A Study of Burstiness in TCP Flows

Srinivas Shakkottai[1], Nevil Brownlee[2], and K. C. Claffy[3]

[1] Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign, USA
email: sshakkot@uiuc.edu

[2] CAIDA, University of California at San-Diego, USA and
Department of Computer Science
The University of Auckland, New Zealand
email: nevil@auckland.ac.nz

[3] Cooperative Association for Internet Data Analysis,
University of California at San-Diego, USA
email: kc@caida.org

Abstract. We study the burstiness of TCP flows at the packet level. We aggregate packets into entities we call "flights". We show, using a simple model of TCP dynamics, that delayed-acks and window dynamics would potentially cause flights at two different timescales in a TCP flow— the lower at the order of 5-10 ms (sub-RTT) and the higher at about 10 times this value (order of an RTT seen by the flow). The model suggests that flight sizes would be small at the lower timescale, regardless of the network environment. The model also predicts that the network conditions required for the occurrence of flights at the larger timescale are either large buffers or large available bandwidths — both of which result in a high bandwidth delay product environment. We argue that these two conditions indicate that the TCP flow does not operate in a congestion control region , either because the source of traffic is unaware of congestion or because there is so much bandwidth that congestion control is not required. We verify our model by passive Internet measurement. Using the trace files obtained, we collect statistics on flights at the two timescales in terms of their frequency and size. We also find the dependence of the sizes and frequency of flights on the Internet environment in which they occurred. The results concur strongly with our hypothesis on the origins of flights, leading us to the conclusion that flights are effective indicators of excess resource in the Internet.

1 Introduction

TCP is the dominant protocol in today's Internet. It has been observed [1, 2] that TCP sometimes sends packets in the form of deterministic aggregations. The timescale at which this phenomenon occurs is at the RTT level, which indicates that we should study it at the packet level in individual flows. We consider the steady state characteristics of TCP at a packet level and investigate the frequency with which TCP flows have recognizable structure that we can label *flight behavior*. Fig. 1 shows a sequence of thirteen packets and we observe deterministic behavior of packet aggregates at two time scales.

Definition 1. *A small time scale flight (STF) is a sequence of packets whose inter-arrival times differ by at most 'T' percent, where 'T' is a fixed threshold value.*

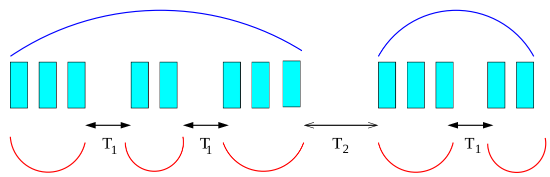

Fig. 1. Illustration of two aggregation levels. Packets may be aggregated into flights at different time scales. At the lower time scale we see five flights, while at the higher time scale we see two.

At the smaller time scale we look at inter-arrival times between single packets; if the inter-arrival times are nearly identical then we say that the packets belong to a single STF. However, observing packets at such a fine resolution obscures the temporal relations that might exist between aggregations of packets. In other words, there may be deterministic behavior between the STFs themselves. In the figure, there are two groups of STFs, within which STFs have nearly identical inter-arrival times.

Definition 2. *A large time scale flight (LTF) is a sequence of aggregations of packets whose inter-arrival times differ by at most 'T' percent, where 'T' is a fixed threshold value.*

By our definition, aggregations of STFs with nearly identical inter-arrival times are defined to be LTFs. We recognize that the terms "small" and "large" are relative. Both terms are with respect to the RTT seen by a flow. The inter-arrival times between packets of an STF are on the order of 5-10 milliseconds (sub-RTT), while the inter-arrival times between STFs are on the order of 40-1000 milliseconds (order of RTT seen by the flow).

