Bursty Fluid Approximation of TCP for Modeling Internet Congestion at the Flow Level.

Daniel Genin
NIST, Gaithersburg, MD
Email: dgenin@nist.gov

Vladimir Marbukh
Senior Member, IEEE,
NIST, Gaithersburg, MD
Email: marbukh@nist.gov

***Abstract*—We propose an improved model for TCP flow level congestion performance. Conventional fluid approximation models yield incorrect results for TCP congestion at the flow level due to throughput overestimation. We introduce a more accurate model incorporating some of the packet level burstiness, by approximating TCP sources as on-off fluid sources. Our model is largely based on the work of D.Anick, D.Mitra and M.M.Sondhi exploring buffer statistics of the superposition of on-off sources. Incorporating some of the TCP burstiness at the packet level while retaining the over all fluid approximation framework allows us to substantially improve the accuracy of the flow level congestion model at the price of a relatively small increase in mathematical complexity. The model is extensively validated against *ns2* simulations and shown to perform better than the M/M/1/B based model typically used in this context.**

***Index Terms*—TCP, flow level congestion, fluid approximation, congestion control**

I. Introduction

It is a well known fact that TCP is currently responsible for transporting the bulk of Internet traffic. This makes models of TCP performance essential tools for network operators as well as application designers. Significant progress has been made in understanding the steady state behavior of multiple TCP flows traversing a network, e.g. [6],[10],[3],[7],[1],[11] to cite just a few of the many publications on the topic. However, in real networks, like the Internet, the number of active flows is constantly changing and, in recent years, particular attention has been given to the study of TCP performance under a steady arrival stream of document transfer flows. This mimics what has been dubbed elastic traffic, comprising web page, music, video, and miscellaneous file downloads. Performance of such flows is measured by their duration and the degree to which they accumulate in the network reducing individual user throughput. Note that this type of models is different from the class of models with a fixed number of continuously transmitting flows. The latter are mainly used to study equilibrium properties and near equilibrium dynamics of TCP on the timescales much shorter than the duration of a typical document download. On the other hand, in the models with elastic traffic, often called flow level congestion models since the quantity of interest is the number of concurrent flows in the network, the number of flows changes as active flows finish and new flows arrive.

Flow level congestion can be studied by running detailed packet level simulators such as *ns2*. This approach while giving highly accurate results is computationally intensive and for realistic networks prohibitively so. Moreover, simulations yield information about specific combinations of network parameters, requiring a large number of simulations to map out performance over the range of operational network conditions. At the opposite extreme are fluid approximation based mathematical models, which assume separation of time scales between flow duration and convergence to equilibrium transmission rate under TCP. These models ignore the details of packet traffic all together, instead assuming efficient, i.e. without waste, link bandwidth partitioning between concurrent flows [5]. The latter approach while lacking the extensive computational requirements of the packet level simulations is also significantly less accurate. Thus, as is often the case we are faced with a trade off between accuracy and computational intensiveness. Our goal is to improve this trade off by introducing a fluid approximation based, and hence computationally undemanding, model, which at the expense of some increase in mathematical complexity performs better than the currently available fluid approximation models.

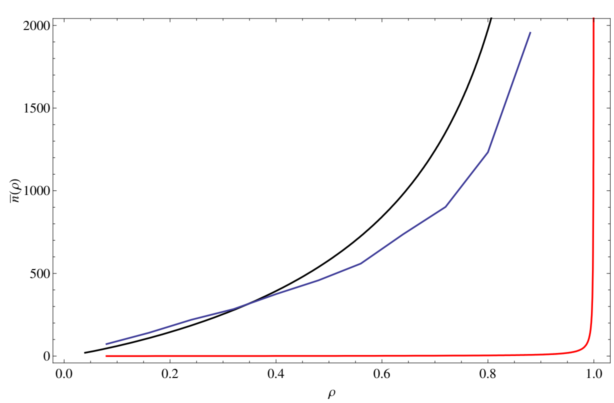


Fig. 1. Number of concurrent flows vs. load based on 1 (red), *ns2* (blue) and the proposed model (black).

