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

Saâd Biaz[†] Nitin H. Vaidya
Department of Computer Science
Texas A&M University
College Station, TX 77843-3112, USA
E-mail: {saadb,vaidya}@cs.tamu.edu
Phone : (409) 845-5007
Fax : (409) 847-8578

Technical Report 99-006 (March 9, 1999)

Abstract

TCP uses packet loss as a feedback from the network to adapt its sending rate. TCP keeps increasing its sending rate regardless of the network congestion state as long as no loss occurs (unless constrained by buffer size). Alternative congestion avoidance techniques (CATs) have been proposed to avoid such "agressive" 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. Such ability may be used to distinghish congestion losses from transmission losses. A previous study shows that these CATs do not yield interesting results for diagnosing the real reason of a loss.

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, we analyze the sample coefficient of correlation between round-trip time and the number of packets in flight for 14,218 connections over 737 Internet paths. These coefficients of correlation were extracted from a set of tcpdump *traces collected by Vern Paxson.*

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 [3]. Packet losses are the only feedback TCP uses to adjust its congestion window size in order to adapt its load to the network conditions.

When a packet loss occurs, TCP assumes that it is a congestion loss and therefore reduces its congestion window size. This response is appropriate if most losses are due to congestion. This is not the case for wireless links where packets may be lost due to transmission errors. In such case, TCP performance can be unnecessarily reduced.

The TCP sender keeps increasing its load regardless of the network congestion state until packet loss occurs (unless limited by buffer size). To avoid such "provoked" losses, several congestion avoidance techniques (CATs) [4, 2, 9] attempt to determine the 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.

In addition, the ability of the CATs to detect queue build-up may be used to differentiate packet losses due to congestion from those due to other causes (such as transmission errors) [1]. We tried to use these techniques to distinguish congestion losses from wireless transmission losses. The results were quite poor [1] raising a natural question about the CATs's ability to detect queue build-

[*] Research reported is supported in part by the Fulbright Program, National Science Foundation grant CDA 9529442, and Texas Advance Technology Program grant 010115-248.

[†] On leave from the Ecole Supérieure de Technologie Fès (MOROCCO).

up under real network conditions. Proposers of the CATs acknowledge that a high randomness in the network may affect the performance of the CATs [4, 2, 9]. However, we did not *a priori* know how the randomness on real networks would impact efficacy of the CATs.

This paper makes an attempt to evaluate the rationale upon which congestion avoidance techniques (CATs) are built. For this purpose, we study the sample coefficient of correlation between the round trip time and the number of packets in flight for 14,218 connections over 737 Internet paths.

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 (number of packets in flight) variations and the observed round trip time variations. It is then interesting to measure the sample coefficient of correlation bewteen the number of packets in flight and the round trip time observed by the sender. To our knowledge, there has been no published work relating the study of this correlation on a large set of connections.

A caveat: The correlation measurements reported here are based on data collected for TCP connections. Dynamics of TCP may have biased the measured correlations in some cases, as discussed later.

Rest of this paper is organized as follows. Section 2 presents the terminology and notations used in this paper. Section 3 summarizes the three congestion avoidance techniques (CATs). Experiments and results are discussed in Section 4. Conclusions are presented in Section 5.

2 Terminology and Notations

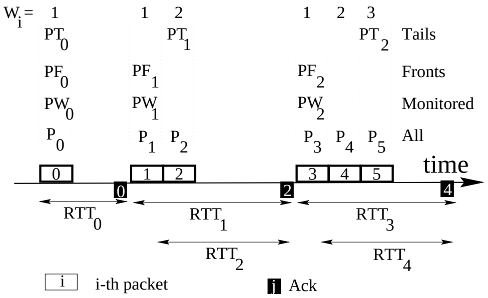

Figure 1. Illustration for the notations

Figure 1 illustrates most of the terms used in this paper. Figure 1 represents a typical beginning of a connection where packets 0 through 5 are sent by a sender. We do not consider here the SYN/ACK packets. 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 the ack 0 is received, the congestion window is increased by one. Therefore, two packets, 1 and 2, will be sent back-to-back (burst). Packet 1 is the *front* and packet 2 is the *tail*. The sender awaits then for an ack until ack 2 is received. Again the congestion window size is incremented and the sender may send a burst of three packets 3, 4, and 5.

- i-th packet P_i: P_i is the $i - th$ packet sent by the sender, but not retransmitted. This is to avoid the ambiguity of computing the round trip time when a packet is transmitted more than once. For each packet P_i, we have the round trip time and the number of packets in flight (defined below). In Figure 1, packets numbered $0, 1, 2 \ldots 5$ are respectively the packets P_0, P_1, $P_2 .. P_5$.

  If a time-out occurs while waiting for the acknowledgement for a packet P_i, then the round-trip time for that packet P_i is not used in our calculations.

  The packets P_i are numbered sequentially starting from 0, excluding the packets which time-out.

  For the i-th monitored packet P_i, we define two parameters below, to be used in presenting the CATs.

- j-th packet PW_j: at any time, only one packet “per window” sent by the sender is monitored (by the CATs).

  In Figure 1, only packets 0, 1 and 3 are monitored. Therefore, packets PW_0, PW_1, and PW_2 are respectively the packets 0, 1 and 3. Note that the set of packets PW_j is a subset of the set of packets P_i.

- k-th packet PF_k: This is a packet which is the *front* packet of a burst of b ($b \geq 1$) packets sent back-to-back. In Figure 1, we have three bursts : a burst with packet 0, a burst with packets $1, 2$, and a burst with 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).

