

The effect of flow capacities on the burstiness of aggregated traffic ^{*}

Hao Jiang and Constantinos Dovrolis

College of Computing, Georgia Tech
hjiang,dovrolis@cc.gatech.edu

Abstract. Several research efforts have recently focused on the burstiness of Internet traffic in short, typically sub-second, time scales. Some traces reveal a rich correlation structure in those scales, while others indicate uncorrelated and almost exponential interarrivals [1]. What makes the Internet traffic bursty in some links and much smoother in others? The answer is probably long and complicated, as burstiness in short scales can be caused by a number of different application, transport, and network mechanisms. In this note, we contribute to the answer of the previous question by identifying one generating factor for traffic burstiness in short scales: *high-capacity flows*. Such flows are able to inject large amounts of data to the network at a high rate. To identify high-capacity flows in a network trace, we have designed a passive capacity estimation methodology based on packet pairs sent by TCP flows. The methodology has been validated with active capacity measurements, and it can estimate the *pre-trace capacity* of a flow for about 80% of the TCP bytes in the traces we analyzed. Applying this methodology to Internet traces reveals that, if a trace includes a significant amount of traffic from high-capacity flows, then the trace exhibits strong correlations and burstiness in short time scales.

1 Introduction

The (layer-3) capacity of a network link is defined as the maximum IP-layer throughput that that link can deliver [2]. The capacity of a network path, C , is defined as the minimum capacity of the links that constitute that path. Consider a packet trace \mathcal{T} collected at a network link \mathcal{L}_T (“vantage point”). Suppose that f is a TCP flow in \mathcal{T} , and that \mathcal{S}_f is the flow’s source. The capacity of the path between \mathcal{S}_f and \mathcal{L}_T is referred to as the *pre-trace capacity* C_p of flow f . Notice that $C_p \geq C$, and so the pre-trace capacity can be viewed as an upper-bound,

^{*} This work was supported by the “Scientific Discovery through Advanced Computing” program of the US Department of Energy (award number: DE-FG02-02ER25517), by the “Strategic Technologies for the Internet” program of the US National Science Foundation (award number: 0230841), and by an equipment donation from Intel. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the previous funding sources.

potentially tight, of the path capacity. This is important, especially when the latter cannot be estimated accurately from passive measurements at the vantage point.

Our primary objective in this paper is to examine the effect of flow capacities on the burstiness of aggregated Internet traffic. To do so, however, we first need to estimate C , or at least C_p , for the flows that constitute \mathcal{T} . There are several path capacity estimation techniques and tools, such as [3–5], but they are based on active measurements. With passive measurements, on the other hand, we are only given a packet trace from a network link. Two passive approaches to estimate the capacity of TCP flows from end-host traces are Paxson’s PBM methodology [6] and Lai’s *nettimer* [7]. The problem that we consider is different, however, because we estimate the capacity of a TCP flow from a uni-directional trace collected at a vantage point in the network, rather than at the sender or receiver of the flow.

The dispersion of two successive packets of the same flow at a network link is the time spacing (interarrival) between the last bit of those packets. Our passive capacity estimation technique is based on packet pair dispersion analysis [4]. However, it differs significantly from the technique presented in [4] in two major ways. First, active capacity probing always sends back-to-back packet pairs, with the two packets of each pair having equal size. In passive capacity estimation, we do not know whether two successive packets were sent back-to-back, and they may not have the same size. Second, in passive capacity estimation we need to differentiate between the end-to-end capacity and the pre-trace capacity. As will be explained in the next section, both capacities may be visible in the distribution of packet pair dispersions of a TCP flow.

The paper structure is as follows. The pre-trace capacity estimation methodology is given in Section II. That methodology has been validated with active measurements, as summarized in Section III. Section IV presents measurements of pre-trace capacity distributions from various traces. The connection between flow capacities and traffic burstiness is shown in Section V.