Flight behavior of TCP has been a matter of considerable debate. In fact there is not even a standard terminology for the phenomenon; other names for flight-like phenomena are *bursts* [3] and *rounds* [4], where "bursts" usually describe phenomena similar to our STFs and "rounds" usually describe phenomena similar to our LTFs. While modeling TCP flows some authors simply assume the flight nature of TCP [4, 5]. As far as we know, there are no published statistics on flight behavior, and no studies investigating the correlation of flight occurrence with the Internet environment in which TCP operates. Also, there do not seem to be any algorithms for identifying the structure of TCP flows — the method used in the only other work we are aware of in the area [6], is dependent on visually classifying flows.

TCP Model

Two facets of TCP design could potentially lead to flights, each one at a different time scale.

1. Since many TCP implementations [1, 2] implement *delayed-acks*, a host may send multiple packets for every *ack* it receives. Implementations of delayed-acking vary in terms of the maximum delay (200-500 ms). Many implementations also require that there be a maximum of one outstanding un-acked packet, nominally leading to acknowledgment of alternate packets. Transmission of such packets back to back at source could result in the observation of STFs at the measurement point if the network delays are relatively constant.
2. TCP follows a window-based congestion control mechanism with self-clocking, i.e., the window size changes and packets are transmitted only when acknowledgments are received. If acknowledgments are received with relatively constant inter-arrival times, it would give rise to STFs being sent with similar inter-arrival times, i.e., LTFs.

Another phenomenon that may occur is that of constant-rate flows.

Definition 3. *A constant-rate flow (CRF) is a large TCP flow in which aggregations of two or three packets are observed with nearly identical spacing between the aggregations.*

From the definition of LTFs, it is clear that CRFs are nothing but large LTFs, where we say that a flow is large if it has over 30 packets. Other names for such flows are "rate-limited flows" and "self-clocked flows" [6].

From the above discussion, the origin of of STFs lies in the fact that delayed-acks acknowledge a small sequence of packets (often alternate packets) resulting in the back-to-back transmission of a small number of packets at the source. It seems clear, therefore, that STFs would naturally be of small size regardless of the network environment that the TCP flow in which they occur sees.

However, the question arises: what network environment would be conducive to LTF behavior? We conjecture that LTFs of large size can exist only in high bandwidth-delay product (BDP) regimes. The reason is that as long as no drops occur, TCP increases its window size by some value depending on whether it is in slow-start or congestion avoidance. Only if the network is able to absorb all the packets in the congestion window of a TCP flow will acks be received at deterministic times at the source, leading to transmission of packets at deterministic times. The absorption may take place in two ways:

1. Suppose that the buffer sizes are large in the path of a flow and bandwidth is limited. Then, regardless of congestion window size, the actual throughput is bandwidth constrained. The large buffer size in effect absorbs the packets

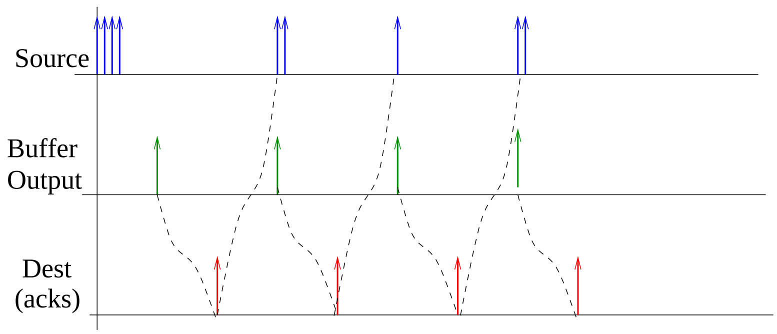

Fig. 2. Illustration of how large buffers in a bandwidth constrained path of a TCP flow lead to LTFs. The congestion window at source gradually increases, but since the buffer absorbs excess packets, the source does not know of the bandwidth constraint.

and delays them so that the source does not see any drops. TCP is unable to estimate the available bandwidth as it is blinded by the large buffer. Fig. 2 depicts the case where there is a large buffer between the source and destination. We have assumed, for illustration purposes, that the delay is large enough to ensure that every packet is acked inspite of the delayed-ack implementation. The source is in congestion avoidance phase and reception of an ack could result in either the source transmitting one packet or an increase in window size with the source transmitting two packets. The source never loses a packet and assumes that excess bandwidth is available. So the window size continuously increases. TCP is thus blind to congestion in this scenario.

