A Discrete-Time Model of TCP Reno with Background Traffic Interference*

Ahmed E. Kamal
Department of Electrical and Computer Engineering
Iowa State University
Ames, IA 50011-3060
U.S.A.
E-mail: kamal@iastate.edu
Telephone: (515) 294-3580
Fax: (515) 294-8432
kamal@iastate.edu

Abstract

This paper introduces a discrete-time model which captures the essential protocol features of the congestion control mechanism used by the TCP Reno protocol, subject to interference from other sources. Under this model, a single target session is modeled according to the TCP Reno mechanism, including fast retransmit and fast recovery. At the same time, other sources are modeled as a background process using a modified discrete batch Markov arrival process. In order to capture all the TCP Reno protocol features, two levels of Markov process modeling are used: a microscopic level, at the packet transmission time boundaries, and a macroscopic one, at the start of the new transmission windows. Several performance measures are derived, and numerical examples which demonstrate the protocol features are presented.

I Introduction

The Transmission Control Protocol (TCP) [1] is the reliable transport layer protocol of the Internet. The purpose of this paper is to introduce an accurate performance model for the TCP Reno version [2] in the presence of interference traffic, which may cause congestion in common routers. Although several models of TCP Reno have already been introduced, this model is different in several aspects. The model keeps track of the *exact* way in which windows evolve under TCP Reno, taking into account the congestion avoidance phase mechanics, the propagation delay, the buffer queueing delay, and time-outs. In addition, interfering traffic is assumed and is modeled aggregately by a modified discrete batch Markovian arrival process (D-BMAP) [3]. This process can be used to model UDP traffic, a collection of TCP sources, etc. The packet dropping mechanism is therefore modeled *exactly*, without assuming any packet loss distribution. That is, a packet will be only dropped when the router is congested.

The paper is organized as follows. The next section provides some background material, in terms of the protocol description, and an overview of the relevant work in the literature. The following section introduces

*This research was supported in part by a Carver Trust grant from Iowa State University

the model, while section IV shows how the transition probabilities can be derived, and the evaluation of the performance measures of interest. Section V introduces several numerical examples. Finally, section VI concludes the paper with a few remarks.

II Background

II.1 The TCP Congestion Control Mechanism:

Under the TCP protocol, every segment that is transmitted by the source is kept in a buffer at the source until it is finally acknowledged by the receiver, or a timer expires. In the latter case, the segment must be retransmitted. The receiver, upon receiving a data segment, whether it is the expected segment or not, always sends back an acknowledgement to the source indicating the sequence number of the byte it is expecting. The TCP congestion control mechanism uses the sliding window mechanism, and is part preventive, and part reactive. In the preventive phase, there are two stages. In the first, called *slow start*, the source starts with a window size of one data segment, and for every acknowledgement it increases the window size by one data segment, until the window reaches a threshold level, which is initially set arbitrarily high. At this point in time, the TCP protocol enters the *congestion avoidance stage* in which the window size is incremented by one maximum segment size every time a window full of segments is transmitted and acknowledged. The reactive phase takes place when a packet is lost inside the network and is not acknowledged. This causes the source to time out, reduce its window size to one maximum segment size, and sets the window threshold to [4]

$$\max(\frac{\text{unacknowledged data}}{2}, 2 * \text{max. segment size})$$

It then enters the slow start stage again, until the window size reaches the threshold, at which time it switches to the congestion avoidance phase, and so on.

II.2 TCP Reno

The TCP Reno version implements two other mechanisms which detect, and react to congestion faster. The first mechanism, called *fast retransmit* detects the loss of a data segment when it receives an acknowledgement followed by three duplicate acknowledgements of the same segment. The lost segment is then retransmitted. The second mechanism, *fast recovery*, follows. The source then halves the window size, sets the threshold to that new window size, increments the window size by three, enters the congestion avoidance stage, and keeps on transmitting new segments. Duplicate acknowledgements keep on incrementing the window size, while a new acknowledgement resets the window to the threshold value. The last event causes the start of the congestion avoidance phase. Therefore, the TCP Reno protocol does not implement the slow start phase, except at the start of a new connection.

II.3 Relevant work

Several models of the TCP congestion control mechanism have been introduced. Reference [5] was the first to introduce models for TCP connections, but they were simple models and did not capture several of the TCP operating features into account. Reference [6] considers a single TCP (Tahoe or Reno) connection in a network with a high bandwidth-delay product, where packet losses occur randomly. A simple analysis of multiple connections was carried out in order to illustrate the bias towards connections with short propagation delays. The work in [7] considers the modeling of different versions of TCP under lossy links. Recently, [8, 9] used a discrete time approach to model a single TCP Reno connection, in which packet losses are independent and random, and [10] extended the analysis in [8] to model the slow start stage. Reference [11] introduced a Markov chain model of TCP Reno over ATM that approximated the protocol operation. Reference [12] used a fluid flow approach to model multiple connections with different round trip times. The work in [13] introduced two separate models: one for the network, and one for an aggregation of multiple on-off sources served by multiple TCP connections. An iterative approach was used for model tunning. In [14] the model in [6] was extended to study the behavior of TCP connections with different types of lossy channels. Using a mixture of simulation and analysis, reference [15] showed the self-similar nature of TCP connections. Reference [16] introduced a model for TCP Reno when losses occur according to a bursty process, and was extended in [17] for the case of a general stationary loss process. Approximations, and bounds for the maximum window size case were introduced, and also approximations for detection of packet losses through time-outs were also presented. Finally, reference [18] used max-plus algebra to model the TCP Tahoe and Reno versions while crossing an arbitrary number of routers, and with different service rates. The effect of cross traffic was modeled using stochastic service rates. The choice of this service rate was not discussed, and the effect of the buffer size was not taken into account. Moreover, the computational cost of formulae increases in a non-polynomial way.

