the TCP throughput follows the variability of the avail-bw at the bottleneck link of its path. This assumption is obviously not true in short time scales (less than a few RTTs), and so the previous conclusions may not be true for short TCP flows.

8. CONCLUSIONS

This paper investigated two classes of throughput predictors for large TCP transfers. FB prediction is an attractive option, given that it does not require intrusive measurements or any prior TCP transfers. We demonstrated however that it can be inaccurate, especially when the transfer attempts to saturate the path, and we explained the main reasons behind these errors. HB prediction, on the other hand, is quite accurate but is feasible only when there is a history of previous TCP transfers in the same path. Although the accuracy of HB prediction does not depend so much on the specific predictor, it does depend on the transfer's maximum congestion window size and on the underlying path. We explained the path dependency based on two factors: the load and the degree of multiplexing in the bottleneck link.

In future work, it would be interesting to examine hybrid predictors, which rely on TCP models as well as on recent history. Another direction would be to develop TCP throughput models that are specifically designed for prediction and that take as input various estimates of the path's load, buffering, and cross traffic nature. In terms of HB prediction, more complex predictors (such as ARIMA models) can be also evaluated, even though our measurements indicate that the prediction error is already quite low in most paths. In addition, efficient mechanisms to acquire or reuse throughput history, by monitoring transfers that flow between two networks rather than two hosts, for example, can also improve the practicality of HB prediction.

9. ACKNOWLEDGEMENTS

We are grateful to the MIT RON project members for providing us access at the RON network. We also thank the anonymous reviewers for their valuable suggestions.

10. REFERENCES

[1] Resilient Overlay Network (RON). http://nms.lcs.mit.edu/ron/, February 2005.

[2] A. Akella, J. Pang, A. Shaikh, B. Maggs, and S. Seshan. A Comparison of Overlay Routing and Multihoming Route Control. In *Proc. of ACM SIGCOMM*, 2004.

[3] D. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris. Resilient Overlay Networks. In *Proc. of ACM Symposium on Operating Systems Principles*, 2001.

[4] N. Cardwell, S.Savage, and T.Anderson. Modeling TCP Latency. In *Proc. of IEEE INFOCOM*, 2000.

[5] Y-H. Chu, S. G. Rao, S. Seshan, and H. Zhang. Enabling Conferencing Applications on the Internet using an Overlay Multicast Architecture. In *Proc. of ACM SIGCOMM*, 2001.

[6] S. Floyd, M. Handley, J. Padhye, and J. Widmer. Equation-Based Congestion Control for Unicast Applications. In *Proc. of ACM SIGCOMM*, 2000.

[7] S. Ben Fredj, T. Bonald, A. Proutiere, G. Regnie, and J. W. Roberts. Statistical Bandwidth Sharing: A Study of Congestion at Flow Level. In *Proc. of ACM SIGCOMM*, 2001.

[8] M. Goyal, R. Guerin, and R. Rajan. Predicting TCP Throughput From Non-invasive Network Sampling. In *Proc. of IEEE INFOCOM*, 2002.

[9] Iperf. http://dast.nlanr.net/Projects/Iperf/.

[10] M. Jain and C. Dovrolis. End-to-End Available Bandwidth: Measurement Methodology, Dynamics, and Relation with TCP Throughput. *IEEE/ACM Transactions on Networking*, 11(4):537–549, 2003.

[11] M. Mathis, J. Semke, and J. Madhavi. The Macroscopic Behavior of the TCP Congestion Avoidance Algorithm. *ACM Computer Communications Review*, 27(3):67–82, 1997.

[12] J. Padhye, V.Firoiu, D.Towsley, and J. Kurose. Modeling TCP Throughput: A Simple Model and its Empirical Validation. *IEEE/ACM Transactions on Networking*, 8(2):133–145, 2000.

[13] M. Pourahmadi. *Foundations of Time Series Analysis and Prediction Theory*. John Wiley and Sons, 2001.

[14] Y. Qiao, J. Skicewicz, and P. Dinda. An Empirical Study of the Multiscale Predictability of Network Traffic. In *IEEE Proc. of HPDC*, 2003.

[15] S. Ratnasamy, M. Handley, R. Karp, and S. Shenker. Topologically-Aware Overlay Construction and Server Selection. In *Proc. of IEEE INFOCOM*, 2002.

[16] V. Ribeiro, R. Riedi, R. Baraniuk, J. Navratil, and L. Cottrell. pathChirp: Efficient Available Bandwidth Estimation for Network Paths. In *Proc. of Passive and Active Measurements (PAM) workshop*, April 2003.

[17] R.S.Tsay. Outliers, Level Shifts, and Variance Changes in Time Series. *Journal of Forecasting*, 1988.

[18] A. Sang and S. Li. A Predictability Analysis of Network Traffic. *Computer Networks*, 39(4):329 – 345, 2002.

[19] B. Sikdar, S. Kalyanaraman, and K. S. Vastola. Analytic Models for the Latency and Steady-State Throughput of TCP Tahoe, Reno and SACK. *IEEE/ACM Transactions on Networking*, 11(6):959–971, 2003.

[20] M. Swany and R. Wolski. Multivariate Resource Performance Forecasting in the Network Weather Service. In *Proc. of Supercomputing*, 2002.

[21] X. Tian, J. Wu, and C. Ji. A Unified Framework for Understanding Network Traffic Using Independent Wavelet Models. In *Proc. of IEEE INFOCOM*, 2002.

[22] S. Vazhkudai, J. Schopf, and I. Foster. Predicting the Performance of Wide Area Data Transfers. In *Proc. of IEEE IPDPS*, 2002.

[23] Y. Zhang, N. Duffield, V. Paxson, and S. Shenker. On the Constancy of Internet Path Properties. In *Proc. of Internet Measurement Workshop*, 2001.