As a reference point we take the flow level congestion model studied in [5], which has gained wide acceptance in the literature. In [5] the number of concurrent flows on a link is hypothesized to behave as the number of customers in an M/G/1 processor sharing queue. Assuming efficient bandwidth utilization by TCP this gives

$$\bar{n} = \frac{\rho}{1-\rho} \tag{1}$$

for the mean number of concurrent flows on the link, where ρ is the offered load. Figure 1 compares $\bar{n}(\rho)$ deduced from *ns2* simulations and the equation (1) for a 1 Gbps link loaded with a Poisson arrival stream of exponentially distributed file transfers of mean size 100KB. We note that we concentrate on the case of a single link as a fundamental test case on which a model claiming any degree of accuracy must perform adequately and on which results about more complicated networks may be built.

It is clear from the diagram that the analytical model (1) significantly underestimates the number of concurrent flows. For comparison we also include the corresponding plot for the proposed model. The main reason for this discrepancy is the assumption of efficient capacity utilization under TCP. As our results show TCP throughput can be as low as 60% of the raw link capacity depending on the propagation delay and router buffer size.

We propose to correct the above shortcoming by bringing some of the packet level dynamics back into the model. The proposed modification is based on the Anick-Mitra-Sondhi(AMS) model introduced in [2] to study queuing statistics of multiplexed on-off sources. Certain aspects of TCP packet dynamics, described in Section III, make the model particularly well suited for representing multiplexing of a large number of TCP flows on a link. In spite of the complexity of the AMS model a relatively simple and accurate approximation can be easily deduced based on the work presented in [2].

The rest of the paper is arranged as follows. We begin with a discussion of existing TCP throughput models and a brief literature review in Section II. In Section III we present the case for using the AMS model followed by an adaptation of the AMS model to the present setting in Section IV. In Section V we derive a simple approximation for packet loss probability based on the dominant eigenmode approximation introduced in [2]. The model is validated in Section VI against *ns2* simulations. We then show that the mean-field approximation for the flow level congestion model based on AMS performs substantially better than (1 and tolerably well when compared with *ns2* simulations. Section VII summarizes our findings and discusses directions for future work.

II. The problem

The main shortcoming of the model in (1) is that it completely disregards packet level dynamics of TCP assuming instead that the *all* available bandwidth is completely divided between concurrent flows. This would be an adequate approximation if TCP was at least approximately efficient in taking advantage of the link capacity. However, the actual throughput of a link multiplexing even a very large number of TCP flows can be far below its raw capacity depending on the round-trip propagation delay and buffer size. The main problem, thus, is to find an accurate approximation for the TCP throughput in terms of the intrinsic link parameters and the number of concurrent flows, where by *throughput* we mean the number of packets successfully delivered per second on a link with the given parameters and a given number of concurrent flows, and by TCP we mean TCP-Reno. (Note, however, that the packet loss probability model derived below may be applicable to other versions of TCP using the “self-clocking” mechanism of sending a new packet only when an acknowledgment is received.) Since what can actually be computed based on the mechanics of the congestion avoidance algorithm is the transmission rate, throughput is computed as

$$throughput = transmission\,rate \times (1 - packet\,loss\,probability) \tag{2}$$

TCP transmission rate is closely linked with packet loss probability on the link since TCP relies on packet loss for detecting network congestion and controlling its transmission rate. The square root relationship between packet loss probability and transmission rate

$$x = \frac{1}{T}\sqrt{\frac{8/3}{p}}, \tag{3}$$

where x stands for throughput and p for packet loss probability is well known and has been substantially validated by live Internet measurements [4], [6]. However, packet loss probability is not an intrinsic property of the link since it itself depends on the transmission rate. Thus, (3) gives only the general form of the fixed point equation determining throughput. To obtain an equation for throughput in terms of the intrinsic link parameters p must in turn be expressed in terms of these same parameters and transmission rate.