III The Model

The model of this paper is based on the following assumptions:

1. We consider a bottlenecked router, and assume that queueing delays, and transmission times at other routers are constant, and are included in a constant factor, τ, which also includes the round trip propagation delay. This is a standard modeling assumption, which has been widely used.

2. Packet lengths are assumed constant, and the packet transmission time is assumed to be the time unit. In addition, the system is assumed to be time synchronous, and the time slot is also equal to the packet transmission time.

3. We consider a *target session* which is modeled as follows:

(a) The target source uses the *Reno* version of TCP. That is, when a packet is discarded, it is detected by the receipt of three duplicate acknowledgements. Or, if the timer expires before such three duplicate acknowledgements are received, the packet loss is also recovered from.

(b) It is assumed that the time-out period, χ, satisfies the following relation

$$\chi > 2\tau + \frac{\text{B}}{\text{transmission rate at bottleneck router}}$$

where B is the buffer size at the bottleneck router. Although this assumption is made for mathematical tractability, it is in line with the actual operation of TCP, and is also satisfied through the use of timers with coarse granularities, e. g., 500 ms.

(c) The target source is assumed to be constantly backlogged, i.e., it always has packets to send.

4. Other sessions are modeled aggregately using a background process, which is modeled as a modified Discrete Batch Markovian Arrival Process (D-BMAP) [3]. The background process consists of a number of phases such that:

   (a) Arrivals from the background process are dependent on the current phase.

   (b) Transitions between phases are also governed by the current phase, and by the number of packets discarded from that process. This last assumption is a modification that we introduce to the D-BMAP process, and it enables one to introduce a correlation between packet losses, and the packet generation process from the background traffic source. This is useful in using the D-BMAP to model other TCP sources, which reduce their transmission rates because of packet losses.

Notice that in the above, no assumption was made about the packet discarding probability or strategy, from either of the target source or the background traffic. Therefore, packet dropping follows the actual mechanics of the network and the router operation, and is solely due to buffer overflow.

The system is assumed to be time synchronous, where the time slot is the packet transmission time at the bottlenecked router, and all time intervals are normalized to the packet transmission time. All packet arrivals and departures are synchronized to the slot boundaries. The system is modeled at two levels, namely a macroscopic and a microscopic levels:

- At the **macroscopic level**, which is the main model, the system is observed at the instants when the target session *starts a new window*, when there is no outstanding packet loss to be recovered from. That is, the system is modeled as an embedded Markovian chain.

- The transitions between different states at the macroscopic level are dependent on all the activities between the embedding points, including window adjustments, packet losses and detection of such losses, round trip delays, including queue size effects, etc. It is not possible to account for all such effects at the macroscopic level only. Therefore, the transition probabilities between two successive

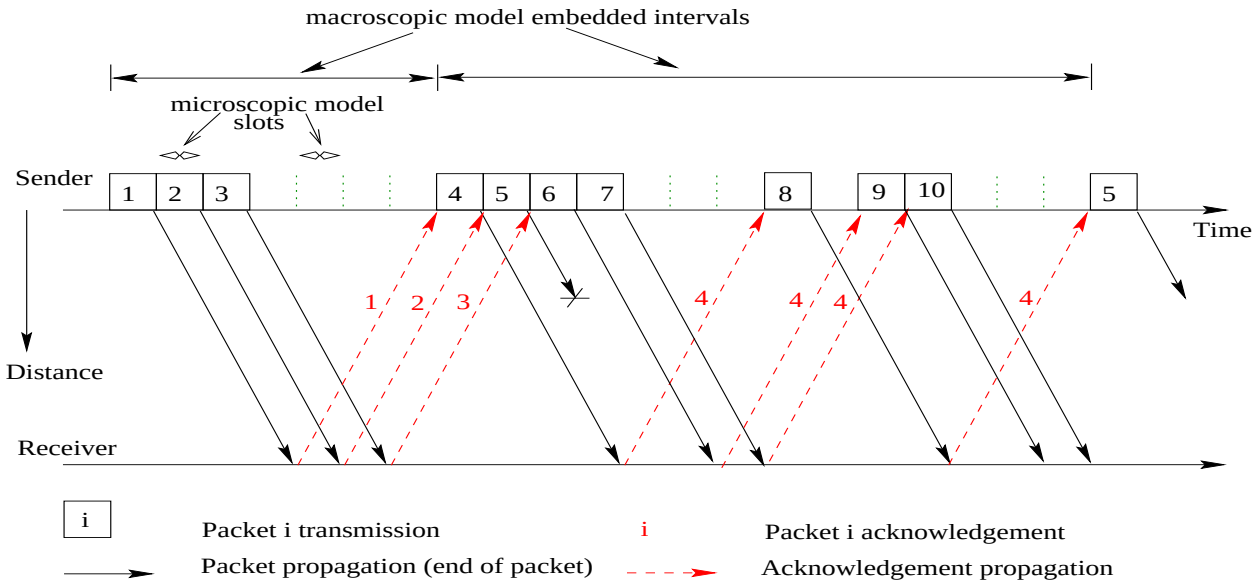

Figure 1: Typical scenario to illusrate the microscopic and macroscopic levels of analysis

embedding points are then computed using the **microscopic model**, by considering the transitions between the system states at all the slot boundaries between these two embedding points.

The system state at the embedding points (**macroscopic model**) consists of the tuple $\Psi = (\Omega, \Theta, \Phi)$, where

Ω the window size of the target session, which can take values from 2 to W_{max}.

Θ the router's queue size, where the maximum buffer size is B

Φ the phase of the background process.

The system state at slot boundaries within an embedded interval (microscopic model) is similarly given by the tuple $\psi = (\omega, \theta, \phi)$, where the window size, ω, is fixed over such an interval. Note that θ and ϕ are taken at the beginning of a slot, just after a packet has been removed from the queue. During the evolution of the phase state variable, ϕ, transitions between phases at the end of a slot, given the number of losses from this process are governed by

$\beta_{ff'}(l) = \Pr(\text{next phase is } f' | \text{ current phase is } f, \text{ and } l \text{ packets have been discarded})$.

