On the Origins of Long-Range Dependence in TCP Traffic

Åke Arvidsson[a], Matthew Roughan[b], and Tobias Rydén[c]

[a]Ericsson Network Core Products, Soft Center VII, SE-372 25 Ronneby, Sweden
Email: Ake.Arvidsson@uab.ericsson.se

[b]AT&T Labs - Research, 180 Park Av., Florham Pk, NJ, 07932, USA
Email: roughan@research.att.com

[c]Dept. of Mathematical Statistics, Lund University, Box 118, SE-221 00 Lund, Sweden.
Email: tobias@maths.lth.se

Since the discovery of self-similarity and Long-Range Dependence (LRD) in data traffic there has been a great deal of research into both its ramifications, and its origins. For instance there are now many studies of the effect of LRD on network performance. However the majority of these studies are "open loop" in the sense that they use the data measurements to construct a model which has the LRD property, and then they use this model as an arrival process for a queuing system. This methodology is clearly suspect when applied to TCP traffic as this traffic is carried over a window flow control which limits its rate when congestion is experienced. Clearly, a closed-loop model that incorporates feedback is required to model such a system. On the other side of the coin, the origin of LRD in traffic has been ascribed to the heavy-tailed nature of the carried traffic. However, once again open-loop models such as the superposition of On/Off sources have been used to produce LRD traffic. Even when the observed traffic shows heavy-tailed On/Off behavior, a simple On/Off model which does not contain feedback should not be used to explain the occurrence of LRD. This paper uses a very simple construction which includes the idea of feedback, and thus provides a more rigorous explanation for why LRD is seen even when there are flow controls involved.

1. Introduction

Measurements of data traffic have universally shown evidence for the related properties of self-similarity and Long-Range Dependence (LRD) (for instance see [1,2], or the many references in [3]). It has been argued that one source of LRD in data traffic is the heavy-tailed size distribution of files which are transferred across networks [4]. However, the theoretical arguments used to illustrate how this might happen via superpositions of heavy-tailed On/Off sources [5–7] typically do not take into account the fact that nearly all file transfers over modern networks occur over TCP, and therefore a flow control.

Flow controls have a drastic effect on carried traffic; they introduce feedback so that the traffic is no longer independent of the network parameters. Ideally the traffic is constrained to exactly fit the capacity of the network, though, of course, this is hard to achieve exactly. Even though TCP only approximates this ideal state, it drastically changes the network performance, particularly where the input traffic is LRD [8].

Despite the drastic differences in performance between closed-loop flow controlled systems, and open-loop systems, both still exhibit LRD in the output traffic, both in measurements of networks, and in simulations [1,4,8]. This paper uses a very simple construction which includes the idea of feedback, and thus provides a more rigorous explanation for why LRD is seen even when there are flow controls involved. The construction is based upon the idea that TCP sources share bandwidth.

Following the Introduction, Section 2 describes the background to this paper, including definitions for LRD, what we mean here by heavy-tails, and how the two are typically related. Section 3 describes a set of instructive simulation results that demonstrate the differences between open-loop, and closed-loop models, and supply the motivation for this work. Section 4 describes our theoretical model for the closed-loop system, explains the origin of LRD in flow controlled traffic, and provides some supplemental explanations for other observations. Finally we conclude the paper in Section 5.

2. Background

2.1. Long-Range Dependence

In this paper we are concerned with second order stationary processes $X(t)$, with constant mean $m = E[X(t)]$, and variance $\sigma^2 = E[(X(t) - m)^2]$, and autocovariance which is a function of the *lag* $k = |t - s|$ only, defined by $r(k) = E[(X(t+k) - m)(X(t) - m)]$. The Fourier Transform of r is known as the *spectral density* and we denote it by f_X.

LRD is commonly defined by the slow, power-law decrease in the autocovariance function: $r(k) \sim c_{\mathrm{r}}|k|^{-(1-\alpha)}$, $k \to \infty$, $\alpha \in (0,1)$, or equivalently as the power-law divergence at the origin of its spectrum: $f_X(\nu) \sim c_{\mathrm{f}}|\nu|^{-\alpha}$, $|\nu| \to 0$, ([9], p. 160). The power-law decay is such that the sum of all correlations (out from any lag) is always appreciable, even if individually the correlations are small. The past therefore exerts a long-term influence on the future.

The main parameter of LRD is the dimensionless scaling exponent α. It describes the qualitative nature of scaling — how behavior on different scales is related. The related parameters, c_{r} and c_{f}, are quantitative parameters which give a measure of the magnitude of LRD induced effects. The parameters may be estimated jointly using the Abry-Veitch wavelet based estimator [10], or separately by a number of other techniques [9].