If the aggregate packet arrival process were a Poisson process the packet loss probability could be approximated by the buffer overflow probability from an M/M/1/B or M/M/1/∞ (if the buffer is very large) queuing model. However, it has been demonstrated that TCP packet arrival process is not a Poisson process even when the number of multiplexed flows is large [8]. In any case, it is easy to see that M/M/1/* models yield incorrect throughput by simply comparing the graphs of the solution to (3) with

$$p(x) = (x/c)^B(1 - x/c)/(1 - (x/c)^{B+1}) \tag{4}$$

or

$$p(x) = (1 - c/x)^+, \tag{5}$$

where c is router capacity, corresponding to buffer overflow probabilities for M/M/1/B and M/M/1/∞ respectively, and results of *ns2* simulations. As Figure 6 shows M/M/1/* model predict that throughput is essentially equal to raw link capacity while *ns2* simulations show that depending on the propagation delay and buffer size the throughput can be as low as 60% of the raw link capacity. In spite of this apparently stark inaccuracy M/M/1/* packet loss models continue to be used in fluid flow approximation models [9].

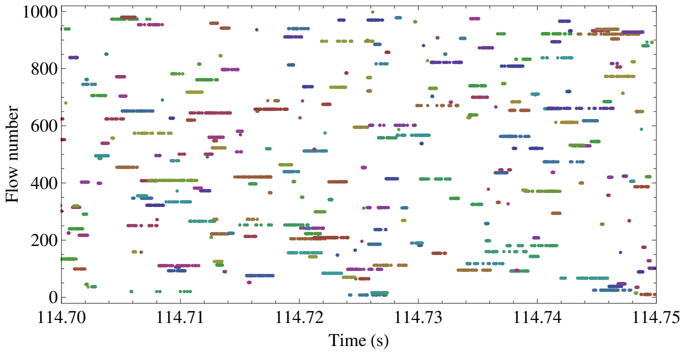


Fig. 2. Packet arrival times separated by flow number

III. Structure of TCP packet arrival process

Poor performance in M/M/1/* based packet loss models is due mainly to the non-Poisson character of the packet arrival process. Under TCP packets tend to arrive in tight batches (or bursts) which greatly increases the number of dropped packets relative to a Poisson stream with the same average rate.

To obtain a more accurate picture of the packet arrival process we studied the *ns2* simulation traces (for configuration details see Section VI). Figure 2 diagrams packet arrival times split vertically by flow number on a timescale of a round trip propagation time. As can be seen from the figure packets from a given flow arrive in tight batches. For the purposes of the discussion we will define a *batch* as a group of packets belonging to the same flow and such that the inter-arrival time between successive packets is less than 1% of the round-trip time.

We make the following observations about the structure of the arrival process.

I) *The aggregate batch arrival process is well approximated by a Poisson process (Fig. 3).* This is a consequence of it being a superposition of a large number of periodic processes with nearly identical periods and random phases. Periodicity in this case refers only to the time of arrival and not to the batch size. Furthermore, the batch arrival rate is very close to RTT/N, where RTT is the round-trip propagation delay and N the number of concurrent flows, suggesting that each batch corresponds to a congestion window's worth of packets sent in reply to the acknowledgments for the previous congestion window. We make this our working hypothesis (to be validated by comparison with *ns2* simulations in Section VI)

II) *The batch size distribution is well approximated by a Gaussian tail distribution* for most parameter values (Fig. 4). Assuming that our hypothesis about the correspondence between batch and congestion window sizes is correct, the parameters of the batch size distribution are determined by the operation of the TCP congestion control. Bacelli et al. [1] have deduced the congestion window size distribution based on the assumption that

