On the Throughput of Nonpersistent Elastic Sessions on an Internet Link with Small Propagation Delay

Arzad A. Kherani, Anurag Kumar, Pinaki Sankar Chandra and Vijay H. Sankar *
Dept. of Electrical Communication Engg.
Indian Institute of Science, Bangalore, 560012, INDIA
Phone: (+91)-80-360 0855, Fax: (+91)-80-360 0991
email: alam, anurag @ece.iisc.ernet.in

Abstract

Recent works on modeling web traffic, engineering an Internet backbone, and providing quality of service in the Internet ([12, 6, 2, 5]) use the realistic model that the number of active file transfers on an Internet link vary randomly with time. In this paper we investigate the use of the M/G/1 Processor Sharing (PS) model to study the performance of TCP controlled nonpersistent sessions sharing a small propagation delay Internet link. For the PS model we define and relate various throughput measures: session average throughput, mean file size divided by the mean transfer time, and the time average bandwidth share per session. For the latter measure there is an explicit formula which is insensitive to the distribution of the file sizes. We use simulations to study how well the time average bandwidth share obtained from the PS model captures the session average throughput performance of TCP controlled nonpersistent sessions with both heavy and light tailed file size distributions. We focus on transfers over a single, low propagation delay link (e.g., campus area or metro area). Our results show that even for the complex case of nonpersistent TCP controlled sessions, the simple formula for the average bandwidth share works remarkably well for predicting the average session throughput. This formula can be used to dimension the link bandwidth given the average offered load.

1 Introduction

Traffic engineering and bandwidth dimensioning in the Internet requires effective models that can predict the performance as seen by network users. At the present time the Internet carries predominantly elastic traffic, and the bandwidth sharing between elastic sessions is

*Address all correspondence to Arzad Alam.

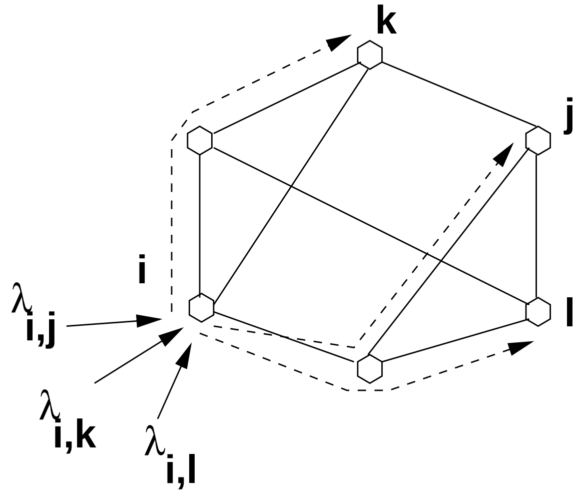

Figure 1: An ISP or enterprise backbone showing the arrival processes of requests for transferring random volumes of data. The traffic for each source destination pair uses links on a fixed route.

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. Most studies of implicit or explicit feedback control of elastic sessions consider only *persistent sources* which have an infinite backlog of data to send ([11, 15, 4, 23, 10, 16, 7]). In real applications, however, the TCP connections over a link are *ephemeral*, i.e., a new TCP connection starts, transmits some finite number of packets and closes the connection. This is the case, for example, when HTTP 1.0 is being used for downloading files, so that a new connection is established for each file transfer request, (unlike HTTP 1.1 where existing connections can be reused for subsequent transfers).

In Figure 1 we show our view of the traffic engineering problem at the backbone of an ISP or an enterprise. Assume that this backbone is the bottleneck for the transfers that it carries. What kind of traffic model should we use as an input to the engineering problem? Since the backbone carries transfers that have finite and random volumes, clearly a model with a fixed number of persistent transfers is not appropriate. For such a model what would be the number of persistent transfers we would use on each route? Some authors (see [6, 3]) have used a traffic model in which a user "logs-in" to the network, and thus starts a session; during this session the user alternates between triggering downloads, and then thinking about what has been downloaded. From measurements at the backbone routers of an ISP, or an enterprise, it will be difficult to determine when such sessions start or end, as the login procedure is only between the client and a local authentication server. Also, the above works assume that a user is transferring (downloading) atmost one file at any instant which will not be true for web traffic generated by a client machine as a web page will comprise several files and the same client machine opens multiple simultaneous connections for download. The models in the above work also does not take into account the possibility of a LAN connected to the backbone link via an access link so that the traffic on the access link is not due to a single machine but is composed of superposition of web traffic generated by several machines

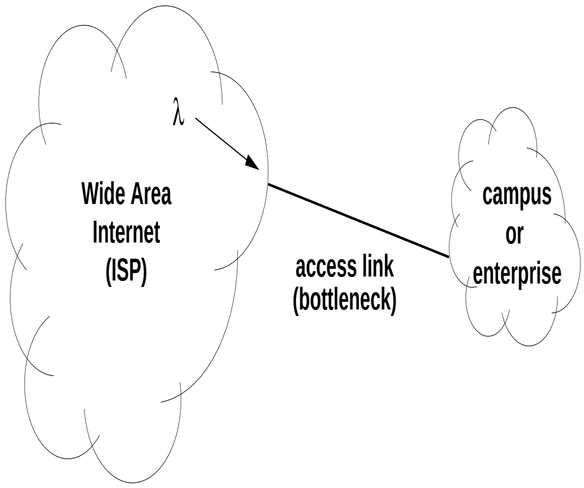

Figure 2: Web or file transfers between an ISP and a campus bottlenecked at an access link.

sharing the same LAN. Thus the backbone will just see a random process of requests for transfers at each of its routers. Each such request will cause the transfer of one or more random volumes of data (files, or web pages). This is depicted in Figure 1. For i, j being edge nodes in the network, $\lambda_{i,j}$ is the rate of arrival of transfers from i to j. All such transfers will take place along the links of a certain route. Such a model was also proposed and studied in [9]. The ISP or enterprise may wish to determine the average transfer throughput on each route, for a given set of load parameters, and may wish to ensure, for example, that the average throughput on each route stays above a certain required level (e.g., 20kiloBytes per second). Such an analysis may be used to groom the routing (adjusting OSPF weights, or adding MPLS LSPs), or even adding new links.

The simplest version of the larger problem described above is depicted in Figure 2, where the access link between the Internet and a campus or enterprise is shown as being the bottleneck. Measurements at the routers at each end of the link can provide statistics of the arrival rates and the transfer volumes. The network administrator may wish to determine the usefulness of leasing another access link to a different ISP, and splitting the traffic across these two links. As another example, consider the problem of determining the capacity of an intracampus link (which is typically of negligible propagation delay) that connects a central computing facility (server) to a department LAN, with the constraint of providing some average minimum access rate.

