Available online at www.sciencedirect.com
ScienceDirect
Computer Networks 51 (2007) 3959–3977
Computer Networks
www.elsevier.com/locate/comnet

On the predictability of large transfer TCP throughput ☆

Qi He *, Constantinos Dovrolis, Mostafa Ammar

College of Computing, Georgia Tech, Atlanta, GA 30332, United States

Received 25 August 2006; received in revised form 5 February 2007; accepted 5 April 2007
Available online 3 May 2007

Responsible Editor: S. Palazzo

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 Resilient Overlay Network (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, the accuracy of HB predictors is highly path-dependent.

Keywords: Network measurements; TCP throughput; Time series forecasting; Performance evaluation

1. Introduction

With the advent of overlay and peer-to-peer networks [1,2], Grid computing, and Content Distribution Networks (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 [3], dynamic server selection [4], 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

☆ This work was supported by the NSF CAREER award ANIR-0347374. An earlier version of this paper appeared in ACM SIGCOMM 2005.
* Corresponding author. Tel.: +1 408 506 7369.
E-mail address: qihe@yahoo-inc.com (Q. He).

1389-1286/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2007.04.013

the Round-Trip Time (RTT) and the presence of random losses [5].

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 we then evaluate empirically over the RON testbed [7]. 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 category of 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 [1], predicting TCP throughput based on the simple "square-root" formula of [8]. 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 [9–12], differing in terms of complexity and accuracy, modeling assumptions, and TCP flavor. In this paper, we prefer to use the main result of [10], referred to as the *PFTK formula*, because it is both simple and accurate. We also experiment with a revised version of this formula. 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. One major 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* [13]. 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. Whereas FB prediction has to use a dif-

ferent throughput model (formula) for each variant of TCP, HB prediction is independent of the specific TCP implementation that is used. 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 [14,15], 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 min, prediction errors are still fairly low. On the negative side, our measurements show that the accuracy of HB predictors is highly path-dependent.

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. We conclude in Section 7.

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 [6,8,10]. 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. [10], and they have proposed a way to estimate p' from p [13]. 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 [16]. In NWS, throughput prediction is based on small (64 KB) TCP transfer probes with a limited socket buffer size (32 KB). Vazhkudai et al. use bulk TCP transfers (1 MB–1 GB) and a large socket buffer (1 MB), performed sporadically (1 min–1 h) [14]. 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 [15]. Their TCP throughput measurements use 1 MB transfers performed every minute, with 200 KB 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 [17]. Some mathematical models (such as MMPP) have been previously used to analyze the predictability of aggregate network traffic [18].

In a recent work by Arlitt et al. [19], the authors studied two approaches for throughput prediction of short TCP flows, similar to FB and HB. Instead of using the PFTK model, their FB approach uses the TCP latency model proposed by Cardwell et al. [9].

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 [8]:

$$E[R] = \frac{M}{T\sqrt{\frac{2bp}{3}}}, \tag{1}$$

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 [10], which improves on the square-root formula especially in the presence of retransmission timeouts and/or a limited maximum window:

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

where T_0 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.

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 [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 (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.

Table 1
Table of symbols

T	RTT experienced by flow[a]
$\hat{T}$	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

[a] The word "flow" in this table refers to the target flow.

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 [20,21]. 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:

$$\hat{R} = \begin{cases} \min\left(\dfrac{M}{\hat{T}\sqrt{\frac{2b\hat{p}}{3}} + \hat{T}_0 \min\left(1, \sqrt{\frac{3b\hat{p}}{8}}\right)\hat{p}(1+32\hat{p}^2)}, \dfrac{W}{\hat{T}}\right) & \text{if } \hat{p} > 0, \\ \min\left(\dfrac{W}{\hat{T}}, \hat{A}\right) & \text{if } \hat{p} = 0, \end{cases} \tag{3}$$

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: $\hat{T}_0 = \max(1\text{ s}, 2\text{ SRTT})$, 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 $\hat{T}$, $\hat{p}$, $\hat{T}_0$, rather than the actual values T, p, T_0, 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 [10], 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. This issue has been also studied in [22]. Comparing TCP sampling with Uniform, Exponential, and Pareto probing interarrivals, they measured that TCP observes a higher average loss rate and longer loss bursts.

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 [13]. Our *ns*2 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 [20]. 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 $\hat{A}$ [23]. 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

We collected measurements on Internet paths that connect hosts of the RON testbed [7]. The RON project currently has 50–60 nodes, mostly at universities and research labs in the US. There are also some at ISPs and other companies, and a few nodes in Europe and Asia. We chose to use RON instead of PlanetLab [24] because the former is not so heavily used. This is important for more accurate measurement, especially when it comes to estimating available bandwidth. Additionally, the RON project does not enforce the type of rate limiting that PlanetLab does.

Unless otherwise noted, the results presented refer to a first set of measurements, collected in May 2004. The measurements were collected over 35 Internet paths. The corresponding RON nodes are located mostly in US universities, with two nodes in Europe and one in Korea. Out of the 35 paths, five are transatlantic paths, one between Korea and New York, 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 the measurements. The capacities for the rest of the paths are at least 10 Mbps. Note that we do not use the RON overlay routing features [7].

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-s measurement of $\hat{p}$ and $\hat{T}$ using a homespun ping utility that generates a 41-byte probing packet every 100 ms, followed by a 50-s TCP transfer (target flow) generated by IPerf [25]. Fig. 1 shows the timeline of the measurements in an epoch. RTT and loss rate estimates are also measured during the TCP transfer. A 50-s TCP transfer on these paths is long enough to ensure that the flow spends a negligible fraction of its lifetime in the initial slow start. Overall, 36750 TCP transfers (and 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 min, while the duration of each trace was about 6 h.

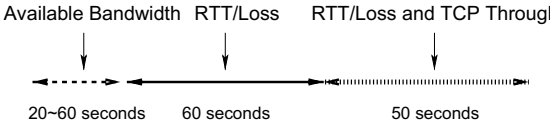


Fig. 1. A measurement epoch. 150 such epochs were recorded during each trace, with 7 traces collected per path.

Where noted, we also present FB results from a second set of measurements, collected in March 2006. Those measurements were collected on 24 Internet paths, none of them present in the first set of measurements, and between 12 RON hosts located in the US. Only one of those nodes was DSL-connected.

IPerf allows us to control the maximum TCP window size W by limiting the socket buffer size. Unless otherwise noted, that parameter was set to $W = 1$ MB, 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 $W = 20$ KB, which limits the transfer to only a fraction of the avail-bw in 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 during 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

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

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

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

where E_i is the relative error of measurement i.

