Prediction of TCP Throughput: Formula-based and History-based Methods

Paper ID: 118 Pages: 12

ABSTRACT

With the advent of overlay and peer-to-peer networks, Grid computing, and CDNs, network performance prediction becomes an essential task. Predicting the throughput of large TCP transfers, in particular, has attracted much attention. In this work, we first classify the existing prediction techniques into two categories: Formula-Based (FB) and History-Based (HB). Within each class, we then develop representative prediction algorithms that we 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 network path (e.g., RTT, loss rate, available bandwidth). Despite the established accuracy of these models, we show that FB schemes can unfortunately lead to large *prediction* errors. The main reason is that throughput models require knowledge of the path characteristics *during* the time the TCP flow is running, while FB predictors can only measure those characteristics *before* the flow starts. The resulting errors are acceptably small, however, if the TCP transfer is window-limited to the point that it does not saturate the underlying path. 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 HB prediction is quite attractive in the sense that even simple HB predictors, such as Moving Average and Holt-Winters, using a history of limited and sporadic samples, are very accurate. On the negative side, HB predictors are highly path-dependent. Using simple queueing models, we explain the cause of such path dependencies based on two key factors: the load on the path, and the degree of statistical multiplexing.

Keywords

Network measurements, TCP throughput, time series forecasting, performance evaluation

1. INTRODUCTION

With the advent of overlay and peer-to-peer networks [1, 7], Grid computing [10], and CDNs [18], performance prediction of network paths becomes an essential task. To name just a few of their applications, such predictions are used in route selection schemes for overlay and multihomed networks [1, 2, 13], dynamic server selection [25], and peer-to-peer parallel downloads [8].

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 much more subject to the randomness in the background traffic [11]. In this work we consider the throughput prediction of a bulk TCP transfer in a particular network path, *prior* to actually starting the flow. For many applications, such as server selection and overlay route selection, a throughput prediction is needed before the flow starts. Note that TCP throughput prediction is different than TCP throughput *estimation*. The latter is performed while the flow *is in progress* with the objective to estimate the TCP throughput or the TCP-Friendly rate of the flow. An example of a TCP throughput estimation scheme is TCP-Friendly Rate Control (TFRC) [9].

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 transfer size, maximum sender/receiver windows, various path characteristics (RTT, loss rate, available bandwidth, cross traffic congestion responsiveness, 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 can be highly intrusive because it saturates the underlying paths, and therefore is widely considered impractical. 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.

Recently, several low-overhead TCP throughput prediction schemes have been proposed in the literature [1, 14, 29, 31, 33]. In this paper, we first classify existing predictors into two broad and fundamentally different categories: *Formula-Based (FB)* and *History-Based (HB)*. We then develop specific FB and HB prediction algorithms, and evaluate their accuracy empirically, with measurements over the

RON testbed [1]. Note that our objective is not to compare FB and HB prediction schemes. 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, and provide insight regarding the factors that affect the predictability of TCP throughput.

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, available bandwidth). For instance, the throughput-optimizing routing component of RON follows the FB approach [1], predicting TCP throughput based on the "square-root" formula [20]. 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 [3, 6, 12, 19, 20, 21], differing in terms of complexity and accuracy, modeling assumptions, and TCP flavor.

The main advantage of FB prediction is that it does not require any history of previous TCP transfers. Also, FB prediction can be performed with relatively lightweight, non-intrusive network measurements of parameters such as RTT and loss rate. Unfortunately, however, we show that FB schemes can have large prediction errors and we reveal the major causes of these 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* [14]. We do observe however, on the positive side, that the prediction errors are much lower, and probably acceptable, if the TCP transfer is limited by the receiver advertised window to the point that it does not saturate its path.

HB approaches, on the other hand, 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. Expediently, this is the case with some applications of TCP throughput prediction, such as overlay networks, parallel downloading and Grid-computing [1, 8, 32].

Our most important result regarding HB prediction is that even simple linear HB predictors, such as Moving Average and Holt-Winters, are quite accurate. Furthermore, in agreement with previous work [31, 34], 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, to detect level shifts and restart the HB predictors. We next show, perhaps surprisingly, that even with a small set of previous transfers with intervals up to 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 an answer to 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? Using simple queueing models, 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 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 this paper is as follows. We summarize the related work in Section 2. In Section 3, we develop an FB predictor and highlight some important issues in that type of prediction. Section 4 presents measurement results for the accuracy of FB prediction. Section 5 introduces several existing HB predictors, and describes two simple techniques that can improve such predictors significantly. Section 6 presents measurement results for the accuracy of HB prediction. Section 7 focuses on two major factors that affect the throughput predictability: the load of the network path, and the degree of statistical multiplexing. We conclude in Section 8.

2. RELATED WORK

One of the motivations for previous work on TCP throughput modeling has been to predict the throughput of a transfer as a function of the underlying network characteristics [9, 20, 21]. 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 model by Padhye et al. [21], and they have proposed a way to derive p' from p [14]. Notice though, their work does not solve the more general problem of estimating the path characteristics during a flow from those observed prior to the flow.

Most of the HB predictors have been proposed previously in the context of Grid computing [29, 31, 32, 33]. One operational system is the Network Weather Service (NWS) project [33]. In NWS, throughput prediction is based on small (64Kbyte) TCP transfer probes with a limited socket buffer size (32Kbyte). Vazhkudai et al. use bulk TCP transfers (1Mbyte-1Gbyte) and a large socket buffer (1Mbyte), performed sporadically (1 minute-1 hour) [31]. 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 [34]. Their TCP throughput measurement uses 1Mbyte transfers performed every minute, with 200Kbyte 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.

Previous work on HB predictors did not distinguish between congestion-limited and advertised window-limited flows [29, 31, 32, 33, 34]. Also, they did not examine the impact of the TCP transfer frequency [29, 32, 33, 34], or they used only a small set of paths [31]. In addition, the previous work has not investigated the underlying path characteristics that can affect the accuracy of HB prediction.

