Is the Round-trip Time Correlated with the Number of Packets in Flight ?*

Saad Biaz
Computer Science and Software Engineering
Auburn University
biazsaa@eng.auburn.edu

Nitin H. Vaidya
Dept. of Electrical and Computer Engineering,
and Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
nhv@uiuc.edu

ABSTRACT

TCP uses packet loss as a feedback from the network to adapt its sending rate. TCP keeps increasing its sending rate as long as no packet loss occurs (unless constrained by buffer size). Alternative congestion avoidance techniques (CATs) have been proposed to avoid such "aggressive" behavior. These CATs use simple statistics on observed round-trip times and/or throughput of a TCP connection in response to variations in congestion window size. These CATs have a supposed ability to detect queue build-up.

The objective of this paper is to question the ability of these CATs to reliably detect queue build-up under real network conditions. For this purpose, the sample coefficient of correlation between round-trip time and the number of packets in flight is analyzed for 14,218 connections over 737 Internet paths. These coefficients of correlation were extracted from a set of *tcpdump* traces collected by Vern Paxson.

The coefficients of correlation measured confirm that the correlation between RTT and window size is often weak.

Categories and Subject Descriptors

C.4 [**Performance of Systems**]: Measurements techniques; C.2.2 [**Computer Communication Networks**]: Network Protocol

General Terms

Congestion control

Keywords

TCP, congestion predictors, roundtrip time, congestion window size, correlation

*Research supported in part by National Science Foundation grant ANI 01-96413.

1. INTRODUCTION

TCP is a popular protocol for reliable data delivery in the Internet. TCP is robust in that it can adapt to disparate network conditions [7]. TCP senders use packet loss as a feedback from the network to adjust a congestion window size to adapt its offered load to the network conditions. A TCP sender keeps increasing its offered load (i.e., congestion window size) until packet loss occurs (unless limited by receiver buffer size). To avoid such "provoked" losses, several congestion avoidance techniques (CATs) [8, 5, 16] attempt to determine the *appropriate* load on the network by using simple statistics on observed round-trip times (RTT) and/or observed throughput of a TCP connection. These techniques attempt to perform congestion avoidance by detecting queue build-up in the network, thus preventing congestion losses. Such congestion avoidance techniques, if reliable, can potentially be used to distinguish packet losses due to congestion from those due to other causes (such as transmission errors) [3]. These techniques were tested in order to distinguish congestion losses from wireless transmission losses. The results were quite poor [3] raising a natural question about the CATs's ability to detect queue build-up under real network conditions. Andren *et. al* [1] also showed that there is a weak correlation between loss and delay. More recently, Martin *et. al* [10, 9] showed that (1) the congestion information contained in RTT variations is not sufficient to reliably predict packet loss and (2) that the impact of one flow on the congestion level of the path is minimal when this flow consumes a small fraction of the resources at the bottleneck.

This paper makes an attempt to evaluate the rationale upon which congestion avoidance techniques (CATs) are built. For this purpose, the coefficient of correlation between the round trip time and the amount of *data in flight* is evaluated for 14,218 TCP connections over 737 Internet paths – amount of *data in flight* is the amount of data sent by a TCP sender for which an acknowledgement has not been delivered to the sender. These data collected in [12] are quite old when many "fast" links were T1 links! Bandwidth on most links has dramatically increased since the collection of this data. But these data are unique because they span many paths with different round trip times.

The CATs are designed based on the assumption that an increase of the load by a user would increase the round trip delay *when queue build-up occurs.* Therefore, it may be inferred that, there should be some correlation between the load (amount of data in flight) variations and the observed round trip time variations. It is then interesting to measure the coefficient of correlation between the amount of data in flight and the round trip time observed by the sender. These results confirm and explain the results obtained by Padhye *et al.* [11] and by Hengartner *et al.* [6].

To derive the TCP throughput model [11], Padhye assumed the independence between the window size and the round-trip time. Padhye *et al.* verified the hypothesis on a small set of TCP connections by measuring the correlation between window size and round-trip time. He observed a weak correlation on high speed links and a strong correlation up to 0.95 on slow links (modem connection with 28.8 Kbps).

