Long range dependence in network traffic and the closed loop behaviour of buffers under adaptive window control

Arzad A. Kherani[a,*], Anurag Kumar[b]

[a] *INRIA, 2004 Route des Lucioles, BP-93, Sophia Antipolis 06902, France*
[b] *Department of Electrical Communication Engineering, Indian Institute of Science, Bangalore 560012, India*

Abstract

We consider an Internet link carrying http-like traffic, i.e., transfers of finite volume files arriving at random time instants. These file transfers are controlled by an adaptive window protocol (AWP); an example of such a protocol is TCP.

We provide analysis for the auto-covariance function of the AWP-controlled traffic into the link's buffer; this traffic, in general, *cannot* be represented by an on–off process. The analysis establishes that, for TCP-controlled transfer of Pareto-distributed file sizes with infinite second moment, the traffic into the link buffer is long range-dependent (LRD).

We also develop an analysis for obtaining the stationary distribution of the link buffer occupancy under an AWP-controlled transfer of files sampled from some distribution. For any AWP, the analysis provides us with the Laplace–Stieltjes transform (LST) of the distribution of the link buffer occupancy process in terms of the functions defining the AWP and the file size distribution. The analysis also provides a necessary and a sufficient condition for the finiteness of the mean link buffer content; these conditions again have explicit dependence on the AWP used and the file size distribution. This establishes the sensitivity of the buffer occupancy process to the file size distribution.

Combining the results from the above analyses, we provide various examples in which the closed loop control of an AWP results in finite mean link buffer occupancy even though the file sizes are Pareto-distributed (with infinite second moment), and the traffic into the link buffer is long range-dependent (with Hurst parameters which would suggest an infinite mean queue occupancy under open loop analysis).

We also study the effect of window reductions due to active queue management and find that window reductions lead to further lightening of the tail of buffer occupancy distribution.

The significance of this work is three-fold: (i) by looking at the window evolution as a function of the amount of data served and not as a function of time, this work provides a new framework for analysing various processes

---

* Corresponding author.
 E-mail addresses: alam@sophia.inria.fr (A.A. Kherani), anurag@ece.iisc.ernet.in (A. Kumar).

related to the link buffer under AWP-controlled transfer of files with a general file size distribution; (ii) it indicates that the buffer behaviour in the Internet may not be as poor as predicted from an open loop analysis of a queue fed with LRD traffic; and (iii) it shows that the buffer behaviour (and hence the throughput performance for finite buffers) is sensitive to the distribution of file sizes.

---

1. Introduction

It was observed in [1] that traffic processes in the Internet display long range dependence. In [2], this phenomenon was traced to the fact that the traffic in the Internet results from the transfer of files that have a heavy-tailed distribution. Models have shown that the transfer of Pareto-distributed files ($P\{V > x\} = \min(1, \frac{1}{x^\alpha})$, $1 < \alpha < 2$) results in a traffic rate process that has an auto-correlation function that decays as $\frac{1}{\tau^{\alpha-1}}$. These observations have been taken to indicate that the buffer occupancy distribution in router buffers will have heavy tails [3]. Such observations are, however, based on an "open loop" analysis of an LRD traffic source feeding a buffer. It has also been noted recently [4] that an understanding of traffic and buffer processes in the Internet should take into account the closed loop nature of Internet congestion control, namely TCP which is an adaptive window protocol (AWP). In this paper, we carry out such an analysis for a particular network scenario.

The Internet carries predominantly elastic traffic; the transfer of such traffic is controlled by TCP [5]. Most of the literature on TCP modelling is concerned with the "throughput" obtained by TCP-controlled file transfers over a single bottleneck link, with or without the assumption of random drops/losses. These works can be divided into two streams; the (chronologically) first stream of work assumes a single bottleneck link that is used to transfer a fixed number of files of very large volumes (see [6–8], and references therein), whereas the second category deals with the performance of TCP-controlled transfer of http-like (finite volume) files where the number of ongoing transfers is randomly varying (see [9], and references therein). An important consideration in the case of http-like traffic is the distribution of file transfer volumes.

Some of the works that fall in the first category attempt to model the behaviour of the link buffer (see [6,10]) but, to our knowledge, there is no such analytical study available for TCP-controlled transfer of http-like traffic. In this paper, we develop a framework for analysing the behaviour of the link buffer, and related processes, assuming that the file transfers are controlled using a general adaptive window protocol, explicitly taking into account the distribution of file transfer volumes.

We consider the scenario shown in Fig. 1, where an Internet link connects clients on one side to servers on the other side. We assume that there is no restriction on the number of simultaneous ongoing transfers. The clients generate file transfer requests and the servers send the requested files using an AWP. The servers and clients are connected to the link by very high-speed access links. Hence, the Internet link is the bottleneck; also shown in the figure is this link's buffer containing data packets from the ongoing file transfers. We make the following system and traffic assumptions:
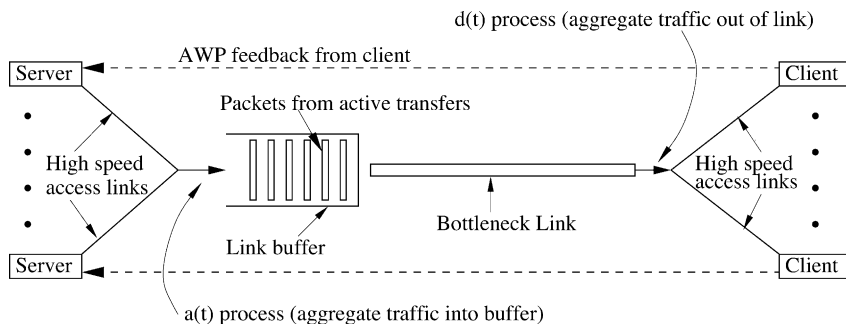
Fig. 1. TCP-controlled file transfers over a single link connecting servers and clients; the link propagation delay is assumed to be negligible, and the link buffer is infinite. In this work, we analyse the $a(t)$ process, and the link buffer occupancy process.

- The end-to-end propagation delay is negligible (in the sense that the propagation delay between the end nodes of the link is much less than one packet service time; for example, this could be a 34 Mbps link interconnecting two locations 15 km apart in a city for a TCP packet size of 1500 bytes, the bandwidth delay product being 0.3 packets).
- The link buffer on the server side is such that there is no packet loss. (It follows that since the file sizes are finite, the window growth is governed solely by the increase phase of the AWP; the window of each transfer remains finite since the volume of the transfer is finite. We wish to study the tail behaviour of the stationary contents of the buffer; such an analysis would provide some insight into the tail drop loss behaviour with finite buffers.)
- The link buffer on the server side implements a per-flow round-robin scheduling discipline with a service quantum of one packet. Examples of such scheme are Deficit Round Robin (DRR, see [11]) and weighted fair queueing (WFQ).
- Each request is for the transfer of a single file, and the files have independent and identically distributed sizes.
- The starting instants of file transfers constitute a Poisson process (of rate λ). (The instants at which new user sessions start is now accepted to be well modelled by a Poisson process (see [4]); our model thus assumes that each session transfers just one file.)
- We first assume that the link buffer does not drop or mark the packets owing to any active queue management mechanisms. This assumption is later relaxed and random marking of packets are also considered.

The first assumption above implies that the link buffer contains all unacknowledged data from the ongoing file transfers (sessions). This also implies that the link is busy whenever at least one session is active.

1.1. Relationship with existing literature

It has been shown [12,13] that for Pareto-distributed file sizes (with tail $\frac{1}{x^\alpha}$), the data departure rate process ($d(t)$ in Fig. 1) is long range-dependent (LRD) with Hurst parameter $\frac{3-\alpha}{2}$. This result follows from the observation that, owing to zero propagation delay, the $d(t)$ process corresponds to the busy idle process of a work conserving queue. Further, $d(t)$ is not affected by the feedback control used. Clearly, however,
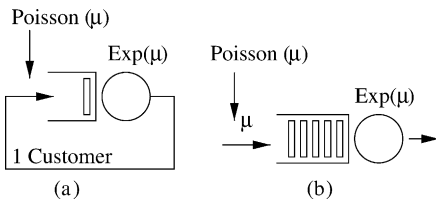
Fig. 2. Two simple queueing models illustrating that for the same input process (Poisson with rate μ) the behaviour of a buffer is entirely different in a closed loop (a) and in an open loop (b). The queue in (b) is unstable (a null recurrent Markov chain).

the input process to the link buffer depends on the feedback control used and hence it is interesting to study the correlation structure of the data arrival rate process into the link buffer (denoted by $a(t)$ in Fig. 1); this is one contribution of the work presented here.

Extensive analysis of Internet data has confirmed that Internet traffic is LRD (see [1]). It has been argued that the LRD behaviour of Internet traffic is related to heavy-tailed file transfer volumes [2]. Recent studies (see [12,14,15]) show that the stationary distribution of a queue fed with LRD traffic will have a non-exponential tail; for example, it has been shown that an arrival rate process auto-covariance that is $O\left(\frac{1}{\tau^{\alpha-1}}\right)$, $1 < \alpha < 2$, leads to a stationary distribution of buffer occupancy that has a tail that is $O\left(\frac{1}{x^{\alpha-1}}\right)$. The above observations are usually combined to conclude that the link buffer occupancies in the Internet will be heavy-tailed. Such observations are, however, based on an "open loop" analysis of an LRD traffic source feeding a buffer. Recent numerical studies [4,16,17] suggest that an understanding of traffic and buffer processes in the Internet should take into account the closed loop nature of Internet congestion control, namely TCP which is an adaptive window protocol. The second contribution of this paper is to carry out such an analysis for the network scenario of Fig. 1 and for a general AWP.

It is easy to see that the behaviour of a buffer for a given input process can be strikingly different in a feedback loop as compared to when the same process is applied to the buffer (i.e., "open loop"). In Fig. 2, we provide a simple example. Fig. 2(a) depicts a closed queueing system where a single customer is fed back to the queue (with a new service requirement distributed as exponential(μ)) as soon as it gets served; the system is clearly stable as there is always a single customer in the system. Note that the customer arrival instants to the queue form a Poisson process of rate μ. Fig. 2(b) depicts an M/M/1 queue with a Poisson arrival process of rate μ, and exponentially distributed service requirement with mean $\frac{1}{\mu}$; this queue is clearly unstable (the queue length process being a null recurrent Markov chain).

It is intuitive that introduction of window reductions due to presence of active queue management scheme at the link buffer would result in a well-behaved buffer occupancy distribution. We study this phenomenon for two specific AWPs and find that the results are in accordance with the intuition.

1.2. Overview and organisation of the paper

Assuming an AWP and a general file size distribution, we study the auto-covariance function of the data arrival rate process into the link buffer (the $a(t)$ process, see Fig. 1). We then analyse the link buffer occupancy process for a general AWP and file size distribution and provide a necessary and a sufficient condition for the existence of the mean buffer occupancy. Combining the results from above two analysis, it is shown that it is possible to have a finite mean link buffer occupancy even when the file size requirements are heavy-tailed and the $a(t)$ process is LRD. This does not contradict the result

of [12,14] as the model analysed there does not include any feedback control from the queue. Next, we consider specific AWPs to study the effect of window reductions owing to random packet markings/drops and find that, as expected, window reductions result in further lightening in the tail of the buffer occupancy distribution while the traffic into the buffer remains LRD.

The paper is organised as follows. In Section 2, we develop a queueing equivalent model of the scenario of Fig. 1, introduce some notation we use in the paper and give some queueing results required later in the work. In Section 3, we introduce some characterising functions associated with an AWP. Section 4 presents a study of the auto-covariance function of the $a(t)$ process. In Section 5, we give the analysis of the link buffer occupancy process. In Section 6, we consider two specific AWPs and study the effect of introducing random marking of packets on the link buffer occupancy process and the $a(t)$ process. Section 7 concludes the paper.

2. Modelling approach

Our model is motivated by the most popular AWP, namely TCP. TCP adapts its transmission window when it receives acknowledgements, detects missing packets, or receives a congestion mark. The window adaptation serves two purposes: (i) to limit the amount of outstanding data (and hence congestion) in the network, and (ii) to promote fair sharing of the bandwidth. Thus, by controlling the window, TCP controls the release of data into the network, and hence the amount of data that stays in the file server. One aspect of our model will capture this feature, i.e., we will model the way the AWP controls the release of data into the buffer and the amount of the file that is left behind on the server.

We also wish to assume that the link's bandwidth is shared equally among the ongoing transfers. Recent literature [9,18] suggests that, *for a zero propagation delay link and even in absence of a per-flow fair scheduling at the link buffer*, in the above situation the TCP mechanism effectively serves the files in a processor sharing (PS) fashion. This suggests that, even if the packets are served in the order of their arrival to the link buffer, the packets from the active files are interlaced in such a manner that the data from these files in the link buffer is served in a round-robin manner. We have observed, however, that such an equal sharing is *not* valid for general file size distributions and breaks down, in particular, for heavy-tailed file volume distributions unless the link buffer implements a per-flow fair queueing (see [19]). Thus, in order to use a processor sharing model to capture the way the link bandwidth is shared among the files being transferred, we need to assume per-flow fair scheduling at the link buffer.