2 Pre-trace capacity estimation

Our pre-trace capacity estimation methodology is applicable only to *TCP data flows*. We expect that a TCP data flow will include several packet pairs, meaning two successive packets sent back-to-back, due to the delayed-ACK algorithm. Based on that algorithm, a TCP receiver should typically acknowledge every second packet, and so the sender responds to every ACK with at least two back-to-back packets (as long as it has data to send).

Consider a TCP flow with pre-trace capacity C_p and path capacity C . In the following, we illustrate that both capacities would be measurable from the dispersion of successive TCP packets, when there is no cross traffic. In the presence of cross traffic, however, it may not be possible to estimate C . To simplify the following example, we assume that the dispersion of ACKs is not affected by

queueing in the reverse path, and that the sender and receiver do not introduce delays in the transmission of data packets and ACKs.

In Figure 1(a) and 1(b), we show the sequence of successive data packets ($\dots, D_k, D'_k, D_{k+1}, D'_{k+1}, \dots$), as well as the corresponding ACKs ($\dots, A_k, A_{k+1}, \dots$), assuming that the dispersion of the TCP flow's packets is not affected by cross traffic. In round i , the sender S sends a window of W_i packets of size L back-to-back. These packets arrive at the receiver R with a dispersion L/C . The receiver responds to every second packet, and so the ACK dispersion is $2L/C$. Upon receiving the first ACK of round i , the sender starts the next round $i + 1$. For each new ACK received, the sender replies with two more back-to-back packets, plus any additional packets due to window increases. If $C_p=C$, the dispersion of successive packets at the vantage point is L/C , as shown in Figure 1(a). If $C_p>C$, the dispersion between packets D_k and D'_k is L/C_p , while the dispersion between packets D'_k and D_{k+1} is $\frac{2L}{C} - \frac{L}{C_p} > L/C_p$. So, within a single round, and if there is no queueing due to cross traffic, the dispersion of successive packets at the vantage point is directly determined by either C_p , or by C and C_p . In that case, it would be relatively simple to estimate both capacities from the location of the two major modes in the distribution of dispersion measurements.

In practice, however, the dispersion of TCP data packets is often affected by cross traffic queueing. Furthermore, increased dispersions in round i can also affect the dispersions in round $i + 1$. Figure 1(c) illustrates this scenario. Cross traffic is introduced at the narrow link in round i , increasing the dispersion between two successive packets to a value X that is unrelated to C and C_p . The dispersion X can be propagated to round $i + 1$, even if there is no cross traffic queueing in that round. On the other hand, even with cross traffic queueing, every new ACK at the sender still triggers the transmission of a back-to-back packet pair. So, we expect that about 50% of the data packets are sent back-to-back, and so their dispersion at the vantage point is independent of previous rounds. The dispersion of packets triggered by different ACKs, however, is more susceptible to cross traffic queueing, because those dispersions are correlated from round to round. Consequently, we expect that the analysis of a TCP trace will provide a strong mode at the dispersion L/C_p , even in the presence of cross traffic queueing, but it may not create a strong mode at the dispersion $\frac{2L}{C} - \frac{L}{C_p}$. This explains why we focus on the estimation of the pre-trace capacity C_p , rather than on the end-to-end capacity C . In the following, *when we refer to "capacity estimation" we mean pre-trace capacity estimation*.

Figure 2 shows the distribution of packet interarrivals for two bulk TCP flows. The interarrivals are normalized by $L/100\text{Mbps}$, where L is the size of the second packet, and then rounded to the nearest integer. Figure 2(a) represents the case $C_p=C$, with $C\approx 100\text{Mbps}$. Note that about 90% of the interarrivals are concentrated in a single mode around the dispersion that corresponds to the capacity. In Figure 2(b), on the other hand, there are two distinct modes. The first represents about 50% of the interarrivals and it corresponds to $C_p\approx 100\text{Mbps}$. The second mode represents about 40% of the interarrivals, and it probably corresponds to the capacity $C\approx 1.3\text{Mbps}$. In practice, it is often the case that even