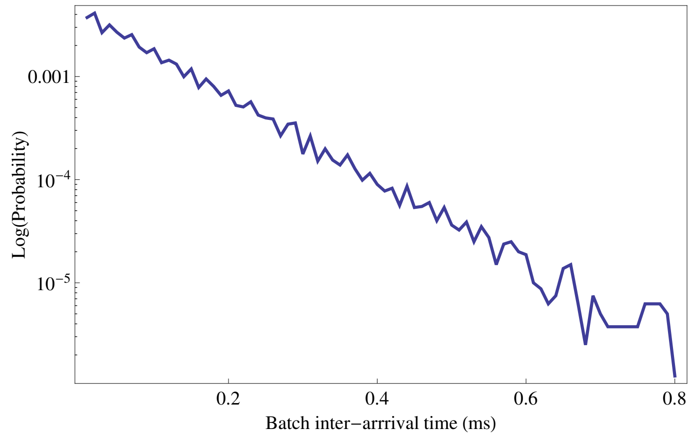
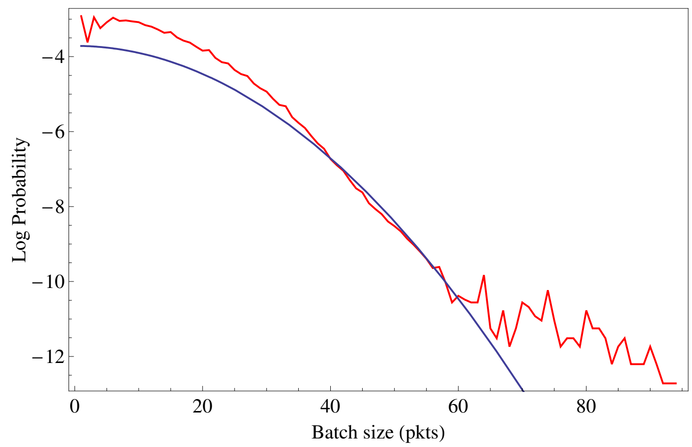


Fig. 3. Sample log probability plot of batch inter-arrival time distribution based on *ns2* packet traces.



(b)

Fig. 4. Sample log plot of *ns2* empirical batch size distribution (red) and the approximating Gaussian tail distribution (blue).

the packet loss process is a variable rate Poisson process. It is interesting to note that the stationary congestion window size distribution obtained in [1] is an infinite sum of functions of the form $\exp(-aw^2)$, where $a > 0$ is a constant coefficient and w is the congestion window size, which are exactly Gaussian tail distributions.

III) *Sizes of consecutive batches in the aggregate stream are very nearly uncorrelated.* Although, the sizes of consecutive batches for a given flow are correlated since if the batch size is equal to the congestion window size then the size of the succeeding batch is either 1 greater or is half of the size of the preceding one. On the other hand, when batch arrivals from all flows are aggregated in a single stream then the sizes of consecutive batches are very nearly uncorrelated.

IV. Anik-Mitra-Sondhi model adaptation

A simple fluid approximation model capable of reproducing the main features of the arrival process described above is the

Annik-Mitra-Sondhi model [2], originally developed to model queuing statistics of ATM multiplexers.

We briefly describe an adaptation of the AMS model to the present setting. The arrival process is modeled as a superposition of N on-off sources. The "on" periods arrive according to a Poisson process with rate $\lambda = 1/RTT$. The duration of "on" periods is taken to be exponentially distributed with parameter μ, $\mu^{-1} = w/C$, where w is the mean batch (or congestion window) size and C is the router capacity. According to the observations of the previous section a more accurate model would probably result if the "on" periods were distributed according to a Gaussian tail distribution but as results of Section VI show, exponential distribution may already provide an adequate approximation and is much easier to deal with mathematically. During an "on" period data arrives at a constant rate C (same as the router capacity). Arriving data is processed and discharged from the queue at the same rate C. The buffer can hold up to B units (packets) of data. Once the buffer is full any subsequently arriving data is discarded.

The mean number of concurrently active sources is easily computed to be $N\lambda/\mu$ since if "on" periods are treated as customers then the system is isomorphic to an M/M/∞ queuing system. Given that source transmission rate is C, we have the necessary stability condition in terms of the offered load

$$\rho = \frac{Nw}{CT} < 1. \tag{6}$$

In TCP terminology this translates into the natural requirement that for the queue to remain finite the aggregate congestion window must be smaller than the bandwidth-delay product.