2. Another possible scenario is when bandwidth is high and delay is moderate. In such a case the link absorbs the packets, and large windows of packet aggregations proceed through the network. There is no congestion in the network and TCP congestion control is not required. This scenario is illustrated in Fig. 3 in the slow start phase. We could draw a similar diagram for the congestion-avoidance phase.

We summarize our main hypotheses and the conjectures that we make in Table 1.

Flights as Indicators of Excess Resource

Why should we study flights? What are they good for? Let us consider the question in detail. Two assumptions that network designers traditionally make are:

1. Link capacities are low and many users contend for their use. The expected load is close to the capacity of the links. Hence the tremendous volume of research on the "single bottleneck scenario".
2. To handle demands close to the capacity, buffer sizes should be of the order of the bandwidth-delay product of the link.

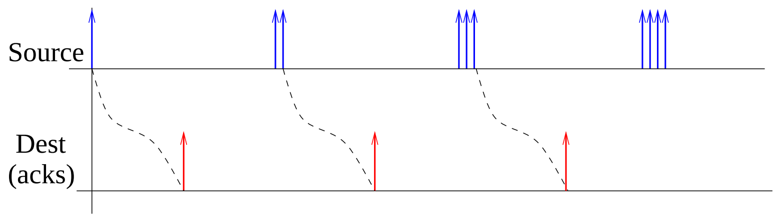

Fig. 3. Illustration of how a large bandwidth and medium delay results in flights in the slow-start phase of TCP. Large bandwidth implies that the source can increase the congestion window to a large size without drops occurring. In this case flights are indicative that congestion control is irrelevant since the network has a large available bandwidth.

Usually such design gives rise to recommendations for large buffer sizes, which in turn has given rise to high bandwidth infrastructure with huge buffer capacities. If the usage assumptions were correct, neither of our two scenarios for flight existence would exist, congestion control would be relevant, and the resource on the Internet would be utilized at high efficiency. On the other hand, the presence of flights is a symptom that we have over designed the Internet — there are enormous resources, in terms of buffer sizes or link capacities, being shared by remarkably few users. In other words, flights are a symptom that TCP congestion control is having no effect, either due to hiding of congestion by buffers, or because there is so much bandwidth that the packets sail through the network. Consistent with the above is the fact that observations of packets on 10 Mb/s Ethernet (for example those in the packet sequence plots in [1, 2]) show clear flight behavior.

Hypothesis 1	STFs arise due to the implementation of delayed-acks.
Conjecture 1	The size of STFs are on the order of two or three packets
Conjecture 2	The frequency of STFs is independent of the network environment.
Hypothesis 2	LTFs arise due to window dynamics of TCP.
Conjecture 3	LTFs could be of large size (potentially several hundred packets)
Conjecture 4	The frequency of LTFs increases with increasing BDP.

Table 1. Summary of our main hypothesis and the conjectures based on them.

Main Results

We use three different packet traces, all from OC-48 ($\approx 2.5Gb/s$) links, and call them BB1-2002, BB2-2003 and Abilene-2002 [7]. Together these packet traces represent a high diversity of IP addresses, applications, geographic locations and access types. For instance, the *BB1-2002* trace shows about 30% of bytes destined for locations in Asia, with flows sourced from about 15% of all global autonomous systems (AS). The *BB2-2003* has even higher diversity with flows from about 24% of all global ASs. The *Abilene-2002* trace has a large fraction of non-web traffic. Since all three traces give nearly identical results, we provide graphs from only one trace: BB1-2002.