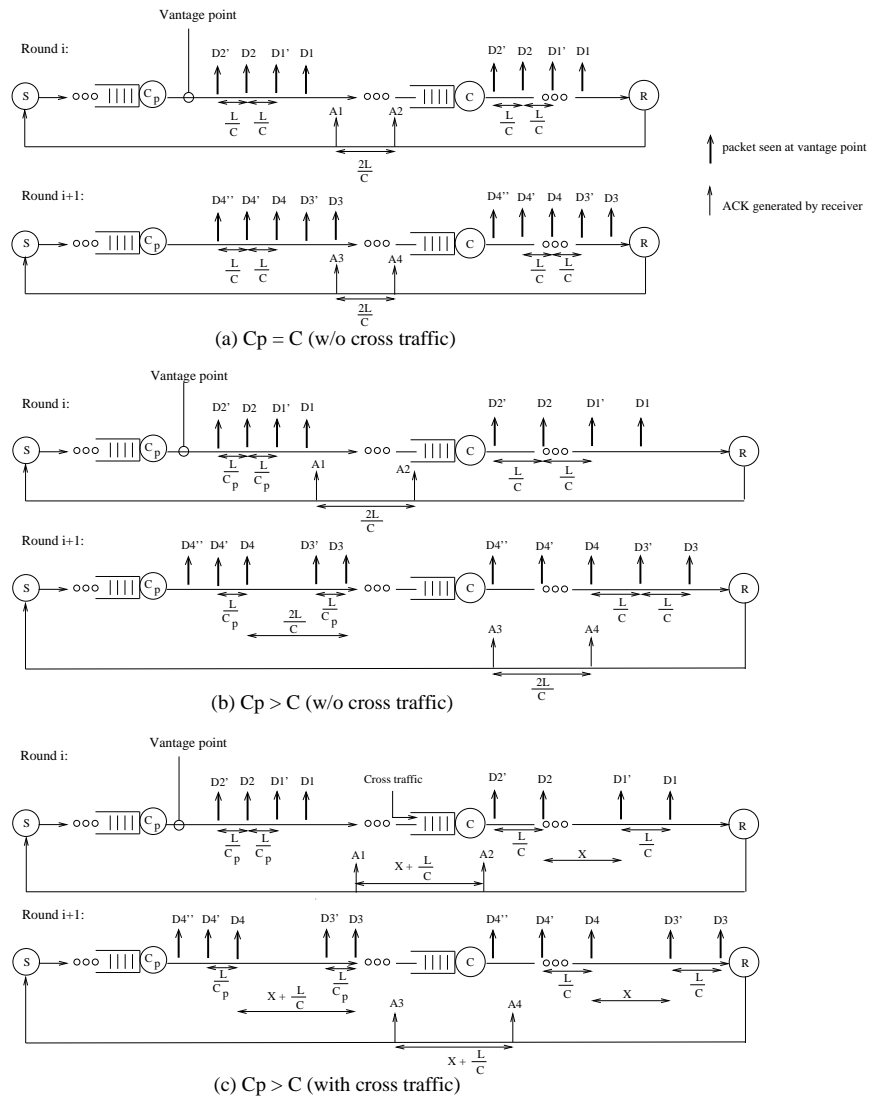


Fig. 1. TCP data and ACK dispersion sequences in three typical scenarios.

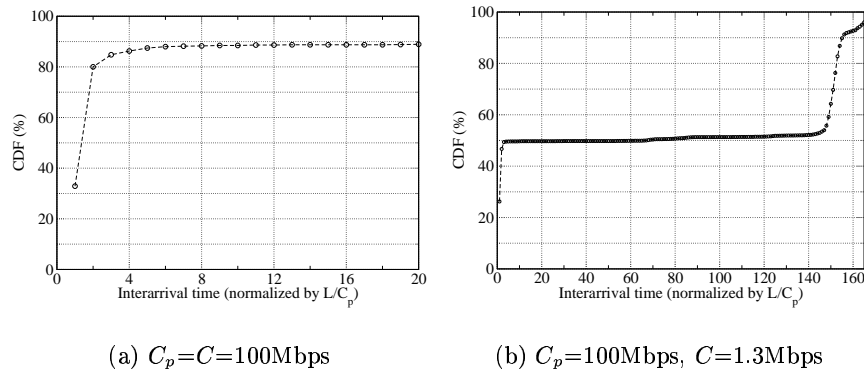


Fig. 2. Cumulative distribution of normalized packet interarrivals

though a single mode with about 50% of the interarrivals is clearly visible, a second strong mode cannot be detected.