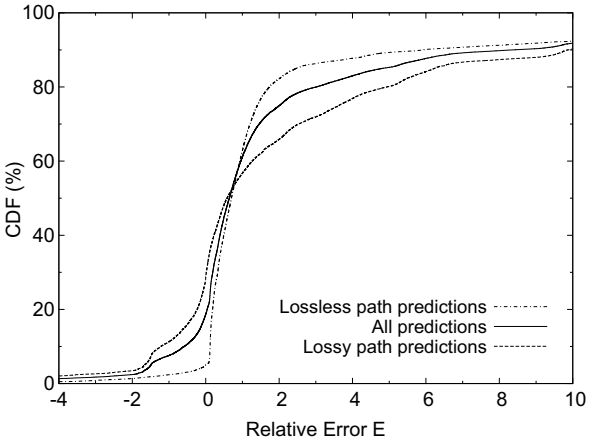


Fig. 2. CDF of E for all predictions, for predictions in lossy paths, and for predictions in lossless paths.

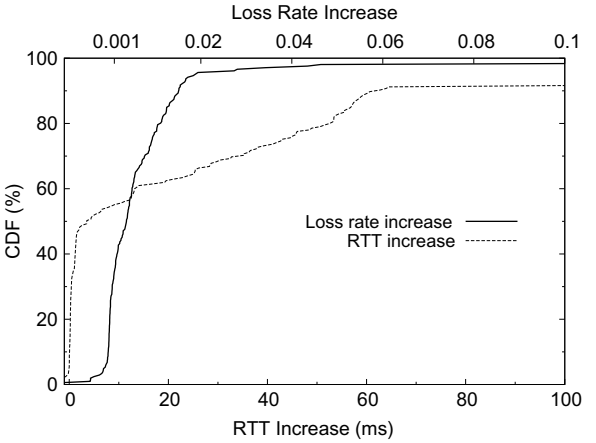


Fig. 3. CDF of RTT and loss rate absolute increase during target flow.

4.2. Results

4.2.1. Prediction error in lossy and lossless paths

Fig. 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* ($E \geqslant 1$). In fact, the overestimation errors are larger than an order of magnitude ($E \geqslant 9$) for almost 10% of the measurements. The underestimation errors are much less dramatic and common, with only 8% 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 Section 3.4. The fact that overestimation is the only major type of prediction error in lossless paths 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.

4.2.2. RTT and loss rate increase during target flow

Returning to the case of lossy paths, the fact that overestimation is much more dramatic than underestimation is probably related to the first issue discussed in Section 3.2, namely that $\hat{T} < T$ and $\hat{p} < p$.

Fig. 3 shows the distribution of the absolute RTT and loss rate increase during the target flow. The increase was 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). Note 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 5 ms and 60 ms. In 10% of the measurements the RTT increase was higher than 100 ms, probably due to congested low-capacity links. The loss rate, on the other hand, increased by 0.1–2% in almost all measurements.

To relate the RTT and loss rate increase to prediction error, we can see, based on the simplified model of Eq. (1), that the relative prediction error is

$$E = \frac{\frac{C}{\hat{T}\sqrt{\hat{p}}} - \frac{C}{\tilde{T}\sqrt{\tilde{p}}}}{\frac{C}{\tilde{T}\sqrt{\tilde{p}}}} = \frac{\tilde{T}\sqrt{\tilde{p}}}{\hat{T}\sqrt{\hat{p}}} - 1,$$

where C represents the constant factor $M\sqrt{\frac{3}{2b}}$. Figs. 4 and 5 show the relative RTT and loss rate increases, respectively, calculated as $\frac{\tilde{T}-\hat{T}}{\hat{T}}$ and $\frac{\tilde{p}-\hat{p}}{\hat{p}}$. The

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

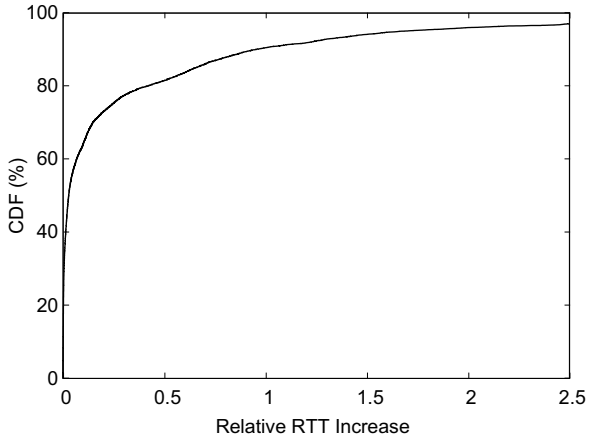


Fig. 4. CDF of relative RTT increase during target flow.

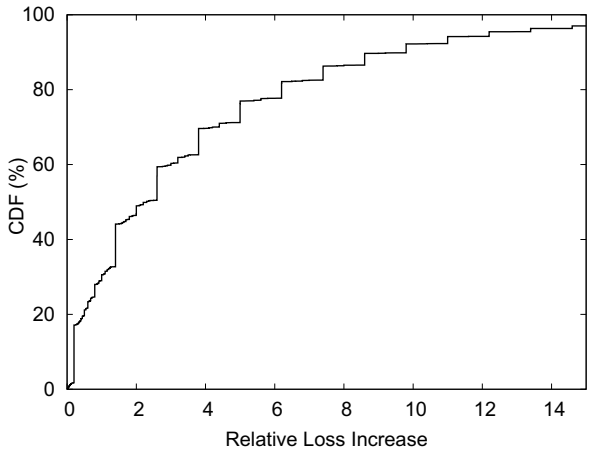


Fig. 5. CDF of relative loss rate increase during target flow.

loss rate measurements refer only to paths that were lossy even before the target transfer ($\hat{p} > 0$).[2] For only 20% of the epochs the relative RTT increase is larger than 0.5 (i.e., $\tilde{T} > 1.5\hat{T}$), contributing a factor of more than 50% in the prediction error.In terms of the loss rate increase, more than 70% of the epochs have a relative increase that is larger than 1.25 (i.e., $\tilde{p} > 2.25\hat{p}$), contributing a factor of more than 50% in the prediction error.

On average, the RTT during the target transfer increased to 1.3 times the pre-transfer RTT, while the loss rate increased to almost 5 times the pre-transfer loss rate. These increase figures translate to an average prediction error of 1.9 (i.e., an overestimation by a factor of 2.9), which is significant considering that the average FB prediction error for these epochs is 3.4.

4.2.3. 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 estimates of the path's RTT $\tilde{T}$ and loss rate $\tilde{p}$, based on periodic probing, 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, as discussed in Section 3.3. In terms of our notation, these errors are caused by the differences between $\tilde{T}$ and T, and between $\tilde{p}$ and p.