We summarize our main results as follows:

1. We propose a simple threshold-based algorithm, which robustly identifies the different time scale aggregation levels.
2. We verify our hypothesis of two distinct phenomena — delayed acks and window dynamics — giving rise to two classes of packet behavior by studying the statistics of each aggregation level.
3. We show how the algorithm naturally leads to a method of identifying CRFs as large LTFs.
4. We further confirm Hypothesis 1 — delayed acks causing STFs — by verifying Conjectures 1 and 2 — that STF sizes are on the order of two to three packets and are independent of network conditions such as round trip time (RTT), bandwidth and BDP. The observation on the size of STFs illustrates that the source transmits a small number (usually 2 or 3) of back to back packets resulting in an STF at the point of measurement.
5. We verify the Conjecture 4 — high BDP regimes permitting LTFs — by studying the variation in LTF lengths as a function of BDP and showing that LTFs that have a much larger number of packets occur at higher BDPs.
6. Finally, using the statistics on LTFs of large size, we verify Conjecture 3 — LTFs can be of large size — and conclude that currently about 12-15% of flows over thirty packets in length in the traces we study are not responding to congestion control, either because the they are unaware of congestion or because there is no congestion on their paths.

2 Algorithms

In this section we describe the algorithms we use for identification of flights. We first consider the case of identifying STFs. Consider a sequence of packets p_1, p_2, p_3, with inter-arrival times (IATs) δ_1 and δ_2 between the first and second pairs of packets, respectively. Then we consider the ratio $g(\delta_1, \delta_2) = |\frac{\delta_2 - \delta_1}{\delta_1}|$. We decide whether a packet belongs to a particular STF depending on whether $g > T$ or $g \leq T$, where T is a threshold value. We use the IAT as a measure of the scale of the flight and call the units *IAT units (IATU)*. Thus a flight of 1 IATU means that the observed IAT was different from the preceding and following IATs. An IATU of 2 would mean that two successive IATs were nearly identical. Fig. 4 depicts two STFs, the first of size 2 IATU and the second of size 3 IATU. Our STF detection algorithm is as follows:

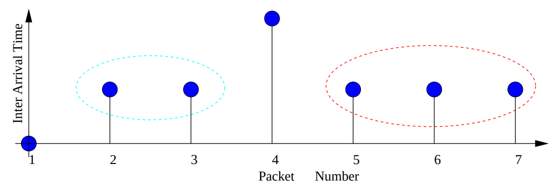

Fig. 4. Illustration of how we find STFs. We group packets $1-3$ together as a 2-inter-arrival time unit flight, and so on. The large gap between packets 3 and 4 appears as a singleton.

```
1. Start with IAT=constant, flight size=1;
2. Compare previous IAT with current IAT and
     calculate g (as defined above).
3. If g is within threshold then
     increment flight size by 1;
   else if flight size >1 start a new
   flight of size 0;
   else start a new flight of size 1;
4. Set previous IAT <- current IAT;
```

The 'else if' line in our above algorithm means that an out-of-threshold IAT indicates the end of a STF, but a sequence of out-of-threshold IATs indicates consecutive 1-IATU STFs. The singleton shown in Fig. 4 indicates such behavior. Our STFs may therefore have 2 packets (1 IATU), 3 packets (2 IATU), 4 packets (3 IATU) and so on. Of course, a 1 IATU STF simply means that at the low time scale, the algorithm did not observe any deterministic behavior.

However, the situation illustrated in Fig. 5 might occur. Here we see a sequence of packet pairs, which are identified by the above algorithm as distinct STFs. But there deterministic behavior between packet pairs at a larger time scale. We would like to have an algorithm that would identify such behavior and aggregate all six packets as an LTF.

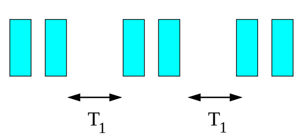

Fig. 5. A sequence of packet pairs in which STFs do not capture complete information.

