Stochastic Models for Throughput Analysis of Randomly Arriving Elastic Flows in the Internet

Arzad A. Kherani and Anurag Kumar
Dept. of Electrical Communication Engg.
Indian Institute of Science, Bangalore, 560 012, INDIA
email: alam, anurag@ece.iisc.ernet.in

Abstract—**This paper is about analytical models for calculating the average bandwidth shares obtained by TCP controlled finite file transfers that arrive randomly and share a single (bottleneck) link. Owing to the complex nature of the TCP congestion control algorithm, a single model does not work well for all combinations of network parameters (i.e., mean file size, link capacity, and propagation delay). We propose two models, develop their analyses, and identify the regions of their applicability. One model is obtained from a detailed analysis of TCP's AIMD adaptive window mechanism; the analysis accounts for session arrivals and departures, and finite link buffers. It is essentially a Processor Sharing (PS) model with time varying service rate; hence we call it TCP-PS. The other model is a simple modification of the PS model that accounts for large propagation delays; we call this model Rate Limited-PS (RL-PS). The TCP-PS model analysis accommodates a general file size distribution by approximating it with a mixture of exponentials. The RL-PS model can be used for general file size distributions.**

We show that the TCP-PS model converges to the standard PS model as the propagation delay approaches zero. We also observe that the PS model provides very poor estimates of throughput unless the propagation delay is very small. We observe that the key parameters affecting the throughput are the bandwidth delay product (BDP), file size distribution, the link buffer and the traffic intensity.

Several numerical comparisons between analytical and simulation results are provided. We observe that the TCP-PS model is accurate when the BDP is small compared to the mean file size, and the RL-PS model works well when the BDP is large compared to the mean file size.

I. Introduction

Traffic engineering and bandwidth dimensioning in the Internet require effective models that can predict the performance as seen by network users. The Internet carries predominantly elastic traffic, and the bandwidth sharing between elastic sessions is controlled by TCP. Hence there is a need for developing performance models that can be used to calculate the throughputs obtained by TCP controlled elastic sessions. These models must capture the realistic situation that the network bandwidth is shared dynamically between a randomly varying number of concurrent elastic flows. The number of concurrent sessions is randomly time varying since sessions randomly arrive, transfer finite volumes of data and depart. Recently the importance of such models has been recognised; see [1], [2], [3], [4], [5], [6], [7]. Obtaining accurate analytical models that work over a wide range of parameters remains a challenging problem, however.

In this paper we seek models for calculating the throughput of TCP controlled finite duration transfers sharing a single bottleneck link; single bottleneck link models have also been studied in [1], [2], [5] and [7]. A single link model would itself be helpful, for example, in dimensioning an enterprise network where many branch sites access data from a high performance data center over a high speed backbone, and hence the performance is limited by each site's access link. Also, modelling the bandwidth sharing on a single link is relatively tractable and may be applicable to modelling a network of links using an approach similar to that in [8], and hence could be useful in traffic engineering and bandwidth dimensioning of internets.

We make the following modelling assumptions: session arrival instants constitute a stationary Poisson process (as observed in [9]); each session needs to transfer a random volume of fluid data, which we will assume to be hyper-exponentially distributed; the sequence of transfer volumes is i.i.d. (independent and identically distributed). Similar modelling assumptions have also been made in [5], [6], [7]. We also assume that the propagation delay for every session is the same; such an assumption is adequate for a situation where the propagation delays of sessions sharing the link are dominated by one large propagation delay. Also in this paper we do not model random discard (e.g., RED) but assume tail drop; note that a recent paper [10] suggests that RED may not be useful for web transfers.

For the above mentioned scenario we observe that a single model, such as Processor Sharing (PS), does not work well for all values of network parameters (link capacity, propagation delay, mean transfer size). We then divide the range of parameter values into two regions (bandwidth delay product (BDP) $<$ mean file size, BDP $>$ mean file size), and develop models to be used for each of these regions based on region specific assumptions and observations. For BDP $<$ mean file size, transfers enter the congestion avoidance phase, and we develop a detailed model capturing the effect of TCP's AIMD mechanism on the total rate with which active sessions are served; we call this the TCP-PS model, where PS stands for processor sharing. For the TCP-PS model we identify renewal epochs embedded in the total rate process and use a "reward" rate analysis to obtain the throughput. For BDP $>$ mean file size, we use the Rate Limited PS (RL-PS) model which behaves like the M/G/∞ queue when the number of active sessions is small so that they do not interact, and is like the PS model when the number of sessions is large.

A. Related Literature

In [1] and [2] the authors have considered a situation in which several Internet subscribers share a single backbone link, and they are attached to this link by lower speed access links. Each subscriber goes repeatedly through a download and "think" cycle. The authors' aim is to develop an approximate analysis for the throughput obtained by subscribers. Assuming zero propagation delay, and ideal bandwidth sharing (i.e., the entire bot-

This work was supported by a research grant from HFCL.

0-7803-7476-2/02/$17.00 (c) 2002 IEEE.

tleneck link rate is applied to the ongoing transfer, and the rate is shared equally), a closed queueing network model applies. In [1] the authors also provide an approximate "correction" to the results so obtained to take care of TCP's adaptive window behavior. They do not model slow start which we observe is important in the case of a large BDP.

In [11] the authors analyse bandwidth sharing with an AIMD control protocol under the assumption of a fixed number of *persistent* sessions; the slow start phase is not modelled. TCP controlled bandwidth sharing with *persistent* connections is also modelled in [12] by applying a general method proposed in [13] to the specific case of TCP Tahoe.

In [14] the authors consider a generalisation of the stochastic model that we have stated above (i.e., Poisson session arrivals and exponential transfer volumes) to the case of a network. Assuming various bandwidth sharing mechanisms, the authors show that the vector process of the number of ongoing sessions on each route is a Markov process. They show that a simple and natural condition is sufficient for the positive recurrence of this Markov chain. The case of a single link when there is no propagation delay on the link has been addressed in [15] using the PS model. The PS model has also been proposed in [16] for a scenario similar to that studied in [1]. [6] explores statistical bandwidth sharing in a network illustrating the impact on stability of certain service differentiation mechanism. [5] uses the Discriminatory Processor Sharing (DPS) as a model for a single bottleneck link shared by several classes identified by their round trip times.

We observe that in the case of a single link, single class network, the models in [6] and [5] reduce to the standard PS model, and hence yield throughputs independent of the link propagation delay which is, as shown in the present work, not correct.

In fact in the present paper we find that for BDP $<$ mean file size (but not very small; BDP $>$ 1 packet) the performance is sensitive to the BDP, to the link buffer size and even to heavy tails in the file size distribution. The TCP-PS model that we develop in this paper can capture all these sensitivities, as the BDP, buffer size and the file size distribution (modeled by a mixture of exponentials) are all parameters in the model. Furthermore, the TCP-PS model converges to the PS model as BDP $\rightarrow 0$; this supports the use of PS model for very small BDP. The TCP-PS model models the details of TCP's AIMD phase. For BDP larger than the mean file size, however, most sessions complete their transfers in slow start, hence TCP's AIMD behaviour plays less of a role, and we find the need for a different model, the RL-PS model.

The paper is organised as follows: Section II introduces some of the notation used in the paper. In Section III we discuss the various measures of throughput for bandwidth sharing among nonpersistent sessions. Section IV gives a brief summary and motivation for the various models used in the paper for calculating throughputs of TCP controlled nonpersistent sessions. Sections VI, VII and VIII give details of the models motivated in Section IV; these sections also provide numerical comparisons of the results obtained from the simulations with the results obtained from the models. Section IX concludes the paper.

II. Notation Used

Figure 1 shows a schematic of the model. All the files undergoing transfer through the link can be viewed as being in a virtual queue to which new files arrive. TCP's mechanisms control the transfer of the packets of the files over the link.