Fig. 6 shows the CDF of the FB prediction error when we apply the ping-based RTT $\tilde{T}$ and loss rate $\tilde{p}$ during the target flow in (3). The CDF refers only to lossy paths, as in Figs. 4 and 5. Note that using $\tilde{T}$ and $\tilde{p}$ makes the relative error significantly lower ($-3 < E < 3$ for about 80% of the predictions) than using $\hat{T}$ and $\hat{p}$. Also, overestimation and underestimation become equally likely and the CDF of E becomes 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. These

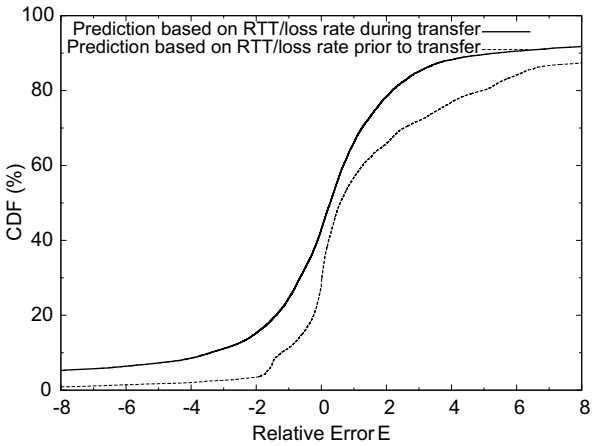


Fig. 6. Prediction errors using $\tilde{T}$ and $\tilde{p}$ (RTT and loss rate measurements during the target flow) and using $\hat{T}$ and $\hat{p}$ (RTT and loss rate measurements prior to the target flow).

[2] Note that the relative loss increase shown in Fig. 5 appears to take discrete values because we used a relatively small number (600) of probing packets.

errors can be attributed to the difference between periodic probing (used to estimate $\tilde{T}$ and $\tilde{p}$) and TCP sampling. Consequently, a second major source of errors in the FB prediction method is that it is difficult to estimate the RTT and loss rate that a TCP connection experiences with active probing.

4.2.4. Variation of prediction error across different paths and traces

Fig. 7 shows the median, as well as the 10/90th 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 it is hard to predict TCP throughput with the FB method, but also it is hard to bound the prediction error that should be anticipated.

Fig. 7 raises the following question: *which paths have the largest prediction errors?* Fig. 8 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 $R \leqslant 0.5$ Mbps have $E > 10$, compared to 0.2% for samples with $R \geqslant 0.5$ Mbps.

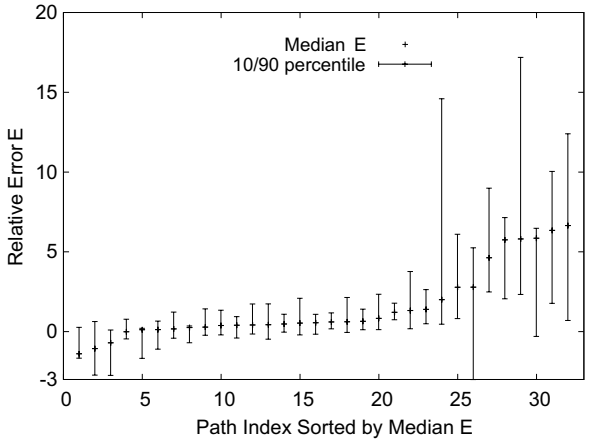


Fig. 7. Variation of the prediction error across different paths.

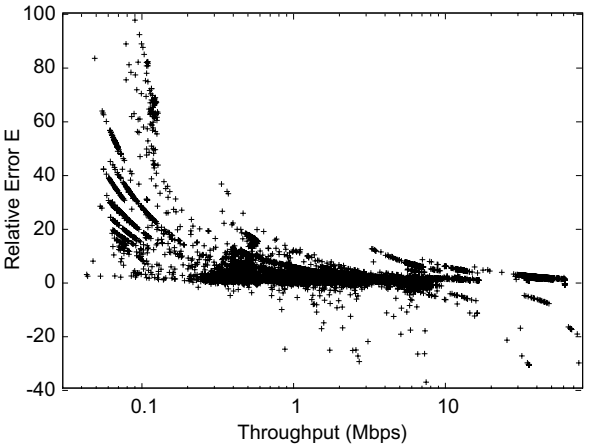


Fig. 8. Actual throughput versus prediction error.

A further analysis of the 10 paths with the largest median prediction errors reveals that two 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 higher 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 significant 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.

4.2.5. Prediction accuracy vs. loss rate

The PFTK model is known to be less accurate when the loss rate is high [10]. This raises the question whether the prediction error is positively correlated to the path loss rate prior to the target transfer.

Fig. 9 is a scatter plot between the loss rate $\hat{p}$ and the FB prediction error. Only lossy epochs are represented in this graph. The major observation here is that the prediction error does not appear to be correlated with the path loss rate.

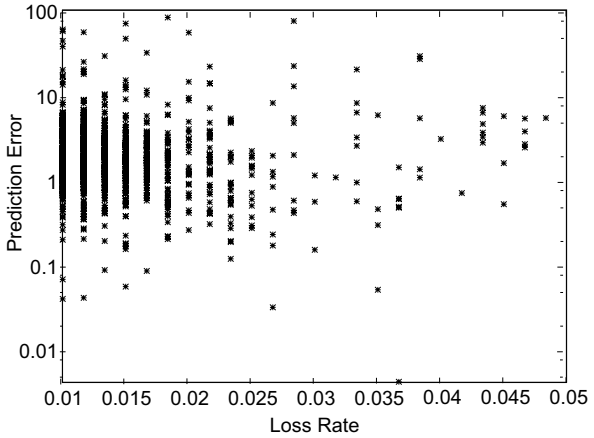


Fig. 9. Prediction accuracy versus the a priori loss rate $\hat{p}$.

4.2.6. Prediction accuracy vs. RTT

It is well-known that TCP flows with longer RTT experience larger throughput reduction upon a congestion event, compared to competing flows with shorter RTT. So, it is worth investigating whether the prediction error is correlated with the RTT prior to the transfer. Fig. 10 is a scatter plot of the RTT $\hat{T}$ and the FB prediction error. Notice, however, that there is no positive correlation between the two metrics.

4.2.7. Prediction accuracy for transfers of different lengths

As the target transfer size increases, it becomes more likely that the load conditions in the path will change, making the FB prediction less accurate. On the other hand, as the transfer size increases the throughput converges to the expected value given by (2). To examine whether there exists a correlation between transfer length and prediction accuracy, the second set of measurements use IPerf transfers that last 120 s, instead of 50 s. These longer transfers allowed us to examine the accuracy of the throughput prediction for the first 30, 60, and 120 s of the target transfer. As shown in Fig. 11, we do not observe a noticeable correlation between prediction error and transfer duration. This observation should of course be limited to flows that are long enough so that the effect of the initial slow start on the average throughput is negligible.