It is common practice to describe LRD through the *Hurst* parameter $H = (1+\alpha)/2$, though in fact H is the parameter of *self-similarity* and is properly used to describe only self-similar processes, which are non-stationary. The connection to LRD is that if a process Y (with finite second moments) is self-similar with parameter $H \in (1/2, 1)$, then its increment process $X(t) = Y(s+t) - Y(s)$ is LRD with $\alpha = 2H - 1$. We follow this convention of writing H instead of α.

2.2. Heavy-tailed distributions

We have noted that there is a relationship between LRD and heavy-tailed distributions. Here we define the latter more precisely using the notion of regular variation. First recall the notation $h(t) \overset{t_0}{\sim} g(t)$ for asymptotic equivalence, which means $\lim_{t\to t_0} |h(t)|/|g(t)| = 1$.

Now define functions as *slowly varying at* t_0 if they satisfy $\lim_{t \to t_0} L(xt)/L(t) = 1$, for every $x > 0$. We can now define a function $h(t)$ as being *regularly varying* at ∞, with index p, if $h(t) \overset{\infty}{\approx} L(t)t^p$, where $L(t)$ is slowly varying at ∞. We refer to a distribution as heavy-tailed with exponent γ if its complementary distribution function is regularly varying with index $p = -\gamma$. Of particular interest here will be the case where $1 < \gamma < 2$, where the mean of the distribution is finite, but the variance is infinite.

2.3. TCP Flow Controls

The majority of Internet traffic (for instance WWW, FTP traffic) is carried over TCP (the Transmission Control Protocol). TCP uses an adaptive window flow control to limit congestion on the Internet. The flow control, described in detail in [11–14], will not be given here, except to note that the intended effect is to limit the rate of each source, or connection so that all active connections may share the available bandwidth. The sharing of these resources is in reality neither fair [15], nor even, and cannot adapt instantly to changing circumstances, however, the success of the current Internet is an indication that to some broad approximation this sharing does work.

2.4. On/Off Sources

One suggested model for the origin of LRD is the superposition of On/Off sources with heavy-tailed On or Off periods [5–7]. More formally, we can model a single source as a renewal process alternating between two states: "On" and "Off", where the generated traffic rates are R and 0, respectively. Moreover, the duration of at least one of the On and/or Off periods has a heavy-tailed distribution with infinite variance, *i.e.* $1 < \gamma < 2$.

On/Off processes with heavy-tailed On periods are asymptotically LRD in themselves [16] with $H = (3 - \gamma)/2$, where the heavy-tailed distribution has exponent γ. However, a better aggregate model may be obtained by superposing a number of sources. The exact way in which the processes are superposed does matter, principally through the renormalisation. Renormalisation is used so that as the number of sources is increased, the average rate remains constant. Three obvious methods of renormalisation are to

(i) reduce the rate R of each source,
(ii) increase the length of Off periods or
(iii) reduce the length of On periods.

As the number of sources goes to infinity, each method results in a different model,

(i) Fractional Brownian Motion with $H = (3 - \gamma)/2$,
(ii) M/G/∞ source model, and
(iii) Lévy-stable motion respectively.

However, the methods suggested here and elsewhere are all open-loop in the sense that the traffic parameters are independent of the network parameters. What is needed is a model in which the rates of each source depend on the number of sources which are On.

The M/G/∞ model [17] is of particular interest here, as it is a variant of this model which we use in Section 3 to compare simple open-loop models of network performance with LRD input to a model incorporating a closed-loop feedback flow control, such as TCP. The M/G/∞ source model assumes that customers arrive as a Poisson process, each with a file to transfer. The distribution of file sizes, $G(\cdot)$, is assumed to be heavy-tailed. The traffic from the source model arrives in proportion to the number of customers present in a M/G/∞ queuing system. In other words, the queuing system represents a number of sources, where the arrival of these sources depends on a Poisson process rather than the an alternating renewal process. Note that in this model, as in the On/Off model, each source still transmits at a constant rate, regardless of how many sources are On.

In the following section we compare simulation results from such an input model, and from an input model with the same service requirements: the same file transfer requests arrive at the same points in time, but whose rates are moderated by a flow control.