A study by Qiao et al. has shown that the predictability of network traffic is highly path dependent [24]. Note, however, that models used to analyze the predictability of aggregate network traffic [4, 27] are not directly applicable to the predictability of TCP throughput.

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 [20]: $\tilde{R} = \frac{M}{T\sqrt{\frac{2bp}{3}}}$, where $\tilde{R}$ is the TCP throughput estimate, M is the flow's Maximum Segment Size, T is the flow's average RTT, b is the number of TCP segments per new ACK, and p is the loss rate that the flow experiences. This model is fairly accurate for bulk TCP transfers in which losses are recovered with Fast-Retransmit. Such analytical results have been very useful in *understanding* the relation between TCP throughput and certain key path characteristics, such as loss rate and RTT.

Another motivation for the research that led to these analytic models was the ability to *predict* the throughput of a TCP flow given estimates of the relevant path characteristics [6, 21]. For instance, the RON overlay network uses the square-root formula to select the route with the highest predicted throughput [1]. In this section, we first present a more complete TCP throughput formula, as well as a related FB predictor. Although several similar models exist in the literature (see [28] and the references therein), we emphasize that our conclusions regarding the limitations of FB prediction are not specific to the particular formula we consider. Then, we explain some important problems in applying FB prediction, the most crucial of which is that the relevant network parameters before the transfer starts can be significantly different than while the transfer is in progress.

3.1 A formula-based TCP throughput predictor

The TCP throughput formula that we consider is the well known "UMass model" of [21], which improves on the square-root formula and addresses retransmission timeout:

$$\tilde{R} = \min\left(\frac{M}{T\sqrt{\frac{2bp}{3}} + T_0 \min(1, \sqrt{\frac{3bp}{8}})p(1+32p^2)}, \frac{W}{T}\right) \quad (1)$$

where T_0 is the TCP retransmission timeout period, and W is the maximum window size (limited, for instance, by the socket buffer size at the sender or receiver). We emphasize that p and T, in the above equation, are the average loss rate and RTT *that this "target flow" experiences.* Notice that the loss rate p may be zero, in which case the flow is lossless and $\tilde{R}$ is given by the term W/T.

Suppose now that we want to apply (1) to TCP throughput prediction. The main problem is that we do not know, when predicting, the loss rate and RTT that the flow will experience during its lifetime. The obvious approach, which is often followed in practice (e.g., in overlay routing [1]), 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 (1). Suppose that $\hat{p}$ and $\hat{T}$ are the loss rate and RTT estimates based on measurements prior to the flow. 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 of the mathematical model that leads to (1).

A problem with the previous approach is that it does not apply to *lossless paths*, i.e., $\hat{p}$=0. In that case, $W/\hat{T}$ can be far from the realized throughput, especially if W is much larger than the bandwidth-delay product of the underlying path. One approach to deal with lossless paths is to predict the TCP throughput as the *available bandwidth (avail-bw)* $\hat{A}$ of the path prior to the TCP flow, when $\hat{A} < W/\hat{T}$. The avail-bw is the non-utilized part of the tight link's capacity[1], and it can be measured non-intrusively [17]. Although 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 [15, 17, 23], and certain overlay routing schemes have proposed using $\hat{A}$ for TCP throughput prediction in lossless paths [1]. 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 reasonable predictor; we refer to such flows as *window-limited.*

To summarize, the FB predictor that we consider is given by the following equation:

$$\hat{R} = \begin{cases} \min\left(\frac{M}{\hat{T}\sqrt{\frac{2b\hat{p}}{3}} + T_0 min(1, \sqrt{\frac{3b\hat{p}}{8}})\hat{p}(1+32\hat{p}^2)}, \frac{W}{\hat{T}}\right) & \text{if } \hat{p} > 0 \\ \min\left(\frac{W}{\hat{T}}, \hat{A}\right) & \text{if } \hat{p} = 0 \end{cases} \quad (2)$$

where $\hat{R}$ is the predicted throughput. In the following, we explain three limitations of the above predictor using basic insight and simple *ns2* simulation scenarios.

3.2 Errors due to the extra load of the target flow

Basic queueing theory tells us that an increase in the utilization of a queue (with non-periodic arrivals) increases the average queueing delay. Similarly, in a queue with a limited buffer, an increase in the utilization can cause a higher loss probability. The increases in both the queueing delays and the loss probability tend to be more significant when the utilization becomes high, close to 100%.

This basic fact can cause major errors in FB prediction. The reason is that the RTT $\hat{T}$ measured prior to the target flow will not reflect the queueing delay increase due to that transfer, and 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 if the target transfer increases the utilization of the tight link significantly. Note that the experimental validation of the UMass model, reported in [21], 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 in the context of prediction.

We use simple *ns*2 simulations to demonstrate how such prediction errors could take place. The simulations use a simple dumbbell topology with the tight link (capacity C, buffer space B) in the center and the TCP flows, both the cross traffic and a target flow, traversing the link. We change the configurations of the tight link and the cross traffic to

[1] Tight link is another term for the path's bottleneck w.r.t. avail-bw.

Path	C (Mbps)	B (pkts)	$\hat{T}$ (ms)	T (ms)	$\hat{R}$	R
1	20	200	16.7	23.4	13.3	9.2
2	50	200	20.1	20.1	10.6	10.6

(A)

Path	C(Mbps)	B (pkts)	$\hat{p}$ (%)	p (%)	$\hat{R}$	R
1	10	50	0.14	0.46	9.2	4.6
2	100	300	0.2	0.22	6.2	6.1

(B)

Table 1: (A) Increased RTT (B) Increased Loss Rate. (R and $\hat{R}$ measured in units of Mbps)

have different paths, on which we perform the same target TCP transfer (100Mbyte). Table 1-(A) gives an example of the discrepancy between T and $\hat{T}$. The target flow as well as a single cross traffic TCP flow are window-limited (W=30Kbyte) so that the tight link does not experience any packet losses (p=0). Note that the capacity of the tight link in Path-1 is lower than in Path-2, causing larger queueing delays in the former. As a result, in Path-1, T is significantly higher than $\hat{T}$, causing a prediction error. Note that based on $\hat{R}$, Path-1 appears better than Path-2, yet this contradicts the actual experience of the target flow. Similarly, Table 1-(B) exemplifies the discrepancy between p and $\hat{p}$. The cross traffic is a single TCP flow on Path-1, and 15 TCP flows on Path-2. None of the flows, including the target transfer, are window-limited. Here, there is a large increase in the loss rate in Path-1 due to the target flow. Although Path-1 has a higher predicted throughput than Path-2, the actual throughput is higher on Path-2.