In [6], Hengartner *et al.* decomposed TCP Vegas into the various novel mechanisms proposed by Brakmo *et al.* [5] and assessed the effect of each of these mechanisms on performance. Hengartner *et al.* show that TCP Vegas' innovative congestion avoidance mechanism has only a minor influence on throughput. Moreover, Hengartner show that TCP Vegas's innovative congestion avoidance mechanism has a slightly **negative** influence on throughput when the bottleneck link has **high bandwidth**. This paper explains this last surprising result.

To our knowledge, there has been no published work relating the study of this correlation on a large set of connections. The rest of this paper is organized as follows. Section 2 presents the terminology and notations used in this paper. Section 3 summarizes three congestion avoidance techniques (CATs). Experiments and results are discussed in Section 4. Conclusions are presented in Section 5.

2. TERMINOLOGY AND NOTATIONS

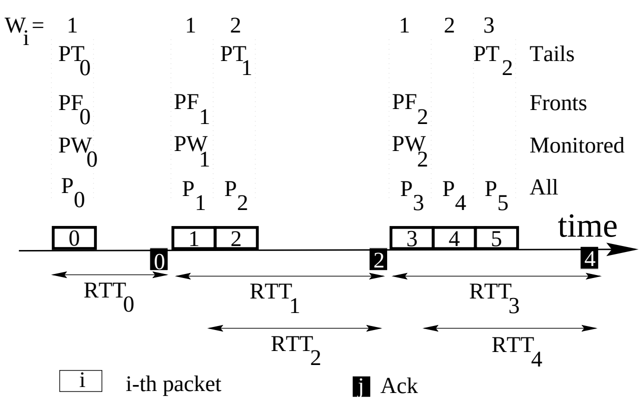


Figure 1: Illustration for the notations

Figure 1 illustrates most of the terms used in this paper; these terms are defined below. Figure 1 represents a typical beginning of a connection where packets 0 through 5 are sent by a sender. SYN/ACK packets are not considered. The sender sends first packet 0 (white rectangle above the time line) and awaits for an acknowledgement (small black packet below the time line) from the receiver before sending packet 1. When ack 0 is received, the congestion window is increased by one Maximum Segment Size (MSS)[1]. Therefore, two packets, 1 and 2, are sent back-to-back (burst). Packet 1 is the *front* and packet 2 is the *tail* of this burst. The sender then waits for an ack, until ack 2 is received. (Note that this example assumes delayed acknowledgement.) Again the congestion window size is incremented and the sender may send a burst of three packets 3, 4, and 5. Notations are defined below.

- packet P_i: P_i is the i-th packet sent by the sender, excluding any retransmitted packets. Essentially, the set of packets P_i contains those packets for which round-trip time (RTT) could be calculated using traced information at the sender. (Note that TCP senders do not, in fact, keep track of the RTT of each P_i.) To avoid the ambiguity of computing the round trip time when a packet is transmitted more than once, retransmitted packets are omitted in the analysis. In Figure 1, packets numbered $0, 1, 2, \cdots, 5$ are, respectively, $P_0, P_1, P_2, \cdots, P_5$. In this figure, there are no retransmitted packets. In the following, a packet P is said "relevant" if P $= P_i$ for some i.

- packet PW_j: TCP sender measures the round-trip time (RTT) for one packet "per window". PW_j is the j-th packet for which an RTT sample is collected. PW_j is referred as a "monitored" packet. In Figure 1, TCP sender monitors only packets P_0, P_1 and P_3 and determines their RTT. Therefore, packets PW_0, PW_1, and PW_2 are respectively the packets P_0, P_1 and P_3.

- packet PF_k: packets sent by a TCP sender are divided into "bursts" – during a burst, packets are sent back-to-back (*line-rate burst*) by the TCP sender. Later in the paper, the method of how a burst is identified is described. PF_k is the k-th *relevant* packet that appears at the front of a burst. In Figure 1, there are three bursts: a burst consisting of packet 0, a burst consisting of packets $1, 2$, and a burst consisting of packets $3, 4, 5$. The front packets for these bursts are respectively $PF_0 = P_0$, $PF_1 = P_1$, and $PF_2 = P_3$.

- packet PT_m: PT_m is the m-th *relevant* packet that appears at the tail of a burst. In Figure 1, the tail packets are $PT_0 = P_0$, $PT_1 = P_2$, and $PT_2 = P_5$.

