A Unified Approach for Analyzing Persistent, Non-Persistent and ON-OFF TCP Sessions in the Internet [*]

Atul Gupta
Freescale Semiconductor
NSEZ, Noida, 201305, India
atul_iisc@ieee.org

Vinod Sharma
Dept. of ECE, Indian Institute of Science
Bangalore, 560012, India
vinod@ece.iisc.ernet.in

Abstract: **Most of the studies on TCP assume a fixed number of persistent TCP connections. A few of the recent studies look at non-persistent TCP connections but with many limiting assumptions. We look at a bottleneck router which has real time UDP traffic, TCP ON-OFF traffic and a stochastic flow of non-persistent TCP connections going through it. Our analysis is able to handle a large number (100s) of TCP connections and still provides the mean file download times of each of these connections when all of them have different packet sizes, round trip times and max window size. Using this approach we are able to model almost all the different traffic types that exist in the Internet today. We also study systems employing RED control, which is recommeded to be used in the Internet. We demostrate the effectiveness of RED for congestion control in realistic Internet scenarios both theoretically and via simulations.**

Keywords: Persistent and non-persistent TCP connections, performance analysis, mean download time, Internet, RED.

1 Introduction

Recently several models of TCP sessions passing through the Internet have been analyzed ([1], [2], [4] - [6], [8], [12] - [19]). These extensive studies have provided detailed performance of the TCP and UDP sessions in terms of their through puts and delays. However, the accuracy of these models in the real networks is limited by the assumptions made and the time scale at which one is looking at the performance. Most of the studies assume persistent TCP sessions, where it is assumed that there are a fixed number of TCP sessions which always have packets to send ([1], [2], [4], [6], [8], [14], [17], [18]). This model can be considered a reasonable approximation if the TCP sessions are sending long files (e.g. FTP sessions). However most of the WAN TCP traffic today, consists of Web traffic [20] which contains small files also. Thus, a model where TCP sessions arrive, send a finite file and then leave (non-persistent sessions) is also being studied ([5], [9], [19]).This model can be useful when an e-mail or FTP server is sending a small file. Another model of a TCP source where the session may be long (persistent) but the TCP source may not always have a file to send is also relevant in the context of the Web traffic using HTTP version 1.1. In this application, the TCP session is kept alive even after a file is downloaded by a user. Then, if the user wants to down-load another file from the same server, the same TCP session can be used. For this application the model of a persistent TCP connection where the source is of ON-OFF type is realistic. This model was (with somewhat different motivation) studied in [5].

In almost all of the above mentioned studies a single bottleneck queue is considered; there is no UDP traffic in the queue and the TCP parameters (propagation delays, maximum window size etc) are same for all the TCP connections. Furthermore, a fluid model (ignoring the packet level details, in particular the packet lengths) is formulated. These assumptions are certainly not realistic and we show that they do have a significant impact on the performance of the UDP and TCP connections. For example, our study shows that the TCP mean download time is (almost) inversely proportional to the packet size. We also show that with RED control, the TCP mean download time is not affected but the UDP mean sojourn time reduces drastically. In [17] the analysis included these extra details but the study was limited to a fixed

---

[*] This work was done when the first author was working with Centre for Development of Telematics, Bangalore.

number of persistent TCP connections which always have data to send. [9], [16] extend the approach of [14] to describe the behavior of finite TCP flow. [9] models the TCP connection establishment phase as well. It gives a detailed modelling of the TCP slow start phase assuming correlated losses. It provides expressions for the average file download time for a TCP connection given its size in packets, connection round trip time and the packet loss probability. [16] provides a more detailed model of TCP slow start phase. It gives a formula for TCP throughput and latency under both independent as well as correlated losses. Both these papers assume a fixed packet loss probability known a priori.

In this paper we extend the approach of [17] to a system where a TCP source is modeled as an ON-OFF source. UDP traffic may also be present. After that we also include in the model a randomly arriving flow of non persistent TCP connections. Unlike in [5], [12], [19], we take into account the packet level behavior and the different packet lengths, propagation delays and maximum window lengths for different TCP connections. Furthermore, we consider these systems with RED control also, which is not studied in these references. In the RED case we are able to handle a packet loss probability which is changing as the RED average queue length parameter changes. First we will start with a fixed number of TCP connections with persistent ON-OFF behavior. We make a Markov chain model and obtain the mean download time of the files. We explicitly consider the effect of different TCP parameters on the mean download time of the files. Since, we have incorporated the parameters of all the TCP connections the dimensionality of the Markov chain is large (2^N, where N is the number of TCP connections). Thus, we next develop an *averaged* model of the system with which we can handle a large number of TCP connections at a moderate computational cost. But we are still able to capture the effect of the different TCP parameters on the mean download times.

Finally we include in the model the non-persistent TCP connections. Unlike the previous studies we include the effect of different propagation delays, packet lengths and maximum window sizes for these connections also. We also model the effect of the slow start phase of the TCP to take care of small file sizes which may spend most of the time in the slow start phase only. To achieve these objectives we will analyze our system by a hybrid fluid/packet level modeling. For each of the above three models, we also study the system when the bottleneck queue deploys RED control [10]. RED is recommended for implementation in routers to manage queue lengths, reduce end-to-end latency and reduce packet dropping within the Internet [7]. Therefore, it is important to study the system with RED control also.

As is clear from the above discussion, for different TCP sessions simultaneously going on in the network, different models may be appropriate : for a long FTP transfer a persistent TCP connection, for Web traffic via HTTP 1.1, persistent TCP connections with ON-OFF behavior and for other applications non-persistent TCP. For all these models we have used a unified approach. The insight obtained from this analysis is used in [11] and [15] to design architectures for providing QoS (Quality of service) to real and non-real time applications.

qwerty

The paper is organized as follows. In section 2 we study the small scale ON-OFF model. Section 3 studies the large scale ON-OFF model. In section 4 we combine the persistent, ON-OFF and the non-persistent TCP connections. Section 5 compares our theoretical results with the simulations.

2 Small Scale ON-OFF Model

In this section we first decribe our model and state the basic assumptions in section 2.1. We also explain our approach in this section. In section 2.2 we carry out the details for the infinite buffer case. Section 2.3 studies the system with RED control.

2.1 Model and our Approach

We consider a system with a single bottle-neck router. An exogenous flow, representing a superposition of UDP flows is passing through the router. We assume that the packets of this flow arrive as a Poisson process with rate λ_U. The packet lengths are iid with a general distribution with mean $E[s]$. We will denote $\lambda_U E[s]$ by ρ. The link speed is normalized to 1. There are N TCP connections sharing the queue with the exogenous stream. A TCP connection can be in an ON or OFF state. Some connections may always stay ON. Such connections will be called Persistent; the others in this section will be called ON-OFF. A connection turns from OFF state to the ON state, when it has generated a file to send. For the i^{th} TCP connection, we use the following notation and assumptions. The files generated are iid with an exponential distribution and mean $1/\mu(i), i = 1, \ldots, N$ (a model used in [19]). For connections which are always ON, we can take$\mu(i) = 0$. The OFF times are iid exponential with mean $1/\lambda(i)$. The total propagation delay is $\Delta(i)$. The packets are of random lengths with an iid distribution and a generic packet is of length $s_T(i)$. The receiver advertised maximum window size is $W_{max}(i)$. The queue is assumed to have an infinite buffer. In this case since there is no packet loss, the window size of TCP connection i increases to $W_{max}(i)$ and stays there. Thus, in the steady state analysis we can ignore slow start.