Motivated by the above, in this work we are interested in the performance of TCP controlled bandwidth sharing in the situation where a single link is shared by TCP controlled sessions that arrive, transfer some data, and depart. In particular, we consider the network scenario shown in Figure 3 where clients are downloading files from the server across the bottleneck link. The propagation delay between the clients and the server is assumed to be small (as is the case if the server and clients are geographically close). By small propagation delay we mean a propagation delay of less than the transmission time of one packet. This would be true for the case where an OC12 link or a Gigabit Ethernet is used to connect the various branches of an enterprise to a central server "farm" located in the same city, and the problem is to find the link capacities to be used so that the download sessions see some minimum average transfer rate. We assume that requests for file transfer on the link arrive in a Poisson process[1]; the request sizes are independent and identically distributed with some distribution. We use both heavy and light tailed file size distributions in this study. We also assume that the propagation delay for every session is the same. Our goal is to study the average bandwidth share obtained by the TCP controlled ephemeral sessions.

Such problems have been under study for the last few years but most of the studies have assumed that there are a fixed number of persistent sessions, i.e., the sessions never depart nor do any new arrivals occur. For this case, the measure of throughput over a time interval is just the ratio of the amount of data transferred by an individual session and the time interval. There are throughput studies for persistent sessions with and without lossy links (see, for example, [16, 15, 4, 11]).

We observe that if the propagation delay on the link is negligible, the active sessions are served in (roughly) a Round Robin manner. It is easier to study the behaviour of a Round Robin server via an ideal Processor Sharing model. So we study, via analysis and simulations, three measures of average session throughputs in the context of a PS model. Our simulation results suggest that the average session throughput is closely approximated by the time average bandwidth share per session (Equation 3) for the PS model. We then use simulations to explore the possibility of such an approximation holding in case of TCP controlled nonpersistent sessions. We find that the closeness between two throughput measures observed in case of PS model extends to the case of TCP controlled bandwidth sharing with nonpersistent sessions as well.

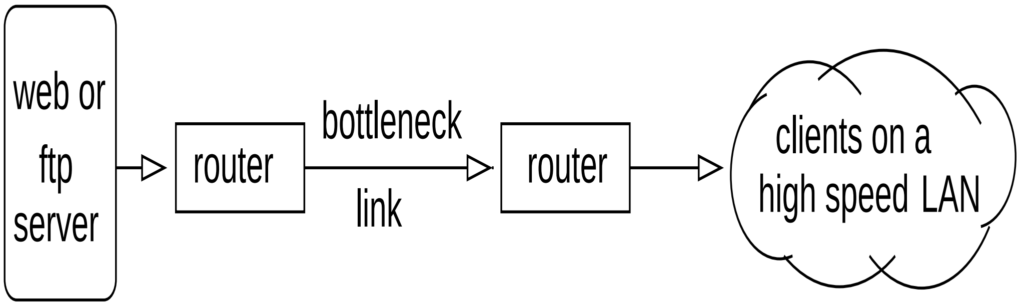

Figure 3: The network scenario considered in this paper

Some prior work related to ours is the following. In [6] and [3] 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 *goodput* obtained by subscribers. Assuming zero propagation delay, and ideal bandwidth sharing (i.e., the entire bottleneck link rate is applied to the ongoing transfer and the rate is shared equally), a closed queueing network model applies. In [6] the authors also

[1] Note that we are *not* assuming that the data packets arrive in a Poisson process. See Paxson [21] for a discussion of this assumption.

provide an approximate "correction" to the results obtained by assuming ideal bandwidth sharing to take care of TCP behavior. We have discussed before that the parameters for such a model are hard to obtained from measurements at the edge device. In [9] the authors consider a generalisation of the stochastic model that we have stated above (i.e., Poisson session arrivals but limited to exponential file size distribution) to the case of a network. Again assuming ideal bandwidth sharing, 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. Our work is not restricted to the assumption of there being atmost one file download over an access link at a time. The performance of TCP sessions shown to depend only on the total occupancy of a link and hence the we can use online measurements to ensure some average throughput. We also use heavy tailed file sizes for the results shown.

The paper is organised as follows. In Section 2 we motivate the use of the PS model for TCP controlled bandwidth sharing over a single link with small propagation delay. In Section 3 we introduce and compare performance measures for average "throughput" in the PS model with ephemeral sessions. In Section 4 we provide the experimental results from the *ns* network simulator.

2 Equal Bandwidth Sharing: The Processor Sharing Model

It has been argued that for sessions in a network the TCP protocol provides roughly proportionally fair bandwidth sharing [8]. In the case of sessions sharing a single link, however, proportional fair sharing is the same as equal sharing. In this section we argue that over a single, small propagation delay link, TCP controlled bandwidth sharing may plausibly be modelled by the PS model. Our results in subsequent sections will support this plausibility.

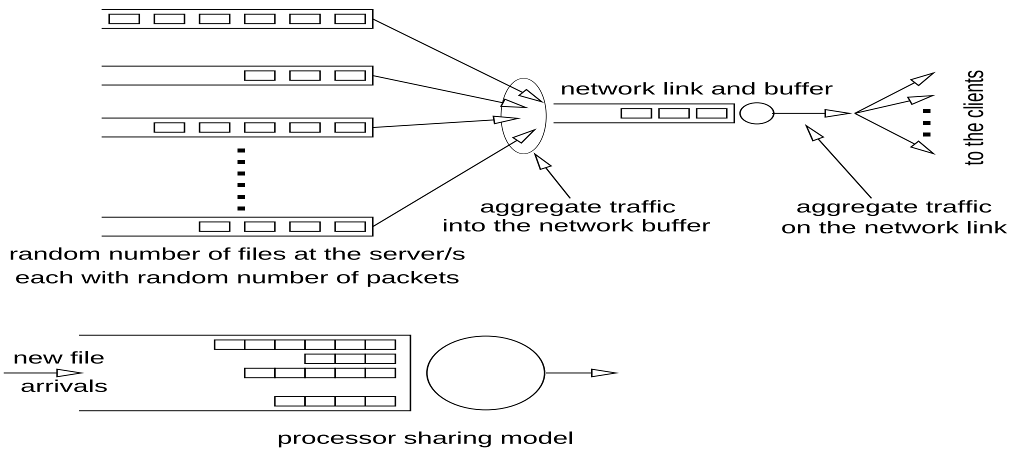

Figure 4: The actual flow of traffic in the network, and the processor sharing model.

Figure 4 shows a schematic of how traffic flows in the network shown in Figure 3.

After each client request reaches the server, we can view the files to be "queued" at the server, waiting as they are gradually transferred to the clients. In the figure, the files have been shown broken up into fixed length packets. With equal bandwidth sharing, when there are n files, each file transfers its packets at $\frac{1}{n}$th of the link rate. The network link carries the superposition of these rate controlled flows from each file transfer. When the propagation delay between server and clients is small, we can assume that each client experiences the same propagation delay. Small propagation delay (say, of less than one packet size) also implies that the sender gets acknowledgement of a packet as soon as the packet is served and hence the sender releases the next packet as soon as one of its packet gets served. Thus there is no time wasted between successive packet arrivals of a session when there is only one session and hence we can view the link as a nonidling server. Thus, considering a *fluid* model, we model the file transfer process, with *ideal* bandwidth sharing, by a PS queue model. The unsent fragments of the files *in the server*, constitute the "customers" queued in the PS model (see the lower part of Figure 4).