3. Performance comparisons

To evaluate the differences between open- and closed-loop models the traffic generator depicted in Figure 1 was constructed in NS2 (network simulator, version 2) [18]. The request generator produces new file transfer requests according to a Poisson process of rate λ and each request is assigned a file size x octets, where x can be either a constant value for all transfers or drawn at random from a Pareto distribution according the procedure in [19]. Each request creates a TCP source and a TCP sink, opens a TCP connection, transfers the file, closes the TCP connection and terminates the source and the sink. The randomness in request arrivals and in data volumes causes the number of transfers in progress to vary randomly over time. We use the NS2 Full (Reno) TCP with default parameters, see [18,20]. Sources and sinks are complete TCP entities: sources send data packets to sinks which respond with acknowledgments. The communication takes place over a common link with bandwidth μ bps and a propagation delay of δ ms. The links are equipped with buffers, with capacity B packets. In the experiments reported below, $\mu = 2.048$ Mbps, $\delta = 40$ ms, the average file size $\overline{x} = 100$ kB, and λ so that the average load before overhead is 50%.

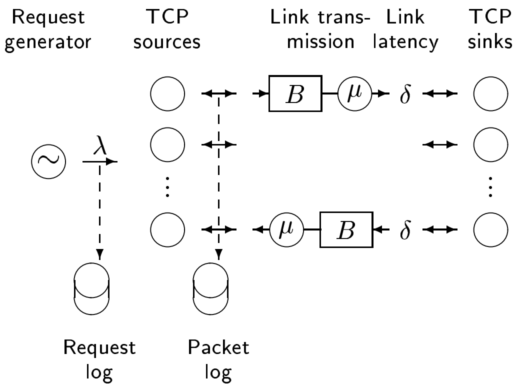

Figure 1. Traffic generation.

Two sets of data are collected, a request log and a packet log. The request log contains the generation time and size of each TCP connection and the packet log contains the generation time and size of each packet. Note that the two sets capture the same traffic although in different terms.

3.1. Closed loop vs. open loop

Closed-loop traffic is modeled by playing back the request log and creating TCP instances which generate packets, Figure 2 (left), while open-loop traffic is modeled by playing back the packet log, as in Figure 2 (right). The bandwidth of the systems is changed to μ' to vary the relative load on the system.

Note that both models are exact in terms of request arrival times and file sizes. The difference is that with a closed-loop model, packet arrival times are determined from the conditions encountered in the system (principally the new bandwidth μ'), but with an open-loop model the packet arrivals are fixed by the conditions in the generator system with bandwidth μ. For more details see [8,19].

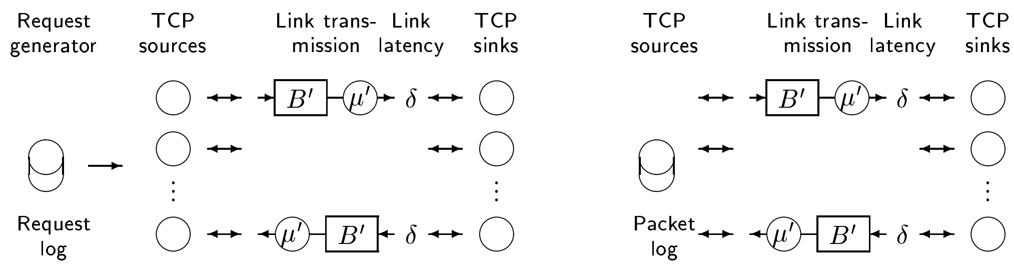

Figure 2. Closed-loop model (left) and open-loop model (right).