- $W(P_i)$, amount of data in flight[2] for P_i: $W(P_i)$ is the amount of data transmitted (including P_i) by the TCP sender for which an acknowledgement is not received by the sender until after P_i is transmitted. On top of Figure 1, the value $W(P_i)$ is provided for each packet P_i. For example, when P_0 is sent, the number of packets in flight is $W(P_0) = 1$ packet. When P_5 is sent, $W(P_5)$ is equal to 3 packets

- Round-trip time $RTT(P_i)$ for packet P_i: $RTT(P_i)$ is the duration from the time when P_i is transmitted, until the time when an acknowledgement for P_i is received by the TCP sender. In Figure 1, for brevity, $RTT(P_i)$ is denoted as RTT_i.

- $Sign(x)$: $Sign(x) = \frac{x}{|x|}$ is respectively equal to 1, 0, or -1 if x is positive, null, or negative.

[1]Ack i acknowledges receipt of data through packet i; for simplicity, this notation is different from TCP syntax.

[2]Although the amount of data is measured in bytes in this analysis, in the discussion, it may occasionally be stated as a number of packets.

- Connection C_l is the l-th TCP connection in the TCP dataset used in this study with l from 1 to 14,218.
- Bandwidth $B(C_l)$ is an estimate of the bandwidth of the bottleneck link for the connection C_l.

Now, some coefficients of correlation are defined for connection C_l to be evaluated during these experiments.

- $\rho(C_l, RTT(P_i), W(P_i))$ is the coefficient of correlation between $RTT(P_i)$ and $W(P_i)$ for the *relevant* packets P_i.
- $\rho W(C_l, RTT(PW_i), W(PW_i))$ is similarly defined for the *monitored* packets PW_i.
- $\rho F(C_l, RTT(PF_i), W(PF_i))$ is similarly defined for the packets PF_i at the *front* of bursts.
- $\rho T(C_l, RTT(PT_i), W(PT_i))$ is similarly defined for the packets PT_i at the *tail* of bursts.

The coefficient of correlation $\rho(C_l, x_i, y_i)$ for connection C_l is defined as [15]:

$$\rho(C_l, x_i, y_i) = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2 \sum_i (y_i - \bar{y})^2}}$$

where $\bar{x}$ and $\bar{y}$ are, respectively, the means over x_i's and y_i's. Since sample coefficient of correlation is a commonly used metric to evaluate the degree of relationship among variables, this metric is chosen for this study as well. The above coefficients of correlation are calculated between values of RTT and W samples. Similar coefficients of correlation for the *direction* of change (increase or decrease) in the values of RTT and W are defined.

- $\rho_v(C_l, Sign(\delta RTT), Sign(\delta W))$ is the coefficient of correlation between $Sign(RTT(P_i) - RTT(P_{i-1}))$ and $Sign(W(P_i) - W(P_{i-1}))$ for the packets P_i
- $\rho W_v(C_l, Sign(\delta RTT), Sign(\delta W))$ is similarly defined for the packets PW_i
- $\rho F_v(C_l, Sign(\delta RTT), Sign(\delta W))$ is similarly defined for the packets PF_i
- $\rho T_v(C_l, Sign(\delta RTT), Sign(\delta W))$ is similarly defined for the packets PT_i

3. THE CATS

The congestion avoidance techniques (CATs) are motivated by the following expectation of network behavior [8]. As illustrated in Figure 2, when network load is small, increasing the load should result in a comparable increase in network throughput with only a small increase in round-trip times (RTT). At some point, when the load is large enough, packets start queuing at the bottleneck. Therefore, increasing the load further should result in a smaller increase in throughput, and a larger increase in round-trip times (this occurs at the "knee" of the load-throughput curve). If the load is increased further, at some point, the network throughput should drop sharply, while round-trip times should become extremely large.

Three CATs are summarized below. The CATs are implicitly based on the notion that there will be some *response* from the network to a congestion window size ("load") *change* for a TCP connection. The CATs measure this response as a function of round-trip times and/or throughput, and recommend reducing or increasing the congestion window based on the observed response. The various functions used in these heuristics are motivated by the curves in Figure 2.