Hence, for small propagation delay links, we can view the service received by TCP controlled sessions similar to that when the sessions are arriving at a Processor Sharing queue with service rate equal to the link capacity. Observe that, as the propagation delay on the link increases, the time wasted for transmission of a packet over the link will increase; this effect will depend on the window size of a session and the number of other sessions sharing the link. In this case, clearly we will not have a work conserving server. Thus, for large propagation delay link, a session will be served as if the server is of some time varying rate $R(t)$, less than the actual link capacity and the PS model used here will not apply directly. This effect of large propagation delay showing up as time varying link rate has been used in analysing TCP throughput with nonpersistent sessions in [22] where we have modeled the evolution of the $R(t)$ process as a Regenerative process.

Note that the restriction to small propagation delays still results in useful situations as, typically, large capacity small length links are used for an intracity connections and for intercity links small capacity is used.

3 Session Throughputs in the PS Model

In the case of a single persistent session using a link, the performance measure widely used is the average rate that the session gets, i.e., if $V(t)$ denotes the amount of data transferred by the session in time t then its throughput is given by $\lim_{t\to\infty} \frac{V(t)}{t}$. Our aim in this section is to examine throughput measure in the case where the sessions arrive and depart.

For a unit capacity link, let,

λ be the session arrival rate,

X_k be the k^{th} session's service requirement,

EX be the mean service requirement,

$\rho = \lambda E(X)$ be the link utilization,

$a_k(d_k)$ be the k^{th} session's arrival(departure) epoch,

$W_k(= d_k - a_k)$ be the sojourn time of the k^{th} session, and

$N(u)$ be the number of active sessions at time u.

Also let (X, W) denote the stationary random vector of the service requirement and the sojourn time of a session (which is known to exist for $\rho < 1$). In terms of above notations the average per session throughput, τ in data units per time unit is given by

$$\tau \;=\; \lim_{n\to\infty} \frac{1}{n} \sum_{k=1}^{n} \frac{X_k}{W_k} \;= E\frac{X}{W}. \tag{1}$$

An other possible throughput measure is given by

$$\phi = \frac{EX}{EW} \tag{2}$$

For an M/G/1 PS queue, we have $EW = \frac{EX}{(1-\rho)}$ yielding $\phi = 1 - \rho$. This is insensitive to the distribution of X. Theorem 3.1 shows that, ϕ is also a lower bound on τ for an M/G/1 PS queue.

Theorem 3.1 *For an M/G/1 PS queue with unit capacity server,* $\tau \geq \phi$.

Proof: We know (see [14]) that the expected sojourn time of a session with a service requirement of x in a PS queue is given by $EW = \frac{x}{1-\rho}$ irrespective of the file size distribution. Now, $E\frac{W}{X} = E(E\frac{W}{X}|X) = \frac{1}{1-\rho}$. Hence, from Jensen's inequality and as $\phi = 1-\rho$, $E\frac{X}{W} \geq \frac{1}{E\frac{W}{X}} = (1-\rho) = \phi$. •

Though for an M/G/1 PS Queue, EW has a nice expression, the *distribution* of the session's sojourn time in the system becomes very cumbersome (see [19] and references therein). This makes τ analytically intractable (in some cases τ can be evaluated numerically as shown in Appendix A). Clearly ϕ, in spite of having a nice expression, is not a good candidate for performance measure as it does not explicitly capture the dependence of a session's sojourn time on its service requirement.

Another measure that we consider is the time average bandwidth share per active session. This and similar measures have been used in [3, 6, 24]. Let the service rate given to session k at instant t be $r_k(t)$; then the average bandwidth share is defined by

$$\sigma = \lim_{t\to\infty} \frac{\int_0^t \frac{1}{N(u)} \left(\sum_{k=0}^{\infty} r_k(u)\right) I_{\{N(u)\geq 1\}} du}{\int_0^t I_{\{N(u)\geq 1\}} du}. \tag{3}$$

For a unit capacity PS server, $r_k(u) = \frac{1}{N(u)} I_{\{N(u) \geq 1\}}$; hence,

$$\sigma = \lim_{t \to \infty} \frac{\int_0^t \frac{1}{N(u)} I_{\{N(u)\geq 1\}} \, du}{\int_0^t I_{\{N(u)\geq 1\}} du} \cdot \tag{4}$$

where the denominator converges to ρ and is included in the expression to compensate for the fraction of time when the server is free, i.e., the fraction of time when there is no customer (and hence no throughput). Assuming that there is a steady state ($\rho < 1$), let $\pi(n)$ denote the probability of n sessions active in steady state. We have $\sigma = \frac{1}{1-\pi(0)} \sum_{n=1}^{\infty} \frac{\pi(n)}{n}$. It is well known [26] that for the PS queue the stationary probability distribution $\pi(n)$ is insensitive to the distribution of service time (in the present problem this translates to insensitivity to the distribution of file sizes), and is given by $\pi(n) = (1-\rho)\rho^n, \quad n \geq 0$. It is then easily seen that,

$$\sigma = \frac{1-\rho}{\rho} \ln \frac{1}{1-\rho} \, . \tag{5}$$

Thus σ has a simple, closed form formula for the PS queue, whereas τ as defined in Equation 1 is difficult to compute for PS Queue. However τ is a better representative of the actual session throughput as perceived by users. It is thus interesting to study the relationship between σ and τ for the PS queue. We do this through the following results and also through simulations of the PS model.

Theorem 3.2 *For a G/GI/1 PS queue with unit capacity server, if* $\lambda \, E(X) < 1$, *then, with probability 1 (w.p.1),*

$$\sigma > \frac{1}{EX} E\left(\frac{X^2}{W}\right) \cdot \bullet$$

Proof: See Appendix C for proof.

Corollary 3.1 *For G/D/1 PS Queue, i.e., with* $D > 0$ *given, and* $X_k = D$, $\forall k \geq 1$, $\sigma > \tau$ *w.p.1.* $\bullet$

Proof: Using Theorem 3.2, $\sigma > \frac{1}{E(X)} E\left(\frac{X^2}{W}\right)$, *w.p.*1. For $X_k = D \;\; \forall k$, we have, *w. p.* 1,

$$\frac{1}{E(X)} \, E\left(\frac{X^2}{W}\right) = D \, E\left(\frac{1}{W}\right) = \tau \cdot$$

Hence, $\sigma > \tau$, *w.p.*1 $\bullet$

Corollary 3.1 states that σ upper bounds τ for the τ for M/D/1 PS queue. We will find from simulations below that $\sigma > \tau$ appears to hold more generally, and further that σ is a good approximation to τ.