The capacity estimation technique is as follows. For a TCP flow f , let $P_f(i)$ be the size of the i 'th data packet, and $\Delta_f(i)$ be the dispersion between packets i and $i+1$. If packets i and $i+1$ have the same size, we compute a capacity sample $b_i = P_f(i)/\Delta_f(i)$. Note that packets with different sizes traverse the network with different per-hop transmission latencies, and so they should not be used by the packet pair technique [4]. As explained in the previous paragraph, we can assume that about 50% of the data packets are sent back-to-back due to the delayed-ACK algorithm, and so they can be used in capacity estimation. The rest of the packets may have been sent with a larger dispersion than L/C_p , and so they can underestimate C_p . Based on this insight, we sort the capacity samples of flow f and drop the lower 50% of them. To estimate C_p , we employ a histogram-based technique to identify the strongest mode among the remaining capacity samples. The center of the strongest mode gives the final capacity estimate \hat{C}_f . The bin width that we use is $\omega = \frac{2(IRQ)}{K^{1/3}}$ (known as ‘‘Freedman-Diaconis rule’’), where IRQ and K is the interquartile range and number, respectively, of the capacity samples.

The algorithm does not produce estimates for interactive flows, ACK flows, and flows with just a few data packets. For such flows, the number of packet pairs can be small and the detection of a capacity mode is quite prone to statistical errors.

3 Validation

We have validated the previous passive estimation technique with active measurements. Specifically, we send a file of size Y with *scp* from various sources around the Internet to a destination at Georgia Tech. At the same time, we collect a trace of TCP data packets using *tcpdump* at the destination host. The trace is then used to estimate passively the capacity of the corresponding path. We also measure the capacity of the same path using *pathrate* [4]. Since the trace

is collected at the end host, the pre-trace capacity is equal to the end-to-end capacity in these experiments. To show the effect of the flow size on the accuracy of the estimation technique, we repeat each experiment for three values of Y : 40KB, 110KB, and 750KB.

Table 1 shows the results of some validation experiments. The passive and capacity estimates are reasonably close, and they correspond to common capacity values such as 256Kbps (upstream DSL), 1.5Mbps (T1), 10Mbps (Ethernet), and 100Mbps (Fast Ethernet). Passive estimation with larger flows obviously helps, even though the results with the 40KB flows are not too inaccurate either.

Name	Location	Passive estimate (Mbps)			Pathrate
		750KB	110KB	40KB	(Mbps)
lulea.ron.lcs.mit.edu	Sweden	97	96-97	94-96	99-101
mazul.ron.lcs.mit.edu	MIT	1.4-1.5	1.4-1.5	1.5-1.7	1.5
magrathea.caida.org	UCSD	97-98	96-99	93-95	94-96
diple.acad.ece.udel.edu	U-Delaware	98	97-98	97-99	94-97
aros.ron.lcs.mit.edu	U-Utah	9.5-9.7	9.2-9.7	11.1-11.2	11.9-12.3
thalis.cs.unipi.gr	Greece	9.1-9.2	8.3-8.5	6.4	9.7-9.8
dsl-64-192-141-41.telocly.com	Atlanta	0.225-0.226	0.226	0.225	0.226

Table 1.

4 Capacity distributions

We performed capacity estimation on several packet traces from edge and backbone network links with a wide range of average rates. The three traces that we use in this paper are publicly available at the NLANR-MOAT site [8], and they are described in Table 2. Note that the capacity estimation technique can provide an estimate for a small fraction of flows (about 4-13%, depending on the trace), but for a large fraction of bytes (about 80%).