In [9], the authors derive the number of segments $E[d_{ss}]$ transferred during the initial slow start as a function of the loss rate p and the total number of segments in the flow d

$$E[d_{ss}] = \frac{(1-(1-p)^d)(1-p)}{p} + 1.$$

Based on this model, we can determine whether a transfer is long enough to neglect the effect of the initial slow start. If not, we need to use an appropriate FB predictor that takes the initial slow start into account. For instance, Arlitt et al. apply the model of [9] to predict the throughput of short TCP transfers [19].

4.2.8. 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

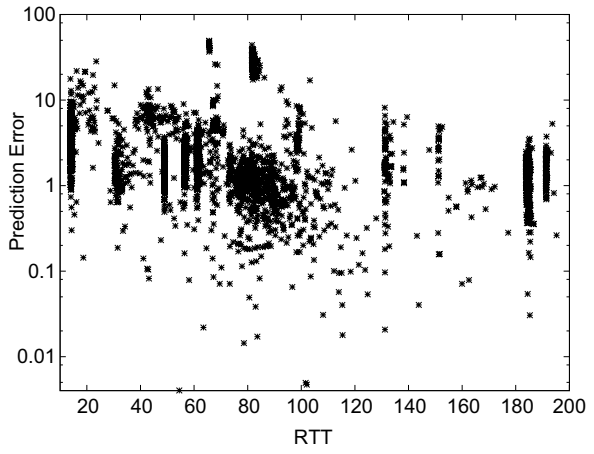


Fig. 10. Prediction accuracy versus the a priori RTT $\hat{T}$.

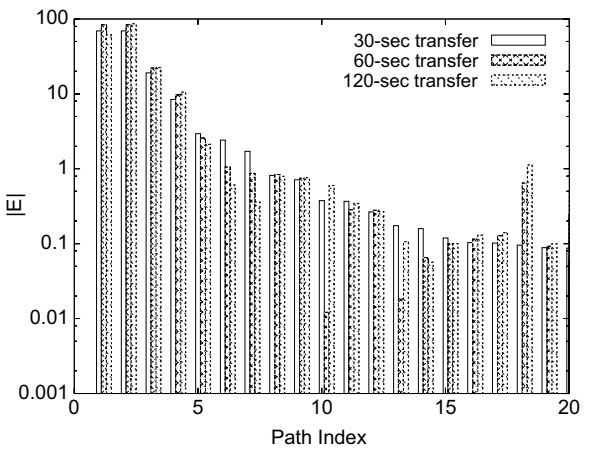


Fig. 11. Prediction accuracy for transfers of different lengths.

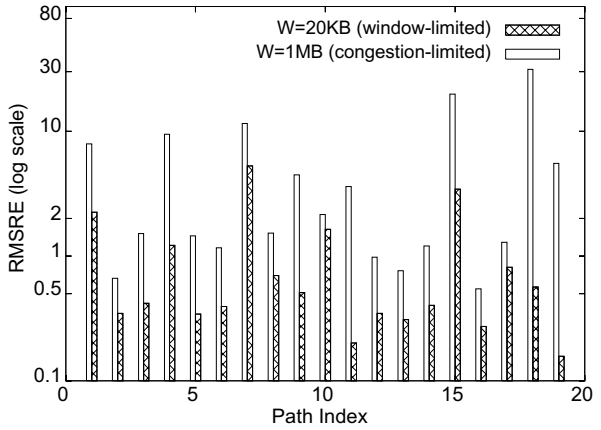


Fig. 12. Prediction accuracy for window-limited versus congestion-limited flows.

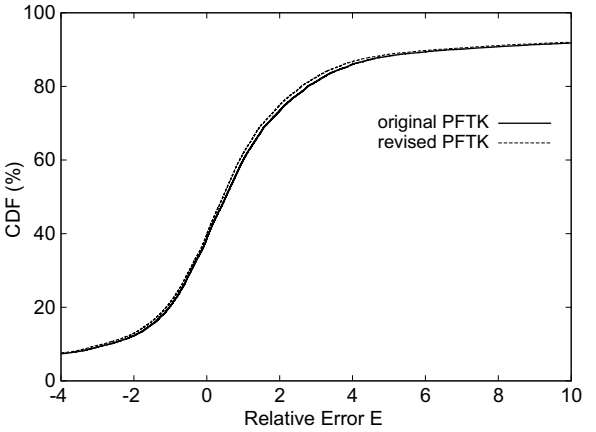


Fig. 13. CDF of E using the revised PFTK model.

answer this question, we extended each epoch with another IPerf TCP transfer with $W = 20$ KB. We verified that this transfer was window-limited on 19 of the 35 paths, and the ratio $W/(\hat{T}\hat{A})$ varied between 0.02 and 0.8. Fig. 12 compares the RMSRE between the transfers with a large maximum window ($W = 1$ MB) and a small maximum window ($W = 20$ KB). Notice the log-scale of the Y-axis. In all paths, the prediction error of window-limited flows is lower, often by a large factor. In particular, 14 out of the 19 paths have an RMSRE that is less than 1.0 for window-limited flows.

We anticipate that for many applications, a prediction error of less than 1.0 would be acceptable. Consequently, applications that care more for throughput predictability than throughput maximization should perform transfers with a limited advertised window so that they do not attempt to saturate the underlying avail-bw. Such applications may involve real-time grid computing, TCP-based streaming, or optimized peer selection in overlay networks.

4.2.9. The revised PFTK model

The PFTK model is not the only available TCP throughput equation, nor the most accurate one. A recent publication identified some errors in the original PFTK model of (2) and derived a corrected version of that equation [26]. Fig. 13 shows the CDF of the prediction errors if we replace the original PFTK formula of (2) with the revised PFTK formula of [26]. Clearly, the difference between the two predictors is negligible compared to the overall errors of FB prediction.

4.2.10. Using history-smoothed RTT and loss rate in FB

The FB predictor that we examined so far uses the most recent measurements of loss rate and RTT to estimate $\hat{p}$ and $\hat{T}$. These estimates however may be subject to noise, and so they may not reflect the true mean of the loss rate and RTT in the path. Will the prediction accuracy improve if we attempt to predict the loss rate and RTT in the path based on recent measurements of these metrics?