3.3 Errors due to the TCP sampling behavior

Even if the target flow would not affect the path's RTT and loss rate, it is still hard to estimate these metrics as they are experienced by the flow. TCP reduces its packet transmission rate when it experiences losses, which means that it tends to sample the RTT and loss rate 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 constant-rate 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, a mathematical model for TCP throughput may be based on certain assumptions that affect the interpretation of parameters such as T or p. For instance, the UMass model assumes that when a packet is dropped, all the remaining packets in that "flight" are also dropped (a "congestion event"). As a result, the parameter p in (1) 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 focus points in [14].

Table 2 shows three different "loss rates", all obtained from the same simulation as in Table 1-(B). In this Table, p_p is a ping-based estimate of the loss rate measured with periodic probing packets (40 bytes every 100ms) during the target flow, p is the (unconditional) loss rate that the target flow experienced, and p' is the congestion event probability estimated from a detailed analysis of the $ns2$ packet trace. Notice the striking difference, more than an order of magnitude, between p_p and the other two metrics. Ping estimates a larger loss rate, due to its non-adaptive sampling behavior we mentioned. The difference between p and p' is also noticeable, although not major. Unfortunately, it is not known how to measure p' or p, prior to the start of the target TCP transfer. For this reason, the existing FB prediction schemes use ping-based loss rate estimates, which are also much simpler to obtain.

Path	p_p	p	p'
1	0.04	0.0046	0.0028
2	0.03	0.0022	0.0015

Table 2: Different loss rate estimates during a TCP flow.

3.4 Errors due to the difference between avail-bw and TCP throughput

As mentioned previously, when the loss rate estimate $\hat{p}$ is zero, it is reasonable to predict the throughput of a target flow based on the avail-bw $\hat{A}$ of the path prior to that flow, as long as the flow is not window-limited. These two metrics, however, can have some important differences, which have been investigated in [23]. Here we simply mention the main conclusions of that study.

First, whether a TCP flow can saturate the avail-bw of a path depends on the buffer space B at the tight link. 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 on the tight link is "congestion responsive", the target flow can capture more than $\hat{A}$, by receiving some of the bandwidth previously occupied by cross traffic flows. An example of congestion responsive cross traffic is TCP 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. Non-TCP traffic, on the other hand, is typically congestion unresponsive.

To summarize, the avail-bw $\hat{A}$ prior to the target flow can be either overestimation or 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 responsiveness 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 argues that FB prediction can be inaccurate under certain conditions. In this section, we show experimental results from Internet paths that confirm and quantify the inaccuracy of FB prediction. First, we describe the measurement set we use throughout this paper.

4.1 Overview of measurement methodology

We collected 245 measurement "traces" on 35 different paths on the RON network [1], with 7 traces collected on each path[2]. Each trace/experiment consists of 150 back-to-back "epochs". An epoch consists of an avail-bw measurement using Pathload [17], followed by a measurement of $\hat{p}$

[2]We prefer the RON testbed to PlanetLab, because the hosts on the latter are often too heavily loaded for accurate network measurement.

and $\hat{T}$ using a homespun ping utility that generates a 41-byte probing packet every 100ms, followed by a 50-second TCP transfer (target flow) generated by IPerf [16] (see Figure 1). RTT and loss rate estimates are also measured during the TCP transfer. The duration of each epoch is about 2-3 minutes. 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 (or epochs) were performed. Notice that the ping probes we use are very lightweight and should not bias the throughput measurements.

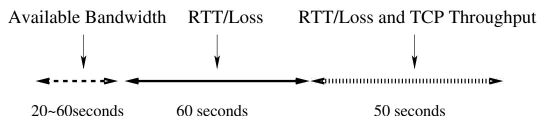


Figure 1: A measurement epoch.

IPerf allows us to directly control the maximum TCP window size W by limiting the socket buffer size. Unless otherwise noted, we used W=1Mbyte, which is large enough to saturate all the paths we experimented with and cause congestion (W=128Kbyte would be sufficient to saturate most paths). To examine the effect of W, we also performed the same measurements with W=20Kbyte, which, as will be shown later, limits the transfer to only a fraction of the avail-bw on many 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 loss rate $\tilde{p}$ and RTT $\tilde{T}$ *during* the transfer. The first three estimates are used in (2) 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.

We define the *relative prediction error* E of a measurement as

$$E = \frac{\hat{R} - R}{\min(\hat{R}, R)} \quad (3)$$

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

To report a single 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)*, defined as

$$\text{RMSRE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} E_i^2} \quad (4)$$

where E_i is the relative error of measurement i.

4.2 Results

Figure 2 shows the CDF of E for all measurements. It also shows separately the CDFs of E for the subset of predictions that is based on the UMass model versus the avail-bw estimate $\hat{A}$. Note that in roughly 40% of all measurements, the prediction is an overestimation by more than 100%. In fact, the overestimation errors are up to an order of magnitude for almost 10% of the measurements. The underestimation errors (E<0) are much less dramatic and common, especially when the prediction is based on $\hat{A}$.