- m-th packet PT_m: This is a packet which is the *tail* packet of a burst of b ($b \geq 1$) packets sent back-to-back. In Figure 1, the tail packets are PT_0 (P_0), PT_1 (P_2), and PT_2 (P_5).

- Number of packets in flight W_i for the i-th packet: W_i is the amount of data transmitted (including the i-th packet) but not yet acknowledged. On top of Figure 1, we provide the value W_i for each packet i. For example, when PT_0 is sent, the number of packets in flight is 1. When PT_2 is sent, W_i is equal to 3.

- Round-trip time RTT_i for i-th packet : RTT_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 sender (see Figure 1).

- Connection C_l : it is the l-th connection with l from 1 to 14,218.

- Coefficients of correlation for connection C_l:

  - $\rho(C_l, RTT_i, W_i)$: this is the sample coefficient of correlation between RTT_i and W_i for the packets P_i (all packets sent, but not retransmitted)
  - $\rho W(C_l, RTT_i, W_i)$ this is the sample coefficient of correlation between RTT_i and W_i for the packets PW_i (one packet per window)
  - $\rho F(C_l, RTT_i, W_i)$: this is the sample coefficient of correlation between RTT_i and W_i for the packets PF_i (burst front)
  - $\rho T(C_l, RTT_i, W_i)$: this is the sample coefficient of correlation between RTT_i and W_i for the packets PT_i (burst tail).

  The sample coefficient of correlation $\rho(C_l, X_i, Y_i)$ ($n = 1 \ldots n$) for two random variables X_i and Y_i ($i = 0 \ldots n$) for connection C_l is defined as [8]:

  $$\rho(C_l, X_i, Y_i) = \frac{\sum_{i=0}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=0}^{n}(x_i - \bar{x})^2 \sum_{i=0}^{n}(y_i - \bar{y})^2}}$$

  where $\bar{x}$ and $\bar{y}$ are respectively the means for variables X_i and Y_i.

- Coefficients of correlation for connection C_l:

  - $\rho_v(C_l, \frac{\delta RTTi}{|\delta RTTi|}, \frac{\delta Wi}{|\delta Wi|})$: this is the sample coefficient of correlation between $\frac{RTT_i - RTT_{i-1}}{|RTT_i - RTT_{i-1}|}$ and $\frac{W_i - W_{i-1}}{|W_i - W_{i-1}|}$ for the packets P_i (all packets sent, but not retransmitted)
  - $\rho W_v(C_l, \frac{\delta RTTi}{|\delta RTTi|}, \frac{\delta Wi}{|\delta Wi|})$:this is the sample coefficient of correlation between $\frac{RTT_i - RTT_{i-1}}{|RTT_i - RTT_{i-1}|}$ and $\frac{W_i - W_{i-1}}{|W_i - W_{i-1}|}$ for the packets PW_i (one packet per window)
  - $\rho F_v(C_l, \frac{\delta RTTi}{|\delta RTTi|}, \frac{\delta Wi}{|\delta Wi|})$:this is the sample coefficient of correlation between $\frac{RTT_i - RTT_{i-1}}{|RTT_i - RTT_{i-1}|}$ and $\frac{W_i - W_{i-1}}{|W_i - W_{i-1}|}$ for the packets PF_i (burst front)
  - $\rho T_v(C_l, \frac{\delta RTTi}{|\delta RTTi|}, \frac{\delta Wi}{|\delta Wi|})$:this is the sample coefficient of correlation between $\frac{RTT_i - RTT_{i-1}}{|RTT_i - RTT_{i-1}|}$ and $\frac{W_i - W_{i-1}}{|W_i - W_{i-1}|}$ for the packets PT_i (burst tail).

  With $\rho_v(C_l, \frac{\delta RTTi}{|\delta RTTi|}, \frac{\delta Wi}{|\delta Wi|})$, we measure only how RTT_i signs of variations are related to W_i signs of variations.

- Bandwidth $B(C_l)$: this is an estimate of the bandwidth of the bottleneck for the connection C_l.

3 Congestion Avoidance Techniques

The congestion avoidance techniques are motivated by the following expectation of network behavior [4]. 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 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 congestion window based on the observed response.

TCP-Vegas [2] 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 is received, the sender calculates the *expected* throughput as,

$$\text{Expected Throughput} = \frac{W_i}{BaseRTT}$$

The actual throughput T_i (as defined earlier), is calculated as $\frac{W_i}{RTT_i}$. Then the difference D is calculated as, D = expected throughput $-$ actual throughput $= \frac{W_i}{BaseRTT} - \frac{W_i}{RTT_i}$. Reference [2] expresses this difference D in terms of *extra packets* in the network, by multiplying D by $BaseRTT$. We define f_{Vegas} as,

$$f_{Vegas} = BaseRTT \times D = W_i \left(1 - \frac{BaseRTT}{RTT_i}\right)$$

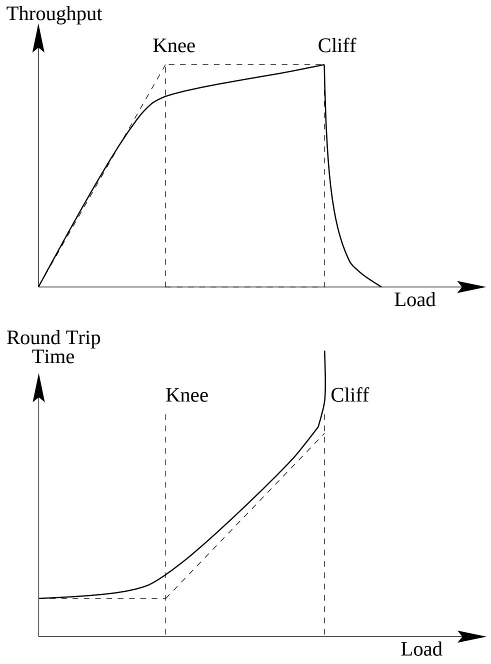

Figure 2. Throughput and RTT versus network load [4]

f_{Vegas} is compared to two thresholds α and β, where $\alpha < \beta$. If $f_{Vegas} < \alpha$, then this congestion avoidance technique suggests that the window size be increased. If $f_{Vegas} > \beta$, it suggests that sender's congestion window size be decreased.