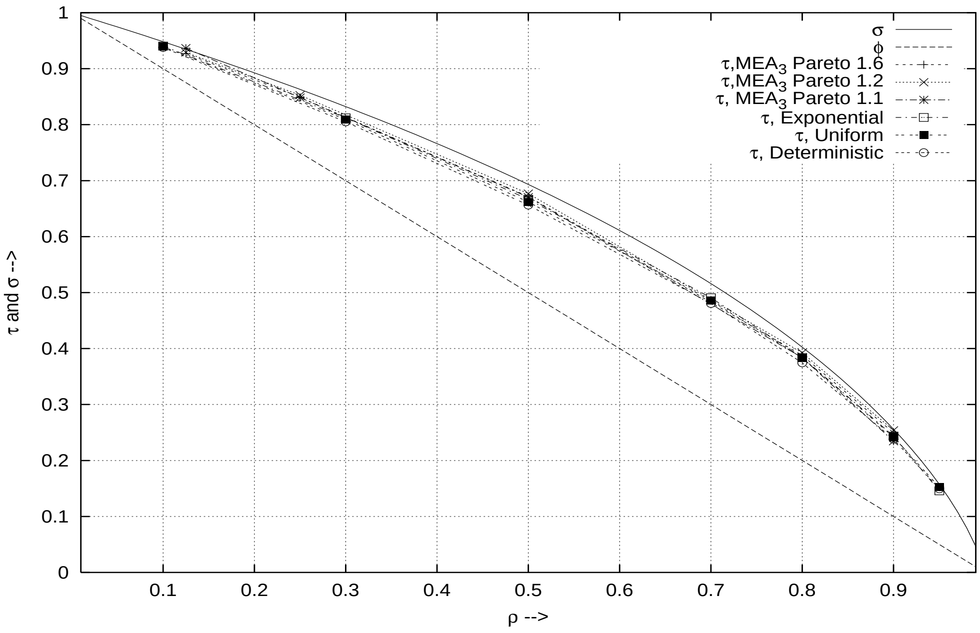

Figure 5: σ (from Equation 5), ϕ ($= 1 - \rho$; see Equation 2) and τ (from fluid simulations of PS model) vs ρ for various file size distributions

Figure 5 shows σ and τ for different file size distributions for an ideal PS queue with a unit capacity server. The values for τ are obtained from simulating the PS queue and σ is obtained from Equation 5. The method used to simulate Pareto file size distribution is that of fitting mixture of exponentials and is discussed further in Appendix B. Numerical evaluation of τ was possible in some cases and is discussed in Appendix A. Figure 5 also shows ϕ (which is also a lower bound on τ for the M/G/1 PS queue from Theorem 3.1). Whereas the measure σ has been proved to be insensitive to the file size distribution, no such results regarding τ is known. It is seen, however, from the figure that the σ curve for the deterministic case is above the corresponding τ curve as the corollary 3.1 claims. We see that τ is very close to σ, and also is not very sensitive to the file size distribution. We have seen in this section that σ, while being computationally simple and insensitive to file size distribution, is also a close approximation to τ for a variety of service distributions including some approximations to heavy tailed distribution. In the next section we explore, via simulations, the possibility of a similar match between τ for TCP controlled ephemeral sessions and the σ given by Equation 5.

4 Comparison with TCP performance on a Single Low Propagation delay Link

As seen in the previous section, σ given by Equation 5 serves as an approximation for τ for an M/G/1 PS queue. As mentioned in the introduction, our goal is to find a simple formula to approximate the throughput (τ) obtained by TCP controlled ephemeral sessions sharing a single link. Owing to the complex nature of the TCP protocol as compared to an M/G/1 PS queue studied in the last section, it is natural to expect that obtaining an exact formula for τ for TCP controlled nonpersistent sessions will be hard. The observations of the previous section and the discussion in Section 2, suggest exploring the possibility of using Equation 5 to approximate τ for TCP controlled sessions. We do this in the present section via simulations using the *ns* [17] network simulator. Throughout this section we assume a FIFO link buffer and no random drop of packets. As observed in [18], we expect the results to be similar if we use link buffer with RED capability. The TCP packet size is fixed at $1500 Bytes$ and the mean file transfer size requirement is fixed at $10 KBytes$. We use various link speeds (C), link propagation delays (δ), and various file size distributions for the results presented in this section. For simulation of Pareto distributions, we have used a mixture of three exponentials as an approximation to Pareto distribution (see Appendix B for details). The values of propagation delay were chosen to be less than a packet transmission time. The following combination of network parameters were used.

1. Figure 6: $C = 128Kbps, \delta = 0$ seconds, link buffer = 32 packets
2. Figure 7: $C = 128Kbps, \delta = 0.01$ seconds, link buffer = 32 packets
3. Figure 8: $C = 1Mbps, \delta = 0$ seconds, link buffer = 64 packets
4. Figure 9: $C = 1Mbps, \delta = 0.005$ seconds, link buffer = 64 packets

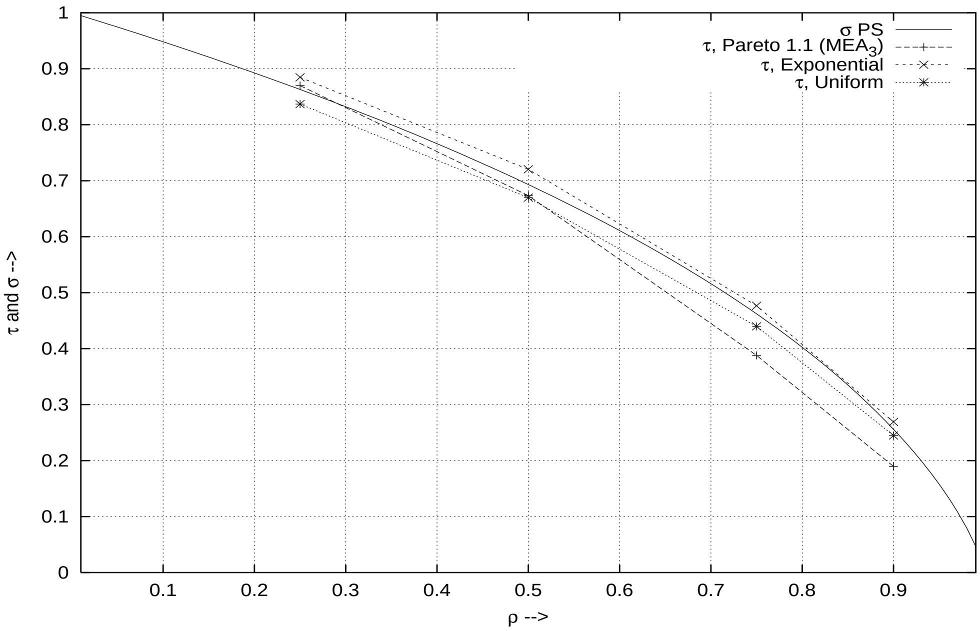

Figure 6: τ as obtained from *ns*, $C = 128Kbps$, buffer size=32 packets and $\delta = 0$ seconds. σ values from Equation 5.