Trace	Link type	Date	Local Time	Rate (Mbps)	TCP flows	Estimate C_p	
						% flows	% bytes
MRA-if-1	OC-12	2002/08/07	20:12:00-20:13:30	180.4	71357	3.8	82.7
MRA-if-2	OC-12	2002/08/07	20:12:00-20:13:30	157.3	118786	8.2	83.4
Auck-1-if-0	OC-3	2001/04/02	14:27:00-14:30:00	2.8	9657	7.8	85.5
Auck-2-if-0	OC-3	2001/06/11	08:56:00-08:59:00	4.8	14017	12.1	81.5

Table 2.

Figure 3(a) shows the distribution of TCP flow capacity estimates for the two interfaces of the MRA-1028765523 OC-12 trace. The cumulative distribution is

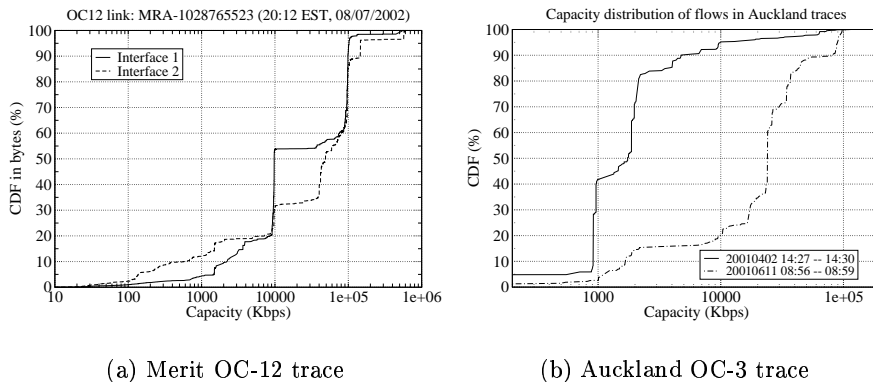


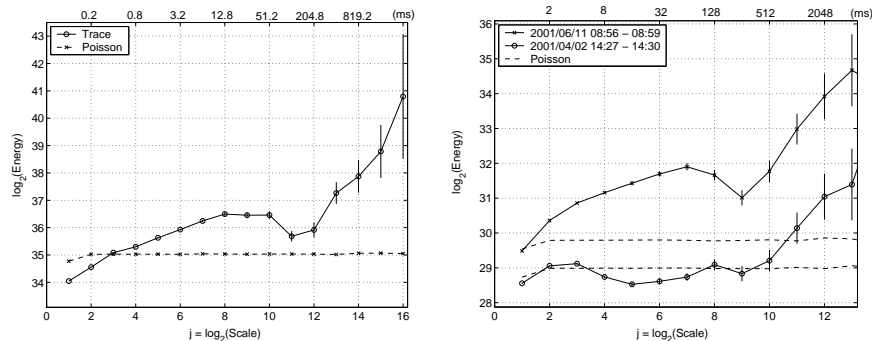
Fig. 3. Capacity distribution in terms of bytes

plotted in terms of TCP bytes, rather than TCP flows. Note that most bytes are generated from flows with capacities 1.5Mbps, 10Mbps, 45Mbps, and 100Mbps. These values correspond to some commonly used links (T1, Ethernet, T3, and Fast Ethernet, respectively). Figure 3(b) shows the distribution of TCP flow capacity estimates for two segments of the Auckland OC-3 trace [8]. Note that the two distributions are quite different. A major difference is that the 2001/06/11 trace carried traffic from TCP flows with significantly higher capacities. Specifically, about 80% of the bytes in that trace were generated from TCP flows with a capacity of more than 10Mbps. On the other hand, more than 80% of the bytes in the 2001/04/02 trace were carried by TCP flows with a capacity of less than 3Mbps.

We have also investigated the correlation between the capacity of a flow and the flow's average throughput and maximum window size. Due to space constraints we do not report the details of that analysis here. The main result, however, is that both correlation coefficients are close to zero, implying that the previous two flow characteristics are independent of the pre-trace capacity, and probably independent of the end-to-end capacity as well. The reason may be that the throughput and window size of bulk TCP transfer are often limited by the receiver's advertised window. The correlation coefficient between C_p and the flow size is also close to zero.