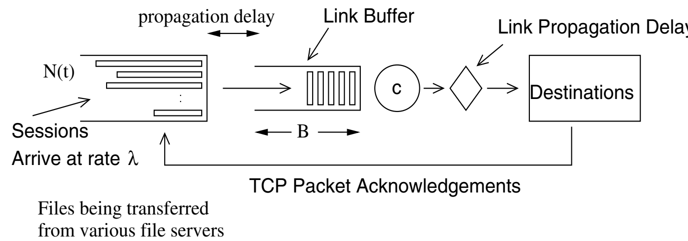

Fig. 1. Schematic of finite volume file transfers sharing a bottleneck link over a wide area network. $N(t)$ is the number of files concurrently being transferred. The packets in transit are in the propagation delay pipe or in the link buffer.

Parameters and Constants

λ the rate of the Poisson process of session arrivals
$\frac{1}{\mu}$ the mean file size
c link capacity (packets per second)
ρ the normalised link load ($=\frac{\lambda}{c\mu}$)
τ round trip link propagation delay (seconds); all sessions have the same τ
p the TCP packet size
b number of data packets that one ack packet acknowledges
B the link buffer size in packets
$R_0(n)$ the value to which the total rate of n TCP controlled sessions sharing a link drops after a buffer overflow; owing to the discretisation of window in terms of packet size, the form of $R_0(n)$ is $R_0(n) = \lceil \frac{0.5c\tau}{n} \rceil \frac{n}{\tau}$; this is because when the number of sessions exceeds the BDP and they suffer loss, they all drop the window to half their share of BDP with the constraint of having a window of at least one packet
a the rate at which an *individual* session increases its data transfer rate in the congestion avoidance phase *(in packets per second per second)*. In the congestion avoidance phase each session increases its window size by $1/b$ packets every τ. Thus the slope a for TCP is $\frac{1}{b}$ packets per τ^2, i.e., $a = \dfrac{1}{b\tau^2}$.

Processes in the Model

$N(t)$ the number of sessions at time t; each session $i, 1 \leq i \leq N(t)$, is transmitting its data at a time varying rate as long as it is in the system
$r_i(t)$ the data transfer rate of session $i, 1 \leq i \leq N(t)$, at instant t.
$R(t)$ the total data transfer rate at instant t; i.e., $R(t) = \sum_{i=1}^{N(t)} r_i(t)$
$r(u)$ the average per session rate at instant u, i.e., $r(u) = \frac{R(u)}{N(u)} I_{\{N(u)>0\}}$

With the above definition of $R(t)$, our modelling approach is to view the files being transferred as being "queued" at a virtual server that applies a time varying service rate to the files (see Figure 2). This service rate $R(t)$ is bounded by the link service rate c. The problem is to analyse the random process $\{R(t)\}$ based on the way TCP sessions adapt their transmission rates.

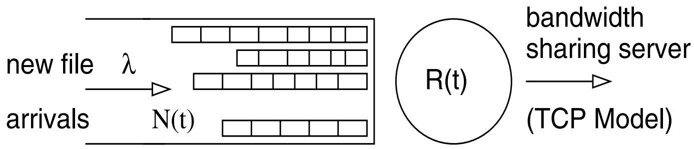

Fig. 2. Schematic of the TCP-PS model. $R(t)$ is obtained from the details of TCP's AIMD behaviour.

Notice that, when $R(t) = c$, with each file getting an equal share, we get the Processor Sharing model.

III. Performance Measures

From the point of view of a user downloading a file, the relevant performance measure is the *file transfer throughput*, i.e., the service volume of the transfer (e.g., number of bytes) divided by its sojourn time in the system. Hence the *average per session throughput* would be a useful performance measure. Let,
$\Psi_k =$ the volume of the k^{th} transfer (in data units),
$W_k =$ total time taken to complete the k^{th} transfer,
Then the average per session throughput θ is given by

$$\theta = \lim_{n\to\infty} \frac{1}{n} \sum_{k=1}^{n} \frac{\Psi_k}{W_k} = E\left(\frac{\Psi}{W}\right). \qquad (1)$$

where the last expression follows from the assumption of existence of the stationary regime, and the associated moments, and with (Ψ, W) denoting the stationary random vector of the transfer volume and the sojourn time of a session.

A throughput measure frequently used in the literature ([7], [6]). is the ratio of average service requirement to the average sojourn time, i.e.,

$$\phi = \frac{E\Psi}{EW}. \qquad (2)$$

Another measure of performance is the *time averaged rate per ongoing session, conditioned on there being at least one ongoing session*, denoted by σ, i.e.,

$$\sigma := \lim_{t\to\infty} \frac{\int_0^t I_{\{N(u)>0\}} \frac{1}{N(u)} R(u)du}{\int_0^t I_{\{N(u)>0\}} du}$$

where $I_{\{N(u)>0\}}$ is 1 if at least one session is active at u and 0 otherwise. [1]

The most commonly used model for bandwidth sharing achieved by TCP controlled nonpersistent sessions is the processor sharing model, where at any instant t the $N(t)$ active sessions get a service rate of $\frac{c}{N(t)}$ (see, for example, [7], [3], [6], [5], with the observation that for a single class, the discriminatory processor sharing (DPS) model [18] is the same as the processor sharing queue).

Comparison of θ, σ and ϕ for the PS model For an M/G/1 PS queue the *average* time a session with a service requirement of s spends in the system is $\frac{s}{1-\rho}$ ([19]). However, the *distribution* of the sessions' sojourn time in the system becomes very cumbersome [20]. This makes θ analytically intractable. However, it is easily seen that for the PS model, $\phi = 1 - \rho$.

[1] See also other related performance measures that have been used in [1], [16], [2] and [17].

If all the ongoing sessions get an equal share of the total rate $R(u)$ (i.e., $r_i(u) = r(u)$)

$$\sigma = \lim_{t\to\infty} \frac{\int_0^t r(u)du}{\int_0^t I_{\{N(u)>0\}}du} = \frac{\lim_{t\to\infty} \frac{1}{t}\int_0^t r(u)du}{\lim_{t\to\infty} \frac{1}{t}\int_0^t I_{\{N(u)>0\}}du} \qquad (3)$$

where the last relation holds if the limits $\lim_{t\to\infty} \frac{1}{t}\int_0^t r(u)du$ and $\lim_{t\to\infty} \frac{1}{t}\int_0^t I_{\{N(u)>0\}}du$ both exist. Thus our performance measure is equal to the time average rate per ongoing session divided by the fraction of time that there is at least one session in the system. We note here that for a Processor Sharing model, $r(u) = \frac{c}{N(u)}$, and it is easily seen that for the PS model σ is given by

$$\sigma/c = \frac{1-\rho}{\rho} \log \frac{1}{1-\rho} \qquad (4)$$

We will use this formula when discussing the results from our TCP model. *Note that this formula is insensitive to the file size distribution.*

While θ may be the most appropriate measure from the user's point of view, as observed, it is analytically intractable even for the PS model. On the other hand while σ and ϕ have simple formulas for the PS model, in general they are different from θ. The following are some definite results on this point.

Theorem III.1: [15] For deterministic file sizes D, i.e., $\Psi_k = D, \forall k \geq 1$, w.p.1, $\sigma > \theta$.

Theorem III.2: [15] For an M/G/1 PS queue with a unit capacity server, ϕ is a lower bound to θ, i.e.,

$$\theta = E\frac{\Psi}{W} \geq \frac{E\Psi}{EW} = \phi = 1 - \rho$$

However, from the fluid simulations of the PS queue (reported in [15]), we have seen that θ is very close to σ, and also is not very sensitive to the file size distribution. It appears from these simulations that irrespective of the file size distributions, θ is between σ and $1-\rho$ and that σ is a good approximation for θ.

Comparison of θ, σ and ϕ for an M/G/∞ queue In Section VIII we propose the use of a rate limited processor sharing (RL-PS) model which converges to the M/G/∞ model as the BDP increases. A recent paper [7] has used the M/G/∞ queue as a model for nonpersistent transfers over a high capacity link with large propagation delay.