Fig. 14 shows the CDF of E by applying predicted values for the RTT and loss rate in the FB predictor. We use a simple Moving Average predictor, based on the last 10 samples. The CDF of E using the original FB predictor is also shown for comparison. We observe that the two predictors

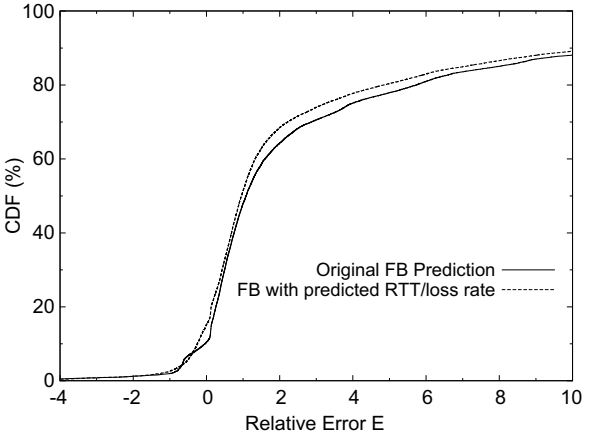


Fig. 14. CDF of E using RTT and loss rate estimates from a smoothed historical average.

are very similar. This means that the errors involved in the estimation of the a priori RTT and loss rate are insignificant compared to the sources of prediction error we analyzed earlier.

4.3. Summary

In this section, we evaluated the accuracy of FB prediction based on Internet measurement results. Our main findings can be summarized as follows:

1. FB prediction can be very inaccurate. About 50% of the predictions in our measurements were wrong by more than a factor of two, while 10% of the predictions were wrong by more than a factor of 10.
2. Overestimation occurs in about 80% of the measurements. In addition, overestimation errors are typically larger than underestimation errors.
3. The increase of the loss rate (primarily) and of the RTT (secondarily) during the target transfer is a major cause of FB prediction errors.
4. The loss rate and RTT estimates based on periodic probing are often significantly different than the corresponding metrics that TCP transfers experience. This is a second major cause of FB prediction errors.
5. The largest prediction errors occur in paths where the TCP throughput is low because there was congestion prior to the target transfer.
6. Window-limited transfers have much more predictable throughput than congestion-limited transfers.

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 [27]; 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

5.1.1. 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).

5.1.2. 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 time series.

5.1.3. 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 $\hat{X}_i^f$ 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

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

X_i in the above equations is the observed value at time i. $\hat{X}_i^s$ can be considered the EWMA of sample values, and $\hat{X}_i^t$ can be considered the EWMA of the difference between two consecutive samples. Assuming that the time series starts at $i = 0$, the initial values of $\hat{X}^s$ and $\hat{X}^t$ are $\hat{X}_0^s = X_0$ and $\hat{X}_0^t = X_1 - X_0$, respectively.

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 isolate 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 [28]. In Fig. 15a–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.

Rigorous methods to detect level shifts and outliers have been proposed in the time series literature [28]. These algorithms however, are mostly based on constructing an ARMA model for the time series, and this typically requires several tens or hundreds of measurements. In the applications that we consider, there are typically only a few previous TCP transfers in the recent past, making the construction of an accurate ARMA model problematic. There are also algorithms to detect the presence of non-stationarities in a time series (for example, the run test or the reverse arrangement test [29]). Note however, that our objective is not to detect any type of non-stationarity. First, outliers or level shifts can be present even in stationary time series. Second, certain types of non-stationarities, such as trends or periodicities, can be captured by the simple linear predictors that we consider next.

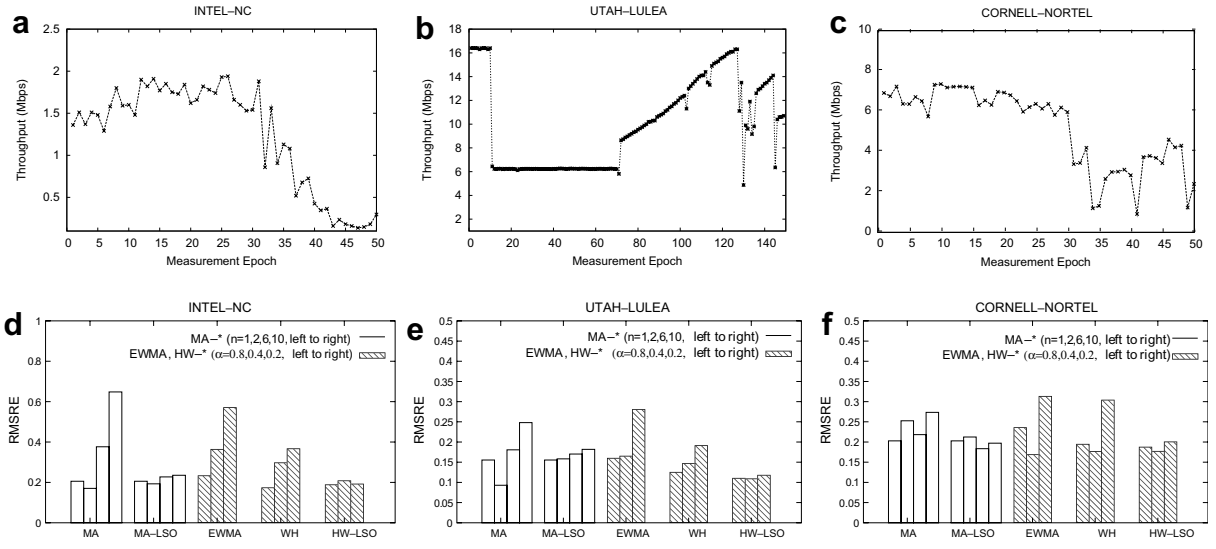


Fig. 15. Examples of TCP throughput traces and the prediction errors (RMSRE) with various predictors: (a) level shift, (b) trend, level shift, and outliers, (c) level shift and outliers, (d) prediction error for trace (a), (e) prediction error for trace (b) and (f) prediction error for trace (c).

We next describe simple heuristics to detect level shifts and outliers that are quite effective in practice. Suppose that $\{X_1, \ldots, X_n\}$ is the sequence of 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 $\{X_1, \ldots, X_{k-1}\}$ are all lower (higher) than the measurements $\{X_k, \ldots, X_n\}$.
2. The median of $\{X_1, \ldots, X_{k-1}\}$ is lower (higher) than the median of $\{X_k, \ldots, X_n\}$ by more than a relative difference χ.
3. $k + 2 \leqslant n$.

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 $\{X_1, \ldots, X_n\}$ by more than a relative difference of ψ. Outliers are discarded from the history of previous measurements.

5.3. Predictor parameters

Fig. 15d–f show the RMSRE for 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. 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 depend significantly on the values of 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: $\beta = 0.2$, $\chi = 0.3$, and $\psi = 0.4$. 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 Section 4. Our objective is to investigate the overall HB prediction accuracy, compare the most promising HB predictors that we experimented with, and examine how the prediction accuracy varies in different paths and with different transfer frequencies.