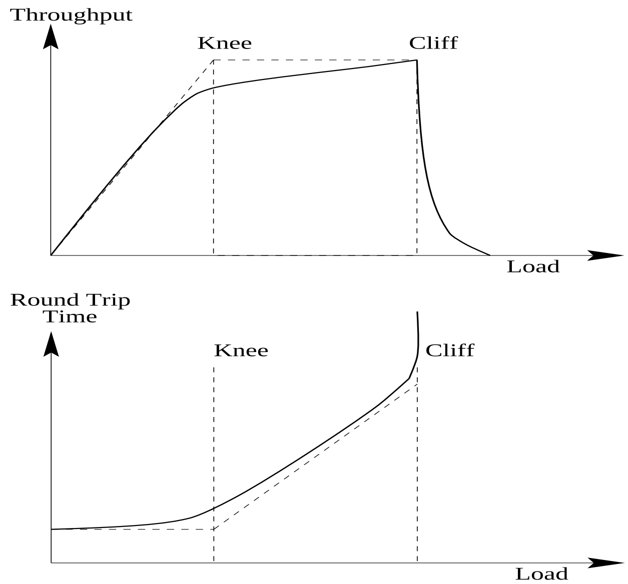


Figure 2: Throughput and RTT versus network load [8]

TCP-Vegas [5] requires a TCP sender to keep track of the $BaseRTT$, defined as the minimum of all RTTs measured during the TCP connection. When acknowledgement for the i-th monitored packet PW_i is received, the sender compares the quantity

$$BaseRTT \left(\frac{W(PW_i)}{BaseRTT} - \frac{W(PW_i)}{RTT(PW_i)} \right)$$

with two thresholds α and β, where $\alpha < \beta$. If the above quantity is less than α, then TCP congestion window is increased; if it is greater than β, the congestion window size be decreased; otherwise, the window is held constant.

Wang and Crowcroft [16] proposed a congestion avoidance technique based on the *Normalized Throughput Gradient* (NTG). This technique evaluates the gain in throughput after an increase of the window size. If the increase of throughput is larger than half the throughput observed for the first packet, then the congestion window may be increased.

Jain [8] proposed a congestion avoidance technique based on *Normalized Delay Gradient* (NDG). This technique looks only at the signs of variations of the round trip time and the congestion window size. If an increase (respectively, decrease) of the window size results in an increase (respectively, decrease) of the round trip time, then the congestion window size is decreased. Otherwise, the congestion window size is increased.

In[3], Biaz and Vaidya hypothesized that good heuristics for detecting congestion (or queue build-up) can potentially be applied to distinguish packet losses due to congestion from those due to transmission errors in wireless networks. However, their analysis showed that the CATs cannot be used to determine the cause of a packet loss with adequate accuracy. Therefore, this paper analyzes the correlation between round-trip time and the amount of data in flight. If this correlation is weak, then the CATs cannot be very useful to draw conclusions about the cause of a packet loss – recall that the three congestion avoidance techniques described above use simple functions of recent RTT and W samples.

4. EXPERIMENTS

Data collected in [12] is used to study end-to-end Internet dynamics. This set of data is old (1994-1995), but is unique by the number of paths probed. Since then, bandwidths on these paths increased tremendously. However, the lessons drawn from this set can be extended to today's bandwidths. Paxson [12] collected two sets of data $\mathcal{N}_1$ and $\mathcal{N}_2$. $\mathcal{N}_1$ and $\mathcal{N}_2$ are *tcpdump*[14] traces collected over 37 sites to study end-to-end packet dynamics. The measurements are extensively described in [12]. This study uses the set $\mathcal{N}_2$ which contains about 20,000 *tcpdump* traces for *bulk transfers* of 100 KBytes between pairs of sites among the 37 sites studied. For each transfer, $\mathcal{N}_2$ contains the *tcpdump* traces both at the sender and at the receiver. This study exploits only the traces at the sender. For technical reasons[3], *tcpdump* traces for only 14,218 transfers were analyzed. These 14,218 TCP connections were between 31 different sites across U.S.A., Europe and Australia. Not all sites send to all sites, therefore these 14,218 TCP connections span only 737 paths. A *tcpdump* trace for a TCP connection allows the computation of the round trip time (RTT) and data in flight (W) for every relevant (P_i) packet. In the following, data for various coefficients of correlation is presented. Recall that some of the coefficients are related to packets at the front or tail of bursts. For this analysis, the bursts are identified by first finding the minimum delay D_{min} between the transmission of two successive packets by the TCP sender. Any two successive packets are considered as sent back-to-back (and, hence, part of the same burst) if the delay between their transmission is less than $1.8 \times D_{min}$. The choice of 1.8 multiplier for D_{min} was somewhat arbitrary, and any other multiplier between 1 and 2 (but not too close to 1) should have sufficed as well. If too close to 1, too few packets appear as back-to-back packets, even during the slow start phase.