It is easily seen that for an M/G/∞ queue, where sessions are served independently at a constant rate, $\sigma = \theta = \phi$ for all values of ρ. Note that above results are insensitive to the sojourn time distribution (same as the file size distribution) for an M/G/∞ model.

It is also easily seen that as $\rho \to 0$, $\sigma \to \phi$ for any system. Thus, the parameters affecting σ are also expected to affect ϕ; in particular, *if σ is sensitive to file size distribution or τ, the same will apply to ϕ.*

Measure used in this paper Based on the observation made above of closeness of σ and θ for the PS model and the M/G/∞ model, in the rest of paper we use σ as the performance measure.

IV. Overview of the Models

As mentioned before, the processor sharing model is frequently used as a model for TCP controlled bandwidth sharing. In Section VI we observe that the applicability of the PS model is restricted to very small BDP links. Based on our investigations we have identified three regions:

- We find that for very small BDP (< 1 packet) the processor sharing model works well.
- For intermediate BDP we have developed a detailed model whose limit is the PS model as the link BDP approaches zero. The PS model assumes that at all times the sessions are served at the full link rate. In practice however owing to tail drop and window reduction by TCP, the actual rate applied to the TCP sessions is less than the link rate and is a random process. We model this rate reduction behaviour but since we assume that the applied rate is shared equally among sessions we call this the TCP-PS model.
- For large BDP most files complete transmission in "slow" start. Interactions between the file transfers and hence a PS-like behaviour sets in only when the number of ongoing sessions exceeds a threshold. For this situation we propose the Rate Limited PS (RL-PS) model in Section VIII.

V. Fixed Parameters Used in Simulations

In all the simulation results that we present in this paper, the packet size of TCP is fixed at $1500Bytes$, the mean file transfer size is fixed at $30KBytes = 20Packets$, and the TCP acknowledgement size is $40Bytes$. Also, unless specifically mentioned, we use the link of capacity $c = 10Mbps$, and a link buffer of $B = 125packets$. We vary the load on the link (ρ) by varying the file transfer request arrival rate (λ), and plot the value of σ (normalised to the link capacity) thus obtained against ρ.

VI. The Processor Sharing Model

The commonly used model for TCP controlled bandwidth sharing is the discriminatory processor sharing model which reduces to processor sharing for a single class (single round trip time) of sessions. As discussed in Section III, for the PS model, σ is given by Equation 4.

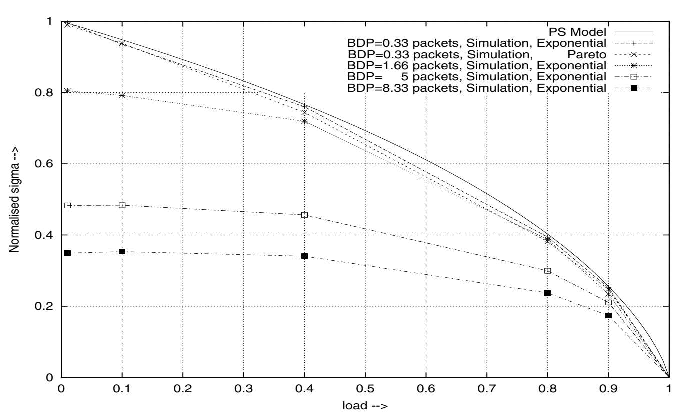

Fig. 3. $\sigma\ vs.\ \rho$ (σ normalised to c) obtained from the PS model and from *ns* simulations for a $10Mbps$ link with various propagation delays. "PS Model" means results from Equation 4. The file size distribution and BDP used are given in the legend.

Figure 3 shows values of σ for different link loads obtained from simulation using *ns* for different propagation delays corresponding to BDPs of $0.33, 1.66, 5$ and 8.33 packets on a $10Mbps$ link; the file size distribution were exponential. Also shown is the simulation result for a BDP of 0.33 packets for MEA_3 approximation of Pareto distributed file sizes (see Section VII-B.1). The figure also shows the values of σ obtained from Equation 4. For BDP of 0.33 packets it is seen that the PS model gives a good estimate of σ for both exponential and Pareto distributed file sizes.

It is easily observed from the figure that the PS model works well only for very small link propagation delays (in the figure the curve corresponding to 0.33 packets BDP) and thus is not applicable to wide area networks. Observe that for larger propagation delays, the PS model overestimates throughput for all values of ρ. Thus there is a need to develop a model that captures the effect of propagation delay on the TCP throughput. This is done in Section VII by developing a model we call TCP-PS.

VII. Small BDP: The TCP-PS Model

The evolution of the total rate of TCP controlled sessions through a single link is controlled by various factors such as the round trip propagation delay, link bandwidth, TCP packet size, and link buffer size. A recent paper ([21]) reports the analysis of the rate evolution of *individual* TCP controlled *persistent* sessions in the congestion avoidance phase with different round trip delays sharing a common bottleneck link. We consider a similar model but in a situation in which sessions arrive randomly, transfer a finite amount of data and then depart.

Figure 4 shows the evolution of the total rate $R(t)$ of TCP controlled nonpersistent sessions under the following simplifying assumptions similar to those in [1]:

1. A session arriving to an empty system undergoes slow start with the slow start threshold set to half the bandwidth delay product (practically possible with the modifications as suggested in [22]).
2. All sessions have the same fixed propagation delay. This assumption is required for the analysis of our detailed model. It is practically appropriate when the propagation delays are dominated by one propagation delay.
3. The total rate at any instant is shared equally by all the active sessions,
4. The effect of slow start is negligible when more than one sessions are active. This is because the BDP is small and the slow start threshold of each session when n are active is $\frac{c\tau}{2n}$ which is much smaller than the BDP.
5. Buffer losses are synchronised; this is a common assumption for tail drop buffers ([1]).
6. The effect of a new session arrival is the same as that of a link buffer overflow; this assumption is equivalent to saying that an arrival immediately causes a buffer overflow. This is because when the BDP is small, most of time there are enough packets in the propagation delay pipe hence the exponential nature of the slow start of a new session arrival results in a quick tail drop.
7. Owing to small BDP, the window size is not limited by the receiver's advertised window.

Figure 4 depicts the way the total rate $R(t)$ of TCP controlled sessions evolve. The figure shows a session arriving to an empty system at time Z_1. The session undergoes slow start with a slow start threshold of half the BDP. Then the session enters the congestion avoidance phase at Z_2. It increases its data sending rate linearly with a fixed slope a packets per second per second un-

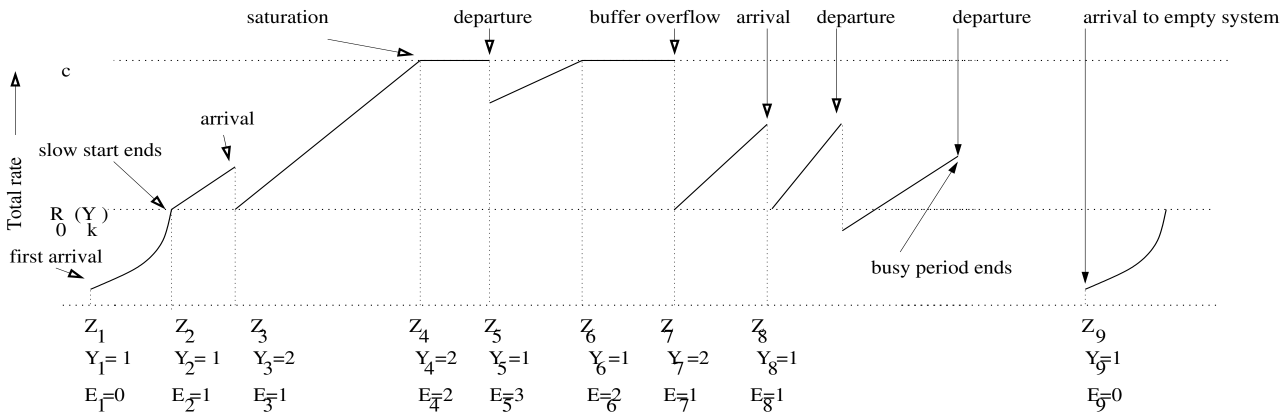

Fig. 4. Figure showing the total rate ($R(t)$) evolution in the TCP-PS model. For simplicity, $R_0(n)$ is shown to be constant.

til an arrival occurs at Z_3. The total rate is then set to the rate corresponding to the total rate just after a buffer overflow with two persistent sessions (the form of this reset rate is given in Section II; for simplicity, the figure shows a constant value of $R_0(n)$). The two sessions now enter the congestion avoidance phase and increase their individual data sending rate linearly, each with slope a. This results in the total rate increasing with the slope of $2a$. This continues until the total rate reaches the link capacity at instant Z_4 after which link buffer starts building up. A session departs at Z_5.