We observe that deterministic behavior in the larger time scale can potentially occur only when the STF algorithm reports that the current IAT is different from the previous one (if not, the current packet would be part of the current STF and we update the STF size by one and proceed to the next packet). Also, since we are interested in large timescales, we need to know if the current IAT is larger than the previous IAT. So, if we keep the most recent large IAT in memory, we may compare it to the current IAT and check if they are within a threshold of each other. If they are, then we update the size of the current LTF by one. We merely need to add the following lines to Step 3 of the LTF algorithm:

```
3. (continued) If current IAT > previous IAT then
    compare current IAT with most recent large IAT;
   If g(current IAT, most recent large IAT) is
    within threshold then increment LTF size by 1;
   else if LTF size>1 start a new LTF of size 0;
   else start a new LTF of size 1;
   Set most recent large IAT  <- current IAT;
```

Looking at Fig. 5 again, we see that the above extension would result in the identification of the packet sequence as one LTF as desired. We remark at this point that the choice of threshold value does not seem to be critical to the algorithm. The reason for this observation is that the timescales of IATs for STFs and LTFs are different. As mentioned in the introduction, the typical IATs between packets of an STF are 5-10 ms, whereas the IATs between aggregations of an LTF are about 10 times this. In our analysis we used several values of T ($\frac{1}{16}$, $\frac{1}{4}$, $\frac{1}{2}$, 1, 2, 4 and 8) with nearly identical results.

3 Frequency and Size of Flights

In this section we show that our flight detection algorithm is successful and also illustrate the fact that considering two aggregation levels of packets yields a clear picture of TCP behavior. We ran the algorithm with different threshold values on the packet traces and show only some illustrative graphs here.

We first consider the statistics of STFs in Fig. 6 and Fig. 7. Recall that the unit of flight size is IATU. We can convert IATU in STFs into packets by recalling that a 1 IATU STF is a packet whose leading and trailing IATs were different, a 2 IATU STF has three packets and so on. Fig. 6 shows the distribution of STF sizes. We see that regardless of the number of packets in the flow, STF sizes are usually quite small — a size 7 IATU occurs in less than 1% of the STFs. Also, we may easily calculate that the mean STF size is 2.5 packets regardless of the number of packets in the flow.

Fig. 7 shows the distributions of number of STFs on a per flow basis. We see that STFs are much more common in flows with a large number of packets.

We consider flight behavior at the timescale of the RTT seen by the flows in Fig. 8 and Fig. 9. As we have just seen, the STFs of which the LTFs are composed

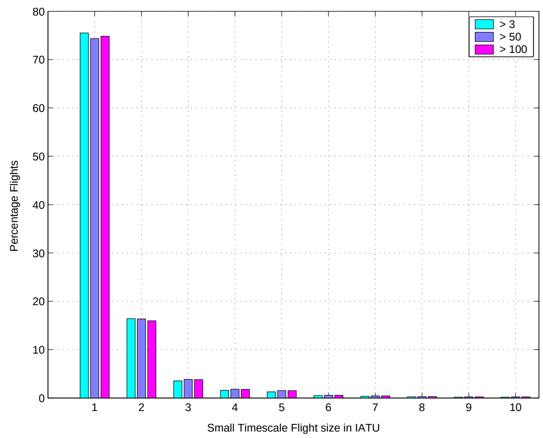

Fig. 6. Small timescale flight size distribution in IATU for BB1-2002. The left column in each set of bars is for flows greater than 3 packets in length; the middle for those greater than 50; and the right column is for flows greater than 100 packets in length. We notice that flights are usually small (in terms of IATU and hence in packets) irrespective of the number of packets in the flow.

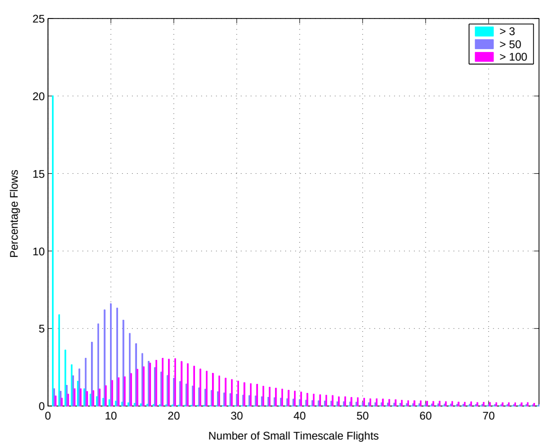

Fig. 7. Number of small timescale flights for trace BB1-2002 on a per flow basis. The left histogram is for flows greater than 3 packets in length, the middle for those greater than 50 and the right histogram is for flows greater than 100 packets in length. We notice that STFs are more common in flows with a larger number of packets.