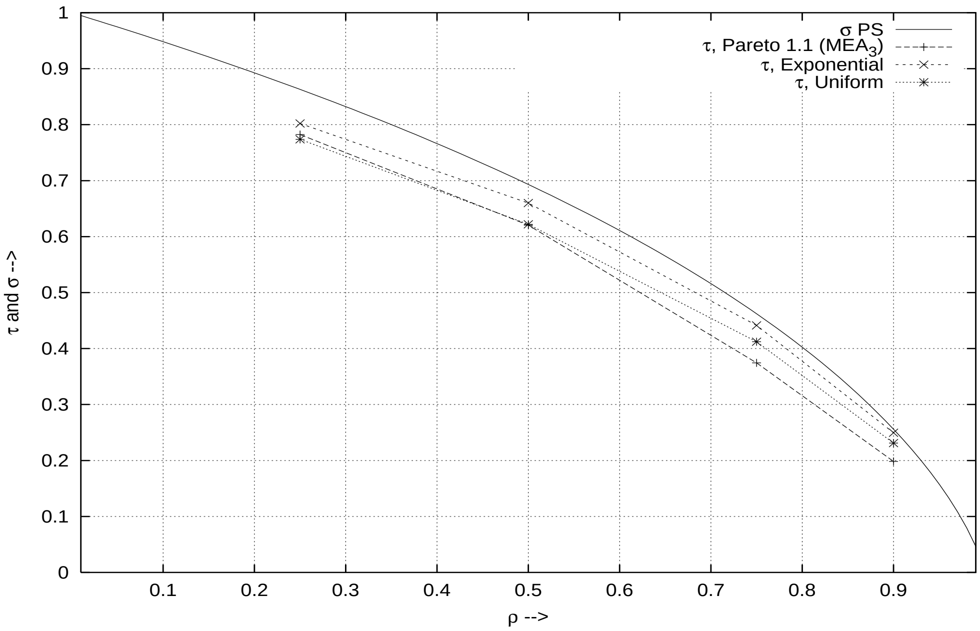

Figure 7: τ as obtained from ns, $C = 128Kbps$, buffer size=32 packets and $\delta = 0.01$ seconds. σ values from Equation 5.

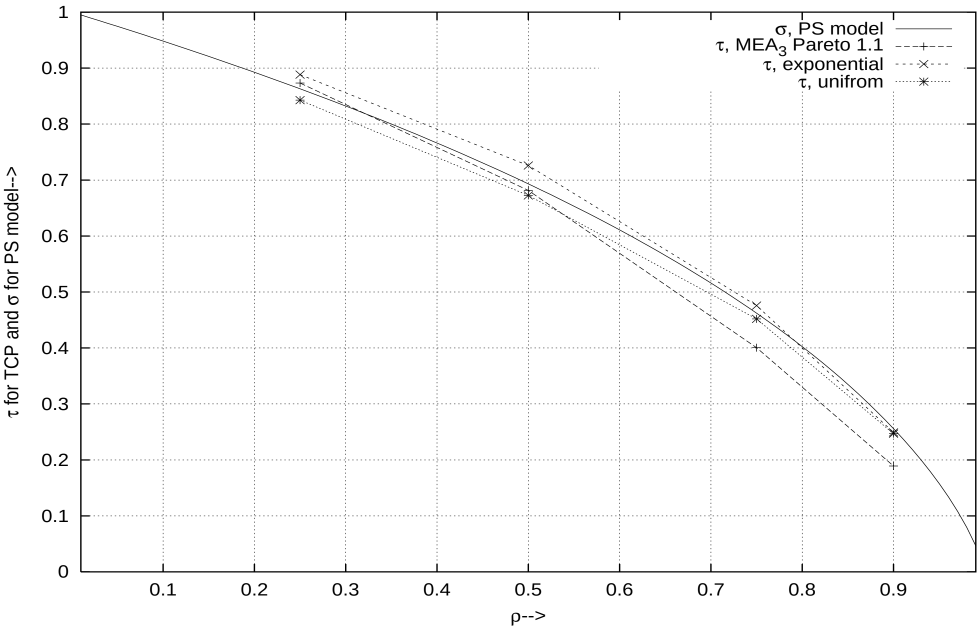

Figure 8: τ as obtained from ns, $C = 1Mbps$, buffer size=64 packets and $\delta = 0$ seconds. σ values from Equation 5.

4.1 Discussion of the Numerical Results

Figures 6, 7, 8, 9 show the values of τ normalised to link capacity for TCP controlled ephemeral sessions sharing a small propagation delay link as obtained from simulations

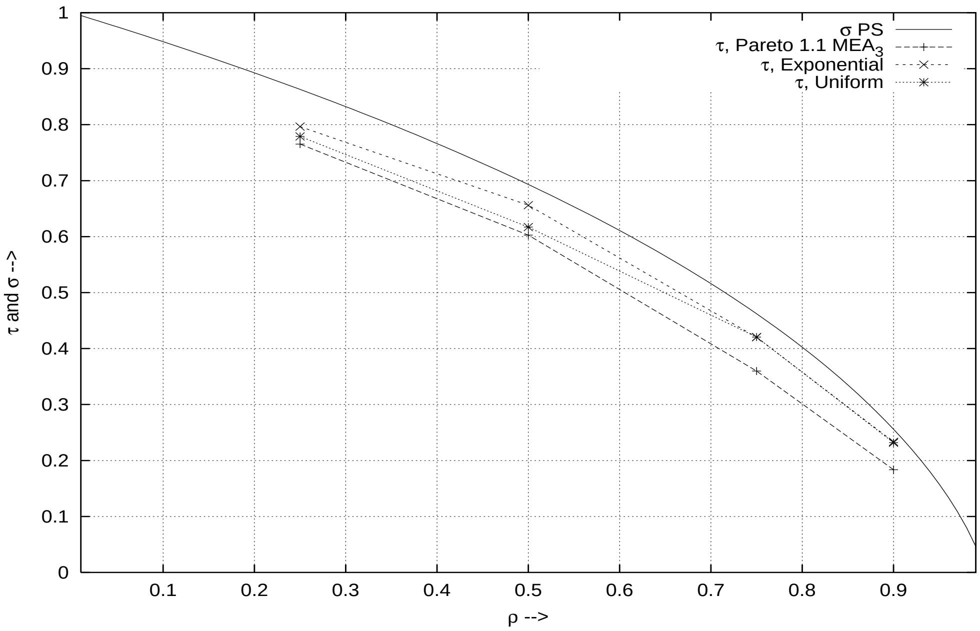

Figure 9: τ as obtained from ns, $C = 1Mbps$, buffer size=64 packets and $\delta = 0.005$ seconds. σ values from Equation 5.

using ns. The various curves also show σ as obtained from Equation 5. The τ curves are obtained for different file size distributions. The common observation from all these figures is that, for small propagation delays (within transmission time of one packet), σ as given by Equation 5 serves as an approximation for τ even in the case of TCP controlled nonpersistent sessions. This is a significant simplification as obtaining τ even numerically in the case of the ideal bandwidth sharing (PS queue) turns out to be hard. On the other hand, σ has a nice expression and depends only on the link load which can be measured on-line. The effect of link buffer on throughputs for $\delta = 0$ seconds is studied in Section 4.2.