Note here that actually the rates of the active sessions would have increased over the interval (Z_4, Z_5), but for ease of analysis we assume that the rates stay constant; this assumption is expected to give an underestimate of the throughput which would be small owing to small BDP.

The departure at Z_5 does not affect the rate of the other active session which (by the above assumption) still has the rate that it had at Z_4. Now since the total rate after departure is below the link capacity, the remaining active session again enters the linear congestion avoidance phase.

The active session increases its rate linearly until the link saturates at Z_6 after which the total rate remains fixed at the link capacity until a buffer overflow occurs at Z_7. The duration of the time to buffer overflow is modeled by assuming that each active session adds another packet to the buffer after one round trip time. The buffer limit B thus determines the time to buffer overflow. The total rate is multiplicatively decreased, and the congestion avoidance phase is entered.

An arrival at Z_8 causes the total rate to reset, and both sessions are in the congestion avoidance phase. A departure in this phase reduces the total rate by an amount equal to the rate of departing session but does not affect the rate of the other active session which remains in congestion avoidance. If an arrival does not come before this remaining session completes its transfer then the busy period ends and a new "cycle" starts at Z_9 where the next arrival takes place. (A formal definition of cycle is given in Section VII-A)

Recalling that $N(t)$ is the number of sessions at time t, define $Y_k := N(Z_k)$. We identify four types of cycles that the process $R(t)$ evolves through. The cycles are denoted by a process $E(t)$, whose value at Z_k (i.e., E_k) are shown in Figure 4. The possible values of E_k and the total rate at the start of each cycle is given in Table I.

E_k	Type of Cycle	$R(Z_k)$
0	Slow Start	$\frac{p}{\tau}$
1	Congestion Avoidance	$R_0(Y_k)$
2	Link Saturation	c
3	Congestion Avoidance entered after departure in Link Saturation Cycle	$c\frac{Y_k}{Y_k+1}$

TABLE I

THE FOUR CYCLES, DENOTED BY E_k, AND THE INITIAL VALUE OF TOTAL RATE $R(Z_k)$ FOR EACH CYCLE.

Recall that p is packet size. Note that $E_k = 3$ is possible only if a departure occurred at Z_k when $E_{k-1} = 2$ and $Y_{k-1} > 1$ thus $Y_k = Y_{k-1} - 1$. The assumption of constant individual rates ($= \frac{c}{Y_{k-1}}$) during the buffer fill up ($E_{k-1} = 2$) cycle implies that value of $R(Z_k)$ for $E_k = 3$ is $c\frac{Y_{k-1}-1}{Y_{k-1}} = c\frac{Y_k}{Y_k+1}$.

Note that, given $E_k = 2$, the length of the cycle in case of no arrival or departure is bounded by the time at which buffer overflow occurs.

We call the above model the TCP-PS model as it assumes processor sharing with the service rate varying in a manner governed by TCP's adaptive window mechanism.

It should be noted that the $E_k = 1$ cycle of the above model represents a class of additive increase multiplicative decrease congestion-feedback based rate control algorithms. Various algorithms of this class may differ with respect to the additive increase slope a, or the reset rate $R_0(n)$. In this paper in order to maintain the generality of the model we will continue to use the notation $R_0(n)$ for the reset rate, and a for the additive increase slope instead of using values specific to TCP implementations.

A. Analysis of the TCP-PS Model

We first develop the analysis for exponential file sizes and then extend the analysis to hyper-exponential file sizes which can be used to approximate heavy tailed file size distributions [23].

We say that an event has occurred when one of the following takes place:

1. An arrival occurs
2. Slow start ends in congestion avoidance
3. A buffer overflow occurs
4. A departure occurs at time t with $R(t) = c$

Thus $Z_k,\ k \geq 1$, denotes the epoch of occurence of the k^{th} event. Define a cycle to be the process in the interval between two consecutive events, and define $X_k := Z_{k+1} - Z_k$. Thus X_k is the length of k^{th} cycle. At Z_k all the ongoing connections reset their transmission rates depending on the value of E_k as shown in Table I. Subsequently each connection starts increasing its rate with a slope a if $E_k \in \{1, 3\}$, or increases exponentially if $E_k = 0$, or remains constant if $E_k = 2$. Also, because file sizes are exponentially distributed, the residual file sizes at Z_k are again exponentially distributed, independent of anything in the past. Hence, X_{k+1}, E_{k+1} and Y_{k+1} depend only on $\{E_k, Y_k\}$. Thus $\{E_k, Y_k\}$ embedded at $\{Z_k\}$ is a Markov chain on the state space $\mathcal{S}$ defined below. Given $E_k = 0$ the possible value of Y_k under the above assumptions is 1. For $E_k = 1, 2, 3$, $Y_k \in \{1, 2, ...\}$ so,

$$\mathcal{S} = \{(0,1), (1,1), (1,2), .., (2,1), (2,2), .., (3,1), (3,2), ..\}.$$

It follows that the process $\{((E_k, Y_k), Z_k), k \geq 1\}$ is a Markov renewal process, with $\{(E_k, Y_k), k \geq 1\}$ being the Markov chain.

Theorem VII.1: The process $\{E_k, Y_k\}$ is positive recurrent for $\rho < 1$.

Proof: Figure 5 shows the transition probability diagram of the $\{E_k, Y_k\}$ process; the expressions for the transition probabilities have not been shown owing to lack of space. Using a Lyapunov function $f(E_k, Y_k) = Y_k^3$ and by bounding the drifts it is proved that the $\{E_k, Y_k\}$ is positive recurrent for $\rho < 1$. We leave out the details of the proof. □

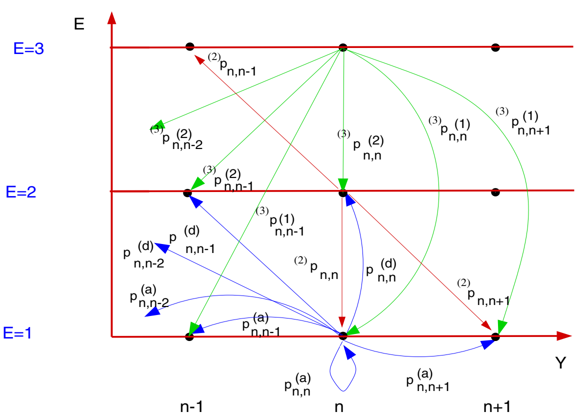

Fig. 5. The transition probability diagram of the $\{E_k, Y_k\}$ process for $Y_k = n > 2$.

For $\rho < 1$, let $\pi(e, n), n \geq 1$, denote the stationary probability distribution of $\{E_k, Y_k\}$.

Notation: in each cycle of the process $R(t)$ that starts with $E_k = e$ and $Y_k = n$, we denote the (conditional) probability law by $P_{e,n}(\cdot)$ and the (conditional) expectation by $E_{e,n}(\cdot)$.