We now describe the stochastic model in detail. File transfer requests arrive in a Poisson process of rate λ to the servers. The transfer volumes are independently and identically distributed with common distribution $V(\cdot)$. An AWP is used to carry out the file transfers. The round trip propagation delay is zero, hence as soon as the link transmits some data for a file, this data is acknowledged to the AWP transmitter at the file server which releases more data (if any) into the buffer. It follows that at each point of time every unfinished transfer has a positive amount of data in the buffer.

The release of data into the buffer is governed by the way the window of the AWP increases with acknowledgements. The window of an ongoing transfer is a function only of the amount of data that it has transferred. The zero propagation delay assumption implies that the contents of the link buffer are the window of the transfer or the residual amount of the file (not yet transferred), whichever is smaller. These windows are served in a round-robin manner as per the assumption of per-flow fair scheduling at the link buffer. The round-robin discipline is simpler to study via the processor sharing model, hence we
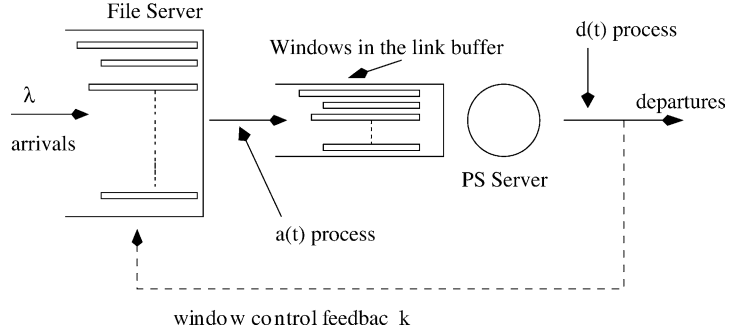
Fig. 3. Queueing equivalent of the network scenario of Fig. 1.

approximate the service of the windows in the link buffer by a PS discipline. Based on this approximation, Fig. 3 depicts the queueing equivalent of the scenario shown in Fig. 1; note that the link buffer has been replaced by a PS server. As each active flow has positive amount of outstanding data (window) in the link buffer and since these windows are served in a PS fashion, assuming that the files are composed of infinitely divisible fluid, *it follows that the ongoing file transfers (as a whole) also get service in a PS manner*.

Note that at any time instant t, an active session would have successfully transferred some data to its client, some of its data would be in the link buffer (this would be the current window size of AWP controlling the transfer of the file), and the remaining data would be in the server waiting to be transferred. At any time instant, we use the term "age" for the amount of data of a file successfully transferred to the client, and by "residual file" or "residual service" we mean the amount of data of the file yet to be transferred, i.e., the sum of the data in the link buffer and that remaining in the server. As data from a file is served (along with other files, in a PS manner), more data is pulled into the link buffer from the file server, so as to replenish the windows and to account for any window growth. Eventually, the amount of data of the file in the server reduces to zero, and the entire residual file is in the link buffer. Note that as long as the file is not fully transferred, a positive amount of it is always in the link buffer. Thus, in term of the PS queue model, the server is the link, and each "customer" in service is split between the file server and that in the link buffer.

Note that, as the files are served in PS fashion irrespective of the AWP used, the evolution of "age" (and hence the residual file) is also independent of the AWP used. At any instant, the AWP used to transfer a file *only* determines the splitting of the *residual file* between the server and the link buffer while the active sessions are still served in a PS manner. Fig. 4 shows the distribution, among the sender and the link buffer, of the data of a file of size v that has transferred u amount of data to the client (i.e., has attained an age u); the AWP used determines only $w(u, v)$ which gives the breakup of the remaining $v - u$ for the link buffer and the sender. To elaborate on this, for a particular fixed sequence of interarrival times and file sizes, the vector valued process of the ages and residual file of the (time varying) active transfers remains the same irrespective of the AWP used to transfer these files but the vector valued process corresponding to the amount of data in link buffer of the individual active transfers will depend on the AWP used. This point will become more clear in Section 5, where we characterise AWPs based on the way they split the file between server and link buffer and thus make clear the distinction between the data of a file in the server and the link buffer.
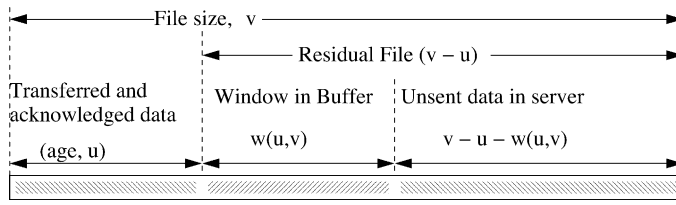
Fig. 4. Figure showing the breakup of the data of a file of size v between the server, the link buffer and the client.

The above general model, which was motivated by observations about the performance of TCP-controlled finite volume transfers, is what we work with in this paper. We will show how to apply it to specific cases of TCP type adaptive window control.

2.1. Notation and some results related to an M/G/1 PS queue

We follow the convention that if Z is a random variable then EZ is its expectation, $Z(\cdot)$ denotes its cumulative distribution function (cdf), $Z^c(\cdot)$ its complementary cdf, $z(\cdot)$ its density (if it exists), $\tilde{z}(s)$ the Laplace–Stieltjes transform (LST) of $Z(\cdot)$, and $\tilde{Z}(s)$ the Laplace transform (LT) of $Z(\cdot)$. We also let $Z_e(\cdot)$ denote the excess distribution of $Z(\cdot)$, and $Z_s(\cdot)$ denotes the spread distribution associated with Z (see [20]). In the context of a queueing system, with the above-mentioned convention, we introduce the following notation:

V, the random variable for the file sizes brought by sessions;
$\rho := \lambda EV$;
$a(t)$, the instantaneous data arrival rate into the queue at time t;
$N(t)$, number of sessions active at time t;
$Y(t)$, total of the residual file sizes at time t;
B, the busy period random variable of an M/G/1 queue;
$x(s) := 1 - \tilde{b}(s)$ (introduced for notational convenience);
B_y, the busy period random variable with initial 'work' y in an M/G/1 queue;
$K_y(t)$, the number of starts of idle periods until time t given $Y(0) = y$.

We know that (see [20])

$$\tilde{b}_y(s) = \mathrm{e}^{-y(s+\lambda-\lambda\tilde{b}(s))} = \mathrm{e}^{-y(s+\lambda x(s))} \tag{1}$$

$$\tilde{b}(s) = \tilde{v}(s + \lambda x(s)) \tag{2}$$

We use the notation $f(t) \sim_{t \to t_0} g(t)$ to mean $\lim_{t \to t_0} \frac{f(t)}{g(t)} = 1$ and write $f(t) \succeq_{t \to t_0} g(t)$ to mean that there exists a function $h(t)$ such that $f(t) \geq h(t)$ for all t and $h(t) \sim_{t \to t_0} g(t)$.

In this work, we frequently use the following known results for a *stationary* M/G/1 PS queue (see [21]). At any time instant t:

- $P\{N(t) = n\} = (1 - \rho)\rho^n$.

- The total file sizes of each of the $N(t)$ ongoing transfers at time t are mutually independent random variables and are distributed as $V_s(\cdot)$, $v_s(x) = \frac{xv(x)}{EV}$, (see [20]).
- Conditioned on the total service requirement of a file transfer being v, its age[1] is uniformly distributed over the interval $[0, v]$.

A recent work [22] reports the following result:

- In the M/G/1 PS system modified by having k permanent jobs with infinite service requirements, for $n \geq k$,

$$\lim_{t\to\infty} P\{N(t) = n\} = (1 - \rho)^{k+1} \binom{n-k}{k} \rho^n \tag{3}$$

3. Characterisation of a general AWP

An AWP can be characterised by the amount of data released by the sender (server) in response to a unit amount of acknowledged data. In general, this quantity will be a function of the size of the file being transferred and the total amount of data successfully received by the client. We introduce the following notation in the context of a general AWP.

Notation:

$R_v(u)$ is the amount of data released by the sender per unit of acknowledged data when a file of size v has attained age u (i.e., u amount of its data has been acknowledged).

$U(v) = \sup\{u : R_v(u) > 0, 0 \leq u \leq v\}$ is the age of a file (of size v) after which an acknowledgment does not result in release of data from file server, i.e., at this time the file server has sent a total of v amount of data to the link buffer. At this point, the receiver would have acknowledged $U(v)$ amount of data and $v - U(v)$ amount of data from the file will be in the link buffer.

$X(u) = \inf\{v : U(v) > u, 0 \leq v < \infty\}$ is the minimum file size for which the protocol will still be sending data to the buffer after it has received acknowledgment of u amount of data.

Note that,

$$\int_{u=0}^{U(v)} R_v(u)\,\mathrm{d}u = v \tag{4}$$

$$U(X(u)) = u, \qquad X(U(v)) = v \tag{5}$$

Thus, U and X are inverse functions.

Call TCP-CA the TCP algorithm with initial slow start threshold set to unity, i.e., the protocol starts with the congestion avoidance phase. Also let TCP-SS denote the TCP algorithm with initial slow start threshold set to infinity, i.e., the protocol starts with and remains in the slow start phase. Note that if no loss occurs, as is the case with an infinite link buffer, TCP-SS is always in the slow start phase and TCP-CA is always in the congestion avoidance phase.

---

[1] The data already transferred to the client by the session by time t.

3.1. The $R_v(u)$, $U(v)$ and $X(u)$ functions for TCP-SS

If the transfer of a file is controlled using only the slow start algorithm of TCP then each unit of data acknowledged results in the arrival of two units of data, thus

$$R_v(u) = 2I_{\{u \leq U(v)\}}$$

Using Eq. (4) with $R_v(u) = 2I_{\{u \leq U(v)\}}$, we get

$$U(v) = \frac{v}{2}$$

Using Eq. (5) with $U(v) = \frac{v}{2}$, we get

$$X(u) = 2u$$

3.2. The $R_v(u)$, $U(v)$ and $X(u)$ functions for TCP-CA

If the transfer of a file is controlled using only the congestion avoidance algorithm of TCP then, when window size is w, each unit of data acknowledged results in arrival of $1 + \frac{1}{w}$ units of data. Also, when the window size is some integer n, the amount of data that has been acknowledged (i.e., the age of file) is $\frac{n(n-1)}{2}$. Thus, the window size corresponding to an age of u is $\frac{-1+\sqrt{1+8u}}{2}$. To avoid complex expressions, since we are interested in asymptotic behaviour, we use the approximation that when the age of file is u the window size is $\sqrt{2u}$. This gives

$$R_v(u) = \left(1 + \frac{1}{\sqrt{2u}}\right) I_{\{u \leq U(v)\}}$$

Using Eq. (4) with $R_v(u) = \left(1 + \frac{1}{\sqrt{2u}}\right) I_{\{u \leq U(v)\}}$, we get

$$U(v) = v + 1 - \sqrt{1 + 2v}$$

Using Eq. (5) with $U(v) = v + 1 - \sqrt{1 + 2v}$, we get

$$X(u) = u + \sqrt{2u}$$

4. Asymptotic behaviour of the auto-covariance function of the $a(t)$ process

In this section, we study the auto-covariance function of the $a(t)$ process (see Fig. 3) when the transfer of files is controlled by a AWP.

When there are n active sessions, owing to the PS model, a unit data served by the link implies each of these n sessions gets a service of $\frac{1}{n}$ data units hence their total data sending rate at instant t will be

$$a(t) = \sum_{i=1}^{N(t)} \frac{R_{v_i}(u_i(t))}{N(t)} \tag{6}$$

where $u_i(t)$ and v_i are, respectively, the total service received by and the total service requirement of ith session active at time t.

Lemma 4.1. *For the stationary system,*

$$Ea(t) = \rho$$

Proof. By conditioning on $N(t)$ and the file size requirements (v_i) of the $N(t)$ ongoing transfers and their ages (u_i), using Eq. (6) and results of Section 2.1,

$$Ea(t) = \sum_{n=1}^{\infty}(1-\rho)\rho^n \int_{v_1=0}^{\infty} \cdots \int_{v_n=0}^{\infty} \int_{u_1=0}^{v_1} \cdots \int_{u_n=0}^{v_n} \left[\sum_{i=1}^{n} \frac{R_{v_i}(u_i(t))}{n} \right] \frac{du_n}{v_n} \cdots$$

$$\times \frac{du_1}{v_1}\, dV_s(v_n) \cdots dV_s(v_1)$$

Since the integrand in square brackets above is linear and symmetric with respect to the indices i, we get

$$Ea(t) = \sum_{n=1}^{\infty}(1-\rho)\rho^n \int_{v_1=0}^{\infty} \cdots \int_{v_n=0}^{\infty} \int_{u_1=0}^{v_1} \cdots \int_{u_n=0}^{v_n} R_{v_1}(u_1(t)) \frac{du_n}{v_n} \cdots \frac{du_1}{v_1}\, dV_s(v_n) \cdots dV_s(v_1)$$

$$= \sum_{n=1}^{\infty}(1-\rho)\rho^n \int_{v_1=0}^{\infty} \int_{u_1=0}^{v_1} R_{v_1}(u_1(t))\, du_1 \frac{dV_s(v_1)}{v_1}$$

Using Eq. (4),

$$Ea(t) = \sum_{n=1}^{\infty}(1-\rho)\rho^n \int_{v_1=0}^{\infty} v_1 \frac{dV_s(v_1)}{v_1} = \rho \quad \square$$

We note that this result is as would be expected; for let $A(t)$ denote the cumulative process for the rate process $a(t)$ (i.e., $A(t) = \int_{u=0}^{t} a(u)\, du$). Then, for the stable system,

$$\lim_{t \to \infty} \frac{A(t)}{t} = \frac{\lambda EV}{c} = \rho$$

(we have taken $c = 1$ packet/s).