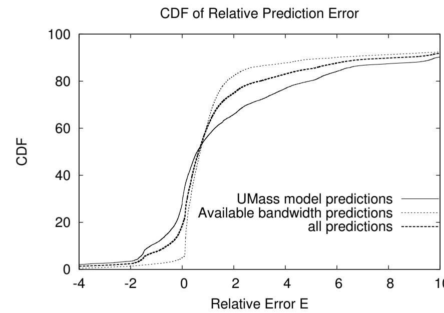


Figure 2: CDF of E.

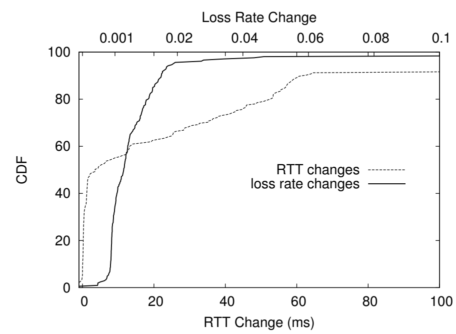


Figure 3: CDF of loss rate/RTT change.

The fact that overestimation occurs more often than underestimation indicates the dominance of the problem discussed in § 3.2, namely $\hat{T} < T$ and $\hat{p} < p$. Figure 3 shows the distribution of the increase 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). Note that in about 50% of the measurements, the RTT increased by more than 5ms. The loss rate increased by 0.1% to 2% in most measurements.

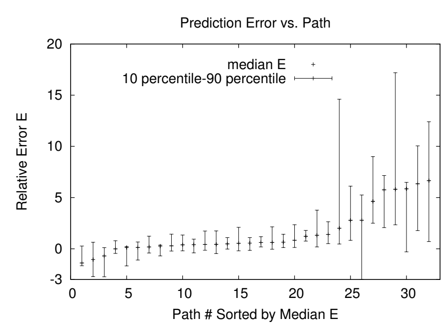


Figure 4: Prediction error variation by path.

Figure 4 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 on each path).

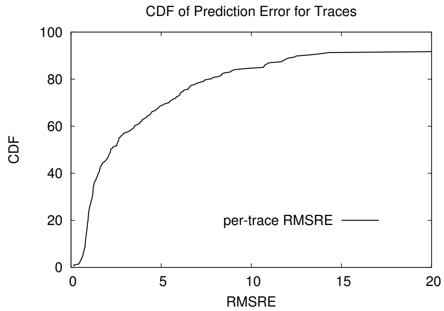


Figure 5: CDF of per-path RMSRE.

With the exception of 4-5 paths that mostly give underestimation errors, most paths give overestimation errors only. Another interesting point is that different paths exhibit very different predictability. About 10 out of the 35 paths have a much larger range of prediction error, extending up to E=10 or larger, indicating that not only the TCP throughput is hard to predict, but also it is hard to bound the prediction error that should be anticipated. We note that there are 3 paths that we did not include in this graph because they have excessive prediction errors.

Figure 5 shows the distribution of RMSRE for the FB predictor. Recall that we calculate an RMSRE for each trace (with 150 successive epochs per trace). About 70% of the traces have an RMSRE that is larger than 1.0.

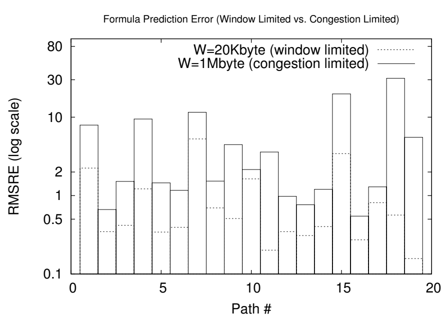


Figure 6: Prediction accuracy for window-limited vs. congestion-limited flows.

An interesting question is whether the FB predictor would be more accurate for window-limited flows (i.e., $W/\hat{T} < \hat{A}$), because the prediction in that case does not depend on a loss rate estimate and $\hat{T}$ would probably not be much different than T in a non-saturated path. To answer this question, we extend each measurement epoch with another IPerf TCP transfer with W=20Kbyte and perform one experiment on each path. 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 6 compares the RMSRE between the transfers with the large window (W=1Mbyte) and the small window (W=20Kbyte). Note the log-scale of the Y-axis. In all paths, the prediction error of window-limited flows was lower, often significantly. Specifically, in 14 out of the 18 paths the RMSRE is less than 1.0 for window-limited flows.

4.3 Discussion

The results of this section showed that FB prediction can be inaccurate, mostly when the RTT and/or loss rate before the transfer are significantly different than while the transfer is in progress. We want to emphasize again that the major prediction errors are not specific to the UMass throughput model. This implies that it is unlikely that other TCP throughput models would have produced more accurate FB predictions.

Our results also suggest that more sophisticated techniques for loss rate and RTT estimation could potentially improve FB prediction significantly. Such estimates, however, should take into account the load that will be exerted by the target flow. More information about the underlying path, such as the capacity or the available bandwidth and buffer size, may help achieving this goal.

5. HISTORY-BASED PREDICTION

A fundamentally different approach is to use throughput measurements of previous TCP transfers in the same path to predict the throughput of the next transfer. This *History-Based* (HB) prediction 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. 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 Holt-Winters). We do not examine more complex linear predictors such as ARMA or ARIMA because the selections of both their order and of their linear coefficients require a large number of past measurements [22]; instead, we envision that TCP throughput HB prediction will be applied in practice 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 two simple heuristics to deal with them.