We make a few comments on the assumptions. A single bottleneck router is the usual assumption (see [1], [5], [8], [17]). Often it is true in practice. But our ideas can also be extended to the multiple router studied in case [17]. Infinite buffer assumption can be a valid approximation if the buffer length is large. Also our model captures many of the essential features of the system i.e. the effect of different propagation delays, packet sizes etc. In section 2.3 we will deploy RED at the queue. Then we will see that the analysis and the formulae we obtain for the infinite buffer queue can be directly carried over to a finite buffer queue with RED control. This is a realistic model of the system with RED control. Now the window size of a connection keeps changing. But we will capture this dynamics in the mean window size that we will compute.

The analysis in this section can be easily extended (with additional computational cost) to the case where the exogenous stream is an MMPP (Markov Modulated Poisson process) and all the exponential distributions are replaced by the phase type distributions [3]. Since any distribution (including the one with heavy tails) can be arbitrarily closely approximated by a phase type distribution, our analysis covers the general distributions. A long range dependent process can also be approximated by a Markov modulated process for any finite time scale of interest. Similar comments hold for later sections also.

To be able to incorporate the effect of packet sizes, propagation delays etc, we will use a hybrid technique. First we make a fluid level Markov model and then use the packet level details to obtain the mean bandwidth a particular TCP connection gets. These are used in the Markov chain to obtain its generator and then its stationary distribution and other performance parameters. In [5] the ON-OFF TCP model is studied when a heavy load condition is assumed: there is at least one packet from each TCP session in the bottleneck queue (because of the pure fluid model used). We relax this assumption to the scenario where the server is never idle. This assumption is much more realistic and will be valid if there are a large number of TCP connections sharing the bottleneck queue. We are able to do this (along with including the effect of different TCP parameters) because we have captured the packet level behavior in our models.

We will use the following result from [17]. If all the TCP connections are ON all the time, then TCP i will receive the throughput

$$\lambda_T(i) = \frac{(1-\rho)E[q_T(i)]}{\Sigma_j E[q_T(j)]E[s_T(j)]} \qquad packets/sec \tag{1}$$

where $E[q_T(i)]$ is the mean number of TCP i packets in the queue under stationarity. To compute $\lambda_T(i)$ explicitly, we need $E[q_T(j)]$ for each j. By Little's law,

$$\lambda_T(i)E[S_T(i)] = E[q_T(i)] \tag{2}$$

where $E[S_T(i)]$ is the mean sojourn time of i^{th} connection in the queue. Also, applying Little's law in the propagation pipe, we get

$$\lambda_T(i)\Delta(i) = W_{max}(i) - E[q_T(i)]. \tag{3}$$

These equations can be solved to obtain $E[q_T(i)]$ for each i. Then

$$E[S_T(i)] = \frac{\Sigma_j E[q_T(j)]E[s_T(j)]}{1-\rho} \qquad and \qquad \lambda_T(i) = \frac{W_{max}(i)}{\Delta(i) + E[S_T(i)]}.$$

Observe that $E[S_T(i)]$ is the same for all i. If $\Delta(i) = 0$ then $E[q_T(i)]$= $W_{max}(i)$.

2.2 Analysis

Let $a(t)$ denote the set of TCP connections which are in the ON state at time t. Let $A(t)$ be a row vector with its i^{th} component $A_i(t) = 1$ if the i^{th} TCP connection is in the ON state at time t and 0 otherwise. To analyze this process, we first consider the files being transmitted as a single entity and make a fluid model. Because a file generated by user i is exponentially distributed with mean $1/\mu(i)$ and other distributions involved are exponential, $\{A(t)\}$ can be considered a Markov chain. If connection i is always ON, then $A_i(t) = 1$ at all times. For such connections, the performance index of interest is the mean throughout under stationarity. For ON-OFF connections, we will be interested in knowing the mean file download time. To compute these quantities, in each state $A(t)$ we want to find out the instantaneous rate at which the file transfer is getting completed at a particular TCP connection. This rate $= \mu(i)\bar{\lambda}_T(i,t)$ depends upon the instantaneous bandwidth $\bar{\lambda}_T(i,t)$ (in bits/sec), which user i gets at time t. At any time t, $\bar{\lambda}_T(i,t)$ depends upon the TCP parameters $W_{max}(j), E[s_T(j)], \Delta(j)$ of the TCP connections in the ON state at time t. To compute $\bar{\lambda}_T(i,t)$, we need packet level analysis. We consider this in the following.

We approximate the throughput of different TCP connections when the system is in state $A(t)$ by the steady state throughput of the TCP connections in a system which has only the TCP connections (and the exogenous stream) that are ON in $A(t)$ (i.e. in the set $a(t)$). This is a reasonable assumption if the connection level dynamics are slow as compared to the packet level dynamics; an assumption which can be quite realistic. This would then imply that in each state $A(t)$ the queue reaches the steady state rather quickly. Hence, this model may not work satisfactorily for very small file sizes, although even in this case it gives satisfactory results under high load conditions. Internet traffic measurements show that for HTTP 1.1 the average file size around 28 KB ([13], [20]) Our simulations will show that these approximations work well for these file sizes and infact even with smaller size.

Under these assumptions, at time t, the throughput $\lambda_T(i,t)$ can be computed from the formulae (1) - (3) by considering only the TCPs in set $a(t)$. Thus for example,

$$\lambda_T(i,t) = \frac{(1-\rho)E[q_T(i)]}{\Sigma_{j\epsilon a(t)} E[q_T(j)]E[s_T(j)]}, \;\; i\epsilon a(t). \tag{4}$$

We use the instantaneous bandwidth obtained above in the fluid model. When the bandwidth obtained by an ON-OFF connection $i\epsilon a(t)$ is $\bar{\lambda}_T(i,t) = \lambda_T(i,t)E[s_T(i)]$, the rate at which its file (being exponential with mean $1/\mu(i)$) is completing its transmission is $\mu(i)\bar{\lambda}_T(i,t)$. This also implies that $\{A(t)\}$ can be considered a Markov chain. We can easily show that it is an irreducible finite state Markov chain and hence is ergodic with stationary distribution (say) π. The dimension of this chain is 2^N. Thus, analyzing this chain is possible for a small N only. We carry out details of this system in this section and will show in Section 5 that we obtain a good match with the simulations. In the next section we will modify this approach and will show that then we can handle a substantially larger system of TCPs.

The generator matrix $Q = (q(A,B))$ of the chain $\{A(t)\}$, obtained via the computation of $\bar{\lambda}_T(i,t)$ above, can be easily shown to be (we use the notation $A+i$ for a state obtained from a jump from state A when the i^{th} ON-OFF connection goes ON; similarly $A-i$ denotes the state when the i^{th} ON-OFF connection goes OFF)

$$q(A, A-i) = \mu(i)\bar{\lambda}_T(i,t), \;\; q(A, A+i) = \lambda(i).$$

Since it is an ergodic chain, its unique stationary distribution can be obtained by $\pi Q = 0$.

We consider the mean file download time $E[S(i)]$ of a particular ON-OFF user i as the performance measure of interest. $E[S(i)]$ is the mean ON time of the i^{th} connection under stationarity. Once π is computed, $E[S(i)]$ is obtained from

$$\frac{E[S(i)]}{E[S(i)] + 1/\lambda(i)} = \Sigma_{A:A_i=1}\pi(A).$$

For persistent connection i, the mean throughput

$$\bar{\lambda}_T(i) = \Sigma_A \bar{\lambda}_T(i,A)\pi(A)$$

where $\bar{\lambda}_T(i, A)$ is the throughput of i in state A.