Also, the probability that n packets arrive from the background process are phase dependent and given by

$\alpha_f(n) = \Pr(n \text{ packet arrivals} \mid \text{current phase is } f)$

such that $\sum_{n=0}^{n_{max}} \alpha_f(n) = 1$, where n_{max} is the maximum number of possible packet arrivals.

In Figure 1 we show an example that illustrates the two abstraction levels, namely, the microscopic and the macroscopic levels. In this example, two embedded intervals at the macroscopic level are shown. In the first, the window size is three packets, and all packets are transmitted successfully. Within this interval, each slot corresponds to one packet transmission time. Notice also that the duration of this embedded period is equal to one packet transmission time, in addition to the delay encountered within the network in the forward and reverse paths.

In the second embedded interval, the window size has become four packets, and packet number 5 is lost. Three duplicate acknowledgements of packet number 4 will invoke the retransmission of the the lost packet. Notice that the embedded interval in this case ends when the lost packet is transmitted.

IV Transition Probabilities

In this section we show how to evaluate the transition probabilities at both levels.

IV.1 Microscopic Analysis

We first analyze the system at the microscopic level.

IV.1.i Microscopic analysis: no loss from the target session

Consider the transition between two successive slot boundaries within an embedded macroscopic interval in which the target session does not encounter a loss. In this case, the target session is partly active for the duration of its window, and its packets are accepted, and partly inactive awaiting acknowledgments.

We use ρ to refer to the ordered pair (θ, ϕ), which is a subset of the state, ψ. Assume that ρ has the value (q, f). Notice that within such an interval, the window size, ω, does not change.

Define the following probability

$x(i, m|j, n, l, \rho)$ = P(within a slot, the buffer accepts i packets from the target session, given that j packets have already been accepted from the same session, and also accept m out of n packets from the background session, given l packets have been accepted, and the system state at the beginning of the slot is ρ)

The conditions $i, j \in \{0, 1\}$, $i + j = 1$ and $m \leq n$ must be satisfied.

We also define $P(accept|l, \rho)$ ($P(discard|l, \rho)$) as the probability of accepting (discarding) a packet given that l packets have been accepted, and the state is ρ. We have the following recursive relations:

$$\begin{aligned} x(1, m|0, n, l, \rho) &= p(accept|l, \rho) \times \left[\frac{1}{n+1} \times x(0, m|1, n, l, \rho) + \frac{n}{n+1} \times x(1, m-1|0, n-1, l+1, \rho)\right] \\ &\quad + p(discard|l, \rho) \times \left[\frac{n}{n+1} \times x(1, m|0, n-1, l, \rho)\right] \end{aligned} \tag{1}$$

$$\begin{aligned} x(0, m|1, n, l, \rho) &= P(accept|l+1, \rho) \times x(0, m-1|1, n-1, l+1, \rho) \\ &\quad + P(discard|l+1, \rho) \times x(0, m|1, n-1, l, \rho) \end{aligned} \tag{2}$$

The two terms in each of the above equations take into account the cases in which the packet from the target source is accepted and rejected, respectively. In equation (1), the first term is further divided into two terms in which the accepted packet can be from either the target source or the background traffic, respectively, and then the queue subsequently accepts 0 or 1 more packets from the target session. In the second term

of the equation, the discarded packet must be from the background process, and we must accept one packet from the target session. In these two equations we use

$$P(discard|l,\rho) = \begin{cases} 1 & l+q^+ = B \\ 0 & l+q^+ < B \end{cases} \tag{3}$$

where $q^+ = \max(q-1, 0)$, and $P(accept|l,\rho) = 1 - P(discard|l,\rho)$.

The probabilities, $x()$, can be calculated recursively using equations (1) and (2), and the initial conditions:

$$x(0,0|1,0,l,\rho) = 1 \quad \text{for } l \geq 0 \text{ and } q^+ + l + 1 \leq B,$$
$$x(i,m|j,n,l,\rho) = 0 \quad \text{for } m > n, \text{ and}$$
$$x(i,m|j,n,l,\rho) = 0 \quad \text{for } i+j+m+l+q^+ > B$$

When the target session is inactive, i.e., awaiting acknowledgements, we define $y(0,m|0,n,l,\rho)$ similar to $x()$, except that no packets are generated or accepted from the target session. We therefore have the following relation, which is derived in a manner similar to equations (1) and (2):

$$\begin{aligned} y(0,m|0,n,l,\rho) &= P(accept|l,\rho) \times y(0,m-1|0,n-1,l+1,\rho) \\ &\quad + P(discard|l,\rho) \times y(0,m|0,n-1,l,\rho) \end{aligned} \tag{4}$$

with the initial conditions:

$$y(0,0|0,0,l,\rho) = 1 \quad 0 \leq l + q^+ \leq B$$
$$y(0,m|0,n,l,\rho) = 0 \quad m > n \;\; or \;\; m + l + q^+ > B$$

Based on the above, we define the transition probability matrix between slot boundaries when a packet from the target session is accepted, $\mathbf{R} = [r(\rho'|\rho)]$, whose elements are the transition probabilities from state $\rho = (q, f)$ to state $\rho' = (q', f')$, and are given by

$$r(\rho'|\rho) = \sum_{n=q'-q^+-1}^{n_{max}} x(1, q'-q^+-1|0,n,0,\rho) \cdot \alpha_f(n) \cdot \beta_{ff'}(n-q'+q^++1)$$

We also define the transition probability matrix between slot boundaries when the target session is not active, $\mathbf{S} = [s(\rho'|\rho)]$, where its elements are given by

$$s(\rho'|\rho) = \sum_{n=q'-q^+}^{n_{max}} y(0, q'-q^+|0,n,0,\rho) \cdot \alpha_f(n) \cdot \beta_{ff'}(n-q'+q^+)$$

IV.1.ii Microscopic analysis: loss from the target session

Slightly different from the above, we define

$z(i,m\|j,n,l,\rho)$ = P(within a slot, i packets are discarded from the target session, given j have already been discarded, and also m packets are accepted out of n packets from the background session, given l have been accepted, and given the system state at the beginning of the slot is ρ).