The auto-covariance function of the $a(t)$ process for a lag of τ is given by

$$r_a(\tau) := Ea(0)a(\tau) - Ea(0)Ea(t) = Ea(0)a(\tau) - \rho^2$$

$$= Ea(0)a(\tau)I_{\{N(0)>0,\, K_{Y(0)}(\tau)=0\}} + Ea(0)a(\tau)I_{\{N(0)>0,\, K_{Y(0)}(\tau)>0\}} - \rho^2 =: J_1(\tau) + J_2(\tau) - \rho^2 \tag{7}$$

where $K_y(\cdot)$ and $Y(t)$ are as defined in Section 2.1. We study the asymptotic behaviour of $r_a(\tau)$ by considering $J_1(\tau)$ and $J_2(\tau)$ separately.

Theorem 4.1. *For AWP-controlled transfer of file sizes of distribution $V(\cdot)$, if there exists a r such that $0 < r \leq \inf_{0 \leq v < \infty} \inf_{0 \leq u < U(v)} R_v(u)$, then*

$$J_1(\tau) \succeq_{\tau \to \infty} (1 - \rho)^2 \lambda r^2 \int_{v=X(\tau)}^{\infty} \int_{u=0}^{U(v)-\tau} du \, dV(v)$$

Proof. Let $V_i(t)$ and $Y_i(t)$, $1 \leq i \leq N(t)$, denote, respectively, the total and residual service requirements of the ith session active at time t. Now,

$$J_1(\tau) = Ea(0)a(\tau)I_{\{N(0)>0, K_{Y(0)}(\tau)=0\}} \geq Ea(0)a(\tau)I_{\{N(0)=1, U(V_1(0))-(V_1(0)-Y_1(0))>\tau\}}$$

which holds because when $N(0) = 1$, $V_1(0) - Y_1(0)$ is the age of the file active at time 0, and $U(V_1(0)) - (V_1(0) - Y_1(0)) > \tau$ implies that the source will still be sending data for this file until time τ. Define the last term to be $J_3(\tau)$.

Note that $\{U(V_1(0)) - (V_1(0) - Y_1(0)) > \tau\} \subset \{V_1(0) > X(\tau)\}$, hence

$$J_3(\tau) = Ea(0)a(\tau)I_{\{N(0)=1, V_1(0)>X(\tau), Y_1(0)>V_1(0)-U(V_1(0))+\tau\}} \tag{8}$$

Plugging the distributions of $N(0)$, $V_1(0)$ and $Y_1(0)$ using results given in Section 2.1, we get from Eq. (8),

$$J_3(\tau) = (1 - \rho)\rho \int_{v=X(\tau)}^{\infty} \int_{y=v-U(v)+\tau}^{v} a(0)E(a(\tau)|N(0) = 1, V_1(0) = v, Y(0) = y)\frac{dy}{v} \, dV_s(v)$$

At τ, there could be other sessions active; these arrive in the interval $(0, \tau)$. Let us continue to use the index 1 at time τ for the session that was active at time 0. Since $Y_1(\tau) > 0$, (and noting that $V_1(\tau) = V_1(0)$)

$$a(\tau) \geq \frac{R_{V_1(\tau)}(V_1(\tau) - Y_1(\tau))}{N(\tau)} = \frac{R_{V_1(0)}(V_1(\tau) - Y_1(\tau))}{N'(\tau) + 1}$$

where $N'(\tau)$ is the number of sessions active at τ *other* than the tagged session which was active at time 0. The inequality is obtained since there could be a positive rate from the other sessions at τ.

By hypothesis, we have

$$0 < r \leq \inf_{0 \leq v < \infty} \inf_{0 \leq u < U(v)} R_v(u)$$

Now, since $Y_1(\tau) > V_1(\tau) - U(V_1(\tau))$, by definition of r we have $R_{V_1(0)}(V_1(\tau) - Y_1(\tau)) \geq r$, hence

$$a(\tau) \geq \frac{r}{N'(\tau) + 1}$$

so, for $y > \tau$,

$$E(a(\tau)|N(0) = 1, V_1(0) = v, Y(0) = y) \geq rE\left[\frac{1}{N'(\tau) + 1}|N(0) = 1, V_1(0) = v, Y(0) = y\right]$$

also, since $V_1(0) - Y_1(0) < U(V_1(0))$,

$$a(0) = R_{V_1(0)}(V_1(0) - Y_1(0)) \geq r$$

Hence, using above inequalities and that $dV_s(v) = \frac{v\,dV(v)}{EV}$,

$$J_3(\tau) \geq (1 - \rho)\lambda r^2 \int_{v=X(\tau)}^{\infty} \int_{y=v-U(v)+\tau}^{v} E\left[\frac{1}{N'(\tau) + 1}|N(0) = 1, V_1(0) = v, Y(0) = y\right] dy\,dV(v)$$

Note that the conditions $N(0) = 1$, $V_1(0) = v$ and $Y(0) = y > \tau$ together imply that the file that was present at time 0 is also present at time τ, and hence owing to the PS model $N'(\tau)$ is independent of v and y. Hence,

$$J_3(\tau) \geq (1 - \rho)\lambda r^2 E\left[\frac{1}{N'(\tau) + 1}|N(0) = 1, Y(0) > \tau\right] \int_{v=X(\tau)}^{\infty} \int_{y=v-U(v)+\tau}^{v} dy\,dV(v)$$

We know that (see Eq. (3) in Section 2.1 with $k = 1$)

$$P\{N'(\tau) = n|N(0) = 1, Y(0) > \tau\} \sim_{\tau \to \infty} (1 - \rho)^2(n + 1)\rho^n$$

Hence,

$$E\left[\frac{1}{N'(\tau) + 1}|N(0) = 1, Y(0) > \tau\right] \sim_{\tau \to \infty} (1 - \rho)$$

The proof follows with a change of variable (using $y = v - u$). $\quad\square$

Pareto-distributed file sizes have the following distribution

$$V^c(x) = \min\left(1, \frac{1}{x^\alpha}\right)$$

with the property that EV is finite iff $\alpha > 1$ and EV^2 is finite iff $\alpha > 2$.

Corollary 4.1. *For Pareto-distributed file sizes transferred using TCP-SS,*

$$J_1(\tau) \succeq_{\tau \to \infty} 4(1 - \rho)^2 \frac{\lambda}{\alpha - 1} \frac{1}{2^\alpha \tau^{\alpha - 1}}$$

Proof. For TCP-SS, $R_v(u) = 2I_{\{u < U(v)\}}$, $U(v) = \frac{v}{2}$ and $X(\tau) = 2\tau$. Use Theorem 4.1 with $r = 2$. $\quad\square$

Corollary 4.2. *For Pareto-distributed file sizes transferred using TCP-CA,*

$$J_1(\tau) \succeq_{\tau \to \infty} (1 - \rho)^2 \frac{\lambda}{\alpha - 1} \frac{1}{\tau^{\alpha - 1}}$$

Proof. For TCP-CA, $R_v(u) = (1 + \frac{1}{\sqrt{2u}})I_{\{u < U(v)\}}$, $U(v) = v + 1 - \sqrt{1 + 2v}$ and $X(u) = u + \sqrt{2u}$. Now,

$$U(v) = (v + 1 - \sqrt{1 + 2v}) = \frac{(\sqrt{1 + 2v} - 1)^2}{2} > \frac{(\sqrt{2v} - 1)^2}{2} = \frac{2v + 1 - 2\sqrt{2v}}{2} > v - \sqrt{2v}$$

Hence, from Theorem 4.1 with $r = 1$,

$$J_1(\tau) \succeq_{\tau \to \infty} (1 - \rho)^2 \lambda \left[\int_{v = \tau + \sqrt{2\tau}}^{\infty} (v - \sqrt{2v}) \, dV(v) - \tau V^c(\tau + \sqrt{2\tau}) \right]$$

$$= (1 - \rho)^2 \lambda \left[\frac{EV}{(\tau + \sqrt{2\tau})^{\alpha - 1}} - \frac{\alpha/(\alpha - 0.5)}{(\tau + \sqrt{2\tau})^{\alpha - 0.5}} - \frac{\tau}{(\tau + \sqrt{2\tau})^{\alpha}} \right] \sim_{\tau \to \infty} (1 - \rho)^2 \frac{\lambda}{\alpha - 1} \frac{1}{\tau^{\alpha - 1}}$$

The proof follows by noting that $f(t) \succeq_{t \to \infty} g(t)$ and $g(t) \sim_{t \to \infty} h(t)$ implies $f(t) \succeq_{\tau \to \infty} h(t)$. □

As is evident from the examples of slow start and congestion avoidance given in Section 3, the $R_v(u)$ function for an AWP, in general, will be of the form $R_\infty(u)I_{\{0 \le u \le U(v)\}}$ so that $\inf_{0 \le v < \infty} \inf_{0 \le u < U(v)} R_v(u) = \inf_{u \ge 0} R_\infty(u)$, where $R_\infty(u)$ is the data sending rate function of a file of infinite volume. If in addition, $R_v(u)I_{\{u \le U(v)\}}$ is bounded, i.e., $R_v(u)I_{\{u \le U(v)\}} \le R$ for some $\infty > R \ge r$, we have the following corollary of Theorem 4.1.

Corollary 4.3. *If an AWP is such that $0 < r \le R_v(u)I_{\{u \le U(v)\}} \le R < \infty$ for $v, u \ge 0$ then,*

$$J_1(\tau) \succeq_{\tau \to \infty} (1 - \rho)^2 \lambda r^2 \left[\int_{v = R\tau}^{\infty} \frac{v}{R} \, dV(v) - \tau V^c(R\tau) \right]$$

Proof. Since $R_v(u)I_{\{u \le U(v)\}} \le R$, we get from Eq. (4),

$$v = \int_{u = 0}^{U(v)} R_v(u) \, du \le RU(v)$$

Thus, $U(v) \ge \frac{v}{R}$. From Eq. (5),

$$X(u) = \int_0^u R_{X(u)}(x) \, dx \le Ru$$

Thus, $X(u) \le Ru$. The proof follows by using above bounds for $U(v)$ and $X(u)$ to further bound the lower bound of $J_1(\tau)$ in Theorem 4.1. □

Corollary 4.4. *If an AWP, such that $0 < r \leq R_v(u)I_{\{u \leq U(v)\}} \leq R < \infty$, is used for transfer of Pareto-distributed files then,*

$$J_1(\tau) \succeq_{\tau \to \infty} \frac{(1-\rho)^2 \lambda r^2}{(\alpha - 1)R^\alpha \tau^{\alpha - 1}}$$

Proof. Follows from Corollary 4.3. □

Theorem 4.2. *For AWP-controlled transfer of Pareto-distributed file sizes,*

$$\rho^2 - J_2(\tau) \rightleftharpoons \frac{\rho^2}{s} - \tilde{J}_2(s) \sim_{s \to 0} \rho \lambda \frac{\alpha^2 \Gamma(-\alpha)}{(1-\rho)^{\alpha-1} s^{2-\alpha}}$$

Proof. We condition as follows:

(1) condition on $N(0)$, the number of ongoing transfers at time 0;
(2) given $N(0)$, condition on the total file size $V_i(0)$ of ith ongoing transfer at time 0, $1 \leq i \leq N(0)$;
(3) given $N(0)$ and $V_j(0)$, $1 \leq j \leq N(0)$, condition on $Y_i(0)$, the residual size of the ith, $1 \leq i \leq N(0)$ file having total size $V_i(0)$;
(4) given $Y(0) = \sum_{i=1}^{N(0)} Y_i(0)$, the total residual file volumes at time 0, condition on θ, $0 < \theta < \tau$, the start of the first idle period after time 0.

This yields

$$J_2(\tau) = \sum_{n=1}^{\infty} (1-\rho)\rho^n \int_{v(n)=0}^{\infty} \int_{y(n)=0}^{v(n)} a(0)$$

$$\times \int_{\theta=0}^{\tau} E(a(\tau)|N(\theta) = 0, N(\theta^-) > 0)b_{\sum_{i=1}^{n} y_i}(\theta)\, d\theta \frac{dy(n)}{v(n)}\, dV_s(v(n))$$

where $v(n)$ is a row vector of dimension n with components $v_1, \ldots, v_n$ and $y(n)$ is a row vector of dimension n with components $y_1, \ldots, y_n$. Also, abusing notation, $\frac{dy(n)}{v(n)} := \frac{dy_n}{v_n} \cdots \frac{dy_1}{v_1}$ and $dV_s(v(n)) := dV_s(v_n) \cdots dV_s(v_1)$, and $b_y(\cdot)$ is the density of B_y.