6.1. Results

6.1.1. Accuracy of HB predictors

Figs. 16 and 17 summarize 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, $\alpha = 0.8$ (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.

As we note earlier, for our dataset, the LSO detection algorithm is not sensitive to its parameters χ and ψ. This is demonstrated in Fig. 18, which compares the CDFs of $|E|$ for MA-5 with several different values for χ and ψ.

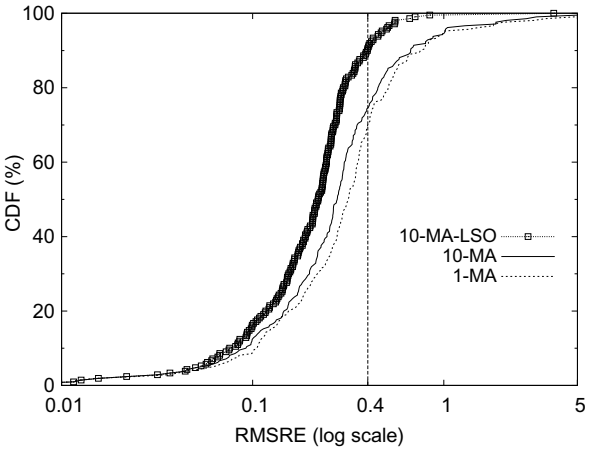


Fig. 16. Moving average prediction error.

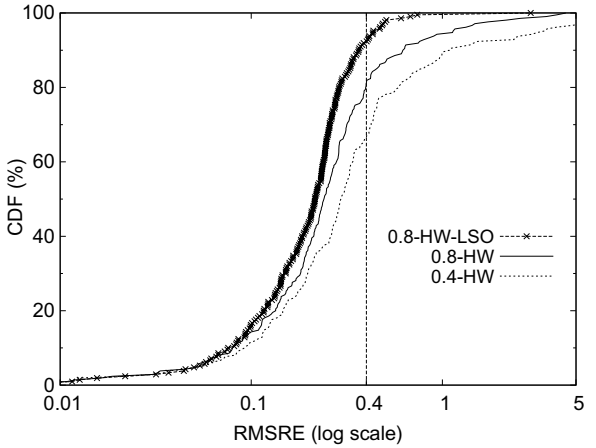


Fig. 17. Holt-Winters prediction error.

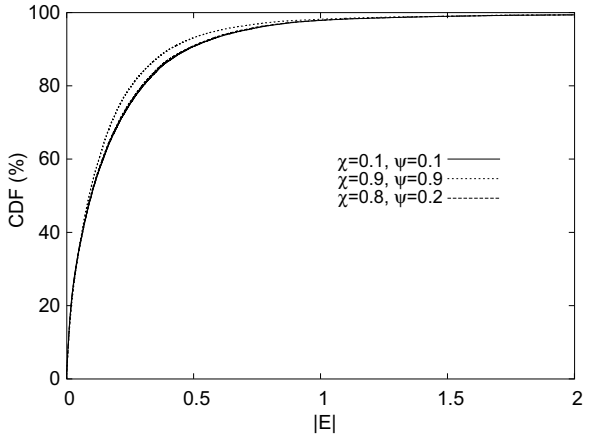


Fig. 18. MA-5 performance with LSO, using different values for χ and ψ.

6.1.2. Comparison of FB and HB predictors

Even though these two classes of predictors are complementary, in some 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, as shown in Fig. 19. 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 over FB prediction.

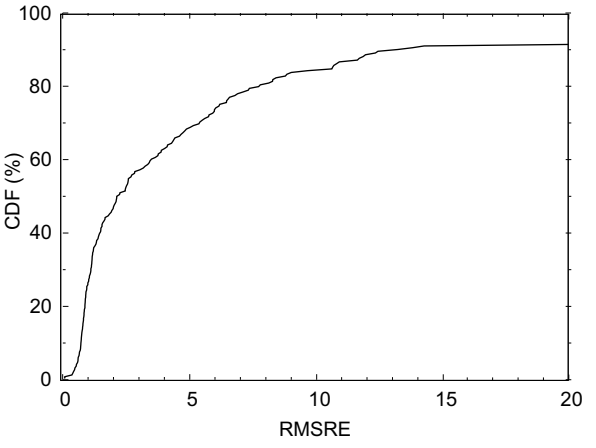


Fig. 19. CDF of RMSRE for FB (for comparison with HB).

6.1.3. RMSRE versus 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 throughput time series.[3] By relating the two metrics, we can then analyze the predictability of TCP throughput based on factors that affect the throughput CoV in a path. This is the approach that we followed in [30].

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. Fig. 20 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. Consequently, at least as a first-order approximation and for the datasets that we analyzed, *the RMSRE prediction error with HW-LSO is approximately equal to the CoV of the corresponding time series.*

6.1.4. Variations in path predictability

Fig. 21 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

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

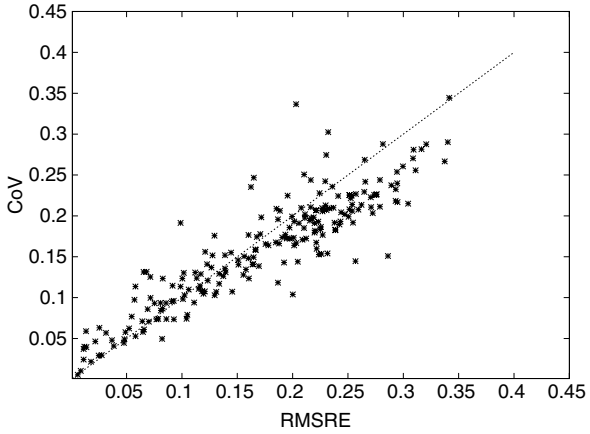


Fig. 20. Prediction error versus CoV of the corresponding time series.

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 significant 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 Fig. 8), we did not observe a significant correlation between the actual throughput and the HB prediction accuracy.

What causes different paths to behave differently in terms of their throughput predictability? We examined the correlation of the prediction error with several path metrics that we could measure: a priori loss rate, a priori RTT, loss rate relative increase, and RTT relative increase. None of the correlations were significant however. One exception was paths with a loss rate larger than 0.5% prior to the target transfer. For those paths, the correlation coefficient between the RMSRE and the loss rate ranges from 0.72 to 0.94. This observation agrees with our earlier finding for FB prediction: it is harder to accurately predict TCP throughput in congested paths.

In an earlier version of this paper [30], we used simple queuing models to show the impact of two additional path characteristics: the utilization and the degree of statistical multiplexing at the path's bottleneck link. The main results of that analysis are that:

1. The prediction error increases with the utilization on the bottleneck link.
2. The prediction error decreases with the number of competing flows on that link, if the utilization remains constant.