V. Many flow asymptotic limit

In spite of certain mathematical complexity of the AMS model it turns out, nevertheless, to be explicitly solvable in the case when the buffer is infinite. The reduction to a linear system of ordinary differential equations (o.d.e.'s) and an elegant solution utilizing a generating function of eigenvectors is lucidly laid out in [2]. The end result is a closed formula for stationary probability of buffer overflow beyond x solely in terms of the parameters of the model — λ, μ, C, and N. We note that there are some minor differences in model parametrization between [2] and the present exposition so not all formulas translate directly.

Serious mathematical difficulties arise when the buffer size is taken to be finite, which, of course, is the case we are most interested in. It is quite likely that explicit formula for buffer overflow probability in this case cannot be found due inherent non-linearity of the problem. In view of these mathematical difficulties we make the crude assumption (justified by numerical simulations) that the buffer overflow probability for finite buffer AMS has the same exponential form as the infinite buffer AMS but with a scaled exponent. Note that this adds only one extra degree of freedom to the model.

In the infinite buffer case the stationary buffer content probability distribution is described by a linear system of o.d.e.'s. The solution of such a system is a linear combination of eigenmodes corresponding to eigenvalues of the coefficient matrix. However, as is often the case the dominant eigenmode carries most of the weight with higher eigenmodes contributing relatively little so that the dominant eigenmode alone is already a good approximation to the exact solution. The dominant eigenmode approximation for buffer overflow beyond x for the infinite buffer case is [2]

$$G(x) \approx \rho^N \left(\prod_{i=1}^{N-\lfloor C \rfloor -1} \frac{z_i}{z_i + r} \right) e^{-rx}, \tag{7}$$

where z_i are the eigenvalues of the linear system, $r = -z_0$ and ρ is the offered load. The eigenvalues z_i turn out to be solutions of a family of quadratic equations parametrized by i (see [2] for details). Since we are interested in the statistics for large N we approximate z_i and ρ by their values in the limit of large N with $C = cN$, where c is the per flow capacity. Passing to the limit we get

$$\begin{aligned} \rho &= \lambda/\mu \\ r &= -\frac{\mu - \lambda}{c} \end{aligned}$$

for very large N. Furthermore, straightforward computations show that products of conjugate eigenvalues (eigenvalues come in pairs corresponding to quadratics) converge to ρ^{-2} as N tends to infinity, so that

$$\rho^N \prod_{i=1}^{N-\lfloor C \rfloor -1} \frac{z_i}{z_i + r} \approx \rho \tag{8}$$

for very large N. Putting this all together and substituting for λ and μ from Section IV we obtain an approximation for probability of buffer overflow as a function of congestion window size w

$$p(w) = \frac{w}{cT} e^{-\alpha(c/w - 1/T)B/c}. \tag{9}$$

where α is the unknown factor compensating for the finite buffer size.

Observe that $p(w)$ has the correct behavior at the extremes:

$$\begin{aligned} \lim_{w \to 0} p(w) &= 0 \\ \lim_{w \to cT} p(w) &= 1, \end{aligned}$$

and is monotonically increasing in w.

VI. Validation

We validate the model derived in Section V by comparing its predictions with *ns2* simulations. The model network we use for validation consists of 1000 TCP sources aggregated at a bottleneck router with 1Gbps capacity and a buffer of size B pkts ranging from 50 pkts to 300 pkts in increments of 50 pkts (Fig. 5). The sources are limited only by the capacity of the router. The main component of propagation delay is the delay on the link connecting the router and the sink node, is denoted by T_0 ms and ranges from 50 ms to 300 ms in increments of 50 ms. The propagation delay T on the access links, connecting

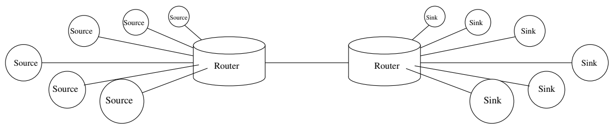


Fig. 5. Simulated network configuration.