To determine if and how bottleneck bandwidth affects the coefficients of correlation, the TCP connections are partitioned into two sets: (a) connections traversing a slow bottleneck link, and (b) connections traversing a fast bottleneck link. Section 4.1 describes how the two sets of connections are identified.

4.1 Partitioning the Set of Connections

Paxson developed a method called *packet bunch mode* [12] to draw an estimate from the tcpdump trace of the bottleneck link bandwidth. This method provides for each connection C_l an estimate $B(C_l)$ of the bottleneck link bandwidth. Figure 3 presents the frequency distribution of $B(C_l)$. The x-axis represents the bandwidth in steps of 8 KBytes/s (8 KBps). For a bandwidth b on the horizontal axis, the value Frequency(b) on the vertical axis represents the fraction of

[3]Tcpdump traces for some sites (e.g, sri[12]) could not be decoded. The packets did not seem consistent, yielding sometimes negative round trip times. Details can be found in [12].

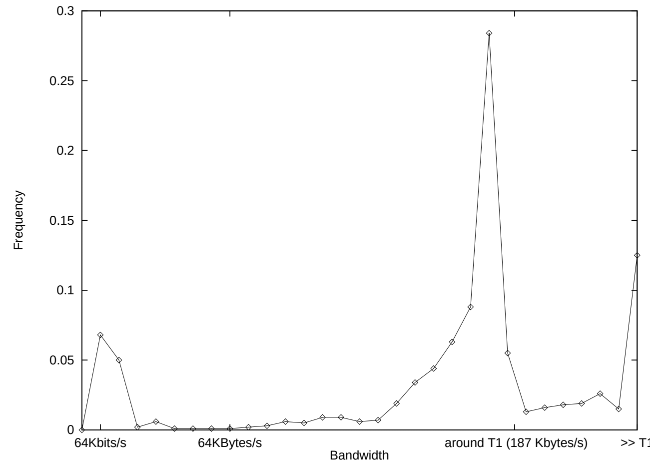
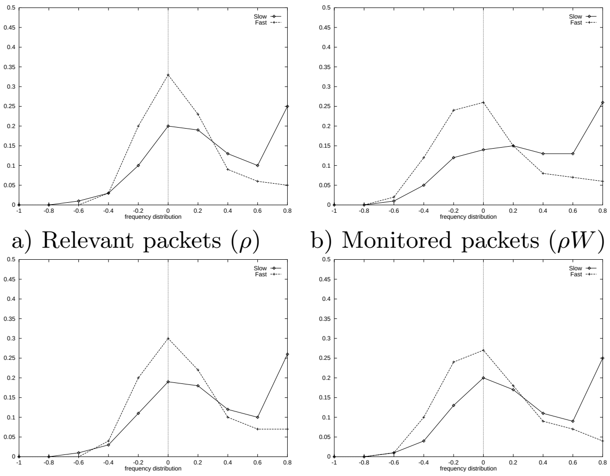


Figure 3: Frequency distribution of connections by bottleneck link speed

connections through a bottleneck bandwidth in the interval $[b - 8KBps, b)$. Figure 3 exhibits two peaks. The rightmost peak is not a real one: it relates to all bandwidths higher than T1 (in this set of bandwidths higher than T1, no bandwidth dominates). They correspond approximately to 64 Kbits/s and the T1 bandwidth. A separation line is drawn between *slow* and *fast* bottleneck links at 64 KBytes/s. Since there are not many connections between the two peaks at 64 Kbits/s and T1, the actual location of the separation line between these two peaks is not very critical (so long as it is not close to either peak). In the rest of this paper, a connection C_l would be said to be a *slow* connection, if its bottleneck link is slow (i.e., $B(C_l) \leq 64$ KBytes/s); else C_l would be said to be a *fast* connection. Similarly, a path (or route) that contains a slow (fast) bottleneck link will be said to be a slow (fast) path. Note that Figure 3 relates to an old set of data: today, bandwidth dramatically increased and peaks at T3 would certainly appear.