Denote the generic marginal random variables for the processes $\{E_k\}$, $\{Y_k\}$, $\{Z_k\}$, and $\{X_k\}$ by E, Y, Z and X respectively. Then the mean cycle time is given by:

$$EX = \sum_{e \in \{0,1,2,3\}} \sum_{n \geq 1} \pi(e, n) E_{e,n} X$$

We have that $EX < \infty$ since the cycle lengths are bounded by an interarrival time. We can now write the numerator of σ (see Equation 3) as

$$\lim_{t \to \infty} \frac{1}{t} \int_0^t r(u) du = \frac{E_\pi \int_0^X r(u) du}{EX}$$

Key Observation: Since we assume that all sessions at time u receive the rate $r(u)$, notice that $E_\pi \int_0^X r(u) du$ is the expectation of the total amount of data transferred in a cycle by the *longest lasting session in that cycle*; this longest lasting session may be active throughout the cycle, or, if the cycle ends in an idle period, the longest lasting session would be the last to leave in the busy period that ends in the cycle (see the cycle that starts at Z_8 and ends at Z_9 in Figure 4).

Notation: let $V_k = \int_{Z_k}^{Z_{k+1}} r(u) du$, i.e., the volume of data transferred in the kth cycle by the longest lasting session.

Then we have

$$\lim_{t \to \infty} \frac{1}{t} \int_0^t r(u) du = \frac{EV}{EX} = \frac{\sum_{0 \leq e \leq 3} \sum_{n \geq 1} \pi(e, n) E_{e,n} V}{\sum_{0 \leq e \leq 3} \sum_{n \geq 1} \pi(e, n) E_{e,n} X}$$

The denominator of σ (Equation 3) can be written as

$$\lim_{t \to \infty} \frac{1}{t} \int_0^t I_{\{N(u) \geq 1\}} du = \frac{E_\pi \int_0^X I_{\{N(u) \geq 1\}} du}{EX}$$

Notation: in the kth cycle, and for each $m \geq 1$, let $U_k^{(m)} = \int_{Z_k}^{Z_{k+1}} I_{\{N(u) \geq m\}} du$.

It follows that

$$\lim_{t \to \infty} \frac{1}{t} \int_0^t I_{\{N(u) \geq m\}} du = \frac{EU^{(m)}}{EX}$$
$$= \frac{\sum_{0 \leq e \leq 3} \sum_{n \geq 1} \pi(e, n) E_{e,n} U^{(m)}}{\sum_{0 \leq e \leq 3} \sum_{n \geq 1} \pi(e, n) E_{e,n} X}$$

Finally we can write

$$\sigma = \frac{EV/EX}{EU^{(1)}/EX} = \frac{EV}{EU^{(1)}} = \frac{\sum_{e,n} \pi(e, n) E_{e,n} V}{\sum_{e,n} \pi(e, n) E_{e,n} U^{(1)}}. \quad (5)$$

Hence to obtain σ we need $E_{e,n}V$, $E_{e,n}U^{(1)}$ and $\pi(e, n)$. In Appendix A we develop the analysis that yields these quantities for $E_{1,n}$; the other quantities can be found in a similar manner.

Hyper-Exponential File Sizes: It is to be noted that the above analysis assumes that the file sizes are exponentially distributed with the same mean. The analysis presented is easily extended to multiple classes of arrivals with the same propagation delay, different arrival rates, and Exponential file sizes with different means by using a vector valued $\{Y_k\}$ process, and some minor modifications which we do not report here. Thus in order to find the throughput for a heavy tailed distribution, we can approximate the desired distribution by a hyper-exponential distribution (as in [23]) and use the extension of our analysis to determine the throughput for the mixture of exponentials. Thus the assumption of exponential file size distribution in the analysis is not a restriction.

We know that $\lim_{t\to\infty} \frac{1}{t}\int_0^t I_{\{N(u)\geq m\}}du = \frac{EU^{(m)}}{EX}$ which is the probability of having at least m sessions active in the system. From this the mean number of active sessions EN can be obtained. *Hence Little's Theorem can be used to find the mean Sojourn Time of each session in the system; thus we can find the other performance measure* ϕ. This is a performance measure frequently used in the related literature (the mean sojourn time has also been used as a performance measure in [14]).

B. Numerical Results: TCP-PS Model

In this section we present the numerical results as obtained from the above analysis and *ns* simulations for a $10Mbps$ link. Figure 6 plots the values of σ from *ns* simulations and the TCP-PS model for the different values of τ used in Figure 3; these values of τ correspond to BDP of $1.66, 5, 8.33$ and 25 packets. The file size distribution in the simulation is exponential. Compare this figure with Figure 3 where we observed that the

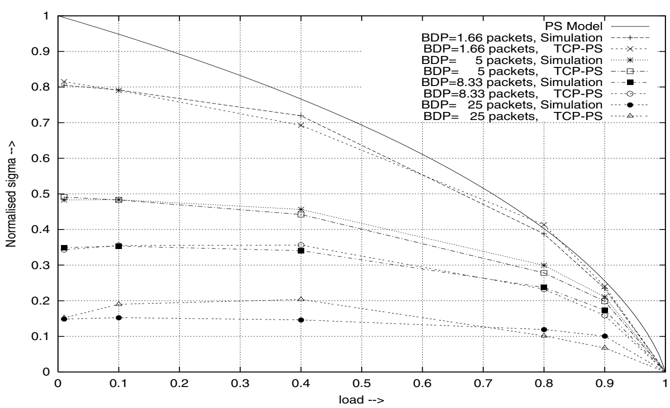

Fig. 6. $\sigma\ vs.\ \rho$ (σ normalised to c) obtained from *ns* simulation and the TCP-PS model, for different propagation delays corresponding to BDP of $1.66, 5, 8.33$, and 25 packets. File size distribution was exponential

PS model is very inaccurate when BDP exceeds 1 packet. The TCP-PS model is seen to give a good estimate of σ for BDP $< 20packets$ which is the mean file size. It is seen that the TCP-PS model is not accurate for BDP close to or exceeding the mean file size as is evident from comparison of simulation and model curve for BDP = 25packets. The TCP-PS model overestimates in this case.

To address the problem for large BDPs, we use the Rate Limited Processor Sharing (RL-PS) model proposed in Section VIII.

B.1 General File Size Distributions

Figure 7 shows the values of σ obtained from simulations and the TCP-PS model using a mixture of three exponentials (denoted MEA_i as an abbreviation for Mixture of Exponential Approximation with i exponentials) to approximate the Pareto distribution ([23]) with its parameter $\alpha = 1.6$ for a BDP of $5packets$, and link buffer of 25 and 75 packets; the mean file sizes of the three exponentials used for the approximation are $3KBytes, 30KBytes$ and $300KBytes$. The figure also shows the MEA_1 curve which is obtained from *ns* simulations for exponentially distributed file size with a mean of $30KBytes$ and link buffer of $25Packets$. Observe from simulation results

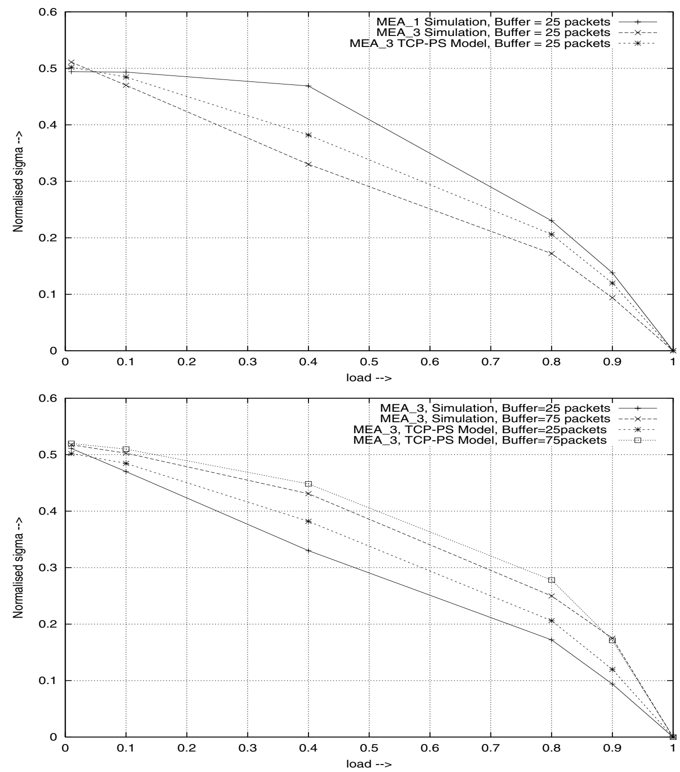

Fig. 7. $\sigma\ vs.\ \rho$ (σ normalised to c) obtained from simulation and TCP-PS model using a mixture of three exponentials to approximate Pareto distribution with $\alpha = 1.6$ for a BDP of $5packets$ and link buffer of 25 and 75 packets.