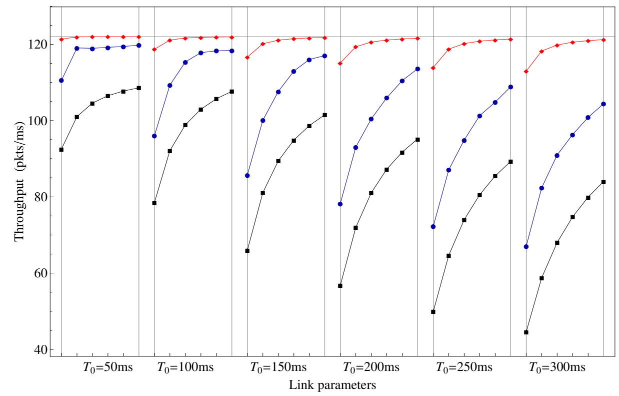


Fig. 6. Throughput computed from *ns2* simulations (blue) and square root law with p given by AMS model(black), M/M/1/B (red). The horizontal line at 122 pkts/ms is the router capacity and also the throughput predicted by the M/M/1/∞ model.

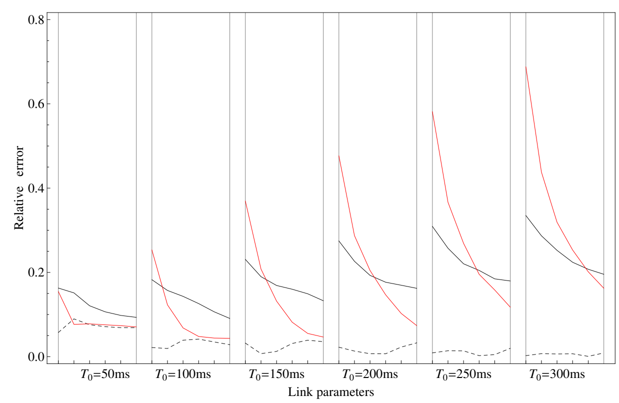


Fig. 7. Relative error for the AMS model with α=1.4 (black), α=4 (dashed black) and M/M/1/B (red).

individual sources to the router, is uniformly distributed in the interval $[0, 10\% T_0]$. This variability in the propagation delay is introduced to avoid phase-locking between flows. All sources are controlled by TCP-Reno with fast retransmit disabled. The sources begin transmitting after a small random delay uniformly distributed in $[0, .1]$ seconds and transmit continuously throughout the simulation. The delay in the start of transmission is also meant to destroy any correlation that may arise had the flows been started simultaneously.

The finite buffer correction factor, α, was calibrated against numerical simulations of the *finite* buffer AMS model executed over a three dimensional grid spanned by T, B and w. The least squares fit over the resulting data produced $\alpha = 1.4$ so that the AMS model packet loss probability is given by

$$p(w) = \frac{w}{cT} e^{-1.4(c/w - 1/T)B/c}. \tag{10}$$

Finally, combining (10) with (3) and using $x = w/T$ we have an equation for the approximate mean congestion window size in terms of the basic link parameters

$$w = \sqrt{\frac{8/3}{(w/cT) e^{-1.4(c/w-1/T)B/c}}}. \tag{11}$$

Figure 6 compares throughput computed from *ns2* simulations (blue), AMS model (black) and M/M/1/B model (red). For the analytical models the throughput was computed as