Wang and Crowcroft [9] 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[4] 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 (resp. decrease) of the window size results in an increase (resp. decrease) of the round trip time, then the congestion window size is decreased. Otherwise, the congestion window size is increased.

The congestion avoidance techniques can potentially be applied to distinguish packet losses due to congestion from those due to transmission errors. However, a previous study shows that the CATs may not be able to determine the cause of a packet loss with adequate accuracy [1]. Therefore, this paper analyzes the correlation between round-trip time and the number of packets in flight. If this correlation is weak, then the CATs would not be very useful to draw conclusions about the cause of a packet loss.

4 Experiments

We use a set of data collected by Vern Paxson [5] to study end-to-end Internet dynamics. Paxson collected four sets of data $\mathcal{R}_1$, $\mathcal{R}_2$, $\mathcal{N}_1$, and $\mathcal{N}_2$. $\mathcal{R}_1$ and $\mathcal{R}_2$ are traces collected with the tool *traceroute* [7] and were used to study routing. $\mathcal{N}_1$ and $\mathcal{N}_2$ are *tcpdump*[7] traces collected over 37 sites to study end-to-end packet dynamics. The measurements are extensively described in [5]. We used the set $\mathcal{N}_2$ which contains about 20,000 *tcpdump* traces for *bulk transfers* of 100 KBytes between 2 sites among the 37 sites studied. For each transfer, $\mathcal{N}_2$ contains the *tcpdump* traces both at the sender and at the receiver. For this study, we use only the traces at the sender. For technical reasons[1], we were able to read and study only the *tcpdump* traces for 14,218 transfers. These 14,218 TCP connections were between 31 different sites across U.S.A, Europe and Australia. Since not all sites send to all sites, these 14,218 TCP connections span only 737 paths.

A *tcpdump* trace for a given TCP connection C_l contains records of the packets for the connection C_l when they appear on the "wire". Each record contains a timestamp and a certain number of bytes covering a part of the packet at the data link layer : it contains the data link layer header, the IP header and the TCP header and eventually a part of the TCP payload.

The *tcpdump* trace for a TCP connection allows us to easily compute the round trip time (RTT_i) for *EVERY* packet P_i not retransmitted. Moreover, we also know, at any time, the number of packets in flight W_i, when packet P_i is sent. This number is not reliable when retransmissions happen. Therefore, we do not take into account round trip times or the number of packets in flight when retransmissions occur.

For each connection C_l, we study four different populations :

- Set of packets P_i : all packets sent during the connection C_l but not retransmitted,
- Set of packets PW_j : only one packet per window PW_j is considered. This packet is the packet TCP would monitor.

[1] We were unable to decode correctly the tcpdump traces, even using the tcpdump tool, for some sites (e.g, sri[5]).

- Set of packets PF_k : if a packet is the front packet of a burst of b ($b \geq 1$) packets, it belongs to this set.
- Set of packets PT_m : if a packet is the tail packet of a burst of b ($b \geq 1$) packets, it belongs to this set.

Note that the populations of packets PW_j, PF_k, and PT_m are subsets of the population of packets P_i.

The objective is to determine whether we can get better information from any of the four populations.

We identify the bursts by first finding the minimum delay D_{min} between the transmission of two successive packets. We consider any two successive packets as sent back-to-back if the delay between their transmission is less than $1.8 \times D_{min}$.

We compute two sample coefficients of correlation for each population:

- $\rho(C_l, RTT_i, W_i)$: this is the sample coefficient of correlation between RTT_i and W_i for connection C_l.
- $\rho_v(C_l, \frac{\delta RTTi}{|\delta RTTi|}, \frac{\delta Wi}{|\delta Wi|})$: this is the sample coefficient of correlation between the signs of variations of RTT_i and the signs of variations of W_i for connection C_l.

The problem with the first coefficient of correlation $\rho(C_l, RTT_i, W_i)$ is that it may be "dominated by outliers (RTT spikes)" [6]. To avoid such problems, we also study the correlation between the signs of variations. $\rho_v(C_l, \frac{\delta RTTi}{|\delta RTTi|}, \frac{\delta Wi}{|\delta Wi|})$ is the sample coefficient of correlation between $\frac{RTT_i - RTT_{i-1}}{|RTT_i - RTT_{i-1}|}$ and $\frac{W_i - W_{i-1}}{|W_i - W_{i-1}|}$. With $\rho_v(C_l, \frac{\delta RTTi}{|\delta RTTi|}, \frac{\delta Wi}{|\delta Wi|})$, we measure only how the signs of variations of RTT_i are related to the signs of variations of W_i.

For each population and each coefficient of correlation, we study the repartition of the connections by their coefficient of correlation.

In order to study the impact of the bandwidth at the bottleneck, we distinguish connections on slow links from those on fast links. The question is how to partition our set of connections. In Section 4.1, we present and justify the partition of the set of connections in two sets : slow links and fast links.