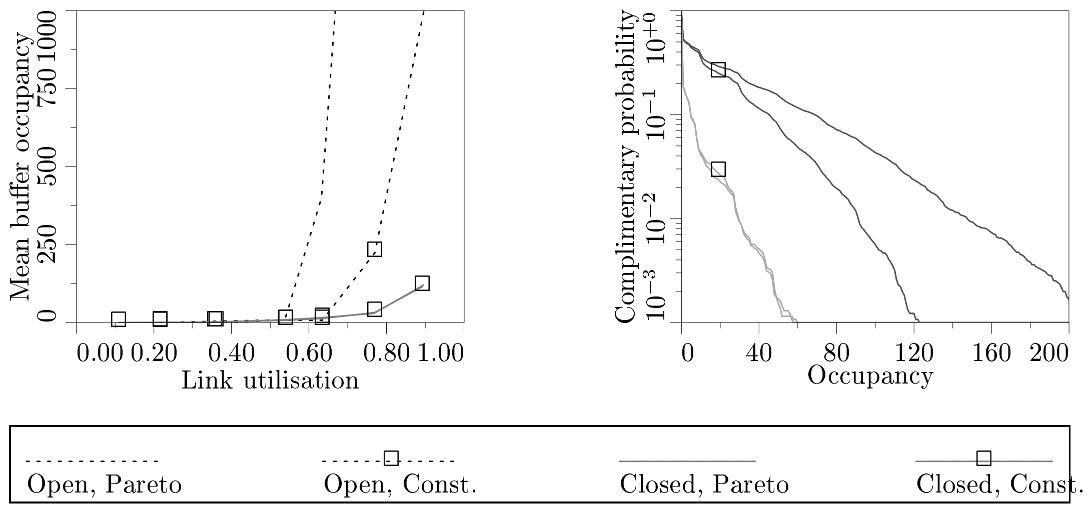

Figure 3. Comparison of closed- and open-loop models (left) and closed-loop buffer length distributions (right).

Figure 3 (left) displays the mean buffer occupancy as a function of link utilization. The curves refer to infinite buffers and file sizes constant or drawn from a Pareto distribution with $\gamma = 1.5$. It is seen that the open-loop model results in very high buffer occupancies, whereas the closed-loop model results are much more modest.

The rightmost graph of Figure 3 confirms these observations. The graph shows the complementary distribution function of the buffer occupancy for a low (37%) and a high (63%) utilization. The distribution functions appear to exhibit exponential decay.

3.2. Long-range dependence

The majority of attention in traffic modeling has been drawn to the packet arrival process. The difference between the performance of the open- and closed-loop models cannot, however, be explained by differences in LRD parameters of arrival processes. Results in [8] show that the Hurst parameter in the packet log is ~ 0.5 for fixed file sizes and around 0.7 for file sizes drawn at random from a Pareto distribution with shape parameter 1.5, and that it apparently increases with utilization.[1] Thus, the packet arrival process in the closed-loop case exhibits similar LRD characteristics to the open-loop case.

An explanation for these results can be found by considering the number of transfers in progress. The graphs of Figure 4 show the autocorrelation of the number of packets in the buffer (left), and the autocorrelation of the number of transfers in progress (middle). We can see that the Pareto cases show slowly decaying tails, whilst the constant cases show rapidly decaying tails. Comparing the left and middle graphs of Figure 4 the two sets of autocorrelation functions are qualitatively similar, and hence we might suspect that the autocorrelation in the buffer might be explained by the number of transfers in progress.

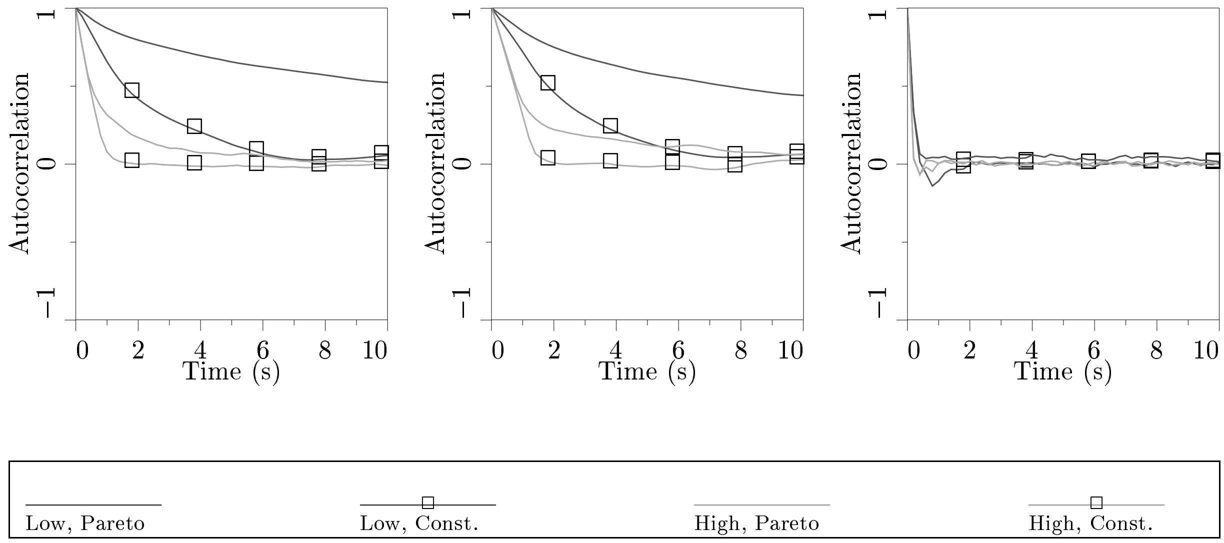

Figure 4. Autocorrelation functions: (left) number of packets in the buffer, (middle) number of transfers in progress (right), the sequence ξ.