Another performance measure of interest is the mean sojourn time $E[S_U]$ of the packets of the exogenous stream in the queue. Under stationarity it can be computed from

$$E[S_U] = \Sigma_{A:A\neq(0,0,\ldots,0)} E[S_U|A(t) = A]\pi(A) + \left(\frac{\lambda_U E[s^2]}{2(1-\rho)} + E[s]\right)\pi((0,0,\ldots,0)),$$

where $E[S_U|A(t) = A]$ can be computed from [17] and is in fact approximately equal to $E[S_T(i)]$ computed above. $A = (0, 0, \ldots, 0)$ indicates that all the TCPs are in the OFF state and then the mean sojourn time of the UDP packets is as in an $M/GI/1$ system with only the UDP load. Since $E[S_U]$ is approximately equal to $E[S_T(i)]$ (because $\pi((0, ..., 0))$ will usually be small) $E[S_U]$ increases if $W_{max}(j)$ increases for any j.

2.3 System with RED Control

When the queue also deploys the RED algorithm, then $W_{max}(i)$ in the above computations will be replaced by $E[W(i)]$, the steady state mean window size when the system has only $a(t)$ TCP connections. This is a mean value kind of approximation. Extensive simulations in [17] and in section 5 justify this approximation. The value of $E[W(i)]$ depends upon the TCP window dynamics and $p(\hat{q})$ i.e. the probability of packet loss in the RED algorithm under steady state, where $\hat{q}$ is the RED average queue length parameter. Given $p(\hat{q})$, various approximations for $E[W(i)]$ are available (see *e.g.,* [14], [18]) . In this paper we have used the approximation developed in [18], applicable for the TCP Reno case. Observe that knowing $E[W(i)]$ for different versions of TCP makes our approach applicable for these other cases as well. Computation of $\hat{q}$ under steady state when the system has only $a(t)$ TCP connections can be done as in [17] using a decomposition approach. Then the steady state of an ODE provides this value. These computations need to be carried out for each set $a(t)$. Then replacing $W_{max}(i)$ by $E[W(i)|a(t)]$ in (4) - (3) will provide $\lambda_T(i, t)$ for each i. These can be used in subsequent computations to obtain the generator Q, the stationary distribution π, $E[S(i)]$ and $E[S_U]$. Simulation results in section 5 will show that we obtain a good approximation of the performance parameters. When RED is deployed we have replaced $W_{max}(i)$ with $E[W(i)]$. Since $E[W(i)]$ is smaller than $W_{max}(i)$, $E[S_U]$ for this system is smaller than the system without RED studied above. Also, if $W_{max}(i) \neq W_{max}(j)$, but both are not too small, we will often have $E[W(i)] \approx E[W(j)]$. Thus for the system with RED control we may not see the effect of different $W_{max}(i)$ on system performance. Furthermore, we will see that for the example in simulations in section 5, $E[S_U]$ with RED is significantly less than $E[S_U]$ without RED but the mean download time for TCP connections remains largely unaffected. Similar conclusions hold for the next two sections.

3 Large Scale ON-OFF Model

We observed above that the state space of the Markov chain $\{A(t)\}$ exponentially explodes as the number of ON-OFF TCP connections N increases. In this section, to reduce the dimensionality, we study the averaged system. However, it is done in such a way that for a particular TCP connection i with given $\Delta(i), W_{max}(i), E[s_T(i)]$ we can compute its mean download time $E[S(i)]$. This can of course be substantially different from the overall mean download time $E[S]$ for the system (as for example computed in [5], [12]) and is actually the performance of interest for the i^{th} user.

In section 3.1 we will explain our approach for this section. Section 3.2 carries out the analysis for the infinite buffer system. In section 3.3 we extend the results to the queue with RED Control.

3.1 Our Approach

The notation will remain as in Section 2. Let N_t be the number of TCP connections ON at time t. The total bandwidth available to the TCP connections at any time is $1 - \rho$. For connection $i, \Delta(i), W_{max}(i)$, the mean file size $1/\mu_i$ and the mean packet size $E[s_T(i)]$ are fixed. But these can be different for different i. In this section we take these quantities as random over the population of N TCP connections ($1/\mu_i$ and $1/\lambda_i$ are random over the ON-OFF connections only). We denote by $1/\mu$ the mean file size and by $1/\lambda$ the mean OFF time where the mean is taken over the whole population of ON-OFF TCP connections. Unlike in section 2, now we will concentrate on the performance of a particular connection i

while the identity of other connections is erased. For connection i the identity of other connections is not important but only the overall effect of them on the instantaneous bandwidth it gets. Thus by erasing their identities but replacing their overall effect by the population averaged TCP parameters, we can obtain the performance of the i^{th} connection. This approximation will be good particularly when N is large. Now $\{N_t\}$ can itself be taken as a Markov chain. Observe that for a system with N ON-OFF TCP connections, the Markov chain $\{A(t)\}$ of the last section has 2^N states while $\{N_t\}$ has $N+1$.

3.2 Analysis for infinite buffer size

At time t, the Markov chain N_t goes to state $N_t - 1$ (for $N_t > 0$) at a rate which can be approximated by $\mu(1-\rho)$ (assuming work conserving, non-idle server) and to state $N_t + 1$ at a rate $(N - N_t)\lambda$. $\{N_t\}$ is a finite state, irreducible Markov chain and we can easily compute its stationary distribution π.

Next we compute the mean download time $E[S(i)]$ for ON-OFF TCP connection i. We also compute the mean queuing delay of the exogenous stream. The general approach will be as in section 2. We first compute the mean instantaneous throughput $\bar{\lambda}_T(i,t)$ for the i^{th} connection and then compute $E[S(i)]$. However, the details differ significantly because of the averaging being done in this section.

Let the i^{th} connection for which we want to compute $E[S(i)]$ be in the ON state at time t. The mean number of connection i packets in the queue at time t will depend upon N_t and hence we write it as $E[q_T(i)|N_t]$. Thus from (4) the expected throughput in the state N_t (in packets/sec) is given by

$$E[\lambda_T(i,t)|N_t] = \frac{(1-\rho)E[q_T(i)|N_t]}{\Sigma_{j=1}^{N_t} E[q_T(j)|N_t]E[s_T(j)]}.$$

We approximate the denominator in this expression by

$$N_t E[E[q_T(j)|N_t]E[s_T(j)]|N_t]$$

where the outer conditional expectation is with respect to the joint distribution of $(\Delta(j), W_{max}(j), E[s_T(j)])$, $j \epsilon N$ and it denotes the overall stationary mean work load of a TCP connection in the queue given that N_t TCP connections are ON. The term $E[q_T(j)|N_t]$ inside the expectation is with respect to the $\Delta(j), W_{max}(j), E[S_T(j)]$ specific to the j^{th} connection. We denote $E[E[q_T(j)|N_t]E[s_T(j)]|N_t]$ by α_{N_t}. Therefore, we obtain

$$E[\lambda_T(i,t)|N_t] \approx \frac{(1-\rho)E[q_T(i)|N_t]}{N_t \alpha_{N_t}}. \tag{5}$$

Similarly, we obtain from (3)

$$E[\lambda_T(i,t)|N_t] = \frac{W_{max}(i) - E[q_T(i)|N_t]}{\Delta(i)}. \tag{6}$$

From (5) and (6) we obtain

$$E[q_T(i)|N_t] = \frac{N_t \alpha_{N_t} W_{max}(i)}{(1-\rho)\Delta(i) + N_t \alpha_{N_t}}. \tag{7}$$

Multiplying both sides with $E[s_T(i)]$ and again taking conditional expectation w.r.t. N_t we obtain

$$\alpha_{N_t} = N_t . \alpha_{N_t} E\left[\frac{W_{max}(i).E[s_T(i)]}{(1-\rho)\Delta(i) + N_t \alpha_{N_t}} |N_t\right]. \tag{8}$$

Since the joint distribution of $(\Delta(i), W_{max}(i), E[s_T(i)])$ for the overall population is supposed to be known, from (8) we compute α_{N_t}. Then from (5) and (6) we obtain

$$E[\lambda_T(i,t)|N_t] = \frac{W_{max}(i)}{\Delta(i) + \frac{N_t \alpha_{N_t}}{1-\rho}}. \tag{9}$$

Equation (9) gives the dependence of the conditional throughput of connection i on its different TCP parameters. In particular, it is directly proportional to $W_{max}(i)$ and $(1-\rho)$ and inversely proportional to $\Delta(i)$. This dependence gets reflected in the unconditional throughput $E[\lambda_T(i)] = \Sigma_n E[\lambda_T(i)|N_t = n]\pi(n)$ of connection i also. The throughput of connection i in *bps* is $E[\lambda_T(i)|N_t]E[s_T(i)]$ and hence is directly proportional to $E[s_T(i)]$. Of course, for persistent TCP connections $E[\lambda_T(i)]$ is the main performance index.