4.1 Partitioning the set of connections

Paxson developed a method called "packet bunch mode" [5] to draw an estimate from the tcpdump trace of the bottleneck link. For each connection C_l, Vern Paxson provides 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 8KBytes/s. For a bandwidth b on the x-axis, the value $f(b)$ on the y-axis represents the fraction of connections which has a bottleneck bandwidth in the interval $[b - 8KBytes/s, b)$. However, the y value for the last point on the x-axis is for connections with a bottleneck bandwidth of more than 240 KBytes/s. There is not much interest to detail the repartition for higher bandwidth than 240 KBytes/s since they represent only 12% of all connections.

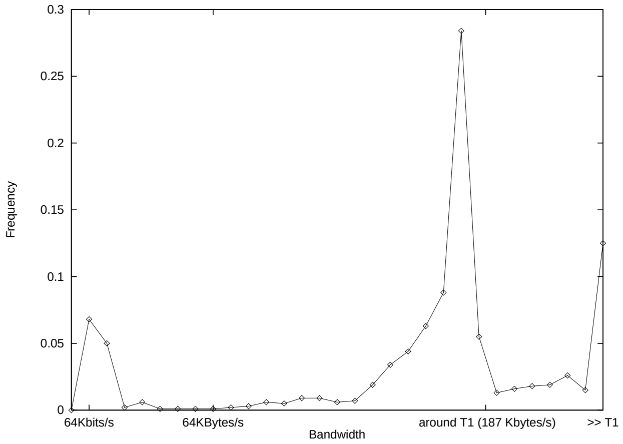

Figure 3. Frequency distribution of connections by bottleneck link speed

Figure 3 exhibits two peaks. They correspond approximately to bandwidths 64 Kbits/s and T1. We can safely draw a separation line at 64KBytes/s because there are not many connections between the two peaks. We may mistake only a small number of connections on fast bottleneck links as connections on slow bottleneck links (and *vice versa*). The small fraction of such "mistakes" should not skew our final results much.

4.2 Results for $\rho(C_l, RTT_i, W_i)$

We present in this section the repartition of the connections by their sample coefficient of correlation. Figure 4 represents the repartition of the connections by their coefficient of correlation $\rho W(C_l, RTT_i, W_i)$ for the population of packets PW_i (i.e, one packet per window is considered). On the x-axis, values vary from -1 to 0.8. in steps of 0.2. For a value x, the y value represents the fraction of connections that have a sample coefficient of correlation in the interval $[x, x + 0.2)$. In each figure,

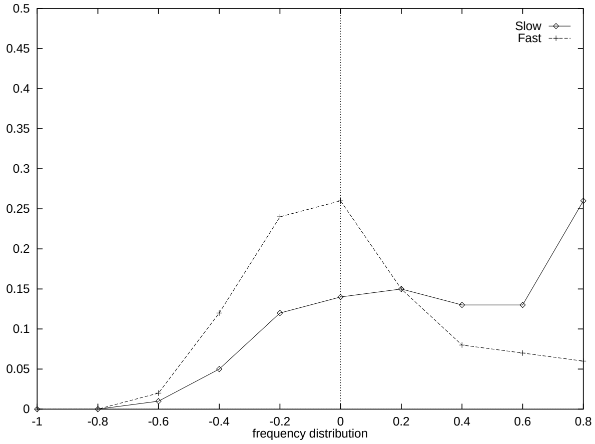

Figure 4. Repartition of $\rho W(C_l, RTT_i, W_i)$ for PW_i

we have two curvess. The curve with “•” is for connections over slow bottleneck links. There are almost 25% (fraction of 0.25) of the connections on slow links that have a coefficient of corrrelation between 0.80 and 1. The second curve with points as “+” is for connections over fast bottleneck links. For fast bottleneck links, only 6% of connections have a coefficient of correlation between 0.80 and 1.

We observe that for slow bottleneck links, there is a significant (35%) proportion of connections with a strong (≥ 0.6) coefficient of correlation $\rho W(C_l, RTT_i, W_i)$. This supports the general opinion that there exists a higher correlation between round trip time and the number of packets in flight on a slow link (than on a fast link). On fast links, only 11% of the connections exhibit a coefficient of correlation larger larger than 0.6.

Figure 5 represents the repartition of the connections by their coefficient of correlation $\rho(C_l, RTT_i, W_i)$ for the population of packets P_i (all packets sent, and not retransmitted). For the population of packets P_i, we get similar results as PW_i. However, the population of packets PW_i (one per window) exhibits a (very slightly) better correlation.

Figure 6 represents the repartition of the connections by their coefficient of correlation $\rho F(C_l, RTT_i, W_i)$ for the population of packets PF_i (all packets sent as the front of any burst of b packets ($b \geq 1$)).

For this population of packets PF_i, we get similar results as for the population of all packets (P_i).