Similar to $x()$, i and j also take values of 0 and 1 and sum to 1.

Therefore, $z()$ can be evaluated for the cases of $m+l+i+j+q^+ < B$ using the following equations:

$$\begin{aligned} z(1,m|0,n,l,\rho) &= p(discard|l,\rho) \times \left[\frac{1}{n+1} \times z(0,m|1,n,l,\rho) + \frac{n}{n+1} \times z(1,m|0,n-1,l,\rho)\right] \\ &\quad + p(accept|l,\rho) \times \left[\frac{n}{n+1} \times z(1,m-1|0,n-1,l+1,\rho)\right] \end{aligned} \tag{5}$$

$$\begin{aligned} z(0,m|1,n,l,\rho) &= P(accept|l,\rho) \times z(0,m-1|1,n-1,l+1,\rho) \\ &\quad + P(discard|l,\rho) \times z(0,m|1,n-1,l,\rho) \end{aligned} \tag{6}$$

and the initial conditions:

$$z(0,0|1,0,l,\rho) = 1 \quad l+q^+ <= B \tag{7}$$

$$z(1,0|0,0,l,\rho) = 1 \quad l+q^+ = B \tag{8}$$

$$z(i,m|j,n,l,\rho) = 0 \quad m > n \ or \ m+l+q^+ > B \tag{9}$$

With the above definition of $z()$, we can define the transition probability matrix between slot boundaries in which the target session encounters a loss as

$$\mathbf{T} = [t(\rho'|\rho)]$$

with

$$t(\rho'|\rho) = \sum_{n=q'-q^+}^{n_{max}} z(1,q'-q^+|0,n,0,\rho) \cdot \alpha_f(n)\beta_{ff'}(n-q'+q^+)$$

IV.2 Macroscopic Analysis

Next, we analyze the system at the macroscopic level.

IV.2.i Macroscopic Analysis: an interval without loss

Now we consider an embedded interval that starts in state $(\Omega,\Theta,\Phi) = (w,q,f)$, and does not encounter any losses from the target TCP session. The transition probability matrix to state (w',q',f'), where $w' = \min(w+1, W_{max})$ will be denoted by $\mathbf{W}_{i+}$.

Let the block matrix row of the matrix $\mathbf{R}$ corresponding the initial queue size being equal to q be denoted by $R(q)$. Therefore, the block matrix row of the matrix $\mathbf{W}_{i+}$ corresponding to an initial queue size being q is also denoted by $W_{i+}(q)$ and is given by

$$W_{i+}(q) = R(q)\mathbf{R}^{i-1}\mathbf{S}^{q^+ + \tau + 2 - i} \tag{10}$$

In the above, it has been assumed that all packets in the window will encounter the same queueing delay at the router, as that seen by the first packet, which is equal to q^+. Equation (10) can be calculated efficiently in a recursive manner.

IV.2.ii Macroscopic Analysis: an interval with loss

Unlike section IV.2.i, this interval does not end with the arrival of the following window, since at the beginning of the following window the packet loss would not have been recovered from yet (see Figure 1). As such, the following window is an extension of the current window, and is used to recover from lost packets. In fact, this interval ends with the arrival of a new window which starts with the retransmission of the discarded packet. Denote by $\mathbf{W}_{i-}$ the transition probability matrix to state (w', q', f'), where $w' = \max(2, \lceil \frac{w}{2} \rceil)$, and $w = i$ is the initial window size. This matrix is used when a packet loss takes place and is detected.

It should be noted that for the case of an initial window size, i, with $i \geq 4$, $\mathbf{W}_{i-}$ consists of several components, depending on whether the loss is detected through duplicate acknowledgements, or through time-out. And, in the former case, there are also two components depending on whether the three packets resulting in the duplicate acknowledgements are transmitted within the same window, or during the next window. These components will be discussed separately below.

Detection of loss through duplicate acknowledgements:

Observe that a packet loss cannot be detected through duplicate acknowledgements if the window size is less than four[1].

We define

$\mathbf{U}(j, \rho' \vert k, \rho)$ = Transition probability matrix from state ρ to state $\rho\prime$, with exactly j packets being accepted from the target session in the router's queue within k slots.

We can therefore use the following recursive relations to compute $\mathbf{U}()$

$$\mathbf{U}(2|k) = \mathbf{T}\mathbf{U}(2|k-1) + \mathbf{R}\mathbf{U}(1|k-1) \quad (11)$$

$$\mathbf{U}(1|k) = \mathbf{T}\mathbf{U}(1|k-1) + \mathbf{R}\mathbf{U}(0|k-1) \quad (12)$$

$$\mathbf{U}(0|k) = \mathbf{T}\mathbf{U}(0|k-1) \quad (13)$$

starting from $\mathbf{U}(0|0) = \mathbf{I}$, $\mathbf{U}(1|0) = \mathbf{0}$, and $\mathbf{U}(2|0) = \mathbf{0}$, where $\mathbf{I}$ is the identity matrix, and $\mathbf{0}$ is the zero matrix.