Observe that if $W_{max}(j), E[s_T(j)], \Delta(j)$ are independent random variables, then the expectation in (8) depends only on the mean of $W_{max}(j)$ (say w_m) and $E[s_T(j)]$ (say s_m) and not on their individual distributions. As an example assume that they are independent and $\Delta(j)$ is uniformly distributed on the interval (a, b), then (8) gives

$$\alpha_{N_t} = \frac{(1-\rho)}{2N_t}\left[\frac{(b-a)(e^{\frac{(b-a)(1-\rho)}{N_t w_m s_m}}+1)}{(e^{\frac{(b-a)(1-\rho)}{N_t w_m s_m}}-1)} - (a+b)\right].$$

Observe that α_{N_t} is the overall stationary mean workload of a TCP connection in the queue, given that N_t TCP connections are ON. Thus the conditional mean sojourn time in the queue is

$$E[S_T|N_t] = \frac{N_t.\alpha_{N_t}}{1-\rho} \tag{10}$$

and the mean sojourn time of the TCP packets in the queue is

$$E[S_T] = \Sigma_{n=1}^{N} E[S_T|N_t = n]\pi(N_t = n|N_t > 0) = \Sigma_{n=1}^{N} E[S_T|N_t = n]\frac{\pi(n)}{1-\pi(0)}.$$

The mean sojourn time of the exogenous stream is

$$E[S_U] = \Sigma_{n=1}^{N} E[S_T|N_t = n]\pi(n) + \left(\frac{\lambda_U E[s^2]}{2(1-\rho)} + E[s]\right)\pi(0).$$

Finally we compute the mean file download time $E[S(i)]$ of a connection i. Let us assume that when the i^{th} connection turns ON, the N_t process is stationary and has the distribution π. This may not be strictly correct but we make this approximation. Let $N_t = n-1$ at the time the i^{th} connection turns ON and then let $E[S(i,n)]$ be the mean time for the i^{th} connection to download its file. Then the mean time for the next jump of the process $\{N_t\}$ is $E[\tau_n] = 1/(\lambda(N-n) + (1-\rho)\mu)$. Also, the probability that this jump corresponds to the completion of transmission of the file of the i^{th} connection is $\mu(i)E[\lambda_T(i,t)|N_t = n]E[\tau_n]$. Therefore, we can write the equations

$$\begin{aligned} E[S(i,n)] &= (E[\tau_n])^2\mu(i)E[\lambda_T(i,t)|N_t = n] + \lambda(N-n)E[\tau_n](E[S(i,n+1)] + E[\tau_n]) \\ &+ [\mu(1-\rho) - \mu(i)E[\lambda_T(i,t)|N_t = n]]E[\tau_n](E[S(i,n-1)] + E[\tau_n]), \\ & n = 1,\ldots,N \end{aligned} \tag{11}$$

where on the right, the first term corresponds to the first jump being due to the i^{th} connection going OFF, the second term due to a TCP connection turning ON and the third term is due to a TCP connection (other than the i^{th}) turning OFF. We can solve this set of simultaneous equations to obtain $E[S(i,n)], n = 1,\ldots,N$. Then

$$E[S(i)] = \Sigma_{n=1}^{N} E[S(i,n)]\frac{\pi(n-1)}{1-\pi(N)}. \tag{12}$$

3.3 System with RED Control

Now we consider the model when the queue employs the RED control. Then depending upon the steady state RED average queue length parameter $\hat{q}^*$, a packet on arrival at the queue will be dropped with probability $p(\hat{q}^*)$. If $p(\hat{q}^*)$ is small enough then the stationary distribution π of N_t remains as in the infinite buffer case. Now the TCP window size can change. In that case as in the last section, instead of $W_{max}(i)$ in the equation in this section, we use $E[W(i)]$, the steady state mean window size for connection i (also for other connections) when the drop probability is $p(\hat{q}^*)$. In particular we first compute $E[\lambda_T(i,t)|N_t = n]$ and then $E[S(i,n)]$ from (11) and finally the mean download time $E[S(i)]$ from (12).

Of course we do not know the $\hat{q}^*$ for our system. For this we follow the ODE approach described in [17], where the steady state value of the ODE provides $\hat{q}^*$. To compute it we require $E[q|\hat{q}]$, the mean total queue length of the system for various values of $\hat{q}$. We do this in the following.

From [17] we obtain that

$$E[q|N_t] = \Sigma_{i=1}^{N_t} E[q_T(i)|N_t] + \frac{\lambda_U}{1-\rho}\Sigma_{i=1}^{N_t} E[q_T(i)|N_t]E[s_T(i)].$$

The second term on the right, as in (5) above is written as $\lambda_U N_t \alpha_{N_t}/(1-\rho)$. For the first term on the right, from (7) we obtain an approximation

$$N_t^2 \alpha_{N_t} E\left[\frac{E[W(i)]}{(1-\rho)\Delta(i) + N_t\alpha_{N_t}}|N_t\right]$$

where we have replaced $W_{max}(i)$ by $E[W(i)]$ (for a given $\hat{q}$, p($\hat{q}$) is known and hence we can compute $E[W(i)]$) . Therefore,

$$E[q|N_t] \approx N_t^2 \alpha_{N_t} E\left[\frac{E[W(i)}{(1-\rho)\Delta(i) + N_t\alpha_{N_t}}|N_t\right] + \frac{\lambda_U N_t \alpha_{N_t}}{1-\rho}. \tag{13}$$

Then $E[q|\hat{q}] = \Sigma_{n=1}^{N} E[q|N_t = n]\frac{\pi(n)}{1-\pi(0)}$. We use this $E[q|\hat{q}]$ in plotting the ODE. The steady state value of the ODE then gives us $\hat{q}^*$.

4 Combined ON-OFF and Non-persistent Model

In this section we assume that along with the ON-OFF TCP sources there is another stream of TCP sources which are Non-Persistent or transitory in nature. The persistent TCP connections can be included as in previous sections and hence we no longer discuss them in this section. The Non-persistent TCP connections are arriving as a Poisson stream with rate λ_N. A connection of this type arrives with a particular set of parameters $W_{max}, E[s_T]$ and Δ, downloads a file with exponentially distributed size and leaves the system. We will assume in the beginning that the queue has an infinite buffer. Later on we will consider the queue with the RED control also. Since for the Non-Persistent connection, only one file is transmitted which may not be large, the slow start phase affects the download time. Therefore, now we will also model the effect of the slow start phase. The analysis of this section can be extended to the case where the Non-Persistent connections arrive as an MMPP process and the file size distributions are phase type, but this extension has not been done in this paper.