Due to the Poisson arrival assumption

$$E(a(\tau)|N(\theta) = 0, N(\theta^-) > 0) = E(a(\tau)|N(\theta) = 0)$$

is a function only of $\tau - \theta$.

Let $\Phi(\tau) := E(a(\tau)|N(0) = 0)$ and denote by $\tilde{\Phi}(s)$ the Laplace transform of $\Phi(\cdot)$. So,

$$E(a(\tau)|N(\theta) = 0, N(\theta^-) > 0) = \Phi(\tau - \theta)$$

Thus,

$$J_2(\tau) = \sum_{n=1}^{\infty} (1-\rho)\rho^n \int_{v(n)=0}^{\infty} \int_{y(n)=0}^{v(n)} a(0) \int_{\theta=0}^{\tau} \Phi(\tau - \theta)b_{\sum_{i=1}^{n} y_i}(\theta)\, d\theta \frac{dy(n)}{v(n)}\, dV_s(v(n)) \qquad (9)$$

Taking the Laplace transform of Eq. (9) and noting that:

- $dV_s(v) = \frac{v\,dV(v)}{EV}$,
- $a(0) = \frac{\sum_{j=1}^n R_{v_j}(v_j - y_j)}{n}$,
- the integral with respect to θ is a convolution of two terms which have LT given by $\tilde{\Phi}(s)$ and $\tilde{b}_{\sum_{i=1}^n y_i}(s)$, and
- $\tilde{b}_{\sum_{i=1}^n y_i}(s) = \prod_{i=1}^n e^{-y_i(s+\lambda x(s))}$,

we get the following expression for the LT of $J_2(\tau)$

$$\tilde{J}_2(s) = \frac{s\tilde{\Phi}(s)}{s} \sum_{n=1}^\infty (1-\rho)\lambda^n \int_{v(n)=0}^\infty \int_{y(n)=0}^{v(n)} \frac{\sum_{j=1}^n R_{v_j}(v_j - y_j)}{n} \prod_{i=1}^n e^{-y_i(s+\lambda x(s))}\,dy(n)\,dV(v(n))$$

It can be seen after some calculations that

$$\tilde{J}_2(s) = (1-\rho)\frac{s\tilde{\Phi}(s)}{s}\lambda \left[\int_{v=0}^\infty e^{-v(s+\lambda x(s))} \int_{u=0}^v e^{u(s+\lambda x(s))} R_v(u)\,du\,dV(v)\right] \frac{s+\lambda x(s)}{s}$$

We know that $\lim_{s\to 0} s\tilde{\Phi}(s) = \rho$. Let $s\tilde{\Phi}(s) = \rho + \kappa(s)$, where $\kappa(s) \to 0$ as $s \to 0$. Also, $\lim_{s\to 0} \int_{u=0}^v e^{u(s+\lambda x(s))} R_v(u)\,du = v$ so, $\int_{u=0}^v e^{u(s+\lambda x(s))} R_v(u)\,du = v + \nabla(s)$ where $\lim_{s\to 0} \nabla(s) = 0$. Thus,

$$\int_{v=0}^\infty e^{-v(s+\lambda x(s))} \int_{u=0}^v e^{u(s+\lambda x(s))} R_v(u)\,du\,dV(v) \sim_{s\to 0} \int_{v=0}^\infty e^{-v(s+\lambda x(s))} v\,dV(v) = -\frac{d}{dz}\tilde{v}(z)\big|_{z=s+\lambda x(s)}$$

For Pareto distribution, $V^c(v) = \min(1, v^\alpha)$, hence

$$\tilde{v}(z) = \alpha\, z^\alpha\, \Gamma(-\alpha, z)$$

where, $\Gamma(\cdot, \cdot)$ is the incomplete Gamma function.[2] Further, by using the series expansion of the incomplete Gamma function (see [23]), we get

$$\tilde{v}(z) = \alpha\, z^\alpha \left\{\Gamma(-\alpha) - \sum_{n=0}^\infty \frac{(-1)^n z^{-\alpha+n}}{n!(-\alpha+n)}\right\} = \alpha\, z^\alpha\, \Gamma(-\alpha) - \alpha \sum_{n=0}^\infty \frac{(-1)^n z^n}{n!(-\alpha+n)}$$

It follows that,

$$\int_{v=0}^\infty e^{-v(s+\lambda x(s))} \int_{u=0}^v e^{u(s+\lambda x(s))} R_v(u)\,du\,dV(v) \sim_{s\to 0} \frac{\alpha}{\alpha-1} - \alpha^2\Gamma(-\alpha)(s+\lambda x(s))^{\alpha-1}$$

---

[2] $\Gamma(a, y) = \int_y^\infty \exp^{-u} u^{a-1}\,du$.

Hence,

$$\tilde{J}_2(s) \sim_{s \to 0} (1 - \rho)s\tilde{\Phi}(s)\lambda \frac{s + \lambda x(s)}{s} \frac{[(\alpha/(\alpha - 1)) - \alpha^2 \Gamma(-\alpha)(s + \lambda x(s))^{\alpha - 1}]}{s}$$

The proof follows by noting that:

(1) For $s \to 0$, $x(s) = sEB + o(s) = \frac{sEV}{1 - \rho} + o(s)$.
(2) $s\tilde{\Phi}(s) \sim_{s \to 0} \rho$.
(3) $EV = \frac{\alpha}{\alpha - 1}$. □

Corollary 4.5. *For AWP-controlled transfer of Pareto-distributed file sizes with $\alpha > 1$,*

$$\rho^2 - J_2(\tau) \sim_{\tau \to \infty} \frac{\alpha^2 \Gamma(-\alpha)}{(1 - \rho)^{\alpha - 1}} \frac{2 - \alpha}{\Gamma(3 - \alpha)} \frac{\rho\lambda}{\tau^{\alpha - 1}}$$

Proof. Follows using Theorem 4.2 and a Tauberian theorem from [13]. □

Note that the result of Theorem 4.2 (and hence of Corollary 4.5) are independent of the AWP used. This also gives the convergence rate in the Key Renewal Theorem [20].

Theorem 4.3. *For transfer of Pareto-distributed file sizes using TCP-SS or TCP-CA,*

$$r_a(\tau) \succeq_{\tau \to \infty} \Delta(\lambda, \alpha) \frac{1}{\tau^{\alpha - 1}}$$

For some function $\Delta(\lambda, \alpha)$ that depends on whether slow start or congestion avoidance is used to transfer the files.

Further, there exists a $\lambda^* > 0$ such that $\Delta(\lambda, \alpha) > 0$ for all $\lambda < \lambda^*$.

Proof. Using Corollaries 4.1, 4.2 and 4.5,

$$r_a(\tau) = J_1(\tau) + J_2(\tau) - \rho^2 \succeq_{\tau \to \infty} \left[\frac{\delta(1 - \rho)^2}{\alpha - 1} - \rho \frac{\alpha^2 \Gamma(-\alpha)}{(1 - \rho)^{\alpha - 1}} \frac{2 - \alpha}{\Gamma(3 - \alpha)} \right] \frac{\lambda}{\tau^{\alpha - 1}} = \Delta(\lambda, \alpha) \frac{1}{\tau^{\alpha - 1}}$$

where $\delta = 1$ for congestion avoidance and $\delta = 2^{2-\alpha}$ for slow start. Note that in both cases δ is independent of λ. Now, for fixed $\alpha > 1$ and $\rho < 1$, the second term in square brackets above is always positive and decreases to 0 as $\lambda \to 0$ while the first term increases to δ (a positive number) as $\lambda \to 0$. Since both these terms are continuous functions of λ, there exists $\lambda^* > 0$ such that $\frac{\Delta(\lambda, \alpha)}{\lambda} > 0$ for all $\lambda < \lambda^*$. □

It follows from Theorem 4.3 that, for $1 < \alpha < 2$ and $\lambda < \lambda^*$, the process $a(t)$ is LRD. Further, if it can be shown that $r_a(\tau)$ has a hyperbolic decay then the $a(t)$ process will be LRD with Hurst parameter $H \geq \frac{3 - \alpha}{2}$.

Theorem 4.4. *For transfer of Pareto-distributed file sizes using an AWP such that $0 < r \leq R_v(u)I_{\{u \leq U(v)\}} \leq R < \infty$, for $v, u \geq 0$,*

$$r_a(\tau) \succeq_{\tau \to \infty} \Delta(r, R, \lambda, \alpha) \frac{1}{\tau^{\alpha-1}}$$

For some function $\Delta(r, R, \lambda, \alpha)$ for which there exists a $\lambda^* > 0$ such that $\Delta(r, R, \lambda, \alpha) > 0$ for all $\lambda < \lambda^*$.

Proof. Follows from Corollaries 4.4 and 4.5 and arguments similar to those in Theorem 4.3. □

Remark. Theorem 4.4 is of particular importance for the cases where the window of the AWP is bounded so that after some age (at which the maximum allowed window is achieved) an acknowledgement of du data result in exactly du to be released by sender thus keeping the window size fixed and hence[3] $R_\infty(u) = 1$. For example, if the AWP is TCP with initial slow start threshold set to a value larger than the maximum allowed window then, for an infinitely long file, the sender will send $2du$ amount of data for each acknowledgement of du amount of data,[4] i.e., $R_\infty(u) = 2$ till the point at which the window size becomes equal to the maximum allowed window and from this age onwards, du amount of data acknowledgement will result in du amount of data released from the transmitter so as to maintain a constant window and hence $R_\infty(u) = 1$.

5. Analysis of the stationary link buffer process

In this section, we present the analysis for obtaining the distribution of the link buffer content process. An explicit expression for the LST of the distribution of the buffer content process is obtained in terms of the file size distribution and the quantities associated with an AWP (see Eqs. (11) and (13)). The structure of the LST thus obtained is used to find a necessary and a sufficient condition for the existence of the mean of the buffer content process.

5.1. Important observations

- The amount of data in the link buffer at any time t is the sum of the windows from all the file transfers ongoing at t.
- Owing to the infinite buffer assumption there is no loss hence an AWP follows a known window increase schedule as a function of the age of the file transfer. This enables us to determine the window (which is also the session's contribution to the link buffer occupancy) for a given age.

---

[3] At this point, it is important to see the relation between the $R_\infty(u)$ function and the window size as a function of age. If $w(u)$ denotes the window size of an infinitely long file then the following relations hold

$$w(u) = w(0) + \int_{x=0}^{u} R_\infty(x)\,dx$$

$$R_\infty(u) = 1 + \frac{dw(u)}{du}$$

Thus, if the maximum allowed window is attained at age u^*, then $\frac{dw(u)}{du} = 0$ for all $u \geq u^*$.
[4] And hence will be increasing the window size by du.

Table 1
The values of $w(n)$ and $W(n)$ for TCP's congestion avoidance and slow start phases

Phase of TCP	$w(n)$ (packets)	$W(n)$ (packets)
TCP-CA	$n + 1$	$\dfrac{n(n + 1)}{2}$
TCP-SS	2^n	$2^n - 1$

The age of an ongoing transfer can be obtained in the following way: the stationary distribution of the number of ongoing transfers $N(t)$ is given by $P\{N(t) = n\} = (1 - \rho)\rho^n$. Conditioned on the number of ongoing transfers at t, the ages of the various ongoing transfers are independent; further, the age of an ongoing transfer is uniformly distributed in the interval $[0, v]$ where v is its total file transfer size (which has distribution $V_s(\cdot)$, see Section 2.1).

5.1.1. Analysis for the buffer content process

Consider an AWP that starts transmission with an initial window of $w(0)$ packets, i.e., on connection startup the server sends $w(0)$ packets into the buffer. Let $w(1)$ be the window after these $w(0)$ packets have been transmitted by the link and acknowledged. For example, in TCP-SS, $w(0) = 1$ and $w(1) = 2$. Generalising, let $w(n)$, $n \geq 1$, denote the window size just after $\sum_{i=0}^{n-1} w(i)$ amount of data of the file is acknowledged. For example, in TCP-SS, $w(0) = 1$, $w(1) = 2$, $w(2) = 4, \ldots, w(n) = 2^n$.

Let the period where the nth window of the sequence $\{w(i), i \geq 0\}$ is getting served be called the $(n + 1)$th cycle. Thus $w(n - 1)$ is the window at the start of the nth cycle and is also the amount of data served in |it nth cycle. The $(n + 1)$th cycle starts when $w(n - 1)$ amount of data is served after start of nth cycle. Let $W(n) := \sum_{i=0}^{n-1} w(i)$ denote the amount of data acknowledged until the start of $(n + 1)$th cycle (see Fig. 5). Note that $W(0) = 0$, by definition. The values of $w(n)$ and $W(n)$ for TCP-CA and TCP-SS are listed in Table 1.