5.1 Simple Linear Predictors

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

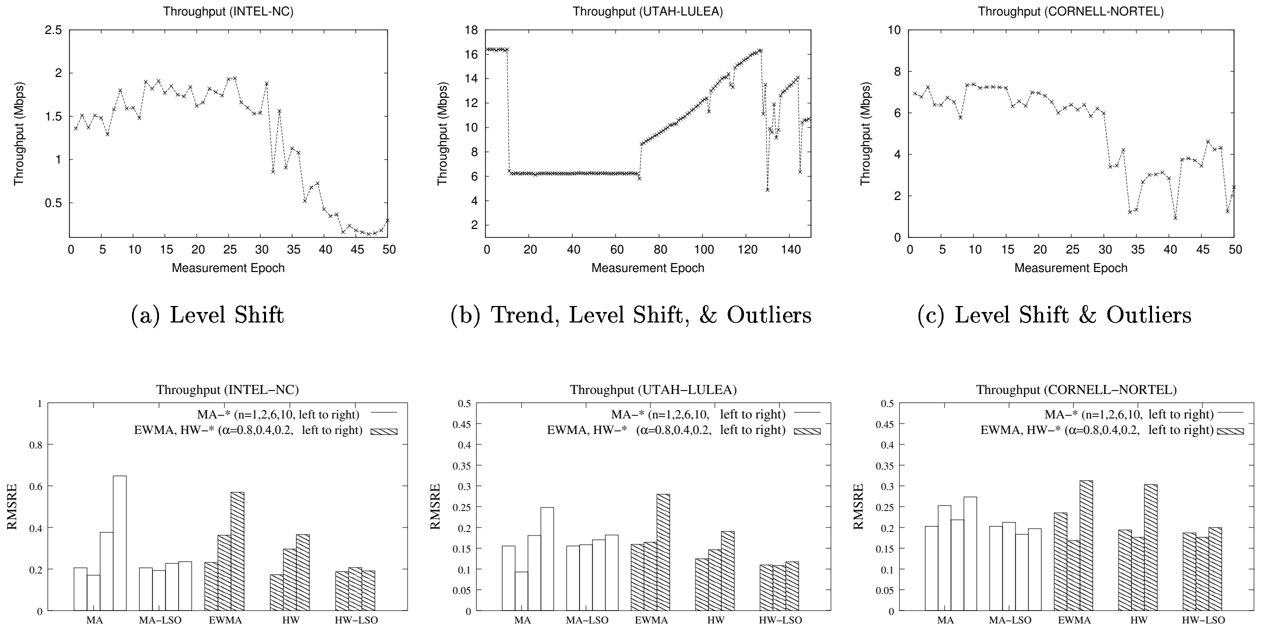
$$\hat{X}_{i+1} = \frac{1}{n} \sum_{k=i-n+1}^{i} X_k$$

  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 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 ($0 < \alpha < 1$). 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 underlying time series.



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

- *Holt-Winters (HW).* The Holt-Winters predictor [5] is a variation of EWMA that attempts to capture the *trend* in the underlying time series, if such a trend exists. The HW 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

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

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

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 find that the largest prediction errors are often caused by level shifts and outliers in the observed time series. Furthermore, if we find a way to deal effectively with the two features, the exact choice of the predictor or of its parameters does not make a significant difference.

A level shift is one type of non-stationarity, and it causes a significant and rather 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, from nearby measurements. Both outliers and level shifts have been studied extensively in the theory of forecasting [26]. See Figures 7(a), 7(b) and 7(c) for 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 our heuristics to detect level shifts and outliers. Suppose that $\{X_1, ...X_n\}$ is the sequence of all the past measurements, ignoring all 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, and 3) $k+2 \leq n$. The last condition avoids 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. A measurement X_k (with $k < n$) is considered an outlier, on the other hand, if it differs by 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 7(d), 7(e) and 7(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 (see the figure legend). For the EWMA and HW predictors, we show results for three values of α. We observed that, at least with our datasets, the RMSRE does not strongly depend on β, χ and ψ, while the following values are close to optimal β=0.2, χ=0.3, and ψ=0.4. On the other hand, the parameters n and α play a major role in the performance when the LSO heuristic is not used. The LSO heuristic decreases significantly the prediction error, and makes the predictors more robust to the choice 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 accuracy of these two predictors will be shown in the next section.

6. HB PREDICTION ACCURACY

In this section, we apply the HB predictors in the previous section on the measurement set that was described in § 4. Our objective is to compare the most promising HB predictors that we experimented with, and to examine how the HB prediction accuracy varies in different paths, with window-limited flows, and with different measurement periods.

6.1 Results

- *HB predictor comparisons.* Figures 8 and 9 summarize the prediction error (in terms of RMSRE) of several MA and HW predictors, respectively. 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 slightly worse. With LSO, there is a significant reduction in the RMSRE of MA predictors. For HW predictors, α=0.8 (0.8-HW) performs visibly better than α=0.4. Further experimentation showed that α=0.8 is close to the optimal for our dataset, and we use this value for HW predictor hereafter unless otherwise noted. 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.

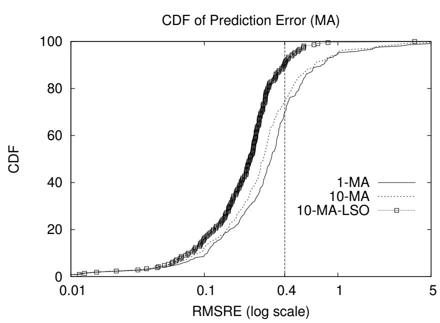


Figure 8: MA prediction error.

- *FB and HB comparison.* Even though these two classes of predictors are complementary, in some cases it may be possible to use either FB or HB predictors. Comparing the RMSRE of the FB predictor (see Figure 5) with that of the HB predictors, we can see that the accuracy of the latter is dramatically better. 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. One may argue that this comparison is not fair for FB, since FB is applicable without the knowledge of any 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.

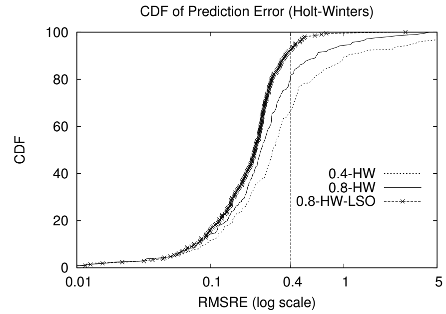


Figure 9: HW prediction error.