The total bandwidth available to the TCP connections is still $1-\rho$. Let $1/\lambda$ be the mean OFF time of the Persistent TCP connections. Let N be the total number of Persistent ON-OFF sources. Let $1/\mu$ denote the mean file size taken over the whole population of (Persistent as well as the Non-Persistent) TCP connections. For a connection i, $W_{max}(i), E[s_T(i)], \Delta(i)$ and $\mu(i)$ are as before. Let the overall statistics of $W_{max}, E[s_T]$ and Δ be known for the total population (Persistent as well as Non-Persistent). The following analysis can be carried out when the statistics of the persistent and the non-persistent population are separately considered, but for simplicity we work with the overall statistics.

Let $N_t(1)$ be the number of Persistent TCP connections which are ON at time t and $N_t(2)$ be the number of Non-Persistent TCP connections which are present in the system at time t. $\{N_t(1), N_t(2)\}$ can be taken as a continuous time countable state Markov chain. Its generator matrix $Q = q((i_1, i_2), (j_1, j_2))$ is given as,

$$\begin{aligned}
q((n_1, n_2), (n_1+1, n_2)) &= (N - n_1)\lambda, \\
q((n_1, n_2), (n_1, n_2+1)) &= \lambda_N, \\
q((n_1, n_2), (n_1-1, n_2)) &= \frac{n_1}{n_1+n_2}\mu(1-\rho), \\
q((n_1, n_2), (n_1, n_2-1)) &= \frac{n_2}{n_1+n_2}\mu(1-\rho), \text{ for } n_1 = 0, \ldots, N, n_2 = 0, 1, 2, \ldots.
\end{aligned}$$

It is an irreducible Markov chain and let its stationary distribution be π. The condition for ergodicity of this Markov chain is $\lambda_N < \mu(1-\rho)$ (see Appendix A for a proof).

Next we want to calculate the throughput for a Persistent TCP connection i with parameters $W_{max}(i), \Delta(i)$ and $E[s_T(i)]$, when the system is under stationarity. Its throughput when $(N_t(1), N_t(2))$ TCP connections are in the system is given by

$$E[\lambda_T(i,t)|N_t(1), N_t(2)] \approx \frac{(1-\rho)E[q_T(i)|N_t(1), N_t(2)]}{(N_t(1)+N_t(2))\alpha_{N_t(1),N_t(2)}},$$

where $\alpha_{N_t(1),N_t(2)} = E[E[q_T|N_t(1), N_t(2)]E[s_T(j)]|N_t(1), N_t(2)]$. In this the outer expectation is with respect to the joint distribution of $W_{max}(j), E[s_T(j)], \Delta(j)$ taken over the total population of Persistent as well as Non-Persistent connections. We can solve for $\alpha_{N_t(1),N_t(2)}$ in the same way as in Section 3. Then,

$$E[\lambda_T(i,t)|N_t(1), N_t(2)] = \frac{W_{max}(i)}{\Delta(i) + \frac{\alpha_{N_t(1),N_t(2)}(N_t(1)+N_t(2))}{1-\rho}}.$$

Again one can see the effect of $W_{max}(i), \Delta(i), (1-\rho)$ and $E[s_T(i)]$ on the throughput of connection i.

For a Non-Persistent TCP connection we use the above formula with a modification, because now we have to take care of the slow start phase. Assume the initial window size to be w_0 and that the receiver sends one ack for each TCP packet it receives. The modifications required when an ack is sent for k packets is obvious. Then the amount of data d (in number of packets) sent after i rounds of slow start is given by

$$d = (1 + 2 + 2^2 + 2^3 + \ldots + 2^{i-1})w_0 = (2^i - 1)w_0.$$

Therefore, $i = log_2(\frac{d}{w_0} + 1)$. The window size after sending d amount of data becomes

$$W_{th} = \frac{d}{2} + \frac{w_0}{2}.$$

In the case when $W_{th} > W_{max}(i)$ then we have to find out how much data was sent in slow start and how much after reaching $W_{max}(i)$. The number of rounds required to reach $W_{max}(i)$ is $log_2(\frac{W_{max}(i)}{w_0}) + 1$. The amount of data transfered by the end of this round is $2W_{max}(i) - w_0$. So the remaining data $d - 2W_{max}(i) + w_0$ will be sent using $W_{max}(i)$. Therefore, the number of rounds required by the remaining data is $\frac{d - 2W_{max}(i) + w_0}{W_{max}(i)}$. Now we define a normalized window size by

$$W_n(i) = \begin{cases} \frac{d}{log_2(\frac{d}{w_0}+1)} & \text{if } W_{th} \le W_{max}(i), \\ \\ \frac{d}{log_2(\frac{W_{max}(i)}{w_0}) + \frac{d+w_0}{W_{max}(i)} - 1} & \text{if } W_{th} > W_{max}(i). \end{cases}$$

which represents the window size if we were to transfer the data using the same number of rounds but with a constant widow size $W_n(i)$. In this paper we have taken $w_0 = 2$. For the Non-Persistent connections we will use $W_n(i)$ in place of $W_{max}(i)$ in the above formula for $E[\lambda_T(i,t)|N_t(1), N_t(2)]$.

From the above relation we observe that $W_n(i)$ depends upon $W_{max}(i)$ and is independent of the other TCP parameters $\Delta(i)$ and $E[s_T(i)]$. Thus, qualitative effect on $E[\lambda_T(i,t)|N_t(1), N_t(2)]$ of a TCP Non-persistent connection i, with parameters $\Delta(i), W_{max}(i), E[s_T(i)]$ and $(1-\rho)$ is the same as for a persistent ON-OFF connection.

Finally we compute the mean file download time $E[S(i)]$ of a persistent TCP connection i. Let us assume that when the i^{th} connection turns ON, the process $\{N_t(1), N_t(2)\}$ is stationary with distribution π. This is an approximation, justified via simulation in the next section. Let (n_1, n_2) be the state after the i^{th} connection turns ON. Let $E[S_i(n_1,n_2)]$ be the mean time for the i^{th} connection to download its file starting from the state (n_1, n_2). The mean time for the next jump of the process $\{N_t(1), N_t(2)\}$ is $E[\tau_{n_1,n_2}] = \lambda_N + \lambda(N-n_1) + (1-\rho)\mu$. Also, the probability that this jump corresponds to the completion of transmission of the file of the i^{th} persistent TCP connection is $\mu(i)E[\lambda_T(i,t)|n_1,n_2]E[\tau_{n_1,n_2}]$. Therefore, for $n_1 = 1, \ldots, N, n_2 = 0, 1, 2, \ldots$.

$$\begin{aligned}
E[S_i(n_1,n_2)] &= (E[\tau_{n_1,n_2}])^2 \mu(i) E[\lambda_T(i,t)|n_1,n_2] \\
&+ \lambda(N-n_1)E[\tau_{n_1,n_2}](E[S_i(n_1+1,n_2)] + E[\tau_{n_1,n_2}]) \\
&+ \lambda_N E[\tau_{n_1,n_2}](E[S_i(n_1,n_2+1)] + E[\tau_{n_1,n_2}]) \\
&+ (\frac{n_1}{n_1+n_2}\mu(1-\rho) - \mu(i)E[\lambda_T(i,t)|n_1,n_2])E[\tau_{n_1,n_2}](E[S_i(n_1-1,n_2)] + E[\tau_{n_1,n_2}]) \\
&+ \frac{n_2}{n_1+n_2}\mu(1-\rho)E[\tau_{n_1,n_2}](E[S_i(n_1,n_2-1)] + E[\tau_{n_1,n_2}]),
\end{aligned} \tag{14}$$

where on the right the first term corresponds to the first jump being due to the i^{th} connection, the second is due to the arrival of a persistent TCP connection, the third due to a non-persistent connection arrival, fourth due to a persistent connection (other than the i^{th}) turning OFF and the fifth term is due to a non-persistent connection turning OFF.