Note that the $w(n)$, $n \geq 0$, are the possible window sizes at the beginnings of cycles; during the cycles the window sequences could pass through other integer values. For example, in TCP-SS, the TCP window can take all integer values but $w(n)$ is restricted to integral powers of 2.

Let $\gamma_m(z)$ denote the *net* input rate into the link buffer from an infinitely long session that has received z amount of service in the $(m + 1)$th window. For TCP-SS, $\gamma_m(z) = 1$ because every dz amount served brings in $2dz$ and dz amount leaves the link buffer. Note that the definition of $\gamma_m(z)$ does not depend on file size as it is defined for an infinitely long file. Also note that, $\gamma_m(z) = R_\infty(W(m) + z) - 1$ in terms of function $R_v(u)$ defined in Section 3.

Recall the function $U(v)$ defined in Section 3. If the age u of a file of size v is such that $u > U(v)$ then the net data input rate into link buffer is -1 because no new data is sent for a unit amount of data served.
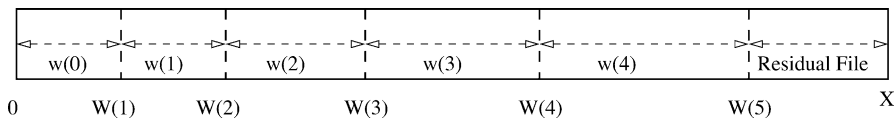

Fig. 5. The figure shows a file of size X packets ($W(5) < X < W(6)$) split into the possible window sizes (at end of cycles) under a general AWP.

Note that if $v \in [W(m), W(m+1)]$, then $U(v) \in (W(m-1), W(m))$. For example, if $v = 5$, and we have TCP-SS, then in the first cycle one packet is sent. Upon the receipt of an acknowledgement for this packet, the window grows to $w(1) = 2$ and two packets are sent during the second cycle. When the acknowledgement of the first of these is received, the window grows to three and two more packets are sent bringing the total number of sent packets to five; we see that $U(5) = 2$ since upon receipt of the second acknowledgement the residual file in the server is fully transferred. On the receipt of the acknowledgement of the third packet, the window grows to four and the second cycle ends. The third cycle involves no transmission from the server and just the last two packets of the file are transferred by the link in this cycle.

The idea now is to find the transform of the distribution of the windows which constitute the link buffer occupancy. Considering the stationary system, at time instant t and for $1 \leq i \leq N(t)$, denote by $G_i(t)$ the window of the ith ongoing transfer. We recall the facts about the stationary M/G/1 PS queue from Section 2.1. As $G_i(t)$ is a function only of the ith ongoing transfer's total file size and age, conditioned on $N(t)$, the $G_i(t)$'s are independent and identically distributed random variables. The total data in the link buffer is

$$Q(t) = \sum_{i=1}^{N(t)} G_i(t) \tag{10}$$

as the whole of the current window worth of data from each file is in the buffer (owing to zero propagation delay assumption).

Thus, the stationary link buffer occupancy is actually a random sum of i.i.d. random variables with common distribution, say, $G(\cdot)$. Using Proposition 2.9 of [24], it follows that if $G(\cdot)$ corresponds to a sub-exponential distribution (see [24]) then so does $Q(\cdot)$. In particular, if the tail of $G(\cdot)$ is regularly varying with parameter β then so is the tail of $Q(\cdot)$.

Denote the LST of $G(\cdot)$, the window size distribution, by $\tilde{g}(s)$. The LST of link buffer occupancy distribution is thus

$$\tilde{q}(s) = \sum_{k=0}^{\infty} \rho^k (1 - \rho) \tilde{g}(s)^k = \frac{1 - \rho}{1 - \rho \tilde{g}(s)} \tag{11}$$

where we have used the fact that the probability of there being n files active is $(1 - \rho)\rho^n$ and that conditioned on $N(t) = n$, the $G_i(t)$'s are independent and distributed according to $G(\cdot)$.

Thus, we can obtain the mean buffer length and the variance of the buffer occupancy once we have obtained $\tilde{g}(s)$. In particular, the mean buffer occupancy is given by

$$EQ = \left(\frac{\rho}{1 - \rho}\right) \left(-\frac{\mathrm{d}}{\mathrm{d}s} \tilde{g}(s)\right)\bigg|_{s=0} \tag{12}$$

Observe that EQ is finite iff $\frac{\mathrm{d}}{\mathrm{d}s} \tilde{g}(s)|_{s=0}$ is finite.

Theorem 5.1. *For an AWP used to transfer files of distribution $V(\cdot)$, the LST of the distribution of a single file's contribution to the link buffer is*

$$\tilde{g}(s) = \sum_{n=0}^{\infty} \int_{v=W(n)}^{W(n+1)} \left[\int_{U(v)}^{v} \mathrm{e}^{-s(v-u)} \frac{\mathrm{d}u}{v} + \int_{W(n-1)}^{U(v)} \mathrm{e}^{-s(w(n-1)+\int_0^{u-W(n-1)} \gamma_{n-1}(z)\,\mathrm{d}z)} \frac{\mathrm{d}u}{v} \right.$$

$$+ \sum_{m=0}^{n-2} \int_{u=W(m)}^{W(m+1)} e^{-s(w(m)+\int_0^{u-W(m)} \gamma_m(z)\,dz)} \frac{du}{v} \Bigg] dV_s(v) \tag{13}$$

Proof. Follows from the results for PS queue given in Section 2.1. We use the fact that the window size of a file of size v, when its age $u < U(v)$ and $u \in (W(m), W(m+1))$, is $w(m) + \int_0^{u-W(m)} \gamma_m(z)\,dz$. Also, the window size of a file of size v is $v - u$ if $u > U(v)$. In Eq. (13), n is used to condition on v (the file size requirement) being in the nth cycle, and m is used to condition on u (the age of the file) being in the mth cycle. The integrand (with respect to v) above contains three terms to take care of the possibility where the age of file is in the last window (i.e., $u > U(v)$) and hence whole of remaining file $v - u$ is in the link buffer. $\square$

Corollary 5.1. *For TCP-SS controlled transfer of files of distribution $V(\cdot)$, the LST of the distribution of a single file's contribution to the link buffer is*

$$G^c(x) \sim_{x \to \infty} V_e^c(2x)$$

Proof. For TCP-SS, $w(n) = 2^n$, $W(n) = 2^n - 1$, $U(v) = \frac{v}{2}$ and $\gamma_m(z) \equiv 1$, hence $w(m) + \int_0^{u-W(m)} \gamma_m(z)\,dz = u + 1$. It follows from Eq. (13) that

$$\tilde{g}(s) = \sum_{n=0}^{\infty} \int_{v=W(n)}^{W(n+1)} \left[\int_{v/2}^{v} e^{-s(v-u)} \frac{du}{v} + \int_0^{v/2} e^{-s(u+1)} \frac{du}{v} \right] dV_s(v)$$

$$= \int_{v=0}^{\infty} \left[\int_{v/2}^{v} e^{-s(v-u)}\,du + \int_0^{v/2} e^{-s(u+1)}\,du \right] \frac{dV(v)}{EV} = \frac{1 + e^{-s}}{sEV} \int_{v=0}^{\infty} [1 - e^{-(s/2)v}]\,dV(v)$$

$$= \frac{1 + e^{-s}}{sEV} \left[1 - \tilde{v}\left(\frac{s}{2}\right) \right] = \frac{1 + e^{-s}}{2} \tilde{v}_e\left(\frac{s}{2}\right) \Rightarrow g(x) = \frac{1}{2}[v_e(2x) + v_e(2x - 1)]$$

where $\tilde{v}_e(s)$ is the LST of the distribution of the excess random variable associated with V (see [20]). The proof follows for large x. $\square$

Corollary 5.2. *For TCP-SS controlled transfer of Pareto-distributed files, $G^c(\cdot)$ is regularly varying with parameter $\alpha - 1$ where α is the shape parameter of Pareto distribution.*

Proof. Follows directly from Corollary 5.1. $\square$

Remark. Using the property that $\gamma_m(z) \geq 0$ for TCP-SS, result of Corollary 5.1 has also been obtained in [25]. Using a similar observation, it is also shown in [25] that for TCP-CA controlled transfer of Pareto-distributed files, $G^c(\cdot)$ is regularly varying with parameter $2(\alpha - 1)$. Thus, the method presented in this section, being applicable irrespective of any structure of $\gamma_m(z)$ function, is more general in nature as compared to that of [25]; Corollary 5.1, when compared to results of [25], serves as validation of the general method introduced in this section.

5.1.2. The mean link buffer occupancy, EQ

The expected link buffer occupancy is, from Eqs. (12) and (13) and using $dV_s(v) = \frac{v\,dV(v)}{EV}$,

$$
EQ = \frac{\lambda}{1-\rho} \sum_{n=0}^{\infty} \int_{v=W(n)}^{W(n+1)} \left[\int_{U(v)}^{v} (v-u)\,du + \int_{u=W(n-1)}^{U(v)} \left(w(n-1) + \int_{0}^{u-W(n-1)} \gamma_{n-1}(z)\,dz \right) du \right.
$$
$$
\left. + \sum_{m=0}^{n-2} \int_{u=W(m)}^{W(m+1)} \left(w(m) + \int_{0}^{u-W(m)} \gamma_m(z)\,dz \right) du \right] dV(v) \tag{14}
$$

The case where $\gamma_m(z) \geq 0$ is of special interest as it includes TCP-SS and TCP-CA. We obtain explicit results for this case. These results are summarised in the following.

Theorem 5.2. *Under a general AWP with $\gamma_m(z) \geq 0, \forall m, z$,*

$$
\frac{\lambda}{1-\rho} \sum_{n=0}^{\infty} w^2(n-1) V^c(W(n)) \leq EQ \leq \frac{2\lambda}{1-\rho} \sum_{n=0}^{\infty} w^2(n) V^c(W(n-1))
$$

Proof. If $\gamma_m(z) \geq 0, \forall m, z$ then the sequence $\{w(n), n \geq 0\}$ is nondecreasing and so, $(w(m) + \int_{0}^{u-W(m)} \gamma_m(z)\,dz) \leq w(m+1)$ for $u \leq W(m+1)$. Thus, we can upper bound the third term in the integrand (with respect to v) in the right-hand side of Eq. (14)

$$
\sum_{m=0}^{n-2} \int_{u=W(m)}^{W(m+1)} \left(w(m) + \int_{0}^{u-W(m)} \gamma_m(z)\,dz \right) du
$$
$$
\leq \sum_{m=0}^{n-2} \int_{u=W(m)}^{W(m+1)} w(m+1)\,du = \sum_{m=0}^{n-2} w(m)w(m+1) \leq \sum_{m=0}^{n-2} w^2(m+1)
$$

Using the same argument, the second term in Eq. (14) is

$$
\int_{u=W(n-1)}^{U(v)} \left(w(n-1) + \int_{0}^{u-W(n-1)} \gamma_{n-1}(z)\,dz \right) du
$$
$$
\leq \int_{u=W(n-1)}^{W(n)} \left(w(n-1) + \int_{0}^{u-W(n-1)} \gamma_{n-1}(z)\,dz \right) du \leq w^2(n)
$$

Thus, using the above inequalities

$$
\int_{v=W(n)}^{W(n+1)} \left[\int_{u=W(n-1)}^{U(v)} \left(w(n-1) + \int_{0}^{u-W(n-1)} \gamma_{n-1}(z)\,dz \right) du \right.
$$
$$
\left. + \sum_{m=0}^{n-2} \int_{u=W(m)}^{W(m+1)} \left(w(m) + \int_{0}^{u-W(m)} \gamma_m(z)\,dz \right) du \right] dV(v)
$$
$$
\leq \int_{v=W(n)}^{W(n+1)} \sum_{m=0}^{n-1} w^2(m+1)\,dV(v) = w^2(n) V^c(W(n))
$$

where the last expression is obtained using some algebra.

The first term in right-hand side of Eq. (14) can be bounded from above as $v - u \leq w(n)$ for $v \in (W(n), W(n+1))$ and $U(v) < u < v$. Hence, the first term is

$$\int_{v=W(n)}^{W(n+1)} \int_{U(v)}^{v} (v - u) \, du \, dV(v) \leq \int_{v=W(n-1)}^{\infty} w^2(n) \, dV(v) = w^2(n) V^c(W(n-1))$$

Combining the upper bounds for the three terms of Eq. (14) and noting that $V^c(W(n)) \leq V^c(W(n-1))$, we get

$$EQ \leq \frac{2\lambda}{1-\rho} \sum_{n=0}^{\infty} w^2(n) V^c(W(n-1)) \tag{15}$$

Now we can lower bound the three terms in right-hand side of Eq. (14). The first and second terms are clearly ≥ 0. Also, if $\gamma_m(z) \geq 0$, then $(w(m) + \int_0^{u-W(m)} \gamma_m(z) \, dz) \geq w(m)$. These observations along with some algebra gives the following lower bound

$$EQ \geq \frac{\lambda}{1-\rho} \sum_{n=0}^{\infty} w^2(n-1) V^c(W(n)) \tag{16}$$

The result now follows from Eqs. (15) and (16). □

Corollary 5.3. *For congestion avoidance controlled transfer of Pareto-distributed file sizes, EQ is finite iff $\alpha > 1.5$.*

Proof. Follows from Theorem 5.2 with $V^c(v) = \frac{1}{v^\alpha}$, $w(n) = n+1$, $W(n) = \frac{n(n+1)}{2}$. □

Corollary 5.4. *For slow start controlled transfer of Pareto-distributed file sizes, EQ is finite iff $\alpha > 2$.*

Proof. Follows from Theorem 5.2 with $V^c(v) = \frac{1}{v^\alpha}$, $w(n) = 2^n$, $W(n) = 2^n - 1$. □

Note that the Corollaries 5.3 and 5.4 are in accordance with our results of Corollary 5.2 and that of [25] where we have seen that $G^c(\cdot)$ is regularly varying with parameter $\alpha - 1$ and $2(\alpha - 1)$ for TCP-SS and TCP-CA, respectively.