- *RMSRE vs. CoV of throughput measurements.* We are interested in the relation between the RMSRE of prediction for a trace and the Coefficient of Variation (CoV) of the corresponding TCP throughput time series. The reason for this comparison will become clear in the following section, where we use the CoV as a metric for the *predictability* of TCP throughput in 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). For the RMSRE, we also exclude measurements that were identified as outliers. Figure 10 shows the CoV and RMSRE for each collected trace, using the HW-LSO predictor. Note the strong positive correlation between the two metrics. The correlation coefficient between them is 0.91. We can thus assume, at least as a first-order approximation, that the RMSRE prediction error with HW-LSO is 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

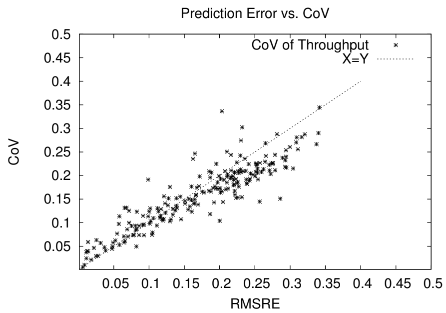


Figure 10: Prediction error versus CoV.

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). What causes different paths to behave so differently? We focus on this question in the next section.

- *Prediction for window-limited flows.* When the target flow is window-limited, it should not be subject to the variation in the avail-bw, so we expect a better predictability. Figure 12 compares the prediction error for window-limited flows (W=20Kbyte) and for congestion-limited flows (W=1Mbyte), using the same traces as in Figure 6. Notice that *window-limited flows have a lower RMSRE, confirming the above insight that the throughput is more predictable when the target flow does not attempt to saturate the path.* The RMSRE reduction is not major, however, especially when the RMSRE for congestion-limited flows is already very low (around 0.1). These remaining errors are probably due to short-term load variations in the underlying path, or random packet losses that the target flow experiences, causing unavoidable variations in the resulting TCP throughput, independent of W.

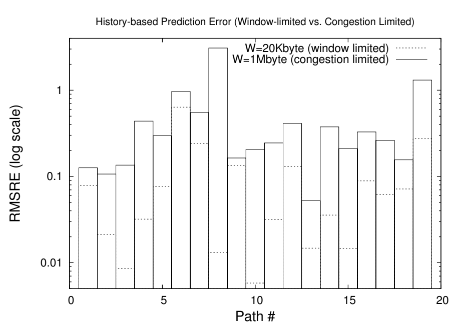


Figure 12: Prediction error for window-limited vs. congestion-limited flows.

- *The effect of the measurement period.* All previous results are based on periodic TCP transfers, performed every 3 minutes. We expect the prediction accuracy to depend on this measurement period. A time series with a larger period spans a wider history horizon, so that the possibility of route changes or major load variations is larger.

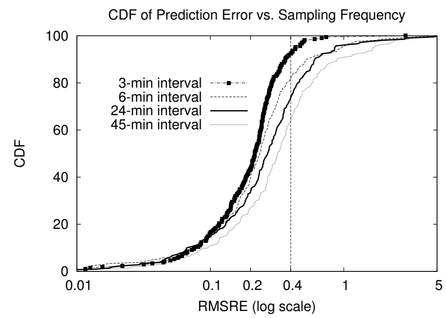


Figure 13: Prediction error with different measurement periods.

To see how the measurement period affects the prediction error, we down-sampled the original traces at different frequencies. We then apply the HW-LSO predictor to the down-sampled traces, producing RMSRE of predictions for measurement periods of 6, 24, and 45 minutes. Figure 13 shows the results. Obviously, *the prediction accuracy degrades as 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 if it relies only on sporadic previous TCP transfers, every few minutes, on the given paths.* Of course we emphasize once more that this conclusion is based only on our dataset, and it is possible that other Internet paths behave quite differently.

6.2 Discussion

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

1. Even a limited history of sporadic previous 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 history of previous TCP transfers in the same path, then HB prediction is 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.

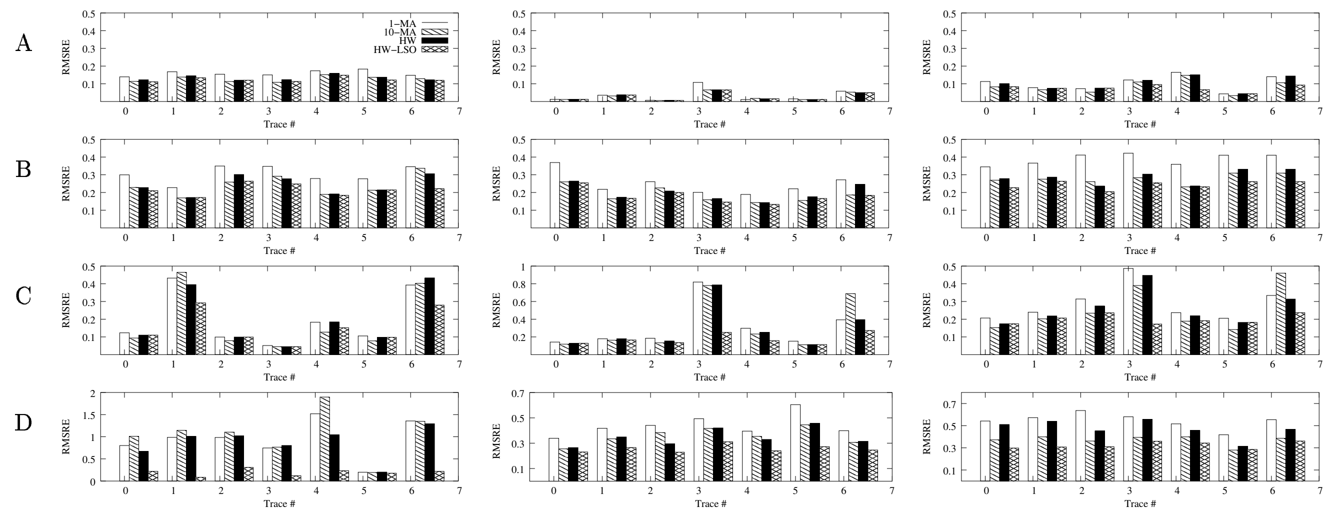


Figure 11: A: Predictable paths (low RMSRE), B: Paths with small and predictable errors (stable RMSRE), C: Paths with small but unpredictable errors (varying RMSRE), D: Unpredictable paths (high RMSRE, notice the different Y-axis ranges).