4.2 Effect of Link Buffer

We performed simulations with different values of commonly used link buffer sizes with a service requirement of Pareto 1.1. Figure 10 shows the values of τ obtained from simulations for $C = 1Mbps$, $\delta = 0$ seconds and for three different values of link buffer. It is observed from the figure that for the commonly used range of link buffer, we do not get significant difference in throughputs.

5 Conclusions

We have studied TCP controlled bandwidth sharing at a single link with small propagation delay via a Processor Sharing model.

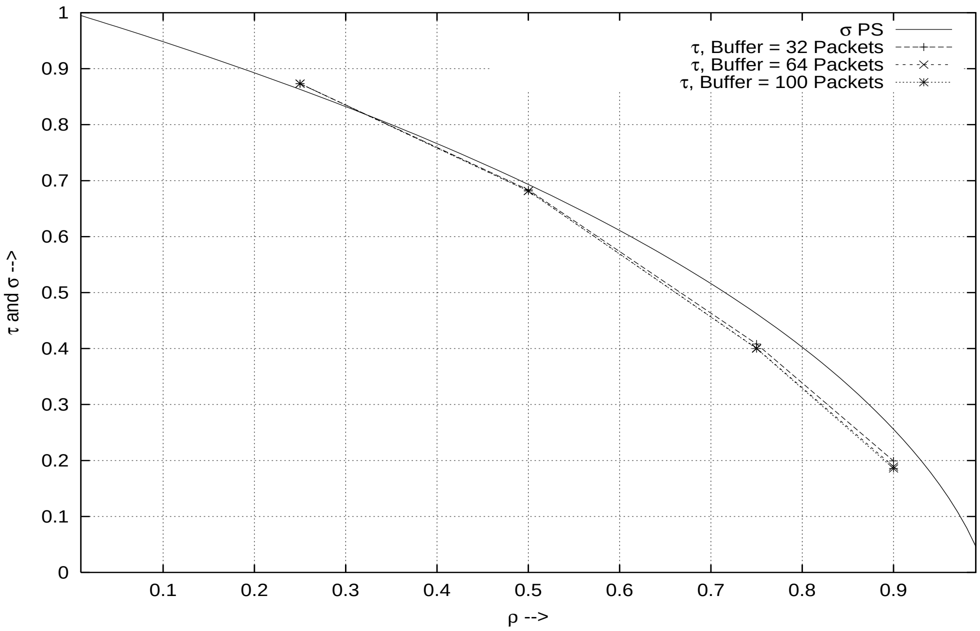

Figure 10: Figure showing effect of buffer size on τ. $C = 1Mbps$, buffer sizes are 32 packets, 64 packets and 100 packets and $\delta = 0$ seconds. σ values from Equation 5.

We introduced and studied three throughput measures in the context of an M/G/1 Processor Sharing queue. We found that the throughput of TCP controlled ephemeral sessions sharing a single small propagation delay link is well approximated and bounded above by the throughput measure σ for which we have a simple closed form formula (Equation 5). This is a significant simplification considering the complex nature of the TCP algorithm. Further, τ is only slightly sensitive to the file size distribution. Though τ can be found numerically for an M/G/1 PS queue (see Appendix A), this requires knowledge of the file size distribution, and the session arrival rates and this numerical evaluation (as proposed in Appendix A) is slow whereas σ requires only the knowledge of the average load (ρ) which can be easily measured on-line.

As mentioned in the Introduction, our interest is in bandwidth dimensioning to provide an average minimum throughput τ to the contesting TCP controlled ephemeral sessions. For this we use the observation that σ approximates, and is a close upper bound to τ. *So, given an average throughput requirement, we can use Equation 5 to find the corresponding normalised link load, and hence a bandwidth that corresponds to this load will provide the required average throughput.*

An important research effort will be to provide corrections to the results obtained from the PS model in order to better predict the performance of TCP controlled bandwidth sharing when large propagation delays are involved (see, for example [6]). Investigating the possibility of using the PS model (or some variation thereof) for TCP controlled sessions on a network with an arbitrary link and route topology, and propagation delays, will also be an important future work.

A Finding τ from Transform of Service Requirement Distribution

In this appendix we show how to numerically compute τ for an M/G/1 PS Queue. By definition,

$$\tau = E\left[\frac{X}{W}\right] = \int_x xE\frac{1}{W(x)}dF(x)$$

where $W(x)$ is the random variable corresponding to the sojourn time of a session with service requirement x; observe that $W(x) \geq x$, and hence $W(x) > 0$ for $x > 0$. Thus we only need to find $E\frac{1}{W(x)}$ to be able to compute τ. Obtaining a closed form expression is hard for general $F(.)$. So, we resort to a numerical evaluation of $E\frac{1}{W(x)}$; we do this via the following result.

Lemma A.1 *For a positive random variable Z with probability distribution $G(.)$ and Laplace-Stieltjes transform (LST) $\tilde{z}(s)$, $E\frac{1}{Z} = \int_{s=0}^{\infty} \tilde{z}(s)ds$.*

Proof:

$$\int_{s=0}^{\infty} \tilde{z}(s)ds = \int_{s=0}^{\infty} \left[\int_{z\geq 0} e^{-sz}dG(z)\right] ds$$

Due to positivity of Z, Tonelli's theorem [25] allows the interchange of the integrals in z and s, hence,

$$\int_{s=0}^{\infty} \tilde{z}(s)ds = \int_{z\geq 0} \left[\int_{s=0}^{\infty} e^{-sz}ds\right] dG(z) = \int_{z\geq 0} \frac{1}{z}dG(z) = E\frac{1}{Z} \qquad \bullet$$

Define the LST of $W(x)$ by $v(s,x) := Ee^{-sW(x)}$. A relation between $v(s,x)$ and the LST of X for an M/G/1 PS queue is given in [20] which can be used for finding $E\frac{1}{W(x)}$ numerically. Thus using the LST of service requirement distribution, τ can be found using numerical integration. We use this method to verify the simulation results of an M/G/1 PS queue.

B Approximating a Pareto Distribution

The tail probability of a Pareto distributed random variable X is

$$P\{X > x\} = \begin{cases} 1, & 0 \leq x < k \\ \left(\frac{x}{k}\right)^{-\alpha}, & x \geq k \end{cases}$$

and mean of X is $EX = \frac{\alpha k}{\alpha - 1}$, $\alpha \geq 1$. For $1 \leq \alpha \leq 2$, the variance of X is infinite resulting in the difficulty of obtaining reliable estimates from simulations [1].