There are two cases to consider when the loss is detected through duplicate acknowledgements:

1. The case in which the loss is detected within the same window. This requires that after the discarded packet there be at least three more packets in the same window. Therefore, for the case of $i \geq 4$, the component of $W_{i-}(q)$ is denoted by W_{i-}^{DS} (for *D*etection within the *S*ame window), and is given by

$$W_{i-}^{DS}(q) = \sum_{n=0}^{i-5} R(q)\mathbf{R}^n\mathbf{T} \sum_{k=2}^{i-n-3} \mathbf{U}(2|k)\mathbf{R}[\mathbf{R}+\mathbf{T}]^{i-k-n-3}\mathbf{S}^{q^+ + \tau + 2 - i}[\mathbf{R}+\mathbf{T}]^{n+1}\mathbf{S}^{k+1}$$
$$+T(q)\sum_{k=2}^{i-2} \mathbf{U}(2|k)\mathbf{R}[\mathbf{R}+\mathbf{T}]^{i-k-2}\mathbf{S}^{q^+ + \tau + k + 3 - i} \quad (14)$$

[1] In the second embedded macroscopic interval in Figure space-time, since the window size is four, loss detection through duplicate acknowledgements is possible. Had the window size been three, or had multiple packets been discarded, then not enough duplicate acknowledgements will be received, and loss detection through this mechanism is infeasible.

The first term in the above equation corresponds to the first lost packet being packet $n+2$, for $n \geq 0$, while the second term corresponds to the first packet in the window being lost.

2. The other case is the one in which the loss is detected within the following window. This requires that, if the first packet discarded is packet k in the first window, where $k > 1$, then the third acknowledgement must be in response to packet number l in the following window, where $l < k$. Therefore, also for $i \geq 4$, the component of $W_{i-}(q)$ for this case is denoted by W_{i-}^{DN} (for *D*etection within the *N*ame window), and can be expressed as

$$W_{i-}^{DN}(q) = \sum_{j=0}^{2} \sum_{n=2-j}^{i-2-j} R(q)\mathbf{R}^n\mathbf{T}\mathbf{U}(j|i-n-2)\mathbf{S}^{q^+ +\tau+2-i} \cdot \sum_{k=2-j}^{n} \mathbf{U}(2-j|k)\mathbf{R}[\mathbf{R}+\mathbf{T}]^{\mathbf{n-k}}\mathbf{S}^{\mathbf{k}+\tau+\mathbf{q}^+-\mathbf{n}+\mathbf{1}} \quad (15)$$

In this equation, a loss must take place in packet number $n+2$, for $n \geq 0$, and fewer than three packets are not discarded from the same window, and after the occurrence of the first loss.

Notice that in equations (14) and (15), we consider transitions between slots whose states correspond to the no loss, loss, or idle from the target session point of view.

Detection of loss through time-out:

Detection of loss through time-out, where the time-out interval is assumed fixed and equal to χ slots, measured from the end of the packet transmission, will take place in one of two cases:

1. The window size is less than four, in which case W_{i-} for $i < 4$ is given by

$$W_{2-}(q) = T(q)[\mathbf{R}+\mathbf{T}]\mathbf{S}^{\chi-1} + R(q)\mathbf{T}\mathbf{S}^{q^+ +\tau}[\mathbf{R}+\mathbf{T}]\mathbf{S}^{\chi-q^+-\tau-2} \quad (16)$$

$$W_{3-}(q) = T(q)[\mathbf{R}+\mathbf{T}]^2\mathbf{S}^{\chi-2} + R(q)\mathbf{T}[\mathbf{R}+\mathbf{T}]\mathbf{S}^{q^+ +\tau-1}[\mathbf{R}+\mathbf{T}]\mathbf{S}^{\chi-q^+-\tau-2} + R(q)\mathbf{R}\mathbf{T}\mathbf{S}^{\tau+q^+-1}[\mathbf{R}+\mathbf{T}]^2\mathbf{S}^{\chi-q^+-\tau-3} \quad (17)$$

In equation (16), the two terms correspond to the first loss occurring in the first, and second packets, respectively. Equation (17), three terms correspond to the first loss occurring in the first, second, and third terms, respectively.

2. The window size is greater than or equal to four, but not enough packets are accepted from the target session (at least three) in order for the three duplicate acknowledgements to be sent back to the source. Therefore, for $i \geq 4$, the corresponding component of $W_{i-}(q)$ is denoted by $W_{i-}^{T}(q)$ is given by

$$W_{i-}^{T}(q) = \sum_{k=0}^{2}\sum_{j=0}^{2-k} R(q) \sum_{n=0}^{i-2} \mathbf{R}^n\mathbf{T}\mathbf{U}(j|i-n-2)\mathbf{S}^{q^+ +\tau+2-i}\mathbf{U}(k|n+1)\mathbf{S}^{\chi-q^+-\tau-1} + \sum_{j=0}^{2} T(q)\mathbf{U}(j|i-1)\mathbf{S}^{\chi-i+1} \quad (18)$$

Based on equations (14), (15), and (18), the matrix $\mathbf{W}_{i-}$ for $i \geq 4$ can be obtained from

$$\mathbf{W}_{i-} = \mathbf{W}_{i-}^{DS} + \mathbf{W}_{i-}^{DN} + \mathbf{W}_{i-}^{T}$$

V Performance Measures

Denote the steady state probability vector by

$$\mathbf{\Pi} = \{\vec{\pi}_1, \vec{\pi}_2, \ldots, \vec{\pi}_{W_{max}}\}$$

where $\vec{\pi}_i$ is the steady state probability vector of the target session having a window size of i. This vector in turn has several component vectors in terms of the state ρ,

$$\vec{\pi}_i = \{\vec{\pi}_{i_0}, \vec{\pi}_{i_1}, \ldots, \vec{\pi}_{i_{\rho_{max}}}\}$$

The transition probability matrix, whose components are derived in the previous section, has a simple structure. For example, for the case of an even W_{max}, the transition probability matrix, $\mathbf{P}$, has the following elements

$$P_{i,j} = \begin{cases} \mathbf{W}_{i+} & j = \min(i+1, W_{max}) \\ \mathbf{W}_{i-} & j = \max(2, \lceil i/2 \rceil) \\ \mathbf{0} & \text{otherwise} \end{cases} \tag{19}$$

We used this structure, and applied the efficient solution method in [19] to solve for the vector Π. Several performance measures can be directly computed from Π, including the distributions of queue size and the window size, as well as their moments. In addition, we can calculate the distribution of losses within windows of different sizes.