4.2 Correlation between RTT and W

Figure 4 presents four plots: one for each population of packets (relevant packets P_i on Figure 4.a, monitored packets PW_i on Figure 4.b, front packets PF_i on Figure 4.c, and tail packets PT_i Figure 4.d). Each plot represents the frequency distribution of the coefficients of correlation for a given population of packets separately for slow and fast paths. On the horizontal axis, values vary from -1 to 0.8. in steps of 0.2. A point at (x, y) on the curve for fast (resp. slow) pathss indicates the fraction of fast (resp. slow) TCP connections that have a coefficient of correlation in the interval $[x, x+0.2)$ is y. For example, on Figure 4.b, observe that approximately 26% of the connections on slow links have a coefficient of correlation between 0.8 and 1. For fast bottleneck links, only 6% of the connections have a coefficient of correlation between 0.8 and 1. First, observe that the curves are quite similar for all types of packet populations. Also observe that for slow bottleneck links, there is a significant (about 35%) proportion of connections with a high (≥ 0.6) coefficient of correlation. This supports the general opinion that there exists a higher correlation between round trip time and the amount of data in flight on a slow link (than on a fast link). On fast links, only about 11% of the connections exhibit a coefficient of correlation larger than 0.6.



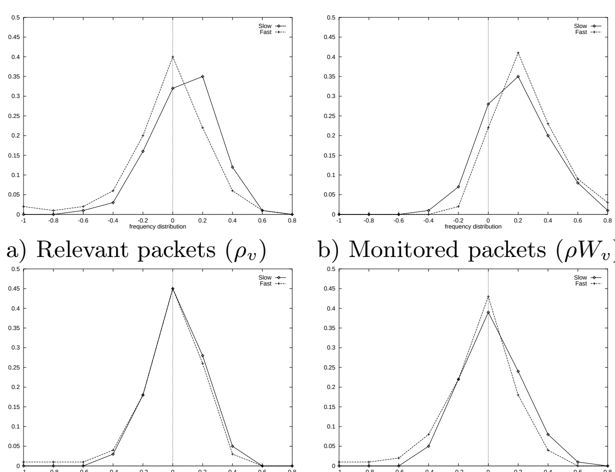
a) Relevant packets (ρ) b) Monitored packets (ρW)

c) Front packets (ρF) d) Tail packets (ρT)

Figure 4: Frequency Distribution of Coefficients of Correlation

4.3 Correlation in *Direction* of Change

The problem with the coefficients of correlation for values of RTT and W is that they may be "dominated by outliers" (i.e., RTT spikes) [13]. To avoid spikes that skew the coefficient of correlation, the correlation between the *direction* of changes is considered. The question is: does an increase (resp. decrease) in the number of packets in flight result in an increase (resp. decrease) in the roundtrip time? To determine if the statistical correlation between the "direction" of change (increase or decrease) in the RTT in response to change in the amount of data in flight is better, four additional coefficients of correlation related to the direction of change were measured. Figure 5 presents our results for the



a) Relevant packets (ρ_v) b) Monitored packets (ρW_v)

c) Front packets (ρF_v) d) Tail packets (ρT_v)

Figure 5: Frequency distribution of coefficients of correlation related to the direction of change.

correlation between the direction of variations. For example, Figure 4.a presents the frequency distribution of the coefficients of correlation between the direction of variations of $RTT(P_i)$ and $W(P_i)$, i.e., $\rho_v(C_l, Sign(\delta RTT), Sign(\delta W))$ for the population of relevant packets P_i. First, we observe that the results are quite similar for all populations of packets, but the curves of Figure 5.b are slightly shifted to the right: correlation with the population PW_i (one packet per window) is higher than with the population P_i of all packets. This confirms that timing all packets is not very helpful. Second, TCP connections with a high coefficient of correlation (more than 0.80) are rare. This shows clearly that our results in Figure 4 were somewhat skewed. Finally, observe that the coefficients of correlations are similar for slow and fast TCP connections. For both, only 30% of the connections have a coefficient of correlation larger than 0.4.