Unlike in section 3, this is an infinite set of equations. However we can truncate n_2 to a suitably large M and solve this set of equations for $E[S_i(n_1,n_2)], n_1 = 1, \ldots, N$ and $n_2 = 0, \ldots, M$. Then,

$$E[S(i)] = \frac{\Sigma_{n_1=1}^{N} \Sigma_{n_2=0}^{M} E[S_i(n_1,n_2)]\pi(n_1-1,n_2)}{1 - \Sigma_{n_2=0}^{M}\pi(N,n_2)}.$$

Next we obtain the mean file download time $E[S_N(j)]$ for the Non-Persistent flow j. In this case the TCP connections arrive as a Poisson process. Therefore, by PASTA, when a Non-Persistent connection arrives it will find the $\{(N_t(1), N_t(2))\}$ process under stationarity with the distribution π. Now we can solve (14) as a set of simultaneous equations to obtain $E[S_j(n_1,n_2)]$, $n_1 = 0, \ldots, N, n_2 = 1, \ldots, M$ for some suitable large M. However, in (14) only the last two terms get modified from the persistent case to the non-persistent case. Moreover the symmetry of the equations and the common statistics ensure that the magnitude of the change is minimal. Hence, we have used the same set of equations and solved them to get $E[S_j(n_1,n_2)]$. Then

$$E[S_N(j)] = \frac{\Sigma_{n_1=0}^{N} \Sigma_{n_2=1}^{M} E[S_j(n_1,n_2)]\pi(n_1,n_2-1)}{1 - \Sigma_{n_1=0}^{N}\pi(n_1,M)}.$$

Now we consider the case when the queue deploys RED control. The analysis goes exactly in the same way as in Section 3. We have to use $E[W(i)]$ in place of $W_{max}(i)$ in the formulas for throughput etc. In particular for the Persistent TCP connections we compute $E[\lambda_T(i,t)|N_t(1), N_t(2)]$ and $E[S_i(n_1,n_2)]$ and then the mean download time $E[S(i)]$. For the Non-Persistent connections we calculate the normalized window size $W_n(j)$ now with respect to $E[W(j)]$ instead of $W_{max}(j)$ to take care of the slow start phase. Finally we get $E[S_j(n_1,n_2)]$ and the mean download time $E[S_N(j)]$.

5 Simulation results

In this section we present simulation results for a small scale system (5 TCP), a large scale system (50 TCP) and a system of persistent ON-OFF and non-persistent TCP connections. We will compare these simulations with the theoretical results obtained in the previous sections. There are no persistent TCPs in these simulations. We have used ns version 2.1b8 for these simulations. Only TCP Reno is considered. The link speed is 10Mbits/sec. The simulations were run for 2000s. This run length provides stable estimates. In all the models the UDP parameters are same. It generates packets of length 750 bytes as a Poisson process with rate 500 packets/sec. For RED simulations $\beta = 10^{-4}$ and p_{max} is kept at 0.1.

For the small scale model we simulated a system with 5 TCP connections and one UDP connection. The TCP connections are persistent, each behaving as an ON-OFF source. The $\Delta(i), W_{max}(i), E[s_T(i)]$, the mean file size $1/\mu(i)$ and the mean OFF time $1/\lambda(i)$ for the 5 TCPs connections are provided in Table 1. An infinite buffer queue and also a finite length queue with RED control are considered. The RED parameters are $T_{min} = 20, T_{max} = 60$. Table 1 provides the mean download times via simulations as well as from theory. In the data observe that TCP2 (almost) takes the same time to download as TCP1 even though it is downloading a file of half the size. This is because of the smaller packet size for TCP2 (all other parameters are same for both the TCPs). This observation justifies the hybrid analysis instead of pure fluid. Also observe that for TCP4 and TCP5, all parameters except the OFF time are same but the download time of TCP5 is more than that of TCP4 (this may appear surprising but is captured by our theory). This is explained by the fact that because the mean OFF time of TCP5 is more, it is in the OFF state for a longer period and hence the other TCPs get better throughput and download times than TCP5. The dependence of the mean download time on the various TCP parameters as observed here are not provided by any of the other studies. In the table also note that the UDP sojourn time has been reduced by about 80% with the usage of RED as prediced by our theory. This supports RED as an effective queue length management technique.

Next we consider the large scale model. For this we have considered a system with 50 TCPs. It is well beyond the reach of the small scale model but has very moderate computational requirements within the large scale model. A much larger number of TCP connections can be handled with this approach. The parameters for different TCP connections have the following distributions: W_{max} is uniformly distributed over (10,20), Δ is uniformly distributed over (0,0.2), $E[s_T]$ is uniformly distributed over (100,500) and these random variables are independent of each other. The mean OFF times and the file sizes for all the connections are same and are taken as $1/\lambda = 3s$ and $1/\mu = 200KB$. For the RED case $T_{min} = 100, T_{max} = 500$.

The mean download time for all the 50 TCP connections via simulations and theory are plotted in Figs 1 and 2. Fig 1 provides the results for the infinite buffer case and Fig 2 for the RED controlled queue. We plot the theoretical (obtained via an ODE) and the simulated $\hat{q}$ and $n\beta$ in Fig 3. The errors in the download time estimation for the RED controlled queue are more than in the infinite buffer case. But for both the systems, errors are within ten percent for most of the TCP connections. Observe that there is significant difference in the mean download times of the various TCP connections although all of them have the same λ and μ. As an illustration, connection 42 with the parameters $\Delta = 0.142, W_{max} = 14, E[s_T] = 111$ has a download time of 29 secs and connection 43 with the parameters $\Delta = 0.036, W_{max} = 19, E[s_T] = 196$ has a download time of 8 secs. None of the previous studies have been able to capture the effect of all three parameters $\Delta, W_{max}, E[s_T]$ on the download times. The mean queue sojourn time of the UDP connections in the infinite buffer case is 114 secs (theory) and in case of RED it is 51 secs (theory). The corresponding values via simulation are 106 secs and 45 secs. Observe that although the UDP sojourn times have been halved there has not been much change in the file down load times of the TCP connections.

Finally we consider the system with combined Persistent as well as Non-Persistent TCP connections. The simulation parameters are $1/\lambda = 0.5, 1/\lambda_N = 0.3, 1/\mu = 27KB$. We have reduced the mean file size to $27KB$ to show that our approach works well for small file sizes as well. The parameters for different TCP connections have the following distributions: W_{max} is uniformly distributed over (10,20), Δ is uniform over (0,0.05) sec and $E[s_T]$ is uniform over (100,500) bytes and these random variables are independent of each other. Number of Persistent ON-OFF TCPs are 50 with flow ids $1, \ldots, 50$. The Non-Persistent traffic has four classes with flow ids $51, \ldots, 54$. Δ for all the four Non-Persistent flow classes is $10ms$ and the W_{max} and the $E[s_T]$ parameters are as follows: 10, 100 for flow id 51; 20, 100 for flow id 52; 10, 500 for flow id 53 and 20, 500 for flow id 54. The value of M in (14) is kept at 30. Fig 4 gives the comparison of the mean download times for the infinite buffer case. For connections 51 and 53, observe that the packet size for connection 53 is five times that for connection 51 and with other parameters remaining the same, the mean file down load time for connection 53 is almost one fifth of that for connection 51. Similarly observe between connection 51 and 52 how the download time decreases with an increase in W_{max}.