$$throughput = (1 - p(x))x, \tag{12}$$

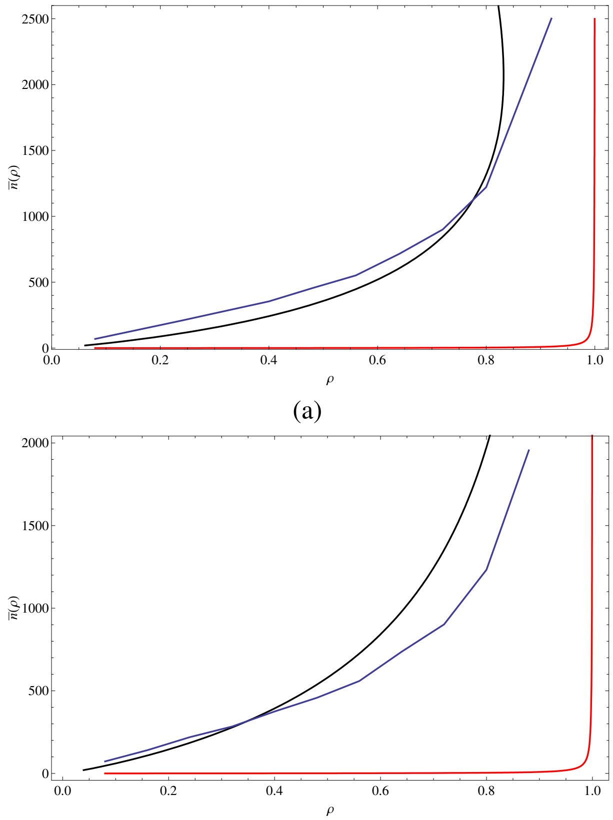
where x is the equilibrium transmission rate computed from the square root law (3). The 36 link parameter pairs of T_0 and B are sorted into six groups of six, first, by increasing T_0 and, then by increasing B. In Figure 6 each pane corresponds to a group of six combinations with the same T_0. This format permits us to display all 36 parameter combinations on an easily readable two dimensional graph. As the figure indicates the AMS model is generally closer to the *ns2* throughput curve, although, for low propagation delay and for large buffer sizes AMS error is similar to or larger than the M/M/1/B error. As we will see shortly, however, where flow level congestion is concerned, overestimating throughput causes much larger errors than underestimating. Note, also, that it is the non-trivial lower bound on TCP throughput that is important in practice since the upper bound is obvious.

Figure 7 compares relative error in predicted throughput for AMS and M/M/1/B models, computed relative to the throughput measured in *ns2* simulations. Although, AMS performs better than M/M/1/B the relative error is still substantial reaching as high as 35% in the worst case. While the AMS packet loss probability formula is a product of a large number of approximations and simplifying assumptions our numerical experiments suggest that the bulk of the error is caused by the passage from the discrete packet model to continuous fluid approximation. We note in passing that increasing α to 4 significantly improves the performance of the model (Figure 7 dashed line),but this parameter choice can only be justified a posteriori by comparison with *ns2* data and we have no theoretical justification for selecting this particular value.

We return now to the question of flow level congestion model. Instead of continuously transmitting flows we now have a steady stream of arriving document transfer flows. Let the flow arrival rate be l flows per second and the mean document size be b bytes then the aggregate load is lb B/s. Let $\tilde{c}(n)$ be the throughput of the link when the number of active flows is n. If the aggregate load does not exceed the capacity of the link the number of concurrent flows will converge to an equilibrium value $\bar{n}$ which must satisfy

$$\tilde{c}(\bar{n}) = lb \tag{13}$$

This is a mean-field approximation which is asymptotically exact as the flow arrival rate and capacity tend to infinity,



(a)

(b)

Fig. 8. Mean number of active flows as a function of utilization for 1Gbps link with $B = 100$ pkts and $T_0 = 100$ ms (a) and $T_0 = 300$ ms (b). Flow size is exponentially distributed with mean 100 KB.

diminishing the relative size of fluctuations about the equilibrium. Thus, given the throughput in terms of the link parameters it is possible to estimate the equilibrium number of flows in the system. Figure 8 shows mean number of flows on a 1Gbps link as a function of utilization, lb/C, under a load of exponentially distributed elastic flows with mean size 100 KB, for *ns2*, the flow congestion model (1) and the AMS based mean-field model. We note that to compute $\tilde{c}$ for the AMS mean-field model we used a modification of the square root law (3) for large p

$$w = \sqrt{\frac{8(1-p)}{3p}} \tag{14}$$

found in [6]. While the AMS based mean-field model is far from perfect it can be seen to predict the equilibrium number of flows in the system much more accurately than (1).

Figure 8 also raises several questions. First, it appears to indicate that for the same average number of flows on the link the TCP throughput is lower when flows come and go than when flows transmit continuously. This can be seen from the AMS model underestimating the number of flows for a given utilization compared with the *ns2* simulations in Figure 8 (a), which corresponds to link parameters for which AMS *underestimates* TCP throughput for continuously transmitting flows by about 15%. If the TCP throughput was the same for elastic traffic as for continuously transmitting flows the order of curves in Figure 8 (a) would have been opposite. The reason for decrease in throughput under elastic traffic could be the cost of setting up and tearing down of TCP connections.