in the figure that for the same link buffer of $25Packets$, the throughput is less for MEA_3. It is also seen from the figure that the TCP-PS model captures the effect of the link buffer well. *Observe that* σ *is not insensitive to the file size distribution; for a fixed value of link buffer,* σ *is less for a heavy tailed file size.*

The TCP-PS model can also yield an estimate of the buffer overflow probability. This is done by having a reward of n packets when there are n sessions active in the $E_k = 2$ cycle and a transition occurs to the $E_k = 1$ cycle due to an overflow. The Markov renewal reward theorem yields the drop probability.

ρ	MEA_1 ($B = 25$)		MEA_3 ($B = 25$)		MEA_3 ($B = 75$)	
	Sim	$tcp{-}ps$	Sim	$tcp{-}ps$	Sim	$tcp{-}ps$
0.01	0.000	0.000	0.000	0.000	0.000	0.000
0.1	0.000	0.000	0.000	0.000	0.000	0.000
0.4	0.000	0.000	0.001	0.000	0.000	0.000
0.8	0.002	0.003	0.008	0.007	0.001	0.002
0.9	0.007	0.006	0.016	0.011	0.003	0.004

TABLE II

PACKET DROP PROBABILITY OBTAINED FROM *ns* SIMULATIONS AND THE TCP-PS MODEL FOR A BDP OF $5packets$ AND DIFFERENT VALUES OF LINK BUFFER (B) FOR MEA_3 AND MEA_1.

Table II gives the drop probabilities obtained from simulation and TCP-PS model for a BDP of $5packets$ and link buffer of 25 and $75packets$ for MEA_1 and MEA_3. It is seen that the

TCP-PS model gives a good estimate of the drop probability. Observe that the drop probability for MEA_3 is more than that for MEA_1 for $B = 25 packets$. This is because of the heavy tailed nature of MEA_3. Also observe from the table that the buffer overflows, and hence timeouts, are rare for small BDPs. Thus neglecting the timeouts in the TCP-PS model is a valid assumption.

C. Convergence to the PS Model

It can be proved that as $c\tau \to 0$, the TCP-PS model converges to the PS model with a server rate c. We leave out the proof of this statement due to lack of space. The intuition behind this result is the following. If $\tau \to 0$, the slope of linear increase of rate, a approaches ∞. Thus the cycles $E_k = 1$ and $E_k = 3$ are left as soon as they are entered so that most of arrivals and departures occur in the $E_k = 2$ cycle where the model behaves like the PS model (see Figure 4).

D. Parameters that Govern Performance

The TCP-PS model has several parameters: the session arrival rate: λ; the mean of the exponentially distributed file sizes: $\frac{1}{\mu}$ (packets); the fixed RTT: τ (seconds); the link speed: c (packets sec^{-1}); the value to which the total rate is reset: $R_0(n)$ (packets sec^{-1}); the increase rate of per session data sending rate: a (in packets sec^{-2}) $a = 1/\tau^2$; the link buffer B in packets.

By normalising time to τ, it is easily seen that the analysis depends on four parameters: $c\tau$, which is the fixed *RTT pipe* size in packets, $(\mu c\tau)^{-1}$, which is the mean file size normalised to the RTT pipe size, $\rho = \lambda/(c\mu)$, the normalised offered load on the link and the buffer B in packets. In Figure 8 we give values of σ from simulation obtained by varying the link capacity (c) and round trip propagation delay (τ) keeping the BDP fixed at $5 packets$ and 8.33 packets. It is seen that the parameters governing the behaviour of TCP controlled sessions are the above identified parameters of TCP-PS model.

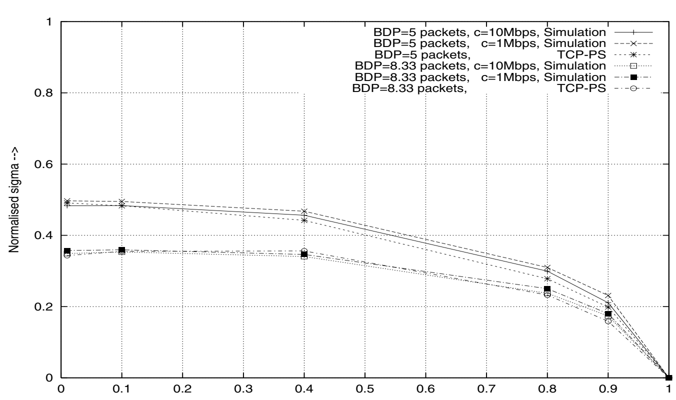

Fig. 8. σ $vs.$ ρ (σ normalised to c) obtained from simulation and TCP-PS model for different BDP values. Two simulations for same BDP were done by varying the link capacity and propagation delay. The link capacity and BDP used in simulations are given in the corresponding legend.

E. Advantage of Analysis Despite the Numerical Computations Involved

The analysis of the TCP-PS model is complex and a valid question is whether a simulation would not suffice. We point out here that analysis has provided following important insights:
- We now know which parameters affect the performance; see Section VII-D.
- The PS model has often been used in this context; our analysis shows that the system in a sense converge to PS-like performance for BDP $\to 0$.

Further, the following general advantages of analytical modeling are well known:
- The exercise of building a model and developing it's analysis provides useful insights into the protocol.
- An analytical model provides a useful validation for a simulation or an experiment.

VIII. LARGE BDP: THE RL-PS MODEL

We have seen that as BDP increases, the PS model no longer applies and we proposed the TCP-PS model to address this issue. We saw that even the TCP-PS model does not work well for large BDP (BDP close to or more than the mean file size). This is because with nonpersistent sessions and large BDP, most session complete their transfer in slow start. Also *most* of the window of a connection is in the RTT pipe rather than in the link buffer. Thus unless the number of sessions becomes large there is little interaction between sessions; this is because the round trip time seen by them is close to the constant part of RTT i.e., τ, the queueing delay at link buffer being small (as compared to τ). Thus it appears that for very high speed wide area links the sessions are served independently motivating the M/G/∞ model (also proposed in [7]). We observe that this reasoning does not apply for *all* loads because if the number of active connections is large enough to fill the link pipe, the link buffer starts filling up resulting in considerable interaction among the sessions. Similar observations have been made in [24]. We propose the following model that converges to the M/G/∞ model for very large BDP.

A. The Rate Limited Processor Sharing (RL-PS) Model

We find the mean sojourn time of a single session sharing the link; this is done using the TCP-PS model for very small load where $\sigma \approx \phi$. Let m_0 denote the mean departure rate (inverse of the mean sojourn time) thus obtained. Let $K := \frac{c\mu}{m_0}$. We model the process $N(t)$ by a birth-death process where the death rate is state dependent and is given by:

$$\mu(n) = \begin{cases} nm_0 & n \leq K \\ c\mu & n > K \end{cases}$$

and the birth rate is the rate of the (Poisson) session arrival process (λ). Observe that as $BDP \to \infty$, $m_0 \to 0$ thus $K \to \infty$, and thus an M/G/∞ model applies in the limit.