5.2. Discussion of results

We make the following remarks on the results obtained till now in this section:

- Note the marked difference in the tail of the buffer occupancy distribution for TCP-SS and TCP-CA. We have seen in Section 4 that for both of these AWPs the $a(t)$ process is LRD with same lower bound on the Hurst parameter. The results of this section thus clearly indicate that feedback control can lead to a lightening of the tail of the link buffer occupancy when compared to an uncontrolled (open-loop) transfer of files. It can also be observed that an aggressive feedback control like the slow start phase of TCP may not result in lightening of the buffer occupancy distribution; in this case the mean buffer occupancy is finite iff the second moment of the (Pareto) file size distribution is finite.
- Corollary 5.3 is interesting in view of the result of Section 4 where it was shown that, for small arrival rates λ, the traffic into the link buffer ($a(t)$ process) is LRD for Pareto-distributed file sizes with
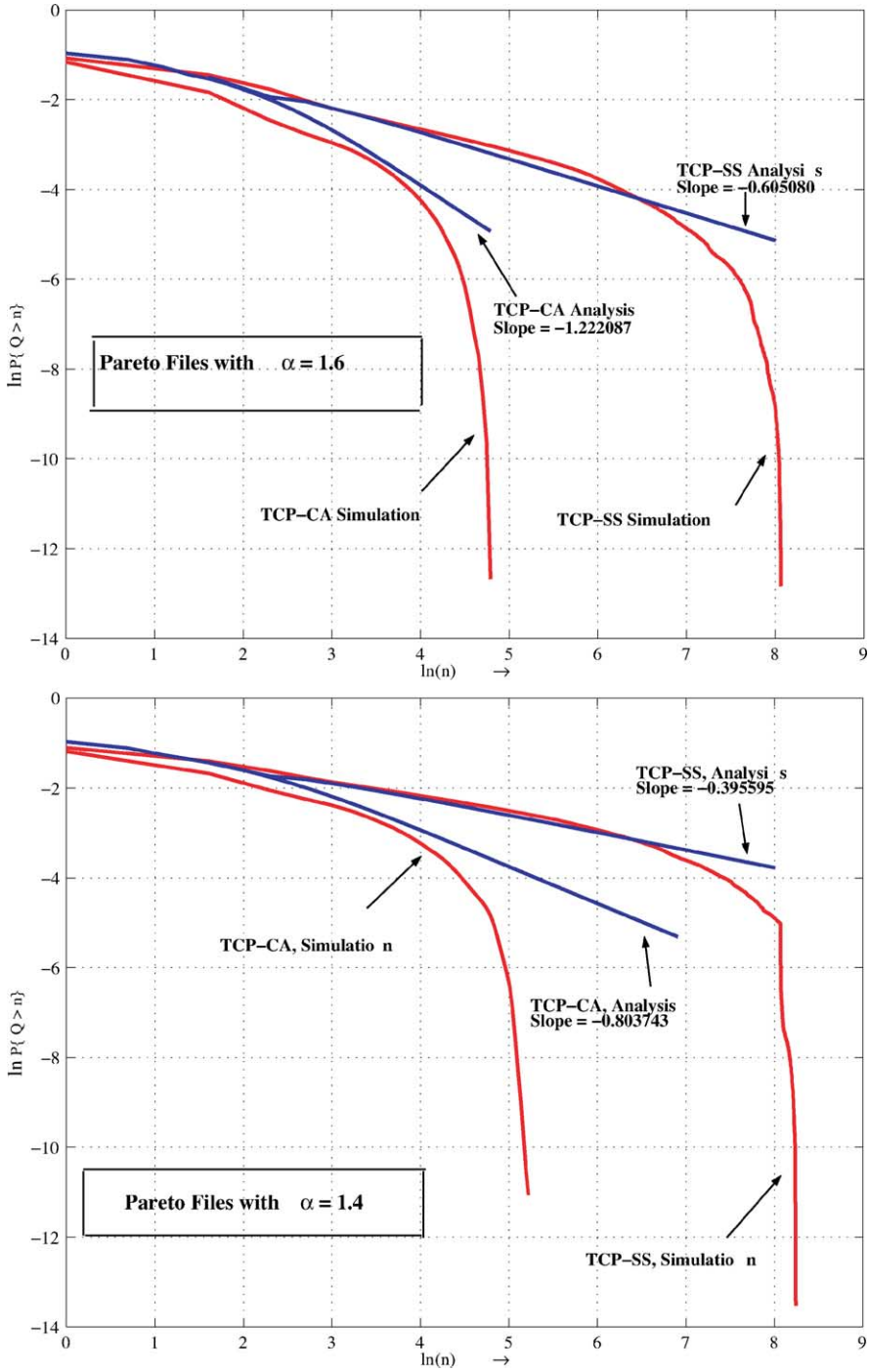
Fig. 6. Plot showing log $Q^c(x)$ vs. log(x) obtained from *ns* simulations for Pareto files with $\alpha = 1.6$ and 1.4 controlled using TCP-CA and TCP-SS. The link load was set to 0.4. Also shown are the curves obtained from numerical computation of $Q^c(\cdot)$ from analysis.
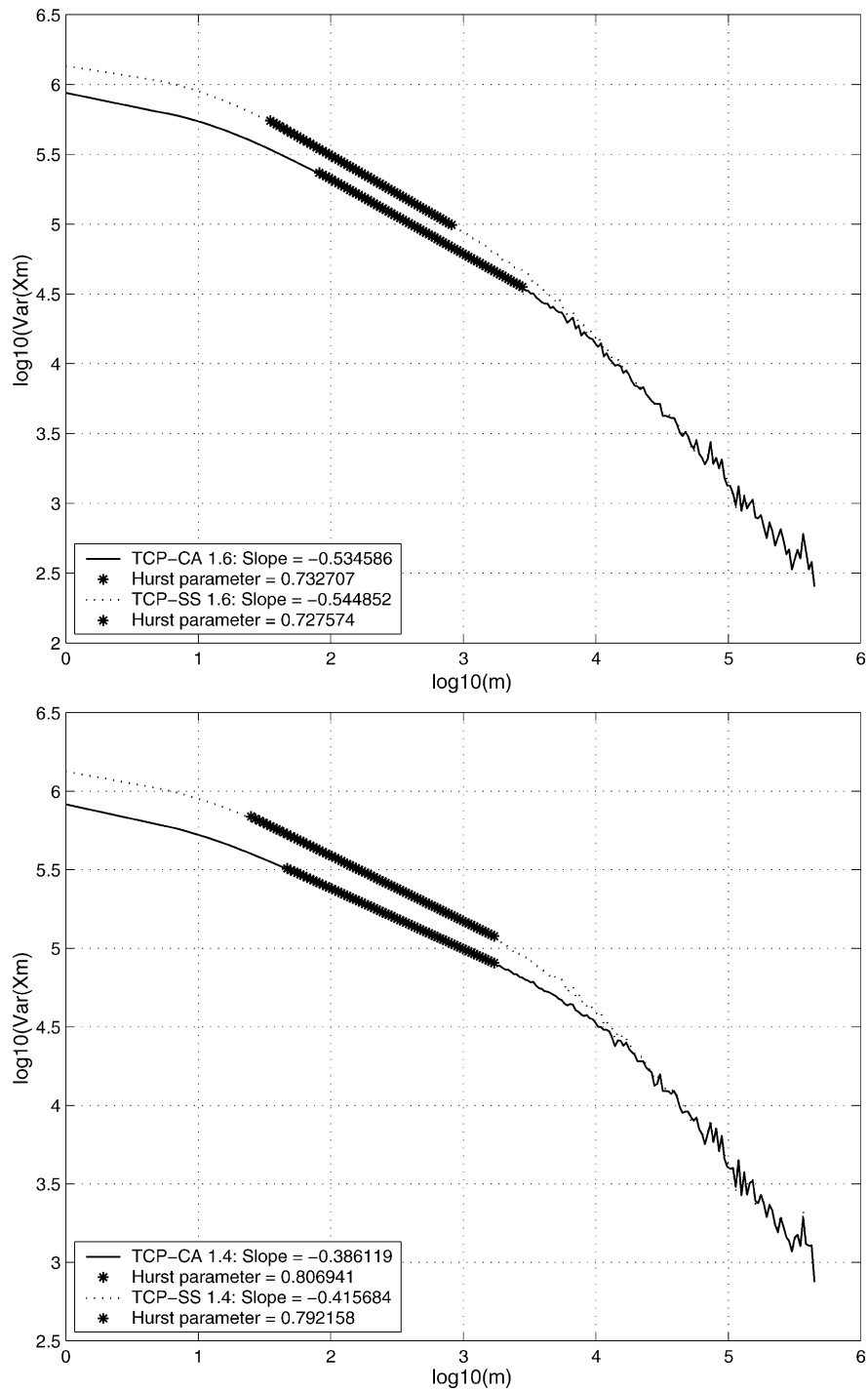
Fig. 7. Variance–time plots for $a(t)$ process obtained from *ns* simulations for Pareto files with $\alpha = 1.6$ and 1.4 controlled using TCP-CA and TCP-SS. The link load was set to 0.4.

$1 < \alpha < 2$. Thus, we now have an example where the traffic into a queue is LRD but the mean queue length is finite; this is because the traffic into the queue is regulated using a closed loop control.

- Note from Theorem 5.2 that if $w(n)$ are bounded then $EQ < \infty$ independent of the file size distribution. Yet it is still possible to have the $a(t)$ process exhibiting long range dependence as seen in the discussions in relation to Theorem 4.4 at end of Section 4. For example, if $w(n) = 1$ for all n then Theorem 4.4 is applicable with $r = R = 1$, thus establishing that $a(t)$ process is LRD for small λ while the distribution of the link buffer occupancy has an exponentially decaying tail, as now $P(Q(t) > q) = P(N(t) > q) = \rho^{q+1}$.

5.2.1. Simulation experiments

We performed *ns* simulations to validate the results of the previous sections. We simulated a single link with capacity 10 Mbps to which requests for transfer of Pareto files arrive according to a Poisson process of rate λ. The mean file size was set to 30 KB and $\lambda = 16.7$ requests/s, so that the link load was $\rho = 0.4$.

In Fig. 6, we plot $\log Q^c(x)$ versus $\log(x)$ obtained from *ns* simulations for the transfer of Pareto-distributed files using the TCP-CA and TCP-SS protocols; the normalised offered load ρ was set to 0.4 and the link buffer implements DRR scheduling. The shape parameters α of the file size distribution were 1.6 and 1.4. Also shown in the figure are the corresponding plots obtained from numerical computation of the buffer occupancy distribution using, along with Eq. (10), the results of Corollary 5.1 (for TCP-SS) and [25] (for TCP-CA). The slopes of the analysis curves are also shown in the figures and are seen to be close to their respective values predicted by the above analysis; for example, the slope for TCP-SS with $\alpha = 1.4$ is -0.3955 which is close to $(\alpha - 1)$ as suggested by the analysis. The plot also confirms the results of Section 5 that with TCP-CA the tail of link occupancy distribution is lighter than that for TCP-SS. Note that the tail behaviour for TCP-SS controlled transfer of Pareto 1.6 files is worse than that for TCP-CA controlled transfer of Pareto 1.4 files. The sharp drops observed at the ends of the simulation curves are due to the finite simulation run lengths and are not considered in the approximation.

Variance–time plots[5] for the $a(t)$ process are shown in Fig. 7 for TCP-SS and TCP-CA controlled transfer of Pareto files with shape parameter $\alpha = 1.6$ and 1.4. The slopes of the plots indicate that the input process is LRD with Hurst parameter $\frac{3-\alpha}{2}$ irrespective of TCP-SS or TCP-CA as proved before.