[1] The apparent increase may be due to the uniform Round Trip Time (RTT) for all sources, which introduces a regularity in the traffic at the scale of the RTT, distorting the simple power-law relationship, and resulting in an over-estimate of the Hurst parameter over a finite data set.

To examine this observation, let $Q(k)$ and $T(k)$ be two stochastic variables referring to the number of packets in the buffer and the number of transfers in progress at time k respectively and assume that $Q(k)$ is a function of $T(k)$ such that

$$Q(k) = E[Q|T(k)] + D[Q|T(k)]\xi(k) \tag{1}$$

where $E[Q|T(k)]$ and $D[Q|T(k)]$ are the mean and standard deviation of Q conditioned on the number of transfers in progress $T(k)$, and $\xi(k)$ is a sequence of random variables. Rewriting (1) gives

$$\xi(k) = (Q(k) - E[Q|T(k)])/D[Q|T(k)] \tag{2}$$

Generally speaking, in the stochastic variable ξ the influence of T has been decreased compared to Q by subtracting the conditional mean and normalizing by the conditional standard deviation such that ξ has zero mean and unit variance irrespective of T.

The rightmost diagram in Figure 4 displays the autocorrelation of ξ. Comparing the curves to those in the previous two figures, it is clear that the LRD of the original packet process is removed with the above operations. In other words, the strong correlation in the packet process need not be seen as an intrinsic property, but rather as a result of the another process, namely that of the number of transfers in progress.

3.3. Session dependence

To confirm this assertion we can examine the conditional expectation of the buffer occupancy as a function of the number of transfers in progress. Figure 5 was obtained by simultaneously sampling transfers in progress and number of packets in the buffer after which the average number of transfers are obtained for each observed number of packets.

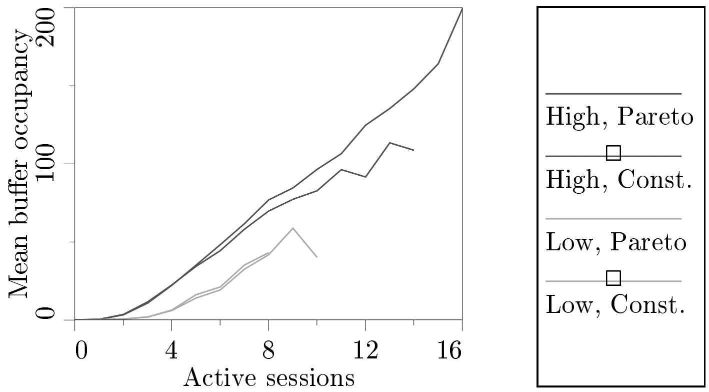

Figure 5. Mean buffer occupancy conditional on the number of transfers in progress.

Two major observations can be made: firstly, there is a striking, almost linear relationship between the numbers of transfers in progress and of packets in the buffer and, secondly, that the curves for the two file sizes more or less overlap for both link loads.

To examine the second observation, let $Q|T = t$ be a stochastic variable referring to the number of packets in the buffer conditioned on there being t transfers in progress. Table 1 gives the result of χ^2-tests of the hypothesis of $Q|T = t$ having the same distribution irrespective of the file size distribution. Comparing the test quantities Σ to the critical values $C_{0.01}(f)$, it is seen that no deviation is significant, *i.e.* the results are in accordance with the hypothesis. The last row gives the number, N, of samples of buffer occupancy with t transfers in progress totaled over both file distributions.

	Low link load		High link load	
	$Q\|T = 3$	$Q\|T = 6$	$Q\|T = 4$	$Q\|T = 8$
Σ	26.2	44.8	44.1	110
f	14	66	49	129
$C_{0.01}(f)$	29.1	89.2	78.2	171
N	3776	243	2451	607

Table 1
Properties of the the two processes.

4. Theory

4.1. A simple processor sharing model

We now seek to explain the large scale behavior (the asymptotic correlations) of the TCP flow control. For this purpose, the fine details are relatively unimportant but it is sufficient to note that TCP shares the bandwidth of a link between a number of connections. We shall model this sharing as equal, but the fact that TCP does not share bandwidth evenly, or fairly does not matter for the analysis to follow.

Given file size analysis such as [4,21], we may assume that the originating applications attempt to transfer heavy-tailed files across the network. However, the flow control limits the rates of each transfer so that (by the assumption above) all rates are equal, and fill the network bandwidth exactly.