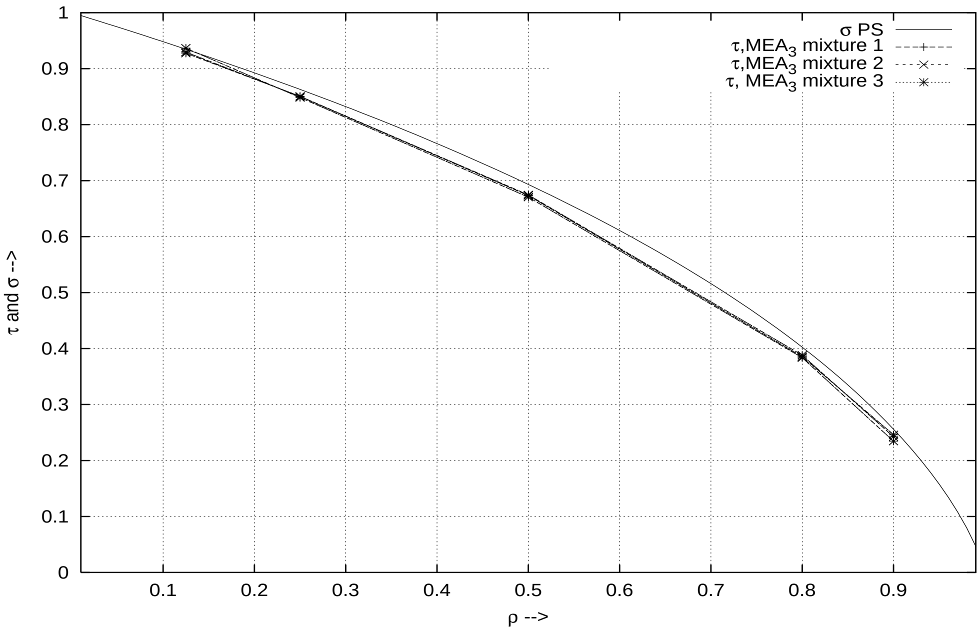

Figure 11: Figure showing τ from simulation of M/G/1 PS queue with mixture of three exponentials approximating Pareto (with $\alpha = 1.1$) service requirement distribution and for different relative positions of the approximating exponentials.

Owing to the above problem, we use a mixture of exponentials to approximate Pareto distribution (for an excellent discussion and a general algorithm on approximating heavy tailed distributions, see [1]). As we are interested in Pareto distribution in the context of web traffic, we use a mixture of three exponentials to approximate the Pareto distribution; the three exponentials corresponding to the "web elephants", "web mice" and "other web animals in between". We also require the mean of each of these exponentials to lie in practical range (for example, a mean of the order of MegaBytes for elephants is reasonable for practical purposes). We call such a mixture an MEA_3 as an abbreviation for "A Mixture of Exponential Approximation" with the index giving the number of exponentials used.

Figure 11 shows the values of τ from simulations of M/G/1 PS queue where the Pareto distribution of file size requirements (with mean file size $10KBytes$ and tail parameter $\alpha = 1.1$) is approximated by mixtures of exponentials with different relative positions of exponentials. For Mixture 1 the mean of largest exponential is $164KBytes$, for Mixture 2, it is $1.64MBytes$ and Mixture 3 corresponds to largest exponential with mean $16.4MBytes$. Also shown is the σ curve from Equation 5. Note that, for this case, we can use the method of Appendix A to obtain values of τ numerically; we do not provide the results of this numerical evaluation as they match the simulation results and only serve the purpose of verifying the simulation results. Figure 11 shows that the relative position of the approximating exponentials do not affect the value of τ. A similar observation is made from Figure 12 showing the τ values from TCP simulations with different relative positions of the approximating exponentials. Hence we use a mixture of three exponentials to approximate Pareto distributed file sizes in the simulation results presented in this work.

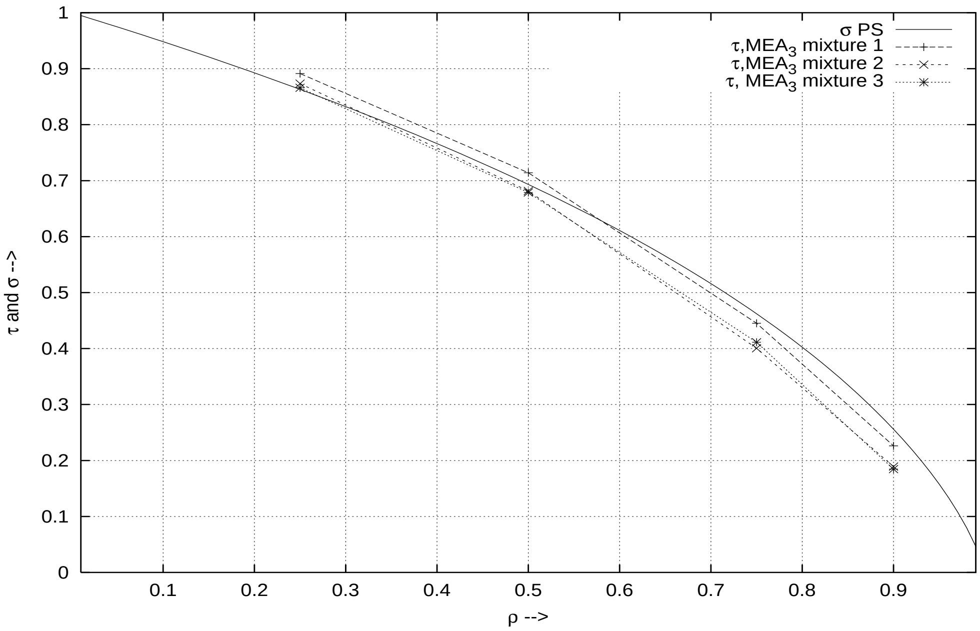

Figure 12: Figure showing τ from simulation of TCP controlled nonpersistent sessions sharing a single zero propagation delay link with mixture of three exponentials approximating Pareto (with $\alpha = 1.1$) service requirement distribution and for different relative positions of the approximating exponentials.

C Proof of Theorem 3.2

Associate with session k the time function

$$f_k(u) = \begin{cases} \frac{1}{N^2(u)} & u \in [a_k, d_k) \\ 0 & \text{otherwise,} \end{cases}$$

Using Brumelle's Theorem [26], we have

$$\lim_{t\to\infty} \frac{1}{t} \int_0^t \sum_{k=1}^{\infty} f_k(u)du = \lambda \lim_{n\to\infty} \frac{1}{n} \sum_{k=1}^{n} \int_0^{\infty} f_k(u)du$$

i.e., using Equation 5,

$$\frac{1}{\rho} \sum_{n=1}^{\infty} \frac{\pi(n)}{n} = \frac{1}{E(X)} \lim_{n\to\infty} \frac{1}{n} \sum_{k=1}^{n} \int_{a_k}^{d_k} \frac{1}{N^2(u)} du \tag{6}$$

Consider the right hand side of Equation 6. Using the Cauchy-Schwartz inequality,

$$\int_{a_k}^{d_k} \left(\frac{1}{N(u)}\right)^2 du \int_{a_k}^{d_k} 1^2 du \geq \left(\int_{a_k}^{d_k} \frac{1}{N(u)}.1du\right)^2, \tag{7}$$

Since $W_k = d_k - a_k$, and, due to the processor sharing service, $X_k = \int_{a_k}^{d_k} \frac{1}{N(u)} du$, we can rewrite Equation 7 as

$$\int_{a_k}^{d_k} \left(\frac{1}{N(u)} \right)^2 du \geq \frac{X_k^2}{W_k}. \tag{8}$$