---

[5] Obtain the time series $X = \{X_i, i \geq 1\}$, where X_i is the number of packets arriving in the interval $(p(i-1), pi)$ where p is a packet service time. Divide the original time series X into blocks of size m and average within each block, that is consider the aggregated series,

$$X^{(m)}(k) = \frac{1}{m} \sum_{i=(k-1)m+1}^{km} X_i, \quad k \geq 1$$

for successive values of m. The index k labels the block. Then take the sample variance of $X^{(m)}(k)$, $k \geq 1$ within each block. This sample variance is an estimator of Var $X^{(m)}$. For a given m, divide the data, $X_1, \ldots, X_N$, into N/m blocks of size m. Calculate $X^{(m)}(k)$, for $1 \leq k \leq N/m$, and its sample variance

$$\widehat{\text{Var}} X^{(m)} = \frac{1}{N/m} \sum_{k=1}^{N/m} (X^{(m)}(k))^2 - \left(\frac{1}{N/m} \sum_{k=1}^{N/m} X^{(m)}(k) \right)^2$$

Repeat this procedure for different values of m and plot the logarithm of the sample variance versus $\log m$. The slope β of the curve is obtained by fitting a line to the points obtained from the plot. The Hurst parameter is then $H = \frac{\beta+2}{2}$.

6. Effect of random packet drops

In this section, we consider the effect of window reductions due to active queue management; for simplicity of analysis we assume random packet *marking* instead of *drops* and assume that the sender AWP is cooperative and responds to the mark signal as if it has detected a packet loss. In Section 5, we have seen that the worst possible behaviour of the tail of the buffer occupancy distribution is obtained when the AWP is aggressive like TCP-SS. In this section, we assume such an aggressive AWP and study the effect of random marking on the buffer occupancy distribution. We consider various schemes of responses to reception of a marked packet, i.e., we consider a multiplicative decrease of the window and also a gentle decrease. This section is meant to support the intuition, in the framework of the model, that with window reductions (due to random packet drops or markings owing to active queue management), the buffer occupancy distribution will have a lighter tail. Similar results have also been obtained in [26].

We use the notation $w(u)$ for the window size of an infinitely long file whose u amount of data has been acknowledged. Note that, owing to random packet marking, $w(u)$ is now a stochastic process. Since we are considering data as fluid, we use a marking *rate* (per unit data) instead of a marking *probability* (per packet), i.e., if the router marks each packet with probability p then, in the fluid model, the amount of data served between two consecutive markings is exponentially distributed with parameter $\ln \frac{1}{1-p} =: \mu$. We assume that $0 < p < 1$ to avoid the consideration of the extreme cases where the analysis is slightly different (the case of $p = 0$ is already presented in previous sections). We are assuming marking instead of packet drops to avoid the unnecessary details of packet retransmissions, and because we are only interested in qualitative results for the case where window reductions occur. Consider n files being served in a PS fashion. At a marking instant, the mark is applied to any one of the files with equal probability. Hence, the marking rate of a file is $\frac{\mu}{n}$ per unit time, assuming a unit rate server. But each file's age is increasing at rate $\frac{1}{n}$ data per unit time. Hence, the marking rate of a file is μ per unit of its age. Further, these marking processes within files are independent. Hence, we can again analyse each active flow separately along the lines our analysis in previous section. The AWP increases the window as a function of age according to $w(u)$; marks can occur in a Poisson process of rate μ over the age evolution; at each mark the window decrease and increase is discussed further. These observations lead to a simple analysis as now we can consider each active flow separately and model only its individual contribution to the link buffer. The modes of evolutions of $w(u)$ we consider in this work are depicted in Figs. 8 and 10.

6.1. The AWP with linear window reduction

In Fig. 8, $w(u)$ increases as it does in TCP-SS, i.e., linearly with age u until a mark occurs. We use a TCP-SS like window buildup as it was observed in [25] that TCP-SS gives the worst possible tail behaviour of $G(\cdot)$ in absence of window reductions (due to marking or drops). We refer to the period, where $w(u)$ increases linearly as an *on-period*. We then assume that after a mark occurs, no new data is sent until whole of the window at which the mark occurred is served; this is what happens in the Tahoe version of TCP (see [7]), except that we are not considering the time wasted in coarse timeout. This assumption ensures that an ongoing transfer always has a positive amount of outstanding data in the link buffer and hence enables us to use the processor sharing results. The phase where $w(u)$ is decreasing and no new data is sent in response to acknowledgments is called an *off-period*. Note also that a mark occurring during the off-period does not affect the trajectory of $w(u)$.
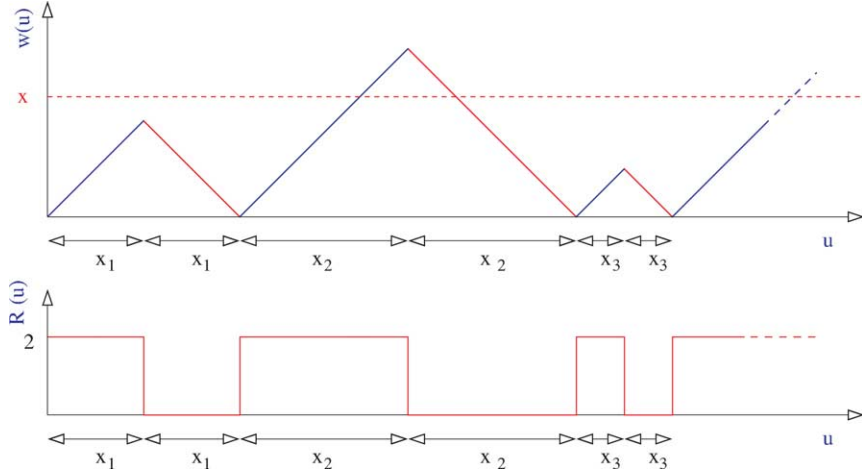
Fig. 8. The $w(u)$ and $R(u)$ functions for the AWP considered in Section 6.1.

In Fig. 8, which depicts the $w(u)$ function for an infinitely long file, the first on-period ends at an age of x_1, where a mark occurs; x_1 is an exponentially distributed random variable with mean $\frac{1}{\mu}$. After a mark at age x_1, the TCP sender stops sending data and enters the off-period. Note that in the off-period the window $w(u)$ decreases at the same rate as the age of the file increases, i.e., $\frac{dw(u)}{du} = -1$. This continues for an additional age of $w(x_1) = x_1$ (for simplicity we have taken $w(x_1) = x_1$ instead of $x_1 + 1$ as is the case for TCP-SS). At $2x_1$ another on-period starts and the evolution process repeats itself. Fig. 8 also shows the $R(u)$ process, the rate of data arrival into the link buffer from an infinitely long session which has attained age u. Note that, owing to the TCP-SS-like window evolution in the on-period, the off-period duration (in terms of age) is same as the corresponding on-period length. Also, $R(u) = 2$ in an on-period and 0 in an off-period.

6.1.1. Buffer occupancy distribution for the AWP of Fig. 8

Let $p_x(u) := P\{w(u) > x\}$. The regenerative behaviour of the $w(u)$ process as observed in Fig. 8 enables us to write a renewal equation for $p_x(u)$ as follows

$$p_x(u) = e^{-\mu z_x(u)} I_{\{u>x\}} + \int_{y=0}^{u} p_x(u-y)\frac{\mu}{2}e^{-(\mu/2)y}\, dy \tag{17}$$

where $z_x(u)$ is the minimum age such that if a mark does not occur in the age interval $[0, z_x(u)]$ then $w(u) \geq x$. Note that $z_x(u)$ satisfies the condition that $w(z_x(u)) = (u - z_x(u)) + x$; this is obtained by putting the condition that a mark occurs at age $z_x(u)$ and results in a window of exactly x at age u (see Fig. 9). In the second term in Eq. (17), y is used to condition on the start of the second on-period which is the regeneration instant for the $w(u)$ process; also used is the fact that the first regeneration epoch (and also the subsequent regeneration cycle lengths) is exponentially distributed with mean $\frac{2}{\mu}$.

Solution to Eq. (17) is (see [20])

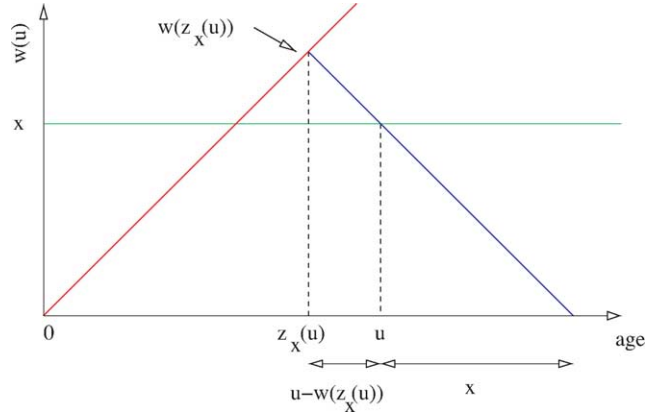$$p_x(u) = e^{-\mu z_x(u)} I_{\{u>x\}} + \int_{y=0}^{u} e^{-\mu z_x(u-y)} I_{\{u-y>x\}} \frac{\mu}{2}\, dy$$

Fig. 9. Figure showing the computation of the $z_x(u)$ function used in analysis of the AWP of Section 6.1.

$$= e^{-\mu z_x(u)} I_{\{u>x\}} + \int_{y=0}^{u-x} e^{-\mu z_x(u-y)} \frac{\mu}{2} \, \mathrm{d}y = e^{-\mu x} I_{\{u>x\}}$$

Now, the above expression assumes an infinitely long file. For a file of size v, the window at age u is $w_v(u) = \min(w(u), v - u) \le w(u)$. Hence, recalling the notation G from Section 5,

$$G^c(x) = \int_{v=0}^{\infty} \int_{u=0}^{v} P\{w_v(u) > x\} \frac{\mathrm{d}u}{v} \, \mathrm{d}V_s(v) \le \int_{v=0}^{\infty} \int_{u=0}^{v} P\{w(u) > x\} \frac{\mathrm{d}u}{v} \, \mathrm{d}V_s(v)$$

$$= e^{-\mu x} \int_{v=x}^{\infty} \int_{u=x}^{v} \frac{\mathrm{d}u}{v} \, \mathrm{d}V_s(v) \le e^{-\mu x}$$

i.e., $G^c(\cdot)$ is bounded by an exponentially decaying function.

Remark. Compare this result with Corollary 5.2 where it is shown that for TCP-SS controlled transfer of Pareto-distributed files and with no random marking $G^c(\cdot)$ is regularly varying with parameter $\alpha - 1$.

6.1.2. Auto-covariance function of the a(t) process for the AWP of Fig. 8

Let $p(u)$ denote the probability that, for an infinitely long file, the $R(u)$ function is 2, i.e., an on-period at age u. The following renewal equation can be written for $p(u)$

$$p(u) = e^{-\mu u} + \int_{y=0}^{u} p(u-y) \frac{\mu}{2} e^{-(\mu/2)y} \, \mathrm{d}y$$

which has solution

$$p(u) = e^{-\mu u} + \int_{y=0}^{u} e^{-\mu(u-y)} \frac{\mu}{2} \, \mathrm{d}y = \frac{1}{2} \left[1 + e^{-\mu u} \right]$$

It is easy to see that Theorem 4.1 can be modified for an AWP which has regenerative behaviour in response to marking as the above protocol, to read.

Theorem 6.1. *If the AWP used to transfer file sizes of distribution $V(\cdot)$, in presence of random drops, has the properties*:

(1) *The AWP alternates between two states, called on-period and off-period.*
(2) *A mark results in the end of the current on-period and the start of the next off-period.*
(3) *The window size at any age is strictly positive irrespective of the state of the AWP.*
(4) *The instants of start of on-periods are regeneration points for the $R_\infty(u)$ function.*
(5) *There exists a r such that $R_v(u) \geq r > 0$ in an on-period and $R_v(u) = 0$ in the off-period,*
Then,

$$J_1(\tau) \succeq_{\tau \to \infty} (1 - \rho)^2 \lambda r^2 \int_{v=X(\tau)}^{\infty} \int_{u=0}^{U(v)-\tau} EI_{\{\text{on at } u\}} I_{\{\text{on at } u_1(\tau)\}} \, du \, dV(v)$$

Proof. The above relation follows using arguments similar to those in the proof of Theorem 4.1. We have used $u_1(\tau)$ to denote the age, at time τ, of the session active at time 0. Here, $U(\cdot)$ and $X(\cdot)$ are deterministic and correspond to the $\mu = 0$ case; this simplification is obtained using the regeneration property of $R_v(u)$ which ensures that (as $R_v(u) = 0$ in off-periods), $\sup\{u : R_v(u) > 0, u \leq v\} \geq U(v)$. $\square$

Corollary 6.1. *For the AWP of Fig. 8,*

$$J_1(\tau) \succeq_{\tau \to \infty} 2(1 - \rho)^2 \lambda \int_{v=X(\tau)}^{\infty} \int_{u=0}^{U(v)-\tau} p(u) \, du \, dV(v) \sim_{\tau \to \infty} \frac{(1 - \rho)^2 \lambda}{2^\alpha (\alpha - 1)} \frac{1}{\tau^{\alpha-1}}$$

(for Pareto-distributed file sizes)

Proof. The first relation above follows from Theorem 6.1 by using $r = 2$ and noting that, due to alternating behaviour of on and off periods in the AWP of Fig. 8, as $0 < \mu < \infty$, and as the consecutive on-period and off-period length are same

$$EI_{\{\text{on at } u\}} I_{\{\text{on at } u+t\}} \to_{t \to \infty} \frac{1}{2} p(u)$$

along with the observation that $u_1(\tau) \to_{\tau \to \infty} \infty$. The second expression of the corollary follows by using $p(u) = \frac{1+e^{-\mu u}}{2}$, $U(v) = \frac{v}{2}$ and $X(\tau) = 2\tau$ and plugging in the expression for Pareto distribution in the first expression. $\square$

Since $J_2(\tau)$ has asymptotic behaviour independent of the AWP used, it follows that Theorem 4.3 (and hence its conclusion) holds with $\delta = 2^{-\alpha}$ for the AWP of Fig. 8.