Figure 8(a), also, shows that at high loads the AMS model departs from *ns2* simulations in another critical way. The reversal in the AMS curve shows that increasing the number of active flows beyond a certain threshold causes the throughput to decrease contrary to what *ns2* indicates. The reason for this behavior is that the transmission rate under AMS model cannot exceed router capacity C, while packet loss probability approaches 1 as the bandwidth per flow tends to zero, leading to a corresponding drop in throughput. This picture might indeed have been accurate if TCP always remained in congestion avoidance mode. In congestion avoidance mode at most one packet per round-trip time can be sent without a corresponding acknowledgment from the sink and the rate of acknowledgment packets clearly cannot exceed the router capacity. At very high loads, however, time-outs and subsequent slow start phases begin to account for a significant portion of transmitted data. This appears to be born out by *ns2* data showing that the mean transmission rate exceeds the router capacity when the number of flows is very large (even when flows are transmitting continuously).

VII. Conclusion

We presented a new model for TCP throughput based on an old model of Anick, Mitra and Sondhi for modeling buffer statistics of ATM multiplexers. Additional accuracy is achieved by accounting for some of the packet level burstiness, while retaining fluid approximation approach yields an explicit formula in terms of the basic link parameters. The presented model provides a lower bound on the TCP throughput and is typically within 20% of the *ns2* simulation throughput, which under most circumstances is better than the commonly used M/M/1/B based models. Using the AMS-based throughput formula in the simple mean-field model of flow level congestion also gives substantially better results than the commonly used formula based on the assumption of efficient bandwidth partitioning.

On the other hand, the AMS model clearly still has a lot of room for improvement. The fluid approximation itself is a major cause of error in the model eliminating which, even partially, is likely to substantially boost the accuracy of the model. At high loads the accuracy of the model will also be improved by accounting for the TCP slow start phase. Finally, more data is necessary to address validity of the AMS mean-field approximation for flow level congestion performance.

References

[1] François Baccelli, David R. McDonald, and Julien Reynier. A mean-field model for multiple tcp connections through a buffer implementing red. *Perform. Eval.*, 49(1/4):77–97, 2002.

[2] M.M. Sondhi D. Anick, D. Mitra. Stochastic theory of a data handling system with multiple sources. *ICC'80; International Conference on Communications, Seattle*, 1, 1980.

[3] D. Tan F. Kelly, A. Maulloo. Rate control for communication networks: shadow prices, proportional fairness and stability. *J. Oper. Res. Soc.*, 49:237–252, 1998.

[4] Sally Floyd. Connections with multiple congested gateways in packet-switched networks. part 1: One-way traffic. *ACM Computer Communication Review*, 21:30–47, 1991.

[5] S. Ben Fredj, T. Bonald, A. Proutiere, G. Régnié, and J. W. Roberts. Statistical bandwidth sharing: a study of congestion at flow level. *SIGCOMM Comput. Commun. Rev.*, 31(4):111–122, 2001.

[6] D. Towsley J. Padhye, V. Firoiu and J. Kurose. Modeling tcp reno performance: a simple model and its empirical validation. *ACM, Transactions on Networking*, 8(2):133–145, 2000.

[7] R. Srikant L. Ying, G. Dullerud. Global stability of internet congestion controllers with heterogeneous delays. *Transactions on Networking*, 14(3):579–591, 2006.

[8] Vern Paxson and Sally Floyd. Wide-area traffic: The failure of poisson modeling. *IEEE/ACM Transactions on Networking*, 3:226–244, 1995.

[9] R. Srikant. *Mathematics of Internet congestion control*. Willey, 2001.

[10] D. Towsley V. Misra, W. Gong. Fluid-based analysis of a network of aqm routers supporting tcp flows with an application to red. *SIGCOMM*, 2000.

[11] G. Vinnicombe. On the stability of end-to-end congestion control for the internet. *Univ. Cambridge Tech. Rep. CUED/F-INFENG/TR.398*, 2001.