As shown in Figure 7, we get similar results for

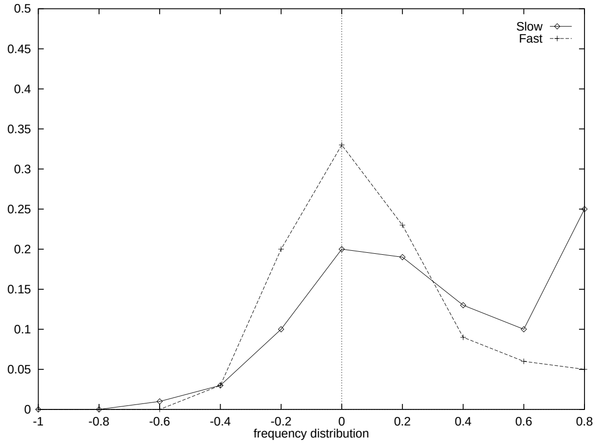

Figure 5. Repartition of $\rho(C_l, RTT_i, W_i)$ for all packets P_i

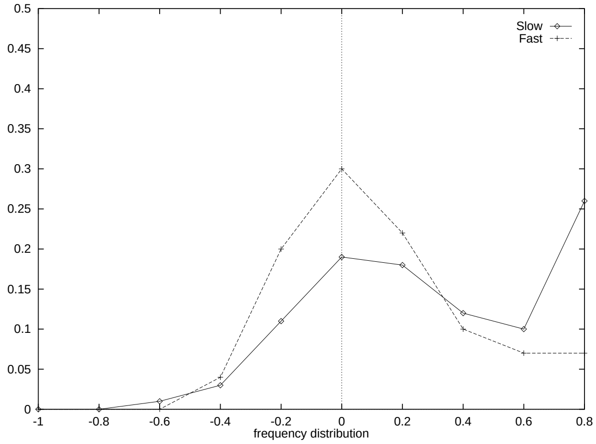

Figure 6. Repartition of $\rho F(C_l, RTT_i, W_i)$ for all “burst front” packets PF_i

the population of packets PT_i (all packets sent as the tail of any burst of b packets ($b \geq 1$)). We get similar results because there are not so many bursts of three or more packets. There is a significant proportion of "bursts" of one packet or two packets. In the case of a burst of one packet, front and tail packet are the same packet. In the case of bursts of two packets, the tail has a smaller RTT (due to delayed acks) and has a number of packets in flight larger by one. Let us further consider this point. Suppose that we are on a slow link of 8 Kbytes/s and that packets are of size 512 bytes. The transmission time of one packet is 62.5 ms. Thus, the RTT for the tail packet should be at least 62.5 ms more than the front packet if queueing happens at the bottleneck. However, with TCP's delayed ack (200 ms) mechanism, the front and tail packets are often acked with the same ack. Therefore, the queueing time experienced by the tail packet is "masked". Note that for fast links, the transmission time is smaller and therefore the previous argument becomes stronger. This explains why the results are similar for front and tail packets.

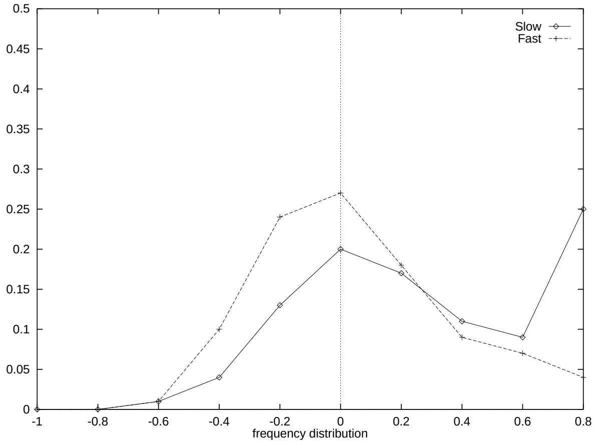

Figure 7. Repartition of $\rho T(C_l, RTT_i, W_i)$ for all "burst Tail" packets PT_i

4.3 Results for $\rho_v(C_l, \frac{\delta RTTi}{|\delta RTTi|}, \frac{\delta Wi}{|\delta Wi|})$

Figure 8 shows the repartition of the coefficients of correlation between the signs of variations of RTT_i and W_i, i.e, $\rho W_v(C_l, \frac{\delta RTTi}{|\delta RTTi|}, \frac{\delta Wi}{|\delta Wi|})$ for the population of packets PW_i (one packet per window).

We observe that the repartition of the coefficients of correlations is similar for slow and fast links. For slow or fast links, 30% of the connections have a coefficient of correlation larger than 0.40.

Figure 9 presents the repartition of the sample coefficients of correlation $\rho_v(C_l, \frac{\delta RTTi}{|\delta RTTi|}, \frac{\delta Wi}{|\delta Wi|})$ for the

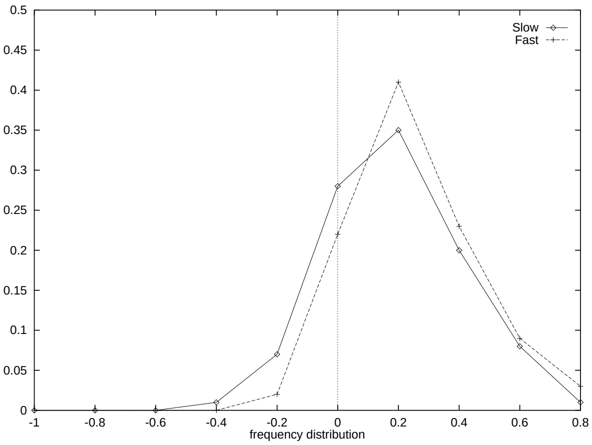

Figure 8. Repartition of $\rho W_v(C_l, \frac{\delta RTTi}{|\delta RTTi|}, \frac{\delta Wi}{|\delta Wi|})$ for PW_i

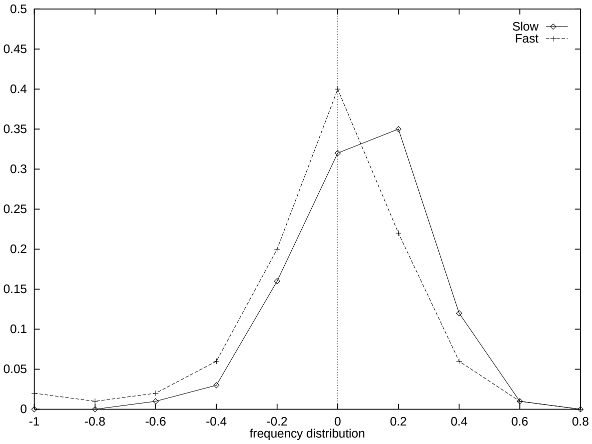

Figure 9. Repartition of $\rho_v(C_l, \frac{\delta RTTi}{|\delta RTTi|}, \frac{\delta Wi}{|\delta Wi|})$ for all packets P_i

population of ALL packets P_i. Figure 9 is similar to Figure 8, but the curves of Figure 8 are slightly shifted to the right. This suggests that there exists a higher correlation with the population PW_i (one packet per window) than with the population P_i of all packets.