B. Finding σ and ϕ for the RL-PS Model

The birth-death process of the RL-PS model is positive recurrent for normalised link load $\rho < 1$. Under the recurrence condition for the above birth death process, it is easy to find the stationary distribution of number of sessions active and thus σ and ϕ. Let $\pi(n)$ denote the stationary probability of having n sessions active. Then, with $\hat{\rho} := \frac{\lambda}{m_0}$,

$$\begin{aligned}
\pi(0) &= \left(1+\sum_{n=1}^{K-1} \frac{\hat{\rho}^n}{n!}+\sum_{n=K}^{\infty} \frac{\hat{\rho}^{(K-1)}}{(K-1)!}\rho^{(n-(K-1))}\right)^{-1} \\
\pi(n) &= \begin{cases} \frac{\hat{\rho}^n}{n!}\pi(0), & n \leq K-1 \\ \frac{\hat{\rho}^{(K-1)}}{(K-1)!}\rho^{(n-(K-1))}\pi(0), & n \geq K \end{cases} \\
\sigma &= \left(\sum_{n=1}^{K-1} \frac{m_0}{\mu}\pi(n)+\sum_{n=K}^{\infty} \frac{c}{n}\pi(n)\right)/(1-\pi(0)) \\
\phi &= \lambda\left(\mu \sum_{n=1}^{\infty} n\pi(n)\right)^{-1} \text{ (By Little's theorem).}
\end{aligned}$$

It can be seen that the parameters for the RL-PS model are the file size distribution, BDP and the normalised link load ρ.

C. Effect of File Size Distribution

m_0 defined above depends on the file size distribution and is not completely determined by the mean file size. An example is that, assuming sessions go away in initial slow start, for a deterministic file size distribution with mean $\hat{\mu}$, ${m_0}^{-1} = \tau * log_2(\hat{\mu}+1)$ whereas for a file size distribution having equal probability mass of 0.5 each at $0.5\hat{\mu}$ and $1.5\hat{\mu}$, the mean file size is $\hat{\mu}$ but ${m_0}^{-1} = \tau * 0.5 * (log_2(1.5\hat{\mu}+1) + log_2(0.5\hat{\mu}+1))$. Note that we have ignored the queueing delay in this formula for m_0. The value of m_0 for above two examples could be very different indicating that σ *is not insensitive to the* file *size distribution*.

D. Numerical Results: The RL-PS Model

Figure 9 plots the values of σ obtained from simulation and the RL-PS model for large BDPs (greater than mean file size which is fixed at $30KBytes$) for two different file size distributions i.e., exponential and MEA_3 approximation of Pareto distribution. It is seen that the match between analysis and simulation is quite good. The flat portion of the curves are indicative of the fact that the sessions are not interacting much in this region and hence the M/G/∞ aspect of the RL-PS model dominates. The nature of the curve for large loads imply that there is significant interaction between sessions in this region and an M/G/∞ model (which predicts load independent constant throughput) without any modification will not apply. It is also again seen from the figure that σ is *not insensitive* to the file size distribution and the RL-PS model is able to capture this sensitivity very well.

IX. Conclusion

We have presented our results on the analytical modelling of TCP controlled bandwidth sharing between randomly arriving finite volume file transfer sessions. The network scenario was: single bottleneck link, finite link buffer, tail drop queue management, and nonzero propagation delay which is the same for all the sessions.

From our results we conclude the following:

1. When the BDP is smaller than the mean file size then the model needs to capture the AIMD behaviour of TCP's transmission rate control. For this we have introduced the TCP-PS (TCP processor sharing) model. For a $30KBytes$ mean file size (20 packets, typical of the Internet) and with light or heavy tails, the TCP-PS model is accurate for BDP < 20 packets (Figure 6). We

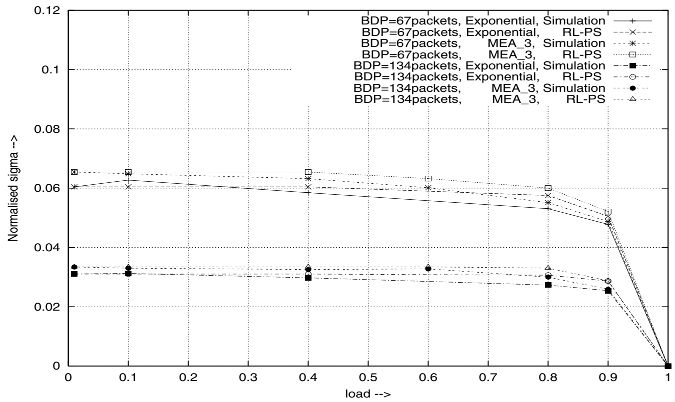

Fig. 9. σ $vs.$ ρ (σ normalised to c) obtained from *ns* simulations and the RL-PS model for BDP of 67 and 134 packets; file size distributions were Exponential and MEA_3 as shown in the legend.

find that for this range of parameters the performance is sensitive to the file size distribution (Figure 6, 7). There are more tail drop loss for a heavy tailed distribution. Our model accurately captures the losses as well (Table II).

2. The standard PS model is the limit of the TCP-PS model as BDP $\rightarrow 0$, and is accurate only for BDP < 1 packet (Figure 3). For large BDPs the actual rate applied to the file transfers is frequently less than the link rate, hence the PS model overestimates the bandwidth share provided to sessions.

3. For BDP much larger than the mean file size, files frequently complete their transfer in slow start. A much simpler RL-PS (rate limited processor sharing) model suffices. We find that for BDP > 25 packets (for the typical file size distribution) the RL-PS model is very accurate (Figure 9). The RL-PS model converges to M/G/∞ for BDP $\rightarrow \infty$.

For the scenario we are concerned with we still do not have an accurate model for BDP close to the mean file size as in this region neither the AIMD behaviour nor the slow start behaviour is predominant. A composite model would be required. Web documents typically comprise several objects and depending on the version of HTTP being used, the file transfer arrival process would be more complex than Poisson. Analytical models that capture these details are also necessary.

In ongoing work we are investigating analytical models for internets with multiple bottleneck links, and path dependent propagation delays as well as the use of such models in traffic engineering and network dimensioning.

Appendix

I. Analysis for Obtaining $E_{1,n}X$

A. An Important Observation

Consider a cycle that begins with n sessions (i.e., $Y = n$) and with $E = 1$; each session begins with a rate $\frac{R_0(n)}{n}$, and increases its rate linearly at the rate a. The departure of a session in a cycle does not disturb the rates of the other ongoing sessions. Notice that, with $Y = n$ and $E = 1$, a transition to $E = 2$ can occur only at those epochs u at which the individual rates equal the ratio of link speed c and $N(u)$; there can be atmost n such epochs. See Figure 10. Note also that one possibility is

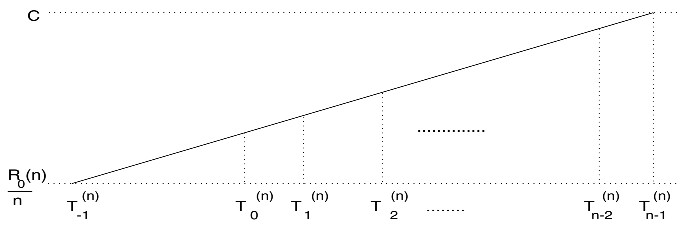

Fig. 10. Evolution of an *individual* connection's rate during a congestion avoidance cycle, starting with n sessions. Figure shows the possible epochs ($T_i^{(n)}, -1 \leq i \leq n-1$) of transition to $E = 2$ cycle.

that no overflow occurs and no arrival occurs, and each session completes its transfer before the total rate reaches c. In this case the cycle includes an idle period and the next cycle starts with $E = 0$ and $Y = 1$.
For simplicity, we suppress the fact that $E_k = 1$ from the notations in what follows.
Notation: we denote these possible transition (to $E = 2$ cycle) epochs by $T_i^{(n)}, 0 \leq i \leq n-1$ where the subscript specifies that i sessions have departed, and the superscript specifies that there were n sessions at the start of cycle. We also define $T_{(-1)}^{(n)} = 0$ and $T_n^{(n)} = \infty$. See Figure 10.