The following performance measures can be computed:

1. Mean queue size $= \sum_{i=2}^{W_{max}} \sum_{\rho, \Theta = q \in \rho} q \vec{\pi}_{i_\rho} \vec{1}$
   where $\vec{1}$ is an appropriately dimensioned column vector of 1's.

2. Also, some probability distributions can be computed including:

   (a) $\mathrm{P(window = i)} = \sum_\rho \vec{\pi}_{i_\rho} \vec{1}$

   (b) The probability of loss given window size $= i$ can be expressed as:

   $$\mathrm{P(loss|window = i)} = \frac{\mathrm{P(loss \text{ and } window = i)}}{\mathrm{P(window = i)}} = \frac{\vec{\pi}_i \mathbf{W}_{i-} \vec{1}}{\sum_\rho \vec{\pi}_{i_\rho} \vec{1}}$$

   and similarly,

   $$\mathrm{P(window = i|loss)} = \frac{\mathrm{P(loss \text{ and } window = i)}}{\mathrm{P(loss)}} = \frac{\vec{\pi}_i \mathbf{W}_{i-} \vec{1}}{\sum_{i=2}^{W_{max}} \mathrm{P(loss \text{ and } window = i)}}$$

   (c) $\mathrm{P(loss\ detected\ by\ duplicate\ ACK|loss\ when\ window = i)} = \frac{\vec{\pi}_i (\mathbf{W}_{i-}^{DS} + \mathbf{W}_{i-}^{DN}) \vec{1}}{\mathrm{P(loss \text{ and } window=i)}}$

(d) P(loss detected by time $-$ out|loss at window = i) $= \frac{\vec{\pi_i}\mathbf{W}_{i-}^{T}\vec{1}}{\text{P(loss and window=i)}}$

(e) P(first loss is at n^{th} position|window = i, loss in window) $= \frac{\vec{\pi_i}\mathbf{R}^{n-1}\mathbf{T}\vec{1}}{\text{P(loss and window=i)}}$

3. **Throughput:** The system throughput is more involved to compute and can be obtained using the definition:

$$\text{Throughput} = \frac{\text{E(number of successfully received packets in a cycle)}}{\text{E(cycle length)}}$$

where a cycle is the duration between two successive embedding points under the macroscopic analysis. There are two cases to consider in computing the numerator and the denominator of the above expression:

(a) **A cycle without loss.**

In this case

$$\text{E(number of packets in a cycle without loss}|\Omega = \text{i}, \Theta = \text{q}) = i\vec{F}(i,q)W_{i+}(q)\vec{1} \tag{20}$$

and

$$\text{E(length of a cycle without loss}|\Omega = \text{i}, \Theta = \text{q}) = (q^{+} + 1 + \tau)\vec{F}(i,q)W_{i+}(q)\vec{1} \tag{21}$$

The vector $\vec{F}(i,q)$ in the above equations is the steady state probability vector of Φ given that $\Omega = i$ and $\Theta = q$. This vector be calculated easily from $\vec{\pi}_i$ and the joint steady state probability of $\Omega = i$ and $\Theta = q$, which is obtained from

$$\sum_{\rho, \Theta = q \in \rho} \vec{\pi}_{i_\rho}\vec{1}$$

(b) **A cycle with loss.**

To compute the cycle length and the number of packets transmitted during this cycle in an exact manner can be computationally expensive. This is especially true since we have to distinguish between a cycle containing a loss which is detected by the expiry of the timer, and a cycle, also containing a loss, but that is detected by the receipt of three duplicate acknowledgements. Instead, we take an approximate, but accurate, approach in handling cycles with losses in which we keep track of the first loss in a cycle exactly. However, further losses are considered to depend on the first loss, but to be independent among themselves.

To compute the cycle length and the number of packets transmitted during this cycle we require several auxiliary probabilities:

- The probability of the first loss at position m, when the window and queue sizes are i and q, respectively, is given by $L_{i,q}(m)$, for $1 \leq m \leq i$. This can be expressed as:

$$L_{i,q}(1) = \vec{F}(i,q)T(q)\vec{1} \tag{22}$$

$$L_{i,q}(m) = \vec{F}(i,q)R(q)\mathbf{R}^{m-2}\mathbf{T}\vec{1} \text{ for } m \geq 2, \tag{23}$$

and

- The probability of loss at position n, given that the first loss was at position m, and that the window and queue sizes are i and q respectively, is denoted by $l_{i,q,m}(n)$, for $1 \leq m < n < i+m$. Notice that n can be greater than i, which means that it is in the following sub-window of packet transmissions corresponding to the first $m-1$ acknowledgements. As such,

$$l_{i,q,m}(n) = \frac{\mathrm{P(loss\ at\ n,\ first\ loss\ at\ m}|\Omega = \mathrm{i}, \Theta = \mathrm{q})}{\mathrm{P(first\ loss\ at\ m}|\Omega = \mathrm{i}, \Theta = \mathrm{q})} \tag{24}$$

where

$$\mathrm{P(loss\ at\ n,\ first\ loss\ at\ m}|\Omega = \mathrm{i}, \Theta = \mathrm{q}) =$$

$$\begin{cases} \vec{F}(i,q)T(q)(\mathbf{R}+\mathbf{T})^{n-2}\mathbf{T}\vec{1} & \text{for } m = 1,\ 2 \leq n \leq i \\ \vec{F}(i,q)R(q)\mathbf{R}^{m-2}\mathbf{T}(\mathbf{R}+\mathbf{T})^{n-1-m}\mathbf{T}\vec{1} & \text{for } 1 < m < n \leq i \\ \vec{F}(i,q)R(q)\mathbf{R}^{m-2}\mathbf{T}(\mathbf{R}+\mathbf{T})^{i-m} & \\ \quad \cdot \mathbf{S}^{(q^+ + 1 + \tau - i)}(\mathbf{R}+\mathbf{T})^{n-i-1}\mathbf{T}\vec{1} & \text{for } 1 < m \leq i < n < i+m \end{cases} \tag{25}$$