For the populations PF_i ("front burst") and PT_i ("tail burst"), the results are presented, respectively, in Figure 10 and Figure 11.

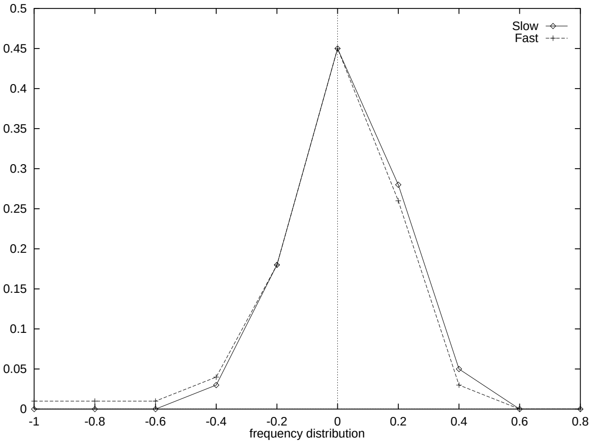

Figure 10. Repartition of $\rho F_v(C_l, \frac{\delta RTTi}{|\delta RTTi|}, \frac{\delta Wi}{|\delta Wi|})$ for all "front burst" packets PF_i

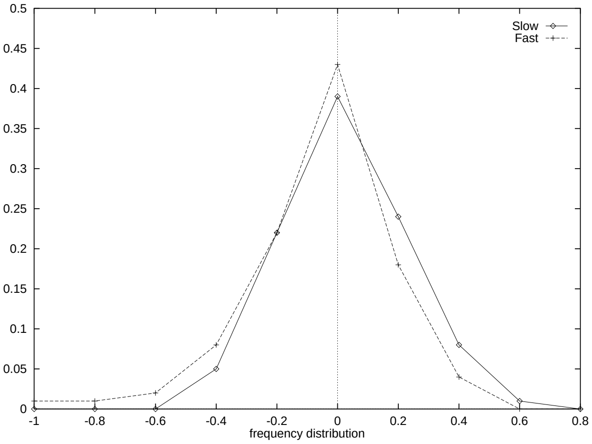

Figure 11. Repartition of $\rho T_v(C_l, \frac{\delta RTTi}{|\delta RTTi|}, \frac{\delta Wi}{|\delta Wi|})$ for all "tail burst" packets PT_i

4.4 Can ρ characterize the path ?

Given a path P, we wondered if the coefficients of correlation for all connections along path P are similar. Unfortunately, they are not. This result is shown in Figure 12. Figure 12 presents the coefficients of correlation of all connections along all 138 "slow" paths. The x-axis represents the path number from 0 to 137. The y-axis represents the coefficient of correlation $\rho W(C_l, RTT_i, W_i)$ for packets PW_i (one packet per window). Each point represents the coefficient of correlation for one connection. For any slow path, the coefficients of correlation span a large interval.

Figure 13 plots $\rho W(C_l, RTT_i, W_i)$ for fast paths for the packets PW_i (one packet per window). In this case, we have 133 paths. We choose these paths because measurements were performed for more than 25 connections. Results are similar to those on slow paths. If we assume that paths (routes between two sites) do not change frequently, then we can conlude that the coefficient of correlation cannot be a characteristic of a path. Results for $\rho W(C_l, RTT_i, W_i)$ on all 598 fast paths are presented in Figure 16.

In Figures 14 and 15, similar results are plotted for coefficient of correlation $\rho W_v(C_l, \frac{\delta RTTi}{|\delta RTTi|}, \frac{\delta Wi}{|\delta Wi|})$ where we consider only signs of variations for the packets PW_i.

Note, however, that the number of connections per path is too small. Definitive conclusions may be drawn if we had a larger number of connections per path. It would be interesting to collect such data.

Observe that for both slow and fast paths, in Figures 14 and 15, the large majority of connections have a positive correlation. Results for $\rho W_v(C_l, \frac{\delta RTTi}{|\delta RTTi|}, \frac{\delta Wi}{|\delta Wi|})$ on all fast paths are presented in Figure 17.

5 Discussion and Conclusion

Suppose that a car has a special control pedal : the car accelerates with probabilty p, and slows down with probability $(1-p)$ whenever you push the gas pedal. The question is : what is the range of values for p which allows us to build a reliable cruise control system ? Intuitively, a value of p between 0.4 and 0.6 would give a very hard time for the designer.