4.4 Can ρ Characterize a Path?

Given a path (i.e., route), an interesting question is whether the coefficients of correlation for all connections along that path are similar. Figure 6 presents the coefficients of correlation of all connections along all 138 "slow" paths. The values on x-axis represent the slow paths numbered sequentially from 0 to 137. For each path number, the coefficient of correlation of all connections on this path is plotted. There are about 15 to 20 points per path along a vertical line.

The y-axis represents the coefficient of correlation ρW^4 for packets PW_i (one packet per window). Each point in the figure represents the coefficient of correlation for one connection. Observe that, for connections traversing the same path, the coefficients of correlation span a large interval on a vertical line. Results for the fast paths are similar and

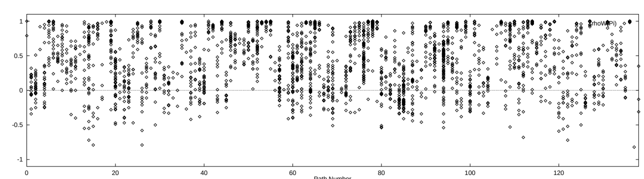


Figure 6: $\rho W(C_l, RTT(PW_i), W(PW_i))$ **for slow paths**

can be found in [4, 2]. If paths (routes between two sites) do not change frequently, then the coefficient of correlation ρW is not a good characteristic of a path because for the *same* path, the coefficients of correlation span a large interval. In Figure 7 , similar results are plotted for coefficient of correlation $\rho W_v(C_l, Sign(\delta RTT), Sign(\delta W))$ where only signs of variations for the packets PW_i are considered.

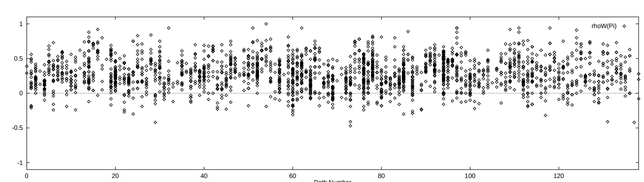


Figure 7: $\rho W_v(C_l, \frac{\delta RTTi}{|\delta RTTi|}, \frac{\delta Wi}{|\delta Wi|})$ **for slow paths**

[4]Precisely $\rho W(C_l, RTT(PW_i), W(PW_i))$.

Observe that for slow paths (as for fast paths [4, 2]) on Figures 7, a large majority of connections have a positive correlation ρW_v when only direction of change in RTT and W is considered.

5. DISCUSSION

Suppose that a car has a special control pedal : the car accelerates with probability p, and slows down with probability $(1-p)$, whenever the control pedal is pushed. The question is *what is the range of values for p which allows the design of a reliable control system*? Intuitively, a value of p between 0.4 and 0.6 would make it hard to design such a system. The design a good congestion avoidance technique under real network conditions must cope with a system under the control of the special pedal as described above. When a TCP sender increases its load (*push the pedal*), the round trip time may either *increase* (as one might expect) or it may also decrease, due to several factors. The most important factor is that RTT observed by a given TCP connection is dependent on other traffic carried by the network, not just on the actions of this TCP connection, especially when this TCP connection consumes a small fraction of the available bandwidth.. Other possible factors include uncertainty of interrupt services on the OS at the endpoints, and vagaries of the transport protocol itself (delayed acknowledgements for example that can add up to 200ms to the RTT). The coefficients of correlation measured confirm that the correlation between RTT and window size is often weak.

To be fair, one may argue that the coefficient of correlation should be expected to be small (near 0) when there is no queue build-up. Whenever the TCP sender increases its load, the round trip time may not increase if there is no queue build-up. This argument may potentially be valid for a very high bandwidth bottleneck link. However, it takes only 4 TCP connections with round trip time 150ms and five packets in flight for each to fill up a T1 pipe (with a 1500 bytes segment size). By a similar argument, if the TCP connection is somehow able to maintain its load below the available bandwidth, then the correlation coefficients could be small. Such a situation could occur, for instance, when the chosen socket buffer size is relatively small.