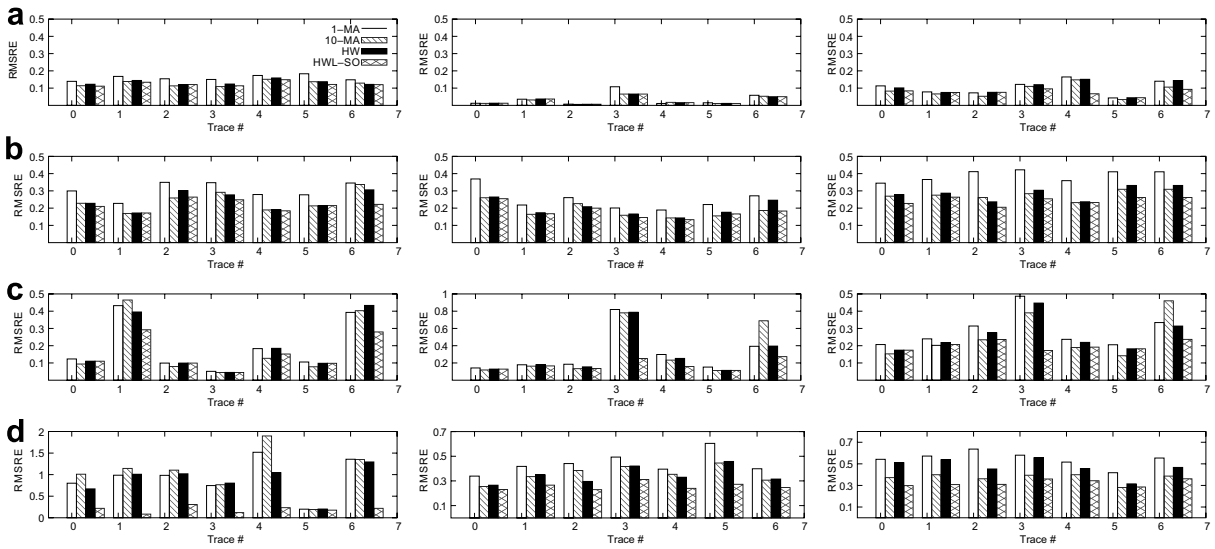


Fig. 21. (a) Predictable paths (low RMSRE), (b) paths with small and predictable errors (stable RMSRE), (c) paths with small but unpredictable errors (varying RMSRE) and (d) unpredictable paths (high RMSRE, notice the different *Y*-axis ranges).

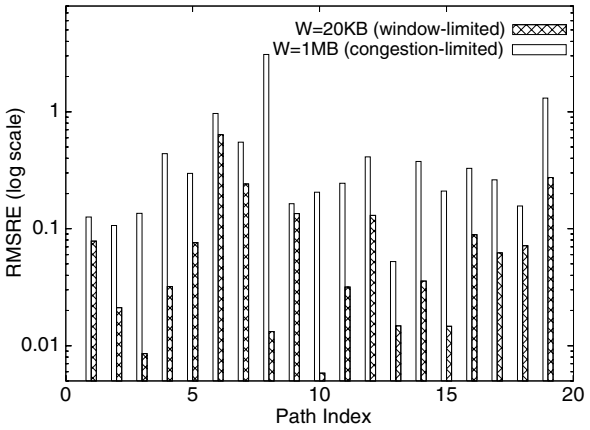


Fig. 22. Prediction error for window-limited versus congestion-limited flows.

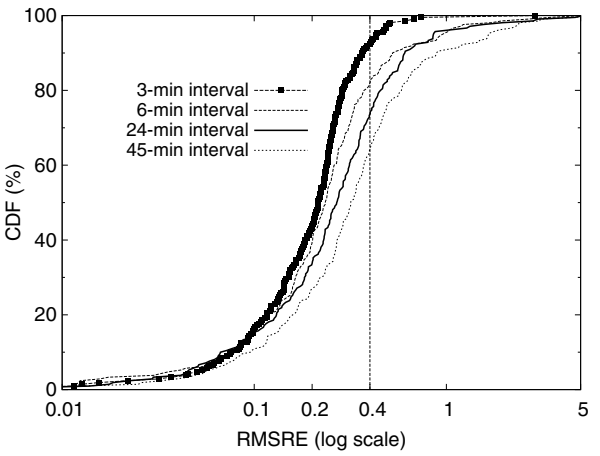


Fig. 23. Prediction error with different TCP transfer intervals.

Unfortunately, we have no means to measure the utilization or the number of competing flows in the bottleneck link. So, we were not able to experimentally verify the previous analytical predictions.

6.1.5. Predictability of window-limited flows

When the target flow is window-limited, it is typically not subject to the dynamic variations of the available bandwidth in the path, and so we expect higher predictability. Fig. 22 compares the prediction error for window-limited flows ($W = 20$ KB) and for congestion-limited flows ($W = 1$ MB), using the same traces as in Fig. 12. Notice that *window-limited flows have a lower RMSRE, confirming the insight that the throughput is more predictable when the target flow does not attempt to saturate the path*. The RMSRE difference is not always major, however, especially when the RMSRE for congestion-limited flows is already quite 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 variations in the resulting TCP throughput independent of W.

6.1.6. The effect of the target flow frequency

All previous results are based on periodic TCP transfers, performed approximately every 3 min. We expect the prediction accuracy to depend on the transfer period/interval. A time series with a larger period spans a wider time 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 min. Fig. 23 shows the results.

As we would expect, *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 90th 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 or even every half hour, 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. Our findings can be summarized as follows:

1. Even a limited history of sporadic TCP transfers (as few as 10 history samples, and as infrequent as 1 transfer every 45 min) 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. With LSO, HB prediction accuracy is not sensitive to the choice of the actual predictor or the predictor's parameters.
4. If HB prediction is feasible, i.e., if there is a history, even though short, of recent TCP transfers in the same path, HB prediction is much more accurate than FB prediction.
5. 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.
6. There is a strong correlation between the RMSRE of HB prediction and the Coefficient of Variation (CoV) of the underlying throughput time series.
7. Similar to FB prediction, the HB prediction errors are lower for window-limited transfers.

7. 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.

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, for example by monitoring transfers that flow between two networks rather than between two hosts, can also improve the practicality of HB prediction.

Acknowledgements

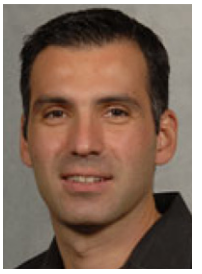
We are grateful to the RON project members for providing us access at their testbed. We also thank the anonymous reviewers for their valuable suggestions.