To design a good congestion avoidance technique under real network conditions, we have to deal with a similar situation as for the special pedal described above. When a user increases its load (push the pedal), the round trip time (speed) may increase as well as decrease because of the actions of the others users, the uncertainty of interrupt services on the OS at the endpoints, and the vagaries of the transport protocol. The coefficients of correlation we measured confirm this. However, one may argue that

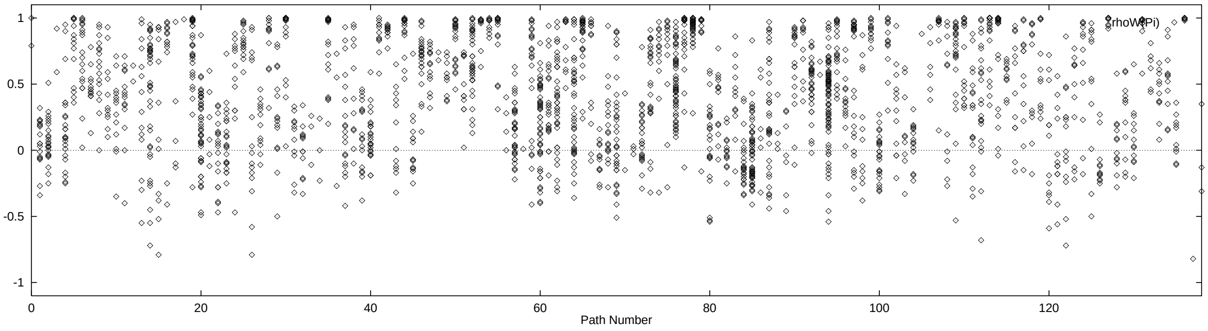

Figure 12. $\rho W(C_l, RTT_i, W_i)$ **for slow paths**

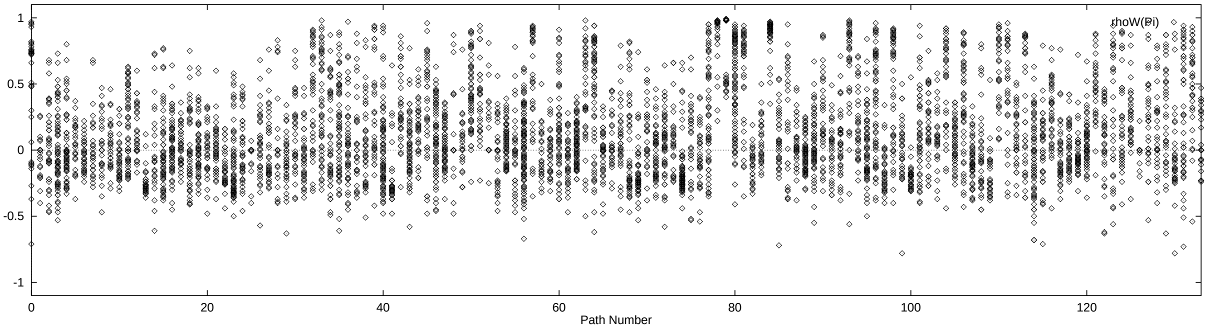

Figure 13. $\rho W(C_l, RTT_i, W_i)$ **for fast paths**

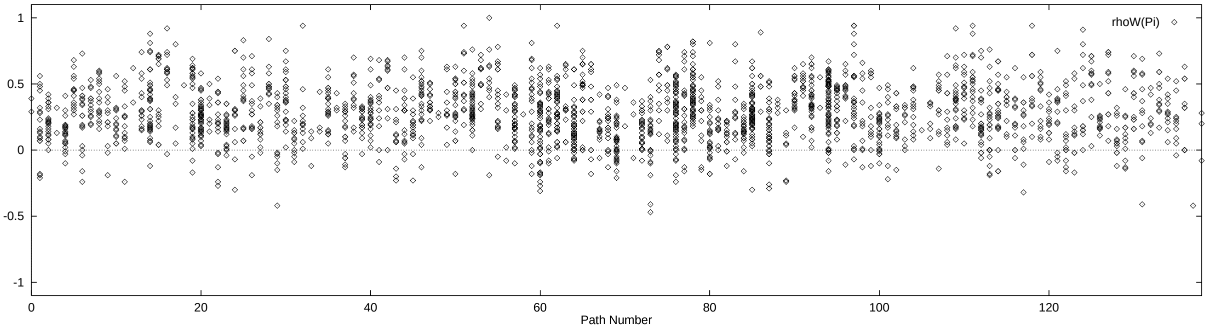

Figure 14. $\rho W_v(C_l, \frac{\delta RTTi}{|\delta RTTi|}, \frac{\delta Wi}{|\delta\ Wi|})$ **for slow paths**

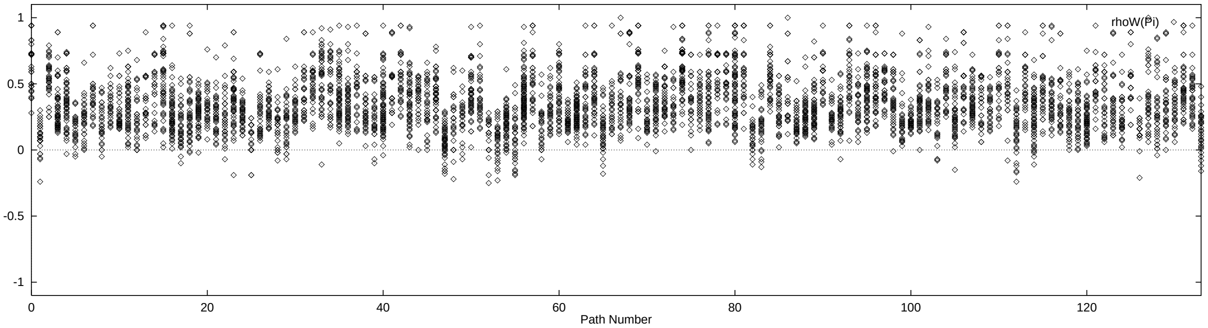

Figure 15. $\rho W_v(C_l, \frac{\delta RTTi}{|\delta RTTi|}, \frac{\delta Wi}{|\delta\ Wi|})$ **for fast paths**

the coefficient of correlation should be expected to be small (near 0) when there is no queue build-up. Whenever the user increases its load, the round trip time does not increase since there is no queue build-up. This argument may be valid for a very fast link, but not with a slow link. Most of the machines on Internet are able to keep a 64 Kbits/s (or even a T1) link busy and drive the queues to build-up. By a similar argument, if the TCP connection is 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 there is no strong relation between the variations of RTT and the sender's earlier variations of its window size. The three CATs presented do relate congestion window size with round trip time to take their decisions. The weakness of these CATs resides in relating RTT and congestion window size.