We now take our cue from the strict inequality condition in Equation 8 to prove that, w.p.1, and, for

$$U_k = \int_{a_k}^{d_k} \left(\frac{1}{N(u)} \right)^2 du \quad \text{and} \quad V_k = \frac{{X_k}^2}{W_k}.$$

Thus we need to show that, w.p.1,

$$\lim_{n \to \infty} \frac{1}{n} \sum_{k=1}^{n} U_k > \lim_{n \to \infty} \frac{1}{n} \sum_{k=1}^{n} V_k.$$

For fixed $\epsilon > 0$, let k_j index those (tagged) sessions that satisfy the following conditions:

1. A tagged session arrives to an empty queue,
2. A tagged session brings at least 44ϵ units of data,
3. Exactly one other session arrives during the sojourn of a tagged session
4. The only session to arrive during the sojourn of a tagged session arrives after the tagged session has transferred half of its data but before it has transferred 3/4 of its data.
5. The arriving session has so much of data to be transferred that it does not finish before the tagged session finishes.

A typical scenario for a tagged session is shown in Figure 13; in this figure the solid line is $N(u)$.

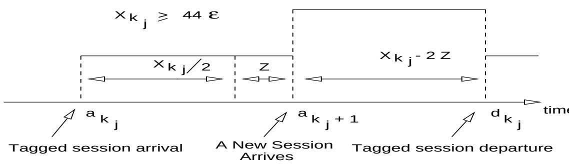

Figure 13: A tagged session, and the single arriving session

It is easily shown that [13], $U_{k_j} - V_{k_j} \geq \epsilon > 0$. Let $m(n) = |\{ k \leq n : \exists j \geq 1 \; s.t. \; k = k_j \}|$ Using the fact that $U_{k_j} \geq V_{k_j} + \epsilon$, and $U_k \geq V_k$ for $k \neq k_j$, we have,

$$\lim_{n \to} \frac{1}{n} \sum_{k=1}^{n} U_k \geq \lim_{n \to \infty} \frac{1}{n} \sum_{k=1}^{n} V_k + \lim_{n \to \infty} \frac{m(n)}{n} \epsilon$$

Letting $G(.)$ denote the distribution function of the amount of data in a session, we have, w.p.1., $\lim_{n\to\infty} \frac{m(n)}{n} =$

$$\pi(0) \int_{44\epsilon}^{\infty} \int_{\frac{x}{2}}^{\frac{3x}{4}} (1-G(x-t)) \left(\exp -\lambda 2(x-t) \right) \lambda \exp -\lambda t \, dt dG(x).$$

Let, the right hand side integral be denoted by p, and observe that $p > 0$. Hence, w.p.1, we have

$$\lim_{n\to\infty} \frac{1}{n} \sum_{k\geq 1}^{n} U_k \geq \lim_{n\to\infty} \frac{1}{n} \sum_{k=1}^{n} V_k + p\epsilon > \lim_{n\to\infty} \frac{1}{n} \sum_{k=1}^{n} V_k.$$

Hence the proof. •

References

[1] Anja Feldman and Ward Whitt. Fitting Mixtures of Exponentials to Long-Tail Distributions to Analyze Network Performance Models.

[2] Anja Feldman et al. IP Network Configuration for Traffic Engineering.

[3] Arthur W. Berger and Yaakov Kogan. Dimensioning bandwidth for elastic traffic in high-speed data networks. *Draft*, 1998.

[4] Patrick Brown. Resource sharing of TCP connections with different round trip times. *Proc. IEEE INFOCOM'2000*, April 2000.

[5] Claudio Casetti and Michela Meo. A new approach to model stationary behavior of TCP connections. *Proc. IEEE INFOCOM'2000*, April 2000.

[6] T.V.Lakshman D.P.Heyman and A.L.Neidhardt. New method for analyzing feedback protocols with applications to engineering web traffic over the Internet. *Proceedings of ACM SIGMETRICS*, 1997.

[7] Konstantin Avrachenkov Eitan Altman and Chadi Barakat. TCP in presence of bursty losses. *Performance Evaluation*, 42, 2000.

[8] A.K. Maulloo F.P. Kelly and D.K.H. Tan. Rate control for communication networks: shadow prices, proportional fairness and stability. *Operations Research*, 49, 1998.

[9] Tae-Jin Lee Gustavo De Veciana and Takis Konstantopoulos. Stability and performance analysis of networks supporting services with rate control-could the Internet be unstable. *Proc. IEEE INFOCOM'1999*, April 1999.

[10] J. C. Hoe. Improving the start-up behavior of a congestion control scheme for tcp. In *ACM Sigcomm*, 1996.

[11] Don Towsley Jitendra Padhye, Victor Firoiu and Jim Kurose. Modeling TCP Throughput: A Simple Model and its Empirical Validation. *SIGCOMM'98*, 1998.

[12] J.W.Roberts. Engineering for quality of service. 1998.

[13] Arzad A. Kherani. Rate based congestion control for ephemeral best effort sessions in packet networks. Master's thesis, Indian Institute of Science, INDIA, January 1999.

[14] Leonard Kleinrock. *Queueing Systems*, volume 2. John Wiley and Sons, 1996.

[15] Anurag Kumar. Comparative performance analysis of versions of TCP in local network with a lossy link. *IEEE/ACM Transactions on Networking*, 6(4), August 1998.

[16] T.V. Lakshman and Upamanyu Madhow. The performance of TCP/IP for networks with high bandwidth delay products and random loss. *IEEE/ACM Transactions on Networking*, 5(3):336–350, June 1997.

[17] Steven McCanne and Sally Floyd. ns-LBL Network simulator. *Obtain via: http://www-nrg.ee.lbl.gov/ns*, 1995.

[18] Mikkel Christiansen et al. Tuning RED for Web Traffic. *Proc. SIGCOMM'2000*, 2000.

[19] J. A. Morrison. Response time distribution for a processor sharing system. *SIAM Journal of Applied Mathematics*, 45:1:152–167, 1985.

[20] T. J. Ott. The sojourn-time distribution in the M/G/1 queue with processor sharing. *Journal Of Applied Probability*, 21:360–378, 1984.

[21] V. Paxson and S. Floyd. Wide area traffic: The failure of poisson modelling. *IEEE/ACM Transactions on Networking*, 3(3):226–244, 1995.

[22] reference suppressed for anonymity.

[23] Robert Morris. TCP behavior with many flows. *Proc. IEEE International Conference on Network Protocols.*, October 1997.

[24] J.W. Roberts and L. Massoulie. Bandwidth sharing and admission control for elastic traffic. *ITC Specialist Seminar, Yokohama*, October 1998.

[25] H. L. Royden. *Real Analysis*, volume 2. Prentice Hall India.

[26] Ronald W.Wolf. *Stochastic Modelling and The Theory of Queues.* Prentice Hall, Englewood Cliffs, 1989.