Round-trip time measured by TCP is imprecise and bears a high random component *independent* of the actions (increasing or decreasing the load) of the sender. The results in this paper suggest that, due to such factors, there is no strong relation between the variations of RTT and the variations of congestion window size. These results explain the surprising result of Hengartner *et al.* [6] about the minor influence of TCP Vegas's novel congestion avoidance scheme on performance.

The measurements of correlation in the *directions* of change in RTT and W (e.g., $\rho_v(C_l, Sign(\delta RTT), Sign(\delta W))$ exhibit a positive correlation between the signs of variations. This is specially true for the *monitored* packets PW_i where more than 88% of the connections have a positive correlation. This confirms that the network response (i.e., RTT) is in general somewhat "sensitive" to the load presented by a TCP connection. However, the correlation is not strong enough to build "smart" congestion avoidance techniques which would **reliably** detect queue build-up using short-term statistics on round-trip times (RTT) and/or window size (W) unless the TCP flow consumes a large fraction of the available bandwidth.

6. ACKNOWLEDGEMENTS

The authors would like to thank Vern Paxson for sharing the data he collected and for answering the many questions they had. Many thanks to the reviewers, whose comments were insightful and helpful. Research reported was supported in part by the Fulbright Excellence Program and the U.S. National Science Foundation grant ANI 01-96413.

7. REFERENCES

[1] J. Andren, M. Hilding, and D. Veitch. Understanding end-to-end internet traffic dynamics. In *Proc. IEEE Globecom Internet Mini-Conference, Sydney, Australia*, pages 1118–1122, Nov. 1998.

[2] S. Biaz. *Heterogeneous Data Networks: Congestion or Corruption?* PhD thesis, Texas A&M University, College Station, Aug. 1999.

[3] S. Biaz and N. H. Vaidya. Distinguishing congestion losses from wireless transmission losses : A negative result. In *IEEE 7th IC3N'98, Lafayette, LA USA.*

[4] S. Biaz and N. H. Vaidya. Is the round-trip time correlated with the number of packets in flight? Technical Report 99-006, Department of Computer Science, Texas A&M University, Mar. 1999.

[5] L. Brakmo and S. O'Malley. TCP-vegas : New techniques for congestion detection and avoidance. In *ACM SIGCOMM'94*, pages 24–35, Oct. 1994.

[6] U. Hengartner, J. Bolliger, and T. Gross. TCP Vegas revisited. In *IEEE Infocom'00*, Mar. 2000.

[7] V. Jacobson. Congestion avoidance and control. In *ACM SIGCOMM'88*, pages 314–329, Aug. 1988.

[8] R. Jain. A delay-based approach for congestion avoidance in interconnected heterogeneous computer networks. *ACM CCR*, 19:56–71, 1989.

[9] J. Martin, A. Nilsson, and I. Rhee. Delay-based congestion avoidance for TCP. *IEEE/ACM Transactions on Networking*, 11(3).

[10] J. M. A. Nilsson and I. Rhee. The incremental deployability of rtt-based congestion avoidance. In *ACM SIGMETRICS'2000, Santa Clara, CA*, pages 134–144, 2000.

[11] J. Padhye, V. Firoiu, D. Towsley, and J. Krusoe. Modeling TCP throughput: A simple model and its empirical validation. In *ACM SIGCOMM '98 conference on Applications, technologies, architectures, and protocols for computer communication*, pages 303–314, Vancouver, CA, 1998.

[12] V. Paxson. *Measurements and Analysis of End-to-End Internet Dynamics.* PhD thesis, University of California, Berkeley, Apr. 1997.

[13] V. Paxson. Private email, Jan. 1999.

[14] W. R. Stevens. *TCP/IP Illustrated: The Protocols (v.1).* Reading, MA: Addison-Wesley, 1994.

[15] K. S. Trivedi. *Probability and Statistics with Reliability, Queueing, and Computer Science Applications.* Prentice Hall, 1988.

[16] Z. Wang and J. Crowcroft. A new congestion control scheme : Slow start and search (tri-s). *ACM Computer Communication Review*, 21:32–43, Jan. 1991.