For the RED case $T_{min} = 100, T_{max} = 500$. Fig 5 gives the mean download times for the RED case. One can see a good match between theory and simulation. Almost all errors are within 10%. One can see that for connection numbers 51 and 52 and similarly 53 and 54, the mean download time has become almost equal. This is in contrast to their values in Fig 4. This is because in the RED case we are using $E[W]$ instead of W_{max} and all the other parameters being same we end up getting the same download times for them. Observe that packet size continues to have a significant impact on the download times. We also plot the theoretical and the simulated $\hat{q}$ vs $n\beta$ in Fig 6.

The error is more for the non-persistent connections especially for larger Δ. The error can be attributed to the manner in which we use the slow start approximation of W_{max}. The error is more prominent when RED comes into picture as can be seen from Fig 5.

6 Conclusions

Made simple analytically tractable models for TCP-persistent, non-persistent and TCP ON-OFF which model the TCP connections for long lived connections, short lived connections and for interactive webtraffic. The different connections can have different propagation delays, packet lenghts and maximum window size. Our analysis can handle several hundred TCP connections. Our formulae for throughputs and mean file download times explicitly show the impact of these parameters on the different connections. In particular we observe that for the persistent and ON-OFF connections, the TCP dynamics can be summarized in its mean window size but for the non-persistent TCP's the show start phase of the TCP plays an important role in deciding above mentioned performance indices. Verified these theortical formulae with NS Simulations and found that they match well with the NS-Simulation results. Our results also show that using RED control does indeed reduce the mean delays of UDP and TCP traffic in the router significantly but does not affect the throughput and mean download time of the TCP connections.

The insight obtained from this study has been used in developing schemes to provide QoS (Quality of Service) to real, and non-real time and interactive applications in Internet in [11] and [15] where all the above three models of TCP are employed.

Appendix A
Proof of ergodicity of the Markov Chain in section 4

Let $\{(N_k(1), N_k(2))\}$ be the embedded discrete Markov chain at the jump times of chain $\{(N_t(1), N_t(2))\}$. We show the ergodicity of this jump chain. This will imply the ergodicity of $\{(N_t(1), N_t(2))\}$.

To prove the ergodicity of $\{(N_k(1), N_k(2))\}$ we use the Lyapunov method where we have to find a non-negative function $f(n_1, n_2)$ such that

$$E[f((N_k(1), N_k(2))|(N_{k-1}(1), N_{k-1}(2))] - f((N_{k-1}(1), N_{k-1}(2)) \leq -\epsilon \tag{15}$$

for some $\epsilon > 0$ and for $(N_{k-1}(1), N_{k-1}(2))$ outside a finite set. Let $f(n_1, n_2) = n_2$ and let the finite set be $\{(n_1, n_2) : n_1 + n_2 \leq M_s\}$, where $M_s > N$. This would then imply that $n_2 \neq 0$ outside this set. Later on we will specify how to pick M_s. To show (15) consider two cases: when $n_1 = 0$ and when $n_1 \neq 0$.

When $n_1 = 0$, (15) can be written as

$$\frac{N\lambda}{D} n_2 + \frac{\lambda_N}{D}(n_2 + 1) + \frac{\mu(1-\rho)}{D}(n_2 - 1) - n_2 \leq -\epsilon.$$

where $D = N\lambda + \lambda_N + \mu(1 - \rho)$. Then,

$$\frac{n_2}{D}(N\lambda + \lambda_N + \mu(1-\rho)) - n_2 < \frac{\mu(1-\rho) - \lambda_N)}{D}.$$

The above holds if $\lambda_N < \mu(1-\rho)$.

When $n_1 \neq 0$, then (15) can be written as

$$\frac{(N-n_1)\lambda}{D} n_2 + \frac{n_1\mu(1-\rho)n_2}{(n_1+n_2)D} + \frac{\lambda_N}{D}(n_2+1) + \frac{n_2\mu(1-\rho)(n_2-1)}{(n_1+n_2)D} - n_2 \leq -\epsilon,$$

where $D = (N-n_1)\lambda + \mu(1-\rho) + \lambda_N$. Then,

$$\frac{n_2}{D}((N-n_1)\lambda + \mu(1-\rho) + \lambda_N) - n_2 < \frac{n_2\mu(1-\rho)}{(n_1+n_2)D} - \frac{\lambda_N}{D}.$$

This implies that $\lambda_N < \frac{n_2\mu(1-\rho)}{(n_1+n_2)}$. Here, by making M_s large enough $\frac{n_2}{(n_1+n_2)}$ can be made arbitrarily close to 1. Therefore the required condition is $\lambda_N < \mu(1-\rho)$.

References

[1] A. A. Abouzeid, S. Roy, M. Azizoglu, *Stochastic modeling of TCP over lossy links*, in Proc. IEEE INFOCOM 2000.

[2] E. Altman, J. Bolot, D. Elouadghiri, M. Erramdani, P.Brown and D. Collange, *Performance modeling of TCP/IP in a wide-area network*, INRIA Tech. Report, March 1997.

[3] S. Asmussen, *Applied Probability and Queues*, John Wiley and Sons, 1987.

[4] F. Baccelli, D. Hong, *The AIMD model of TCP sessions sharing a common router*, in 39th Annual Allecton conf. on Communication Control and Computing, Allecton Park, 2001.

[5] A. W. Berger and Y. Kogan, *Dimensioning bandwidth for elastic traffic in high-speed data networks*, IEEE/ACM Transaction Networking, Vol 8, 2000, 634-654.

[6] T. Bonald, *Comparison of TCP Reno and TCP Vegas: efficiency and fairness*, Performance Evaluation, Vol. $36-37, 1999, 307-322$.

[7] B. Braden *et al.*, *Recommendations on queue management and congestion avoidance in the Internet*, Internet RFC 2309, April 1998.

[8] P. Brown, *Resource sharing in TCP connections with different round trip times*, in IEEE INFOCOM, 2000.

[9] N. Cardwell, S. Savage, T. Anderson, *Modelling TCP Latency* , in IEEE INFOCOM, 2000, Tel Aviv, Israel, March, 2000.

[10] S. Floyd and V. Jacobson. *Random Early Detection Gateways for Congestion Avoidance*, IEEE/ACM Transactions on Networking, Vol. $1, 1993, 397-413$.

[11] V. Kamble and V Sharma,*A simple approach to provide QoS and fairness in Internet*, in Proc. International Conf. on Signal Processing and Communications, (SPCOM04), Bangalore, Dec. 2004.

[12] A. A. Kherani, A. Kumar, *Stochastic models for throughput analysis of randomly arriving elastic flows in the Internet*, in Proc. IEEE INFOCOM 2002.

[13] B. A. Mah, *An empirical model of HTTP network traffic*, in Proc. of IEEE INFOCOM 1997, pp 592-600.

[14] J. Padhye, V. Firoiu, D. Towsley and J. Kurose, *Modeling TCP throughput: A simple model and its empirical validation*, Proc. SIGCOMM 98, pp 303-314.

[15] V. Reddy, V Sharma and M. B. Suma, *Providing QoS to TCP and real time connections in the Internet*, Queueing Systems, Vol 46, 2004, 461-480.

[16] B. Sikdar, S. Kalyanaraman, K. S. Vastola, *An Integrated Model for the Latency and Steady-State througput of TCP connections*, Performance Evaluation, vol. 46, no. 2-3, pp. 139-154, September 2001..

[17] V. Sharma and P. Punyaslok, *Stability and Analysis of TCP connections with RED Control and Exogenous Traffic*, to appear in Queing Systems, Vol 48, 193-235, 2004. (A shorter version appeared in Proc. IEEE Conf. Globcom 2001).

[18] V. Sharma and A. Gupta, *Performance Analysis of Routers with TCP and UDP Connections with Priority and RED Control*, In International Conference on Computer Communication, ICCC 2002, Mumbai, India. For further details also see *A Unified Approach for Analyzing Persistent, Non-Persistent and ON-OFF TCP sessions with RED control and Exogenous Traffic*, M. Sc. Thesis, Electrical Communication Engineering Dept., I.I.Sc, Bangalore, June, 2002.