It is easy to see that $T_i^{(n)} = \frac{c}{a}(\frac{1}{n-i} - \frac{R_0(n)}{cn})$.

B. Analysis for Obtaining $E_n X$

Let the number of sessions at the beginning of a cycle be $n \geq 1$. We seek the complementary distribution $P_n(X > x)$, from which $E_n X$ can be obtained. For $0 \leq i \leq n$, and $x \in (T_{i-1}^{(n)}, T_i^{(n)}]$, we see that $X > x$ if and only if:
1. there is no arrival in $(0, x]$,
2. *and* there are at least $j+1$ departures in $(0, T_j^{(n)}]$ for each $j = 0, ..., i-1$.
Notation: Let $Q_1, Q_2, \ldots, Q_n$ denote i.i.d. exponential(μ) distributed random variables, representing the residual file sizes of the n sessions at the beginning of the cycle. Let $Q^{(n,1)}, Q^{(n,2)}, \ldots, Q^{(n,n)}$ denote the (ascending) ordered residual file sizes. Let, for $-1 \leq i \leq n-1$, $S_i^{(n)}$ denote the amount of data that an *individual* session can transfer in $(0, T_i^{(n)})$, given that $X > T_i^{(n)}$. Clearly, $S_i^{(n)} = \frac{R_0(n)}{n} T_i^{(n)} + \frac{a}{2}(T_i^{(n)})^2$.
Define, for $1 \leq i \leq n-1$

$$I_i^{(n)} = P_n\{Q^{(n,j)} \in (0, S_{j-1}^{(n)}), 1 \leq j \leq i+1\}$$

and also $I_0^{(n)} = 1$. It is then clear from the observations in Section A-A that, for $0 \leq i \leq n$, and $x \in (T_{i-1}^{(n)}, T_i^{(n)}]$,

$$P_n\{X > x\} = e^{-\lambda x} . I_{i-1}^{(n)}$$

Hence

$$\begin{aligned} E_n X &= \int_0^\infty P_n\{X > x\} dx \\ &= \sum_{i=0}^{n-1} \int_{T_{i-1}^{(n)}}^{T_i^{(n)}} P_n\{X > x\} dx + \int_{T_{n-1}^{(n)}}^\infty P_n\{X > x\} dx \\ &= \sum_{i=0}^{n-1} I_{i-1}^{(n)} \int_{T_{i-1}^{(n)}}^{T_i^{(n)}} e^{-\lambda x} dx + I_{n-1}^{(n)} \int_{T_{n-1}^{(n)}}^\infty e^{-\lambda x} dx \\ &= \frac{1}{\lambda} \sum_{i=0}^{n-1} I_{i-1}^{(n)} (e^{-\lambda T_{i-1}^{(n)}} - e^{-\lambda T_i^{(n)}}) + I_{n-1}^{(n)} \frac{e^{-\lambda T_{n-1}^{(n)}}}{\lambda} \end{aligned}$$

The second term in the second equality above is the case in which all sessions depart before reaching the link rate, and the cycle ends with a new busy period. Thus we need only $I_j^{(n)}, 0 \leq j \leq n-1$, to obtain $E_n X$. We obtain these as a by-product of calculation in finding $E_n V$ which we donot give here.

REFERENCES

[1] D.P.Heyman, T.V.Lakshman and A.L.Neidhardt, "New Method for Analyzing Feedback Protocols with Applications to Engineering Web Traffic over the Internet.," *Performance Evaluation Review*, vol. 25, no. 1, 1997.
[2] Arthur W. Berger and Yaakov Kogan, "Dimensioning Bandwidth for Elastic Traffic in High-Speed Data Networks," *IEEE/ACM Transactions on Networking*, vol. 8, no. 5, pp. 643–654, 2000.
[3] J.W. Roberts and L. Massoulie, "Bandwidth Sharing: Objectives and Algorithms," in *INFOCOM'99*, 1999.
[4] G. Fayolle, A. de la Fortelle, J-M. Lasgouttes, L. Massoulie, and J. Roberts, "Best-Effort Networks: Modeling and Performance Analysis via Large Networks Asymptotics," in *IEEE INFOCOM'01*, 2001.
[5] T. Bu and D. Towsley, "Fixed Point Approximations for TCP Behaviour in an AQM Network," in *SIGMETRICS'01*, 2001.
[6] T. Bonald and L. Massoulie, "Impact of Fairness on Internet Performance," in *SIGMETRICS'01*, 2001.
[7] S. Ben Fredj, T. Bonald, A. Proutiere, G. Regnie, and J. Roberts, "Statistical Bandwidth Sharing: A Study of Congestion at Flow Level," in *SIGCOM'01*, 2001.
[8] P. S. Chanda, A. Kumar, and A. A. Kherani, "An Approximate Calculation of Max-Min Fair Throughputs for Non-Persistent Elastic Sessions," in *Internet Performance Symposium (IEEE Globecom'01)*, 2001.
[9] S. Floyd and V. Paxson, "Difficulties in Simulating the Internet," *IEEE/ACM Transactions on Networking*, vol. 9, no. 4, pp. 392–403, 2001.
[10] Mikkel Christiansen et al, "Tuning RED for Web Traffic," in *SIGCOMM'2000*, August 2000.
[11] F. Baccelli and D. Hong, "The AIMD Model for TCP Sessions Sharing a Common Router," in *39th Annual Allerton Conf. on Communication, Control and Computing*, Allerton Park, October 2001.
[12] M. Garetto, R. L.Cigno, M. Meo, M. A. Marsan, "A Detailed and Accurate Closed Queueing Network Model of Many Interacting TCP Flows," in *IEEE INFOCOM'01*, 2001.
[13] Renato Lo Cigno and Mario Gerla, "Modeling Window Based Congestion Control Protocols with Many Flows.," *Performance Evaluation*, vol. 36-37, 1999.
[14] Gustavo De Veciana, Tae-Jin Lee and Takis Konstantopoulos, "Stability and Performance Analysis of Networks Supporting Services with Rate Control-Could the Internet be Unstable," in *Proc. IEEE INFOCOM'1999*, April 1999.
[15] Arzad A. Kherani Anurag Kumar, Pinaki S. Chanda and Vijay H. Sankar, "On Throughput of Nonpersistent Elastic Sessions on an Internet Link with Small Propagation Delay," Manuscript, available at http://ece.iisc.ernet.in/~ anurag/pubm.html, 2001.
[16] J.W. Roberts and L. Massoulie, "Bandwidth Sharing and Admission Control for Elastic Traffic," in *ITC Specialist Seminar*, October 1998.
[17] H. Chen, O. Kella and G. Weiss, "Fluid Approximations for a Processor-Sharing Queue," *Queueing Systems, Theory and Applications*, 1997.
[18] G. Fayolle, I. Mitrani and R. Iasnogorodski, "Sharing a Processor Among Many Jobs," *Journal of ACM*, vol. 27, pp. 519–532, 1980.
[19] Leonard Kleinrock, *Queueing Systems*, vol. 2, John Wiley and Sons, 1996.
[20] J. A. Morrison, "Response Time Distribution for a Processor Sharing System," *SIAM Journal of Applied Mathematics*, vol. 45:1, pp. 152–167, 1985.
[21] Patrick Brown, "Resource Sharing of TCP Connections with Different Round Trip Times.," in *Proc. IEEE INFOCOM'2000*, April 2000.
[22] J. C. Hoe, "Improving the Start-up Behavior of a Congestion Control Scheme for TCP," in *ACM SIGCOM*, 1996.
[23] Anja Feldman and Ward Whitt, "Fitting Mixtures of Exponentials to Long-Tail Distributions to Analyze Network Performance Models," *Performance Evaluation*, vol. 31, 1998.
[24] R. Venkatesh, "Performance Analysis of Aggregates of Non-Persistent TCP flows in Very High Speed Networks," M.S. thesis, ECE, Indian Institute of Science, 2002.