5. The predictability of an HB predictor is higher when the transfer is window-limited. Consequently, if prediction accuracy is more important than throughput maximization, then the TCP flow should have a limited advertised window such that it does not saturate the underlying path.

7. TWO PREDICTABILITY FACTORS

The empirical results in the previous section raise the question: *what makes TCP throughput much less predictable in certain network paths than in others?* In this section, we focus on this question and identify two major factors, load and the degree of multiplexing, that affect the accuracy of HB prediction in a path. We consider simple queueing models that provide a framework for reasoning about the relationship between TCP throughput predictability and these factors. Instead of examining directly the predictability of a TCP throughput time series, which would be quite hard to do analytically, we relate TCP throughput to the background traffic and examine the predictability of the network traffic time series.

First, we show the connection between the relative prediction error and the Coefficient of Variation (CoV) of the underlying time series. Consider a second-order stationary time series X with mean μ_X, variance σ_X^2, and covariance $\gamma_X(k)$ ($\gamma_X(0)=\sigma_X^2$). According to the Yule-Walker forecasting model [22], an autoregressive one-step predictor based on the n most recent samples of X has the following prediction error variance:

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

where X_i and $\hat{X}_i$ are the actual and predicted values of X, respectively, at time i, and $\{a_{X,n}(i), i = 1, \dots n\}$ 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) [3]) is given by:

$$\frac{\sqrt{\mathrm{Var}[e_n]}}{\mu_X} = \sqrt{\mathrm{CoV}_X^2 - \frac{\sum_{k=1}^{n} a_{X,n}(k)\gamma_X(k)}{\mu_X^2}}, \qquad (5)$$

where $\mathrm{CoV}_X = \sigma_X/\mu_X$. The key point here is that *the relative prediction error increases with the CoV of the underlying time series.* Also recall the observation from Figure 10: the RMSRE with the HW-LSO predictor and the CoV of the corresponding time series have a correlation coefficient larger than 90%. Consequently, in the following we are interested in the effect of load and degree of multiplexing on the CoV of the TCP throughput time series, instead of examining directly the effect of these factors on prediction error.

7.1 Effect of utilization or offered load

Consider a link of capacity C, modeling the bottleneck of a path. We next examine the same question through two different models: first, an IID process for the aggregate traffic on a given time scale 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

We use this model to examine the variability of an available bandwidth time series, which, as mentioned in § 3, can be used as a first-order approximation of TCP throughput on lossless paths. Suppose that the *arriving* traffic rate on a given time scale T is modeled by 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 on the same time scale. For a bufferless link, the observed rate process is given by $Z = \begin{cases} Y & \text{if } Y < C, \\ C & \text{if } Y \geq C \end{cases}$ and so the probability dis-

[3]Notice that although NRMSE is not exactly the same as RMSRE, they are reasonably close as long as u_x does not vary significantly, say spanning an order of magnitude, in a time series. This is true for most paths.

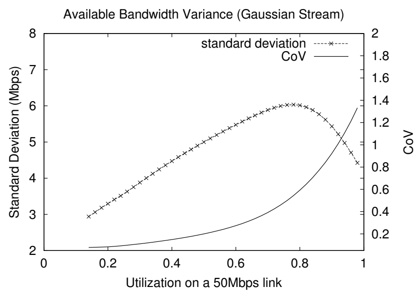


Figure 14: Gaussian process.

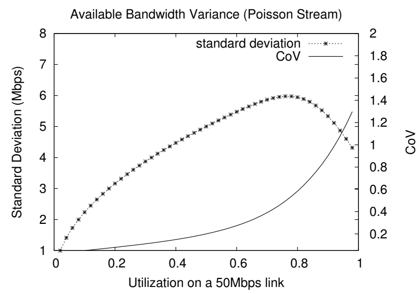


Figure 15: Poisson process.

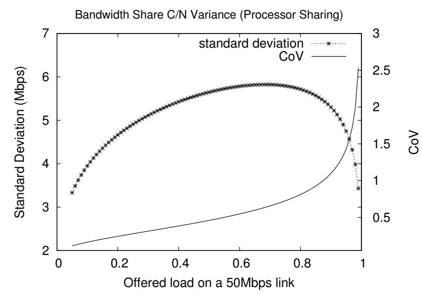


Figure 16: Processor Sharing model.

tribution function of Z is trivially obtained from that of Y. The avail-bw process is given by $A{=}C - Z$, and its CoV is

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

We used Mathematica to derive $\text{CoV}(A)$ for two offered load processes Y: a Gaussian process and a Poisson process. The resulting $\text{CoV}(A)$ (as well as the std-deviation of A) is shown in Figures 14 and 15, respectively, as a function of the link utilization $\rho{=}(C - A)/C$. The key observation is that *the CoV of the avail-bw process increases with the link utilization, and so we should expect a higher relative prediction error under heavier load conditions.* As an interesting side-note, observe that the standard deviation reaches a maximum as the load increases, and then it decreases. The reason is that, in heavy-load conditions, the link is almost always utilized meaning that there is little *absolute* variation in the avail-bw process. This point has been studied in more depth by Tian et al. in [30].

7.1.2 Processor Sharing model with Poisson session arrivals

The previous model does not capture what happens on a congested link, on which the avail-bw is zero. In this model, we consider the traffic as a stream of independent and identically distributed 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, suppose that the link behaves 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, this a more appropriate model for a congested link [11]. An arriving flow, modeling the TCP target transfer, will obtain the same throughput $r(N)$ as any other active flow. Note that, in this model, we are not interested in the CoV of the avail-bw process, 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 16 shows the standard deviation and CoV of $r(N)$ as a function of the offered load ρ. The observation is the same as that from the previous IID traffic model. Specifically, *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" on links with higher degrees of multiplexing, i.e., with a larger number of simultaneously active flows. Using a simple queuing model, we aim to understand whether this intuition applies to TCP throughput as well, and if so, under what exact terms.