5 Capacity and traffic burstiness

We employ wavelet-based energy plots to analyze the correlation structure and burstiness of traffic in a range of short time scales [9,10]. Since the Poisson stream (i.e., independent exponential interarrivals) is traditionally viewed as benign while traffic with stronger variability is viewed as bursty, we use the Poisson process as a reference point in the following analysis. The energy plot of a Poisson process with rate λ is a horizontal line at $\log_2(\lambda T_0)$, where T_0 is the minimum time scale of the energy plot. If the energy of a traffic process X_j at scale $T_j=2^j T_0$ is higher than the energy of a Poisson process that has the same



(a) MRA-if-1

(b) Auck-1-if-0 & Auck-2-if-0

Fig. 4. Energy plots of three traces

average rate with X_j , then we say that X_j is *bursty at scale T_j* . Otherwise, we say that X_j is *smooth at scale T_j* .

Figure 4(a) shows the energy plot of a highly aggregated backbone trace, which carries thousands of flows at any point in time. We focus in time scales up to 100msec ($j \leq 10$). The correlation structure and burstiness of the trace in longer scales is determined by Long-Range Dependency effects that have been studied in depth in earlier work [11]. The trace is clearly bursty, compared to Poisson traffic, in all time scales between 1-100 msec. This may be surprising from the perspective of the theory of point processes, because that theory predicts that the superposition of many independent flows tends to a Poisson process [12]. There is no actual contradiction however. The previous superposition result applies to flows with rate R/N , where N is the number of aggregated flows, i.e., it assumes that the flow interarrivals become “sparser” as the degree of aggregation increases. That is not the case, however, in typical packet multiplexers; flows are aggregated in higher capacity links without artificially increasing the interarrivals of each flow.

Figure 3 shows that a major part of the previous trace (about 40% of the bytes) are generated from 100Mbps flows, i.e., flows with comparable capacity to the 622Mbps capacity of the monitored OC-12 link. These high-capacity flows are not small relative to the aggregate, neither in terms of size (not shown here), nor in terms of rate. Consequently, we should expect that their correlation structure and burstiness can significantly affect the burstiness of the aggregate traffic.

To elaborate on the previous point, we examine the energy plot of the two Auckland traces from Figure 3(b). As previously shown, the 2001/06/11 trace carries traffic from significantly higher capacity TCP flows than the 2001/04/02 trace. Figure 4(b) shows the corresponding energy plots. The 2001/06/11 trace is clearly bursty, while the 2001/04/02 trace remains below the Poisson energy level. We note that the two traces are similar in other aspects, including flow RTTs, number of active flows, flow size distribution, and average utilization.