References

[1] D. Andersen, H. Balakrishnan, F. Kaashoek, R. Morris, Resilient overlay networks, in: Proceedings of ACM Symposium on Operating Systems Principles, 2001.
[2] Y.-H. Chu, S.G. Rao, S. Seshan, H. Zhang, Enabling conferencing applications on the internet using an overlay multicast architecture, in: Proceedings of ACM SIGCOMM, 2001.
[3] A. Akella, J. Pang, A. Shaikh, B. Maggs, S. Seshan, A comparison of overlay routing and multihoming route control, in: Proceedings of ACM SIGCOMM, 2004.
[4] S. Ratnasamy, M. Handley, R. Karp, S. Shenker, Topologically-aware overlay construction and server selection, in: Proceedings of IEEE INFOCOM, 2002.
[5] S. Ben Fredj, T. Bonald, A. Proutiere, G. Regnie, J.W. Roberts, Statistical bandwidth sharing: a study of congestion at flow level, in: Proceedings of ACM SIGCOMM, 2001.
[6] S. Floyd, M. Handley, J. Padhye, J. Widmer. Equation-based congestion control for unicast applications, in: Proceedings of ACM SIGCOMM, 2000.
[7] Resilient Overlay Network (RON), February 2005. <http://nms.lcs.mit.edu/ron/>.
[8] M. Mathis, J. Semke, J. Madhavi, The macroscopic behavior of the TCP congestion avoidance algorithm, ACM Computer Communications Review 27 (3) (1997) 67–82.
[9] N. Cardwell, S. Savage, T. Anderson, Modeling TCP latency, in: Proceedings of IEEE INFOCOM, 2000.
[10] J. Padhye, V. Firoiu, D. Towsley, J. Kurose, Modeling TCP throughput: a simple model and its empirical validation, IEEE/ACM Transactions on Networking 8 (2) (2000) 133–145.
[11] B. Sikdar, S. Kalyanaraman, 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) (2003) 959–971.
[12] N. Celandroni, Comparison of FEC types with regard to the efficiency of TCP connections over AWGN satellite channels, IEEE Transactions on Wiresless Communications 5 (7) (2006) 1735–1745.
[13] M. Goyal, R. Guerin, R. Rajan, Predicting TCP throughput from non-invasive network sampling, in: Proceedings of IEEE INFOCOM, 2002.
[14] S. Vazhkudai, J. Schopf, I. Foster, Predicting the performance of wide area data transfers, in: Proceedings of IEEE IPDPS, 2002.
[15] Y. Zhang, N. Duffield, V. Paxson, S. Shenker, On the constancy of internet path properties, in: Proceedings of Internet Measurement Workshop, 2001.
[16] M. Swany, R. Wolski, Multivariate resource performance forecasting in the network weather service, in: Proceedings of Supercomputing, 2002.

[17] Y. Qiao, J. Skicewicz, P. Dinda, An empirical study of the multiscale predictability of network traffic, in: IEEE Proceedings of HPDC, 2003.
[18] A. Sang, S. Li, A predictability analysis of network traffic, Computer Networks 39 (4) (2002) 329–345.
[19] M. Arlitt, B. Krishnamurthy, J. Mogul, Predicting short-transfer latency from TCP arcana: a trace-based validation, in: Internet Measurement Conference, 2005.
[20] M. Jain, C. Dovrolis, End-to-end available bandwidth: measurement methodology, dynamics, and relation with TCP throughput, IEEE/ACM Transactions on Networking 11 (4) (2003) 537–549.
[21] V. Ribeiro, R. Riedi, R. Baraniuk, J. Navratil, L. Cottrell, pathChirp: efficient available bandwidth estimation for network paths, in: Proceedings of Passive and Active Measurements (PAM) Workshop, April 2003.
[22] B. Melander, M. Bjorkman, Trace-driven network path emulation, Technical Report, Uppsala University, 2002.
[23] G. Appenzeller, I. Keslassy, N. McKeown, Sizing router buffers, in: SIGCOMM, 2004.
[24] PlanetLab, June 2003. <http://www.planet-lab.org>.
[25] Iperf. <http://dast.nlanr.net/Projects/Iperf/>.
[26] Z. Chen, T. Bu, M. Ammar, D. Towsley, Comments on modeling TCP reno performance: a simple model and its empirical validation, Transactions on Networking (2005).
[27] M. Pourahmadi, Foundations of Time Series Analysis and Prediction Theory, John Wiley and Sons, 2001.
[28] R.S. Tsay, Outliers, level shifts, and variance changes in time series, Journal of Forecasting (1988).
[29] J. Bendat, A. Piersol, Random Data: Analysis and Measurement Procedures, John Wiley and Sons, 1986.
[30] Q. He, C. Dovrolis, M. Ammar, On the predictability of large transfer TCP throughput, in: SIGCOMM, 2005.

Qi He received her Ph.D. in Computer Science from Georgia Tech in 2005. She received the S.B. and S.M. degrees from Tongji University and Fudan University, China, in 1995 and 1998 respectively, both in Computer Science. She is currently with IBM, focusing on workload characterization, system performance modeling, and capacity planning. Her research interests also include network performance evaluation, and Internet protocols and services.



Constantinos Dovrolis is an Assistant Professor at the College of Computing of the Georgia Institute of Technology. He received the Computer Engineering degree from the Technical University of Crete (Greece) in 1995, the M.S. degree from the University of Rochester in 1996, and the Ph.D. degree from the University of Wisconsin-Madison in 2000. He was an assistant professor of Computer and Information Science at the University of Delaware from January 2001 to July 2002. His research interests include Internet protocols and technologies, network measurements and their applications, overlay and multi-homed networks, intelligent route control, router buffer sizing, service provisioning and traffic engineering, routing security, and biology-inspired network architectures. He received the NSF CAREER award in 2003.

Mostafa H. Ammar is a Regents' Professor with the College of Computing at Georgia Tech where he has been since 1985. His research interests are in the area of computer network architectures, protocols and services. He received the S.B. and S.M. degrees from the Massachusetts Institute of Technology in 1978 and 1980, respectively and the Ph.D. in Electrical Engineering from the University of Waterloo, Ontario, Canada in 1985. For the years 1980–1982, he worked at Bell-Northern Research (BNR), Ottawa, Canada. He was the co-recipient of the Best Paper Awards at the 7th WWW conference for the paper on the "Interactive Multimedia Jukebox" and the 2002 Parallel and Distributed Simulation (PADS) conference for the paper on "Updateable Network Simulation". He served as the Editor-in-Chief of the IEEE/ACM Transactions on Networking from 1999 to 2003. He is a Fellow of the IEEE and a Fellow of the ACM.