Consider again a model of IID Poisson session arrivals. Instead of the Processor Sharing model (which leaves no avail-bw), suppose 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 [11] with mean and variance $E[N] = \text{Var}[N] = (\lambda\theta)/r$.

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

$$\text{CoV}[A] = \text{CoV}[C - Y] = \frac{1}{\sqrt{E[N]}} \frac{\rho C}{C(1-\rho)} \qquad (6)$$

Suppose that we keep the utilization ρ constant, but decrease the session service rate r so that the average number of sessions on the link $E[N]$ increases. Equation (6) 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, when the utilization remains constant.*

7.3 Summary

This section used simple analytical arguments to confirm the following points:

- the relative prediction error increases with the CoV of the underlying time series,
- the CoV of the avail-bw process (on a non-congested link) or the CoV of a flow's throughput (on a congested link) increases with the offered load on the link,
- 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. We note that we have also validated the previous points using simulations of both TCP and non-TCP traffic. We do not include those simulation results due to space constraints.

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 history of prior TCP transfers. We demonstrated however that it can be inaccurate, especially when the transfer attempts to saturate the path, and we explained the reasons for these errors. HB prediction, on the other hand, is quite accurate but it 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 actual 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 on the tight link of the path. In future work, it would be interesting to examine hybrid predictors, which rely on TCP models in the absence of recent history. Another direction would be to develop TCP throughput models that are specifically designed for prediction, and that take as inputs various estimates of the path load, buffering, and cross traffic congestion responsiveness. 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, probably for any practical purposes, in most paths.

9. REFERENCES

[1] D. Anderson, H. Balakrishnan, M. Kaashoek, and R. Morris. Resilient Overlay Networks. In *SOSP*, 2001.

[2] D. Anderson, A. Snoeren, and H. Balakrishnan. Best-Path vs. Multi-Path Overlay Routing. In *IMC*, 2003.

[3] F. Baccelli and K. B. Kim. TCP Throughput Analysis under Transmission Error and Congestion Losses. In *Infocom*, 2004.

[4] S. Basu and A. Mukherjee. Time Series Models for Internet Traffic. Technical report, Georgia Tech, 1995.

[5] J. Brutlag. Aberrant Behavior Detection in Time Series for Network Monitoring. In *USENIX LISA XIV*, 2000.

[6] N. Cardwell, S. Savage, and T. Anderson. Modeling TCP Latency. In *Infocom*, 2000.

[7] Y. Chawathe, S. Ratnasamy, L.Breslau, N. Lanham, and S. Shenker. Making Gnutella-like P2P Systems Scalable. In *SIGCOMM*, 2003.

[8] B. Cohen. Incentive Build Robustness in BitTorrent, 2003.

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

[10] I. Foster, C. Kesselman, and S. Tuecke. The Anatomy of the Grid: Enabling Scalable Virtual Organizations. *Journal on Supercomputer Applications*, 2001.

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

[12] M. Garetoo, R. L. Cigno, M. Meo, and M. Marson. A Detailed and Accurate Closed Queueing Network Model of Many Interacting Flows. In *Infocom*, 2001.

[13] D. Goldenberg, L. Qiu, H. Xie, Y. Yang, and Y. Zhang. Optimizing Cost and Performance for Multihoming. In *SIGCOMM*, 2004.

[14] M. Goyal, R. Guerin, and R. Rajan. Predicting TCP Throughput From Non-invasive Network Sampling. In *Infocom*, 2002.

[15] N. Hu and P. Steenkiste. Evaluation and Characterization of Available Bandwidth Probing Techniques. *JSAC Special Issue in Internet and WWW Measurement*, 2003.

[16] iperf. http://dast.nlanar.net/Projects/Iperf.

[17] M. Jain and C. Dovrolis. End-to-end Available Bandwidth: Measurement Methodology, Dynamics and Relation with TCP Throughput. In *SIGCOMM*, 2002.

[18] B. Krishnamurthy, C. Wills, and Y. Zhang. On the Use and Performance of Content Distribution Networks. In *IMW*, 2001.

[19] S. Low, L. Peterson, and L. Wang. Understanding TCP Vegas: A Duality Model. In *SIGMETRICS*, 2001.

[20] M. Mathis, J. Semske, J. Mahdavi, and T. Ott. The Macroscopic Behavior of TCP Congestion Avoidance. *Computer Communication Review*, 1997.

[21] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose. Modeling TCP Throughput: A Simple Model and its Empirical Validation. In *ACM SIGCOMM*, 1998.

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

[23] R. Prasad, M. Jain, and C. Dovrolis. Socket Buffer Auto-Sizing for High-Performance Data Transfers. Technical report, GIT-CERCS, 2004.

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

[25] S. Ratnasamy, M. Handley, R. Karp, and S. Shenker. Topologically-Aware Overlay Construction and Server Selection. In *Infocom*, 2002.

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

[27] A. Sang and S.Q. Li. A Predictability Analysis of Network Traffic. In *INFOCOM*, 2000.

[28] B. Skidar, S. Kalyanaraman, and K. Vastola. Analytic Models for the Latency and Steady-State Throughput of TCP Tahoe, Reno and SACK. *Transaction on Networking*, 2003.

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

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

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

[32] S. Vazhkudai and J. Schopt. Predicting Sporadic Grid Data Transfers. In *HPDC*, 2002.

[33] R. Wolski, N. Spring, and J. Hayes. The Network Weather Service: A Distributed Resource Performance Forecasting Service for Metacomputing. *Journal of Future Generation Computing Systems*, 1998.

[34] Y. Zhang, N. Duffieled, V. Paxson, and S. Shenker. On the Constancy of Internet Path Properties. In *IMW*, 2001.