Such models are not new — in fact [22] suggest a similar model (though with a finite number of sources) for estimating the performance of TCP/IP networks. However, this paper does not consider the characterization of the resultant traffic, namely, the origin of LRD in this traffic.

To formally define our model: assume that we are considering a bottleneck at which workload arrives as a Poisson process. We assume that the workload, representing a file to transfer across the network, arrives all at once, and has a heavy-tailed distribution. Then the workload in the system can be modeled by the M/G/1 processor sharing queue.[2]

The properties of the M/G/1 queue are well-known. For instance, in the case of processor sharing, the average queue length is insensitive to the workload distribution, except through the mean, and the number of customers in the system will be distributed geometrically, just as in the M/M/1 queue [23, pp. 226–229].

[2]Note that due to the assumption that all of the workload arrives at once, the workload in the system will not really represent the number of packets at a real router. The workload represents the part of the files still remaining to transfer, the packets of which could be distributed over the network, or still waiting to be transmitted at the origin.

We now consider the busy period, *i.e.* the period from the entrance of a customer to the empty system, until the workload once again reaches zero. During the busy period the sum of the outputs of all of the sources adds (within the limitations of the assumptions above) to the bandwidth of the link. That is, the queuing system will itself appear as an On/Off source which has rate equal to the bandwidth of the bottleneck, associating the busy period with the On period, and the Off period with the idle period.

It follows immediately from the Poissonian arrivals that the idle periods will be exponentially distributed. For the busy periods it turns out that there is a simple theorem [24, Theorem 8.10.7, p. 388] which relates the heavy-tails of the workload (or file size) distribution to the distribution of the busy period.

Theorem: *For a stable M/G/1 queue of traffic intensity ρ, slowly varying function $l(\cdot)$, and $\gamma \geq 1$, the following are equivalent:*

$$1 - G(x) \overset{\infty}{\approx} l(x)x^{-\gamma} \tag{3}$$

$$P(T > x) \overset{\infty}{\approx} (1-\rho)^{-\gamma-1} l(x) x^{-\gamma}. \tag{4}$$

where T is the length of the busy period, and $G(\cdot)$ is the service-time distribution.

That is, if the service-time distribution is heavy-tailed with exponent $\gamma \geq 1$, then the busy-period distribution will also be heavy-tailed with the same exponent. Therefore when the workload, or file size distribution has a heavy-tail with exponent $1 < \gamma < 2$, the busy-period will have a heavy-tail with the same exponent. Hence the traffic is LRD with Hurst parameter $H = (3-\gamma)/2$ (as in the On/Off case).

4.2. Queue and buffer lengths

We now turn to the observed queuing behavior. It is not surprising that the queue lengths observed in the closed-loop case are short compared with those in the open-loop case — this is the one of the major intentions of a flow control. The flow control of TCP is based on an adaptive window size. It is thus clear that the buffer size distribution is limited by the number of transfers in progress and their window sizes. There are thus two major sources of variation in the buffer size distribution, namely[3]

(i) the variation in the number of transfers in progress and

(ii) the variation in window sizes.

As noted above, the number of transfers in progress will follow a geometric distribution. Hence, if the majority of variation in buffer sizes comes from the variation in the number of transfers in progress, it is not at all surprising that, as suggested by Figure 3, the buffer length distribution should be approximately exponential.

Similarly, the autocorrelation in the number of transfers in progress has been given in Daley [25] in his results for the autocorrelation of the number of customers present in the M/G/1 queue. This explains the results of Figure 4, giving the autocorrelation for the number of transfers in progress, and hence by the same argument above the autocorrelation of the number of packets in the buffer.

The simulation experiments above showed that the average buffer length was proportional to the number of transfers in progress. It has been suggested elsewhere that the av-

[3]Note there is a third source of variation, namely the burstiness induced in the traffic by the TCP mechanism, even if the window size were fixed. This is a result of a number of factors such as ACK compression too detailed to attempt to model here.

erage window size be used as a performance measure describing TCP file transfers [26,27], and the simulation results here are indeed suggestive that it is a sufficient statistic for approximating the queuing systems performance.

However, although these arguments are useful qualitatively, the simulations above argue against using these result to predict precise performance. In particular, examining the right hand graph of Figure 3, we note that for high utilization, the graphs for the constant case and the Pareto case differ significantly. Noting that the model above is insensitive to the file-size distribution (except through the mean) and therefore the two lines should be very close (as indeed they are for the low load case) it is concluded that the other sources of variation play important roles in determining the behavior of the buffer.