Now we can express the contribution of cycles with losses to the mean cycle time and the mean number of packets transmitted within those cycles according to the window size:

i. Ω **=2 or 3**, in which the loss is always detected by the time-out mechanism.

$$\mathrm{E(number\ of\ successful\ packets\ in\ a\ cycle\ containing\ a\ loss}|\Omega = \mathrm{i}, 2 \leq \mathrm{i} \leq 3, \Theta = \mathrm{q})$$
$$= \sum_{j=1}^{i} L_{i,q}(j)[j - 1 + \sum_{k=1}^{i-1}(1 - l_{i,q,j}(j+k))] \tag{26}$$

$$\mathrm{E(length\ of\ a\ cycle\ that\ contains\ a\ loss}|\Omega = \mathrm{i}, 2 \leq \mathrm{i} \leq 3, \Theta =, \mathrm{q})$$
$$= \sum_{j=1}^{i}(\chi + j - 1)L_{i,q}(j) \tag{27}$$

ii. $\Omega \geq 4$, in which case the loss can be detected either by the time-out mechanism, or the triple duplicate acknowledgement mechanism. The cycle length and the number of packets transmitted in a cycle in such cases are calculated similar to equations (26) and (27), except that we have to take into account all combinations of losses in the cycle:

$$\mathrm{E(number\ of\ successful\ packets\ in\ a\ cycle\ with\ loss}|\Omega = \mathrm{i}, 4 \leq \mathrm{i}, \Theta = \mathrm{q})$$
$$= \sum_{j=1}^{i} L_{i,q}(j) \sum_{k=0}^{i-1}(j - 1 + k)\mathrm{P(k\ successful\ packet\ transmissions)} \tag{28}$$

$$\mathrm{E(length\ of\ a\ cycle\ with\ loss}|\Omega = \mathrm{i}, 4 \leq \mathrm{i}, \Theta =, \mathrm{q})$$
$$= \sum_{j=1}^{i} L_{i,q}(j)(\chi + j - 1)\mathrm{P(less\ than\ 3\ successful\ transmissions\ after\ first\ loss)}$$
$$+ \sum_{j=1}^{i} L_{i,q}(j) \sum_{k=j+1}^{i} (k + q^+ + 1 + \tau)$$
$$\cdot \mathrm{P(3rd\ successful\ transmission\ after\ first\ loss\ is\ at\ k)}$$
$$+ \sum_{j=1}^{i} L_{i,q}(j) \sum_{k=i+1}^{j+j-1} [2(q^+ + 1 + \tau) + k - i]$$
$$\cdot \mathrm{P(3rd\ successful\ transmission\ after\ first\ loss\ is\ at\ k)} \tag{29}$$

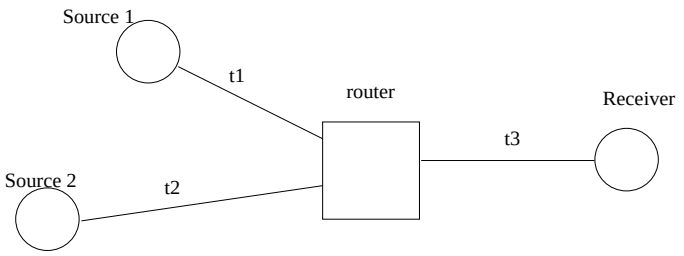

Figure 2: Sample network

Although the number of combinations in the above equations can be large, but the calculation of their respective probabilities is straightforward and fast since they only involve the scalar probabilities $L_{i,q}$ and $l_{i,q,m}(n)$.

Now, based on equations (20), (26) and (28),

$$\begin{aligned}&\text{E(number of successful packets in a cycle)}\\&\quad=\sum_i\sum_q[\text{E(number of packets in a cycle without loss}|\Omega=\text{i},\Theta=\text{q})\\&\quad+\text{E(number of packets in a cycle with loss}|\Omega=\text{i},\Theta=\text{q})]\Pr(\Omega=\text{i},\Theta=\text{q})\end{aligned}\tag{30}$$

and from equations (21), (27) and (29)

$$\begin{aligned}&\text{E(cycle length)}\\&\quad=\sum_i\sum_q[\text{E(length of a cycle without loss}|\Omega=\text{i},\Theta=\text{q})\\&\quad+\text{E(length of a cycle with loss}|\Omega=\text{i},\Theta=\text{q})]\Pr(\Omega=\text{i},\Theta=\text{q})\end{aligned}\tag{31}$$

VI Numerical Examples

In this section we present several numerical examples based on the above model.

We first verify the accuracy of the model via simulation. We used the network shown in Figure 2, which consists of two sources connected to a router with a buffer size of 20 packets. Both sources send to the same destination node, which is also connected to the same router. The first source has a maximum window size of 16 packets, while the second source has a maximum window size of 8 packets. The propagation delays, $t2 = 4$, and $t3 = 4$, both in terms of packet transmission times. The propagation delay between source 1 and the router, $t1$, assumes two values, namely 4 and 8 packet transmission times [2]. For this network, two models were constructed, one for each of the two sources, while the other source was modeled approximately using a Markov modulated Poisson process (MMPP). For example, for the model in which source 1 is the target source, source 2 was modeled as an MMPP generating background traffic. The parameters of the

[2]This network corresponds to a network with links operating at a DS-3 rate of 44.736 Mb/s, packet lengths of 1400 bytes, and propagation delays $t1 = 1$ and 2 ms, $t2 = 1$ ms, and $t3 = 1$ ms.