are an average of 2.5 packets in length. We may thus get an estimate of the number of packets in an LTF by multiplying its IATU size by this number. Fig. 8 shows the size distribution of LTFs. The statistics are quite different from the STF size distribution that we analyzed earlier. Flights are much more common at the larger timescale. The graph follows a distribution that is proportional to $\frac{1}{LTF\ size}$. Thus, even at this timescale, decay of flight sizes is fairly quick. Finally, we plot the distributions of number of LTFs on a per flow basis in Fig.

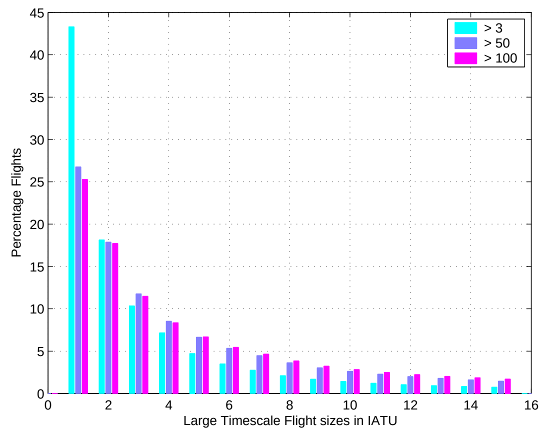

Fig. 8. Large timescale flight size distribution in IATU for BB1-2002. The left column in each set of bars is for flows greater than 3 packets in length; the middle for those greater than 50; and the right column is for flows greater than 100 packets in length. We notice that the LTF size distribution varies proportionally to $\frac{1}{LTF\ size}$.

9. We see that as with STFs, LTFs are much more common in flows with a large number of packets.

We draw the following conclusions from the flight statistics observed above:

1. Our initial hypotheses from our model of TCP were that there would be two distinct aggregation levels at different timescales caused by delayed acks and TCP window dynamics. The hypotheses are borne out by the fact that we usually see short STFs, normally consisting of two or three packets, indicating delayed acks. We also see much larger LTFs indicative of windows of packets transmitted in pairs and triplets (i.e., as STFs) with similar spacings between the aggregations.
2. Over 75% of flows having over 50 packets contain LTFs. We identify fairly large LTFs of up to 16 IATUs, i.e., with an average of 40 packet or more, thus verifying Conjecture 3 — that LTFs could potentially be large (and hence be identified as CRFs). What we observe in such cases are aggregations of two or three packets being transmitted at a constant rate. Thus, our algorithm

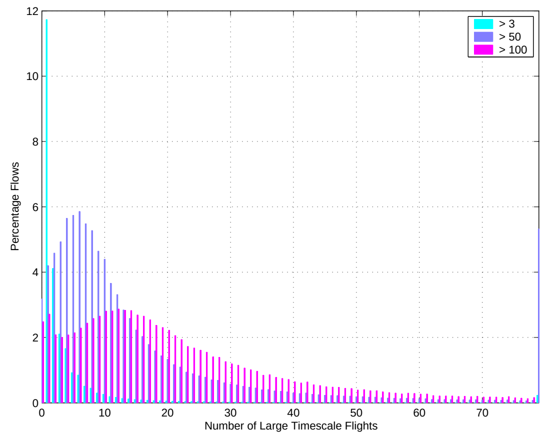

Fig. 9. Number of large timescale flights for trace BB1-2002 on a per flow basis. The left histogram is for flows greater than 3 packets in length, the middle for those greater than 50 and the right histogram is for flows greater than 100 packets in length. We notice that LTFs are more common in flows with a larger number of packets.

offers a simple means of identifying CRFs. If we consider a flow to be a CRF if it has over 30 packets in equally spaced aggregations, then about 12-15% of flows are constant-rate flows. These flows are clearly not limited by PC clock speed, as Brownlee and Claffy also observe in [8].