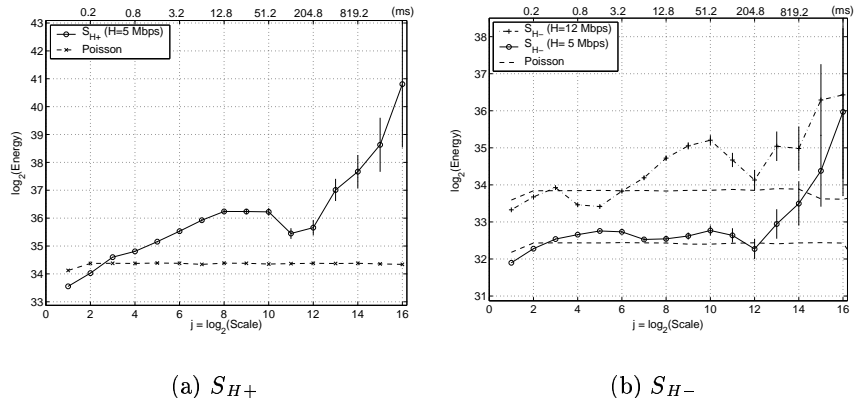


Fig. 5. Energy plots of S_{H+} and S_{H-}

To further examine the link between short scale burstiness and flow capacities, we separate the TCP flows of the trace MRA-if-1 for which we have a capacity estimate in two subsets: S_{H+} and S_{H-} . S_{H+} consists of all flows with capacity larger than a threshold H (in Mbps), while S_{H-} includes the remaining flows that have a lower capacity. The average rate of S_{H+} and S_{H-} are 119Mbps and 30Mbps, respectively. The energy plots of S_{H+} and S_{H-} are shown in Figure 5. If the threshold H is between 1-10Mbps, the resulting energy plots are not so sensitive to the exact value of H , and so we set $H=5$ Mbps. Notice that the energy plot of S_{H-} is about at the same level with that of the corresponding Poisson process, meaning that the lower capacity flows do not generate significant burstiness. On the other hand, S_{H+} has much higher energy than the corresponding Poisson process, as shown in Figure 5(a), confirming our earlier conjecture that high capacity flows cause significant burstiness in short scales. Note that if we set $H > 10$ Mbps, then both S_{H+} and S_{H-} will be characterized as bursty. Finally, it should be mentioned that the file size distributions of S_{H+} and S_{H-} are similar. S_{H+} includes 1937 flows, with 86% of them being larger than 10KB. S_{H-} includes 773 flows with 90% of them being larger than 10KB. Consequently, the difference in the burstiness of the two subsets cannot be attributed to their flow size distribution.

6 More recent results

In more recent work [13], we have further investigated the connection between flow capacities and short time scale burstiness. The main result of that work is to explain the origin of such burstiness based on TCP self-clocking. Specifically, we have shown that, under a certain condition on the flow's window size and bandwidth-delay product (that is proportional to the flow capacity), TCP self-clocking can generate a two-level ON/OFF interarrival structure. That structure results in considerable traffic burstiness and strong correlations in sub-RTT time scales.

7 Acknowledgment

This work would not be possible without the traces collected from the NLANR Passive Measurement and Analysis (PMA) project. The NLANR PMA project is supported by the National Science Foundation Cooperative agreements ANI-9807479 and ANI-0129677 and by the National Laboratory for Applied Network Research.

References

1. Zhang, Z.L., Ribeiro, V., Moon, S., Diot, C.: Small-Time Scaling behaviors of Internet backbone traffic: An Empirical Study. In: Proceedings of IEEE INFOCOM. (2003)
2. Prasad, R.S., Murray, M., Dovrolis, C., Claffy, K.: Bandwidth Estimation: Metrics, Measurement Techniques, and Tools. *IEEE Network* (2003)
3. Carter, R.L., Crovella, M.E.: Measuring Bottleneck Link Speed in Packet-Switched Networks. *Performance Evaluation* **27,28** (1996) 297–318
4. Dovrolis, C., Ramanathan, P., Moore, D.: Packet Dispersion Techniques and Capacity Estimation. Technical report, Georgia Tech (2004) To appear in the *IEEE/ACM Transactions on Networking*.
5. Pasztor, A., Veitch, D.: Active Probing using Packet Quartets. In: Proceedings Internet Measurement Workshop (IMW). (2002)
6. Paxson, V.: End-to-End Internet Packet Dynamics. *IEEE/ACM Transaction on Networking* **7** (1999) 277–292
7. Lai, K., M.Baker: Measuring Bandwidth. In: Proceedings of IEEE INFOCOM. (1999) 235–245
8. NLANR MOAT: Passive Measurement and Analysis. <http://pma.nlanr.net/PMA/> (2003)
9. Abry, P., Veitch, D.: Wavelet Analysis of Long-Range Dependent Traffic. *IEEE Transactions on Information Theory* **44** (1998) 2–15
10. Veitch, D.: Code for the Estimation of Scaling Exponents. http://www.cubinlab.ee.mu.oz.au/darryl/secondorder_code.html (2001)
11. Park, K., W. Willinger (editors): *Self-Similar Network Traffic and Performance Evaluation*. John Willey (2000)
12. Cox, D.R., Isham, V.: *Point Processes*. Chapman and Hall, London (1980)
13. Jiang, H., Dovrolis, C.: The Origin of TCP Traffic Burstiness in Short Time Scales. Technical report, Georgia Tech (2004)