[19] G. De. Veciana, T. J. Lee and T. Konstantopoulos, *Stability and Performance analysis of networks supporting Elastic services*, IEEE/ACM Transactions on Networking, Vol 9, No. 1. Feb 2001.

[20] K. Thompson, G. J. Miller and R. Wilder, *Wide-Area Internet Traffic Patterns and Characteristics*, IEEE Network, Nov/Dec 1997.

TABLE 1: SMALL SCALE MODEL ON-OFF, 5 TCP CASE,
MEAN DOWNLOAD TIMES FOR TCP's (sec) AND
MEAN SOJOURN TIME OF UDP (ms).

		TCP1	TCP2	TCP3	TCP4	TCP5	UDP
	Δ	0.01	0.01	0.015	0.020	0.020	$E[S_U]$
	W_{max}	30	30	30	60	60	
	$E[s_T]$	1000	500	500	1000	1000	(ms)
	$1/\lambda$(s)	1	1	1	1	2	
	$1/\mu(KB)$	1000	500	100	1000	1000	
Infinite Buffer	Theo.	3.95	3.57	0.88	2.55	2.76	122.0
	Siml.	3.91	3.73	0.94	2.57	2.70	119.0
Fin. Buff. RED	Theo.	2.87	2.55	0.66	3.47	3.70	21.9
	Siml.	2.89	2.64	0.72	3.58	3.77	18.4

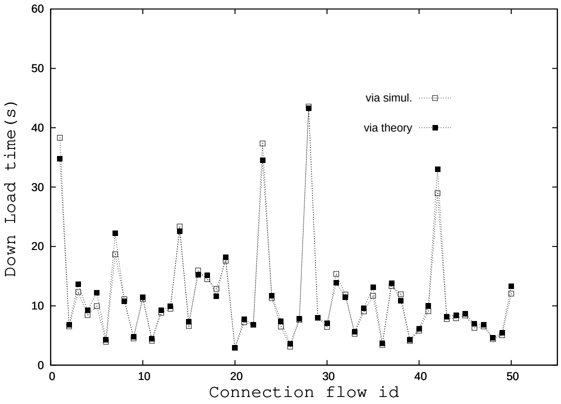

Figure 1: 50 TCP ON-OFF model for the infinite buffer case, $1/\lambda = 3s, 1/\mu = 200KB$. Observe that the model captures download times varying over an order of magnitude.

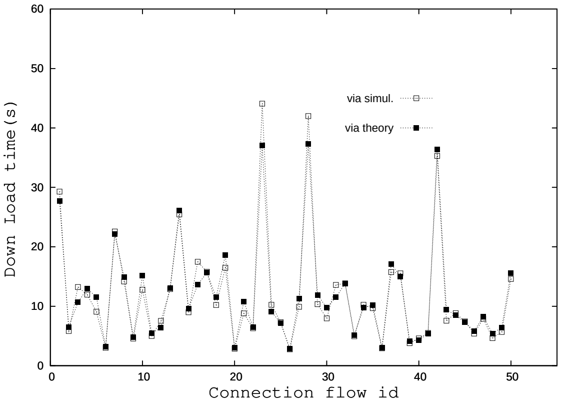

Figure 2: 50 TCP ON-OFF model, finite buffer with RED $1/\lambda = 3s, 1/\mu = 200KB, T_{min} = 100, T_{max} = 500, \beta = 10^{-4}$.

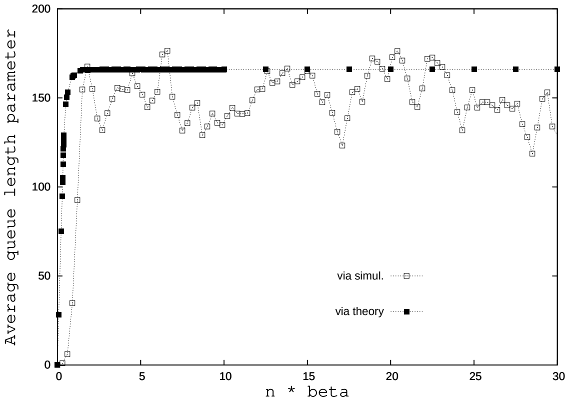

Figure 3: $\hat{q}$ vs $n\beta$, 50 TCP ON-OFF model, with RED, $1/\lambda = 3s, 1/\mu = 200KB$.

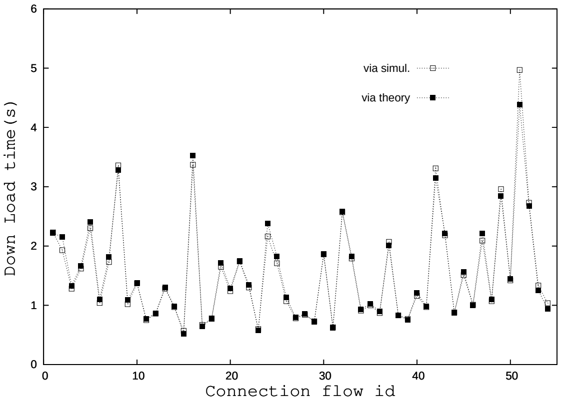

Figure 4: Combined model for the infinite buffer case. Flow ids $1 \ldots 50$ are the Persistent flows and flow ids $51 \ldots 54$ are the Non-Persistent flows, $1/\lambda = 0.5, 1/\lambda_N = 0.3, 1/\mu = 27KB$.

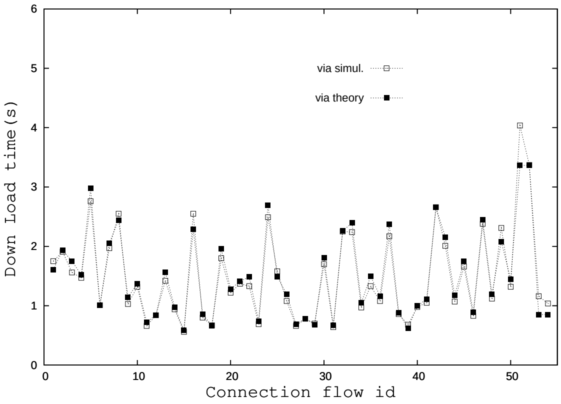

Figure 5: Combined model for the RED case. Flow ids $1 \ldots 50$ are the Persistent flows and flow ids $51 \ldots 54$ are the Non-Persistent flows, $1/\lambda = 0.5, 1/\lambda_N = 0.3, 1/\mu = 27KB$.

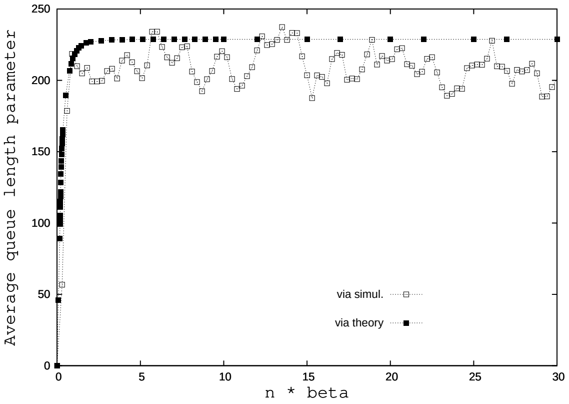

Figure 6: $\hat{q}$ vs $n\beta$, Combined model with RED, $1/\lambda = 0.5, 1/\lambda_N = 0.3, 1/\mu = 27KB, T_{min} = 100, T_{max} = 500, \beta = 10^{-4}$.