4.3. Extensions

4.3.1. Maximum rate

In reality there will always be a maximum rate for an individual source, due either to window-size, or access-bandwidth limitations. There are two cases to consider:

- when the total access rate is greater than the link bandwidth and
- when the total access rate is less than than the link bandwidth.

The first case is modeled above. The second case is that of superposed On/Off sources. The complexity lies in the fact that the total access rate varies with the number of active sources. A model has been proposed to deal with this in [22,28], but note that for our main purpose, that of explaining the origin of LRD in flow controlled traffic, we need only note that in either case LRD occurs, with the same parameters.

4.3.2. Uneven sharing

Another extension to the work above is to consider the *unfair* sharing of resources. Extension to the above should be possible based on the work described in [29,30], which considers the generalized processor sharing model (such as weighted fair queuing) with heavy-tailed sources. In these papers a slightly different model is used, the queue is fed by N, On/Off sources. The generalized processor sharing divides the capacity by weights which specify the minimum bandwidth guaranteed to each source. Bandwidth in excess of these minimums (occasioned by the fact that the sources are not all continuously on) is redistributed amongst the sources with a backlog. The authors note that this model might be used for TCP where the queue does not represent a buffer, but rather the workload remaining (which might not be localized at the buffer). These results are of interest because they allow unequal weights, thereby allowing for the case where TCP does not share evenly due to external factors (different RTTs for example).

As we noted above, the M/G/∞ model is the limit of the superposition of N equally weighted sources where $N \to \infty$, and the Off periods increases in length with N so that the average rate for the superposition remains constant. Hence, it would be reasonable to assume that the above results can be duly extended (though this is not yet proved).

4.3.3. On/Off sources

We have considered a case which might represent a system with infinitely many sources, and which therefore generates arrivals as a Poisson process. However, we might also wish to consider the case where finitely many sources contribute traffic. Similar results to those used above exist for queues fed by On/Off sources with heavy-tails [31], showing that the index of regular variation for the busy-period is the same as that of the heaviest-tailed source. Thus the results above carry through to the case with finitely many sources.

This result is consistent with the observations that when traffic is broken into individual traffic sources they appear as On/Off sources with heavy-tailed distribution times. If this was done with the On/Off source model above, then the output traffic broken into sources would also appear as On/Off sources with heavy-tails, though the rate during the On period would not be constant, but it is not necessarily constant in real traffic either.

5. Conclusion

The main result of this paper is an explanation for the origin of LRD in TCP traffic. This is different from the previously considered case where simple On/Off sources are multiplexed because in our case the rate of each source depends on the state of the system, rather than being constant, as in the superposition of On/Off sources model. Therefore the performance of the system is substantially different, although LRD is present.

The method allows supplemental arguments which explain other features seen in simulations of such systems, such as the exponential queuing distributions for closed-loop systems, and the form of the autocorrelation of the buffer content.

It is important to note that these result do not deny that LRD has an impact on performance, even for TCP traffic — witness the right hand graph of Figure 3 which shows a clear difference in the performance for the Pareto case, and the case with a constant file-size distribution. The important point is that many performance studies including LRD have neglected to include the *very* important effects of flow controls.

REFERENCES

1. W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, "On the self-similar nature of Ethernet traffic," *IEEE/ACM Transactions on Networking*, vol. 2, pp. 1–15, Feb 1994.
2. V. Paxson and S. Floyd, "Wide-area traffic: The failure of poisson modeling," *IEEE/ACM Transactions on Networking*, vol. 3, no. 3, pp. 226–244, 1994.
3. W. Willinger, M. Taqqu, and A. Erramilli, in *Stochastic Networks: Theory and Applications,* eds. F. P. Kelly and S. Zachary and I. Ziedins, ch. A Bibliographical Guide to Self-Similar Traffic and Performance Modeling for Modern High-Speed Networks, pp. 339–366. Clarendon Press (Oxford University Press), Oxford, 1996.
4. M. E. Crovella and A. Bestavros, "Self-similarity in World Wide Web traffic: Evidence and possible causes," *IEEE/ACM Transactions on Networking*, vol. 5, December 1997.
5. W. Willinger, M. S. Taqqu, R. Sherman, and D. V. Wilson, "Self-similarity through high-variability: Statistical analysis of Ethernet LAN traffic at the source level," *Proceedings of the ACM/SIGCOMM'95*, 1995.
6. M. S. Taqqu, W. Willinger, and R. Sherman, "Proof of a fundamental result in self-similar traffic modeling," *Computer Communications Review*, vol. 27, pp. 5–23, 1997.
7. W. Willinger, V. Paxson, and M. S. Taqqu, "Self-similarity and heavy tails: Structural modeling of network traffic," in *A Practical Guide to Heavy Tails: Statistical Techniques and Applications* (R. Adler, R. Feldman, and M. S. Taqqu, eds.), pp. 27–53, Birkhauser, Boston, 1998.