3. From the statistics on the number of flights seen in flows we conclude that many flows are composed of fairly small deterministic packet aggregations at the large timescale, which indicates that the congestion window in these flows grows only up to 10-12 packets before feedback from the network causes it to reduce. Thus, large time scale structure is lost with the growth of TCP congestion windows.

4 Relationship with Network Environment

We now study the relation between flights and the characteristics of the path that a flow traverses, namely the round trip time (RTT), the bandwidth and the BDP. Our usage of the term RTT is to indicate the entire path delay inclusive of queuing delays. We measure RTT by the syn-ack method whose validity has been largely established in [9].

We first consider STFs. We already know that the majority of them are two packets in size, irrespective of flow size. We would like to know if any network characteristics affect STFs larger than two packets. If their origin has to do with delayed acks, i.e., the source constrains them to be small, then the variation of their occurrence with the network parameters should not be significant. In Fig. 10, we show a probability histogram ($\times 100\%$) of STFs larger than two packets in size in different RTT regimes. We see that the chance of seeing an STF larger than two packets is about 1% regardless of the RTT.

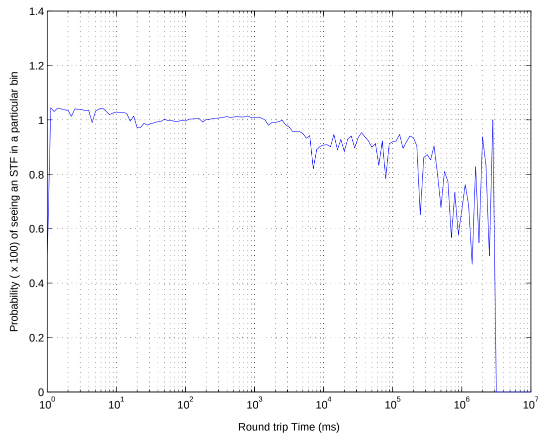

Fig. 10. Fraction of traffic having STFs larger than two packets in each RTT regime for BB1-2002. The variation is about 0.4% indicating the independence of STFs and RTT

We next examine the relation between STFs and bandwidth (Fig. 11). By 'bandwidth', we mean the total bytes transferred divided by the lifetime of the flow. Here too the probability ($\times$100%) of seeing at least one STF larger than two packets in size for any particular bandwidth is nearly constant at 1%. We also analyzed the frequency of seeing at least one STF with more than two packets with regard to BDP and found that the probability ($\times$100%) of this event too was about 1%, with a variation of less than 0.5%.

We now study the variation of LTF sizes as a function of BDP, which we show in Fig. 12. We see that on average, LTF sizes are higher at higher BDP. The graph peaks at 10 kb with an average LTF size of 5 (i.e. most flows in this regime had LTFs consisting of of 12 or more packets). The number of available points is small after the peak and the graph dips sharply. The above facts support Conjecture 4 — high BDP is conducive to LTFs being large — since it means that the network has the capacity to absorb the large windows of packets.

5 Conclusion

We studied deterministic temporal relations between TCP packets using several packet traces, which were from different backbone fibers and represent a large fraction of ASs and prefixes. Such traces give an indication of Internet traffic characteristics, since phenomena occurring at the edges are reflected in temporal relations between packets at the measurement point. We studied aggregation

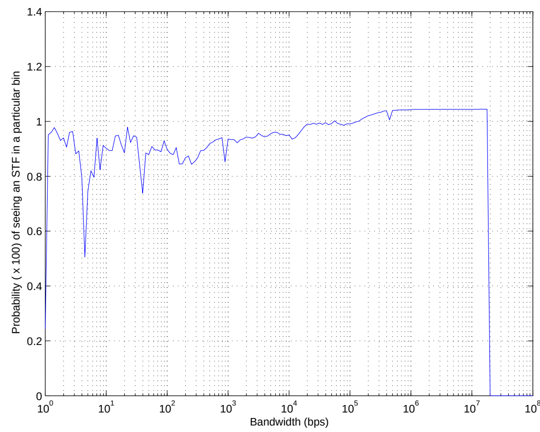

Fig. 11. Fraction of traffic having STFs in each bandwidth regime for BB1-2002. The flat nature indicates that the probability of seeing an STF is independent of bandwidth.