The measurements of $\rho_v(C_l, \frac{\delta RTTi}{|\delta RTTi|}, \frac{\delta Wi}{|\delta Wi|})$ exhibit a positive correlation between the signs of variations. This is specially true for the packets PW_i (one packet per window) where more than 88% of the connections have a positive correlation. This confirms that the network is "sensitive" to the load. But, the correlation is not strong enough to build "smart" congestion avoidance techniques which would detect reliably queue build-up.

Note that the number of connections studied is quite limited : an average of 20 connections per path. Moreover, the transfer of 100 KB is too small. It may be possible to get better correlation for long lived connections. However, with transfers of only 100 KB, the $\mathcal{N}_2$ data set already takes a huge amount of memory.

For the purpose of this study, the amount of data collected may be drastically reduced. It would be interesting to have the same set-up as Vern Paxson, but without collecting all the tcpdump traces. We would need an agent which saves only the coefficients of correlation, and not the entire trace. With such a set-up, it would be possible to determine if the coefficient of correlation is a characteristic of the path, by measuring a large number of connections per path.

6 Acknowledgements

We would like to thank Vern Paxson for sharing the data he collected and for answering the many questions we had. Research reported is supported in part by the Fulbright Program, National Science Foundation grant CDA 9529442, and Texas Advance Technology Program grant 010115-248.

References

[1] S. Biaz and N. H. Vaidya. Distinguishing congestion losses from wireless transmission losses : A negative result. In *IEEE 7th Int'l Conf. on Computer Communications and Networks*, Oct. 1998.

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

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

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

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

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

[7] W. R. Stevens. *TCP/IP Illustrated: the protocols (v.1)*. Addison-Wesley, 1994.

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

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

7 Appendix

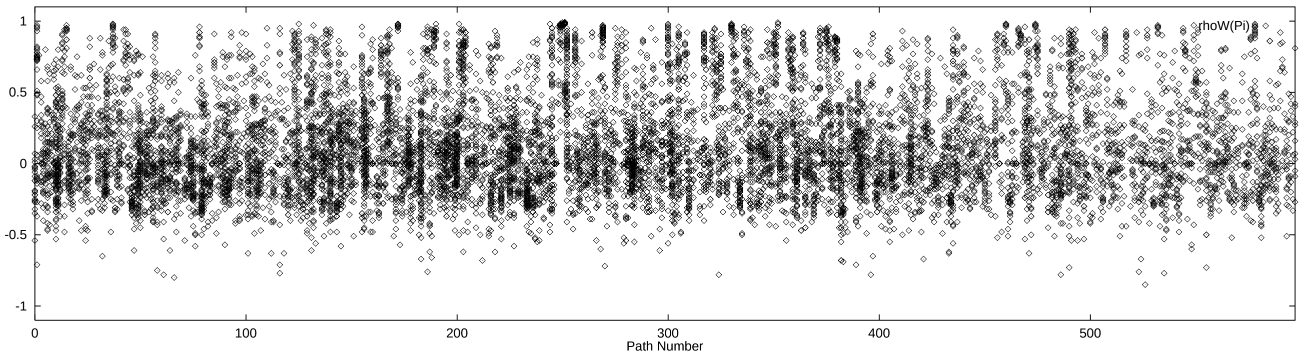

Figure 16. $\rho W(C_l, RTT_i, W_i)$ **for all fast paths**

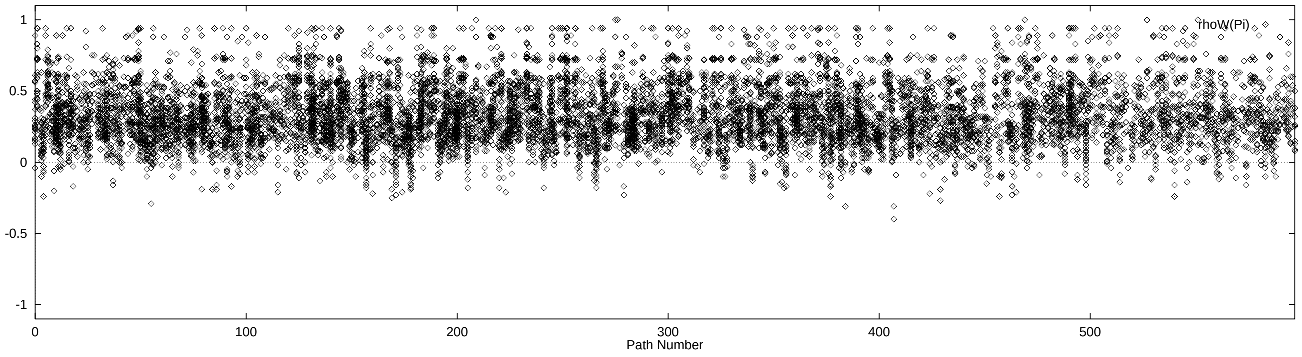

Figure 17. $\rho W_v(C_l, \frac{\delta RTTi}{|\delta RTTi|}, \frac{\delta Wi}{|\delta\ Wi|})$ **for all fast paths**