8. Å. Arvidsson and P. Karlsson, "On traffic models for TCP/IP," in *Proceedings of the International Teletraffic Congress – ITC-16*, (Edinburgh, UK), June 1999.
9. J. Beran, *Statistics for Long-Memory Processes.* Chapman and Hall, New York, 1994.
10. P. Abry and D. Veitch, "Wavelet analysis of long-range dependent traffic," *IEEE Trans. on Info. Theory*, vol. 44, no. 1, pp. 2–15, 1998.
11. G. R. Wright and W. R. Stevens, *TCP/IP Illustrated, Volume 2.* Addison-Wesley Publishing Company, 1995.
12. V. Jacobson, "Congestion avoidance and control," *Communication Review*, vol. 18, no. 4, pp. 314–329, 1988.
13. "Transmission Control Protocol." IETF RFC 793, September 1981.
14. M. Allman, V. Paxson, and W. Stevens, "TCP congestion control." IETF Network Working Group RFC 2581, 1999.
15. S. Floyd, "Connections with multiple congested gateways in packet-switched networks, part I: One way traffic," *Computer Communications Review*, vol. 21, no. 5, 1991.
16. B. K. Ryu and S. B. Lowen, "Point process approaches to the modelling and analysis of self-similar traffic - part i: Model construction," in *IEEE INFOCOM'96*, vol. 3, (San Francisco, California), pp. 1468–1475, March 1996.
17. M. M. Krunz and A. M. Makowski, "Modeling video traffic using $M/G/\infty$ input processes: A compromise between Markovian and LRD models," *IEEE Journal on Selected Areas in Communications*, vol. 16, pp. 733–748, June 1998.
18. "UCB/LBNL/VINT network simulator - ns (version 2)." `http://www.isi.edu/nsnam/ns/`.
19. P. Karlsson and Å. Arvidsson, "TCP/IP user level modeling for ATM," in *Proc. Sixth IFIP Workshop on Performance Modeling and Evaluation of ATM Networks*, (Ilkley U.K.), 1998.
20. M. Abdulaziz, "A study of TCP/IP traffic modelling." Department of Information Technology, Mid Sweden University, 2000.
21. G. Irlam, "Unix file size survey - 1993," `http://www.base.com/gordoni/ufs93.html`.
22. D. Heyman, T. Lakshman, and A. Neidhardt, "A new method for analysing feedback-based protocols with applications to engineering web traffic over the internet," in *SIGMETRICS 1997*, pp. 24–38, 1997.
23. L. Kleinrock, *Queueing Systems II: Computer Applications.* John Wiley and Sons, 1975.
24. N. Bingham, C. Goldie, and J. Teugels, *Regular Variation.* Cambridge University Press, Cambridge England, 1987.
25. D. Daley and R. Vesilio, "Long range dependence of inputs and outputs of some classical queues," *Fields Inst. Commun.*, vol. 28, 2001.
26. M. Mathis, J. Semke, J. Mahdavi, and T. Ott, "The macroscopic behavior of the TCP congestion avoidance algorithm," *Computer Communication Review*, vol. 27, pp. 67–82, July 1997.
27. J. Padhye, V. Firoin, D. Towsley, and J. Kurose, "Modeling TCP throughput: A simple model and its empirical validation," in *ACM SIGCOMM'98*, 1998.
28. D. Heyman, T. Lakshman, and A. Neidhardt, "Engset-model based method for analysing feedback protocols with applications to web traffic engineering," in *DIMACS Workshop on Performance of Realtime Applications on the Internet*, 1996.
29. S. Borst, O. Boxma, and P. Jelenković, "Asymptotic behavior of generalized processor sharing with long-tailed traffic sources," in *INFOCOM'2000*, 2000.
30. S. Borst, O. Boxma, and P. Jelenković, "Generalized processor sharing with long-tailed traffic sources," in *ITC'16*, 1999.
31. O. Boxma, "Regular variation in a multi-source fluid queue," ITC 15, vol. 2b, pp. 391–402, Elsevier, Amsterdam, 1997.