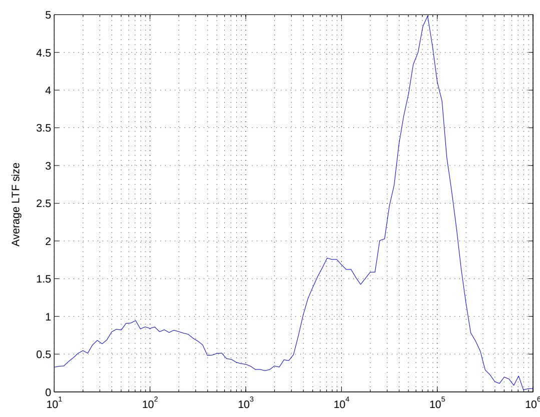

Fig. 12. Variation of LTF sizes as a function of BDP for BB1-2002. The plot peaks at high BDP. Note that the number of data points drops sharply after the peak, and hence the graph ends here.

of packets at two time scales (of the order of 5-10 ms and 50-1000 ms) in order to verify our hypothesis that two distinct facets of TCP structure should give rise to two different types of temporal relations. In doing so we proposed a simple threshold algorithm for identification of flights. Our TCP model predicted that high BDP environments would be conducive to CRFs. Such an environment could exist only if the network were over-provisioned — either in terms of large buffers or large bandwidth — and hence CRFs are indicative of excess resource in the network. Through statistics on flight sizes and frequency, we verified Conjectures 1 and 3 — that STFs should be short and LTFs can be long. We then verified Conjectures 2 and 4 — that STFs should not depend on the network environment, whereas LTFs should be benefited by large BDPs — by studying the correlations between flights and different network parameters. We thus showed that Hypothesis 1 and 2 — delayed-acks giving rise to STFs and window dynamics giving rise to LTFs — are valid. We concluded that about $12-15\%$ of Internet flows in our traces do not operate in a congestion control region. In the future we would like to study how the occurrence of flights changes over the years both on the backbone as well as access links, to understand flights as indicators of excess network resource. Such a study would give us an idea of whether congestion on the Internet has been increasing or decreasing over time.

Acknowledgment

The authors would like to acknowledge Andre Broido (CAIDA) for his participation in many of the discussions that led to this work. This research was funded in part by NSF grant ANI-0221172.

References

1. Stevens, R.: TCP/IP illustrated, Vol.1, Addison-Wesley (1994)
2. Paxson, V.E.: Measurements and Analysis of End-to-End Internet Dynamics. PhD dissertation, University of California, Lawrence Berkeley National Laboratory (1997)
3. Sarvotham, S., Riedi, R., Baraniuk, R.: Connection-level analysis and modeling of network traffic. In: Proceedings of IMW 2001. (2001)
4. Padhye, J., Firoiu, V., Towsley, D., Krusoe, J.: Modeling TCP throughput: A simple model and its empirical validation. In: Proceedings of ACM SIGCOMM '98. (1998)
5. Zhang, Y., Breslau, L., Paxson, V., Shenker, S.: On the Characteristics and Origins of Internet Flow Rates. In: Proceedings of ACM SIGCOMM. (2002)
6. Downey, A.: TCP Self-Clocking. Technical Report TR-2003-002, Olin College, Computer Science (2003)
7. Abilene: Trace obtained from the NLANR PMA webpage (URL below) (2002) http://pma.nlanr.net/Traces/long/ipls1.html.
8. Brownlee, N., Claffy, K.C.: Understanding Internet Streams: Dragonflies & Tortoises. IEEE Communications Magazine (2002)
9. Aikat, J., Kaur, J., Smith, F., Jeffay, K.: Variability in TCP round-trip times. In: Proceedings of IMW. (2003)