6.2. The AWP with multiplicative window reduction

The AWP of Fig. 8 is very conservative in responding to marking as the window size reduces to 0 and then the window buildup restarts independent of the past history.

We now consider another, more aggressive, response to marking, i.e., multiplicative decrease of the window as shown in Fig. 10. The window under this AWP evolves as follows: the AWP behaves like TCP-SS until the first mark occurs at age x_1. At age x_1^+, a multiplicative decrease of the window takes
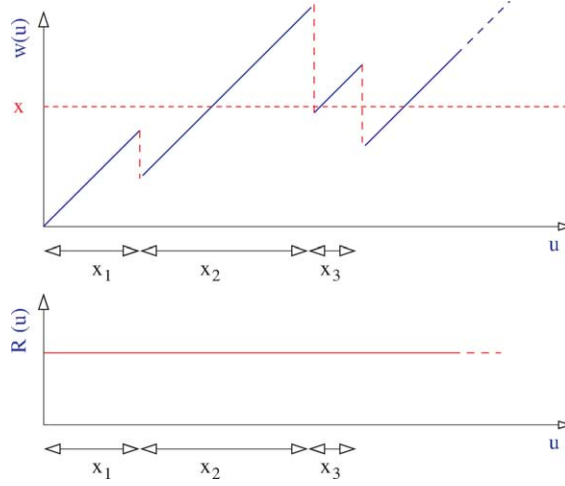
Fig. 10. The $w(u)$ and $R(u)$ functions for the AWP considered in Section 6.2.

place and the window size is reduced to $\frac{w(x_1)}{2}$. Note that this is possible if the following mechanism is used: a mark at window size w results in drop of all of the data of the corresponding file in the link buffer and the AWP controlling the transfer of the file reduces it's window size to $\frac{w}{2}$ instantaneously (thus transferring $\frac{w}{2}$ amount of data to the link buffer). Note that this mode of window evolution is very similar to the Reno version of TCP.

The exact analysis of buffer occupancy distribution becomes hard as the evolution of the window depends on all the previous marking epochs. But it is possible to analyse for the expected window of active sessions. Let $w(u)$ be the random variable corresponding to the window of an infinitely long file at an age of u and let $Ew(u)$ be its expectation. Following the analysis of [27] it is possible to write down the following differential equation for $Ew(u)$[6]. We let $M(u)$ denote the Poisson process corresponding to the instants of marks, i.e., $dM(u) = I_{\{a \text{ mark occurred in the interval } (u, u+du)\}}$. With this notation, we get the following differential equation governing the evolution of the $Ew(u)$ function

$$dEw(u) = du - \frac{Ew(u)}{2}\mu\,du \tag{18}$$

where the first term is for the increment of the window by an amount du when du amount of data is acknowledged (without marking) and the second part takes care of the multiplicative decrease in window size as a result of a mark occurring. We have also used the fact that $M(u)$ is a Poisson process of rate μ.

The above differential equation when solved with the boundary condition of $Ew(0) = 0$ (for simplicity), yields

$$Ew(u) = \frac{2}{\mu}(1 - e^{-(\mu/2)u}) \leq \frac{2}{\mu}$$

---

[6] The only difference in the analysis of [27] and the one to be presented here is that we are interested in window as a function of age while [27] considers window as a function of time for the case where only one session is active.

Note that the above expression is for an infinitely long file and the window at any age of a finite length file is bounded by $w(u)$, it follows that (recall G and Q defined in Section 5) $EG \leq \frac{2}{\mu}$ and hence, using Wald's lemma, $EQ = EN.EG \leq \frac{\rho}{1-\rho}\frac{2}{\mu}$ which is finite for all $\mu > 0$ *irrespective of the file size distribution*. It can also be shown in a similar manner that the second moment of $w(u)$ process is bounded above by a fixed quantity.

As the AWP of Fig. 10 has $R_v(u) = 2$ for the duration the sender is transmitting data to the link buffer, Theorem 4.4 holds for this AWP with $R = r = 2$.

7. Conclusion

We have developed a framework for the analysis of processes related to the bottleneck link buffer under an adaptive window protocol controlled transfer of randomly arriving finite volume files. The most important example of an AWP is TCP. The key idea used in the analysis was to look at the window size of the AWP controlling the transfer of a file as a function of the amount of data served from the file.

We have analysed the auto-covariance function of the process corresponding to the aggregate traffic into the bottleneck link buffer. Bounds on the asymptotic behaviour of the auto-covariance function is given which have explicit dependence on the AWP used and the file transfer volume distribution.

Also analysed is the stationary behaviour of the bottleneck link buffer occupancy under the same scenario. An explicit expression for the stationary distribution of the link buffer occupancy was obtained, and was again seen to have a dependence on the AWP and the file size distribution.

It was shown that, for Pareto-distributed file transfer volumes with shape parameter α:

(1) When TCP's congestion avoidance or slow start algorithms are used, the traffic into the link buffer is long range-dependent (LRD) for small file transfer request arrival rates and for $1 < \alpha < 2$.
(2) Under the congestion avoidance phase of TCP, the tail of the distribution of the link buffer occupancy process is regularly varying with index $2(\alpha - 1)$. This also implies that the mean link buffer occupancy is finite iff $\alpha > 1.5$.
(3) Under the slow start phase of TCP, the tail of the distribution of the link buffer occupancy process is regularly varying with index $(\alpha - 1)$. This means that the mean link buffer occupancy is finite iff $\alpha > 2$.

Thus, we have given an example (Pareto-distributed file sizes with $1.5 < \alpha < 2$ transferred under the congestion avoidance phase of TCP) where, in the presence of LRD input to the link buffer, the buffer occupancy has finite mean. This suggests that the impact of long range dependence of Internet traffic may not be as severe as is usually predicted by means of an open loop analysis.

We have also considered the effect of window reductions (owing to random packet markings) on the link buffer occupancy process. It was seen that, for an AWP which increases its window aggressively like the TCP does in its slow start phase followed be a response to packet marking which is either gentle as in TCP Tahoe or aggressive as in TCP Reno, the mean link buffer occupancy remains finite *irrespective of the file size distribution* while the traffic arrival process into the link buffer could still be long range-dependent.

The characterisation of the tail of link buffer occupancy we have developed could lead to an explanation of the sensitivity with distribution of TCP throughput performance with finite volume transfers (as observed in [9]).

The work reported in this paper is for a zero propagation delay link; it will be interesting to study how the results presented in this paper change as the propagation delay increases.

Acknowledgement

References

[1] W.E. Leland, M.S. Taqqu, W. Willinger, D.V. Wilson, On the self-similar nature of Ethernet traffic (extended version), IEEE/ACM Trans. Netw. 2 (1) (1994) 1–15.

[2] M.E. Crovella, A. Bestavros, Self-similarity in world wide web traffic: evidence and possible causes, IEEE/ACM Trans. Netw. 5 (6) (1997) 835–846.

[3] A. Erramilli, O. Narayan, W. Willinger, Experimental queueing analysis with long-range dependent packet traffic, IEEE/ACM Trans. Netw. 4 (2) (1996) 209–223.

[4] S. Floyd, V. Paxson, Difficulties in simulating the Internet, IEEE/ACM Trans. Netw. 9 (4) (2001) 392–403.

[5] Van Jacobson, Congestion avoidance and control, in: ACM SIGCOMM, 1988, pp. 314–329.

[6] T.V. Lakshman, U. Madhow, The performance of TCP/IP for networks with high bandwidth delay products and random loss, IEEE/ACM Trans. Netw. 5 (3) (1997) 336–350.

[7] A. Kumar, Comparative performance analysis of versions of TCP in local network with a lossy link, IEEE/ACM Trans. Netw. 6 (4) (1998) 485–498.

[8] J. Padhye, V. Firoiu, D. Towsley, J. Kurose, Modeling TCP throughput: a simple model and its empirical validation, in: SIGCOMM'98, 1998.

[9] A.A. Kherani, A. Kumar, Stochastic models for throughput analysis of randomly arriving elastic flows in the Internet, in: IEEE Infocom 2002, New York, June 2002.

[10] H. Chaskar, T.V. Lakshman, U. Madhow, TCP over wireless with link level error control: analysis and design methodology, IEEE/ACM Trans. Netw. 5 (3) (1999) 336–350.

[11] M. Shreedhar, G. Varghese, Efficient fair queuing using Deficit Round Robin, IEEE/ACM Trans. Netw. 4 (3) (1996) 375–385.

[12] D. Heath, S. Resnick, G. Samorodnitsky, Heavy tails and long range dependence in on/off processes and associated fluid models, Math. Operations Res. 23 (1998) 145–165.

[13] A. Kumar, K.V.S. Hari, R. Shobhanjali, S. Sharma, Long-range dependence in the aggregate flow of TCP-controlled elastic sessions: an investigation via the processor-sharing model, in: National Conference on Communications, New Delhi, 2000, available at http://ece.iisc.ernet.in/~anurag/pubm.html.

[14] B. Tsybakov, N.D. Georganas, Self-similar traffic and upper bounds to buffer-overflow probability in an ATM queue, Perform. Eval. 32 (1) (1998) 57–80.

[15] I. Norros, A storage model with self-similar input, Queueing Syst. 16 (1994) 387–396.

[16] Y. Joo, V. Ribeiro, A. Feldmann, A.C. Gilbert, W. Willinger, On the impact of variability on the buffer dynamics in IP networks, in: 39th Annual Allerton Conference on Communication, Control and Computing, 2001.

[17] A. Arvidsson, M. Roughan, T. Ryden, On the origins of long-range dependence in TCP traffic, in: ITC-17, 2001.

[18] J.W. Roberts, L. Massoulie, Bandwidth sharing: objectives and algorithms, in: INFOCOM'99, 1999.

[19] A.A. Kherani, A. Kumar, On processor sharing as a model for TCP controlled http-like transfers, in: International Conference on Communication (ICC), 2004.

[20] R.W. Wolff, Stochastic Modelling and the Theory of Queues, Prentice-Hall, Englewood Cliffs, 1989.

[21] F.P. Kelly, Reversibility and Stochastic Networks, Wiley, New York, 1979.
[22] W. Whitt, The M/G/l processor-sharing queue with long and short jobs, Unpublished Report, available at http://www. research.att.com/wow/A12.html.
[23] I.S. Gradshteyn, I.M. Ryzhik, Table of Integrals, Series, and Products, Academic Press, New York, 1980.
[24] K. Sigman, A primer on heavy-tailed distributions, Queueing Syst. Theory Appl. 33 (1999) 261–275.
[25] A.A. Kherani, A. Kumar, The lightening effect of adaptive window control, IEEE Commun. Lett. 7 (6) (2003) 284–286.
[26] M. Borkovec, A. DasGupta, S. Resnick, G. Samorodnitsky, A single channel on/off model with TCP-like control, available at http://www.orie.cornell.edu/sid/NewForLinking/technical_reports.html.
[27] W. Gong, D. Towsley, V. Misra, Stochastic differential equation modeling and analysis of TCP windowsize behavior, in: Performance'99, 1999.

Arzad A. Kherani obtained his Masters and PhD degrees in telecommunication engineering from Indian Institute of Science, Bangalore, India in 1999 and 2003, respectively. His post-PhD work was done at Tata Institute of Fundamental Research, Mumbai, India (March 2003 till July 2003) and at the MAESTRO project at INRIA, Sophia Antipolis, France (from August 2003 till present). His research interest include stochastic analysis and optimization problems arising in telecommunication systems.

Anurag Kumar obtained his BTech degree in electrical engineering from the Indian Institute of Technology at Kanpur. He then obtained the PhD degree from Cornell University. He was then with Bell Laboratories, Holmdel, NJ, for over 6 years. Since 1988 he has been with the Indian Institute of Science (IISc), Bangalore, in the Department of Electrical Communication Engineering, where he is now a Professor, and is also the Chairman of the department. His area of research is Communication Networking; specifically, modeling, analysis, control and optimisation problems arising in communication networks and distributed systems. He is a Fellow of the Indian National Academy of Engineering (INAE) since 1998. He serves on the editorial board of IEEE Communications Surveys and Tutorials. He is a coauthor of the textbook *Communication Networking: An Analytical Approach*, published by Morgan Kaufmann in 2004.