	Source 1		Source 2	
	Analysis	Simulation	Analysis	Simulation
Scenario 1 ($t_1 = 4$)	0.684002	0.6666	0.339428	0.3333
Scenario 2 ($t_1 = 8$)	0.572	0.592	0.422	0.407

Table 1: Throughput values for the sources in Figure 2 using the model, and the *ns-2* simulator

MMPP are based on first order statistics which are obtained from the other model, in which source 2 is the target source. The two models were run in an alternating manner, and iteratively, until convergence was obtained. Although convergence was based on the mean buffer size of the router, other criteria could have been used. The throughput values obtained from the model, and from the *ns-2* simulator [20] are shown in Table 1. As shown in the table, the results from the model are very close to those from simulation, and the error does not exceed 5%.

We next consider the effect of different parameters, such as the buffer size, the propagation delay, the background interference traffic burstiness, as well as its responsiveness to lost segments. The background interference traffic is called p_b-responsive if it backs off immediately with probability p_b when a packet is discarded due to congestion. UDP traffic can therefore be regarded as 0-responsive[3]. TCP traffic sources back off, but the effect of backoff, and rate reduction, does not appear at the router until after the segment loss is detected at the source. We can therefore model this effect using p_b-responsive sources, where $0 < p_b < 1$. In the examples below, we consider three levels of responsiveness of interference traffic, namely, 0, 0.5 and 1.

In the first example, shown in Figure 3, we show the throughput achieved by the target TCP source for two different router buffer sizes, namely 15 and 30 packets. The maximum window size of the target source is 16 packets, the round trip propagation delay is 25 packets[4], and a time-out interval of 75 packets times. In the figure, the target TCP source throughput is shown versus the load offered by the interference traffic. The interference traffic is modeled as a group of five on-off sources, where the distributions of both the on and the off periods are exponential. During the on period, which has a mean duration of 5 slots, each source generates a segment in a slot with probability 1. The length of the off period is adjusted in order to achieve the desired load level.

It is clear from the figure that as the interference load increases, the target source throughput decreases almost linearly with the load. Also, as the responsiveness level of the background traffic increases, the target node throughput will increase. However, the throughput achievable under the 1-responsive case is not much better than that under the 0-responsive case. This can be attributed to the fact that when the queue is full, it is very likely that it will drop packets from most active sessions, including the target source. Therefore, the target session will reduce its window in all cases. However, with the 1-responsive case, there is a slightly

[3]Results from the ns-2 simulator with UDP background traffic are very close to those obtained from the model for the 0-responsive case.

[4]This is equivalent to about 675 Km one-way separation between the source and the destination, assuming a DS-3 (44.736 Mb/s) link, and 1500-byte IP packets.

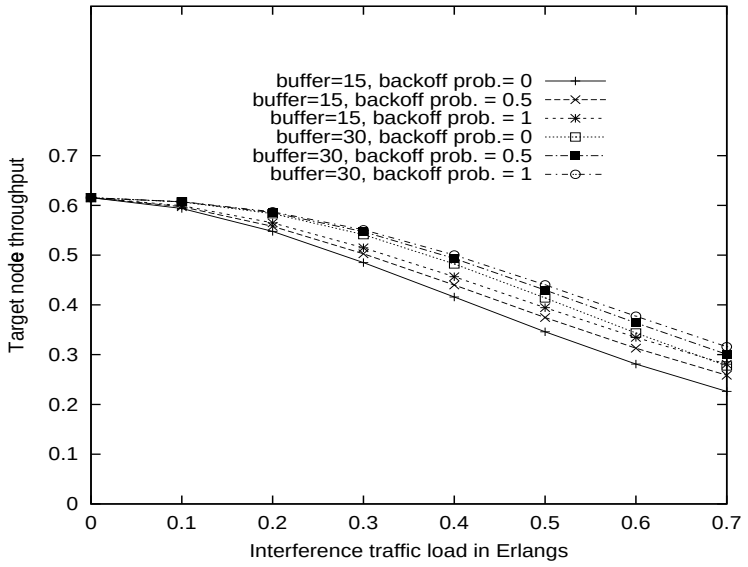

Figure 3: Target TCP source throughput: window size = 16, round trip delay = 25 and time-out interval = 75; two buffer sizes: 15 and 30 packets.

greater probability that a packet will not be dropped from the target source since the background sources have already backed off. Increasing the buffer size from 15 to 30 packets increases the achievable throughput by about 10% under heavy background load. It will therefore take a substantial increase in the buffer size in order to achieve any measurable improvement in the target source throughput.

In the next example, shown in Figure 4, we maintain the same parameters of Figure 3, for the case of a buffer size of 15 packets. We also maintain the mean duration of the off periods, except that we reduce the background traffic burstiness inside the on period. Within the on period, the interval between successive packets is exponentially distributed with a mean of 5 slots, and the average number of packets within this period is kept at 5. Surprisingly enough, the reduced burstiness does not help the target source at all. In fact, the target source achievable throughput is reduced. Taking the dynamics of the TCP protocol into account, suppose the background packets arrive back to back. Then, they may cause successive packets from the target source to be discarded, and the window size will be reduced by one half. However, when the background traffic packets arrive with a time separation, then it is likely that these packets will overlap more than one window, especially under heavy load, and when the active period and the propagation delay are of the same order. Thus, they cause the window size of the TCP source to be reduced twice, hence reducing the throughput. Notice also that in this case increasing the responsiveness of the background traffic has a slightly more positive effect on the TCP source throughput, which is due to the fact that the background traffic refrains from sending any more packets that may interfere with the following window(s).

We have also studied the effect of changing the window size Figure 5. While we keep all the parameters of the example in Figure 3, under the 15 packet buffer size case, we change the target TCP source maximum

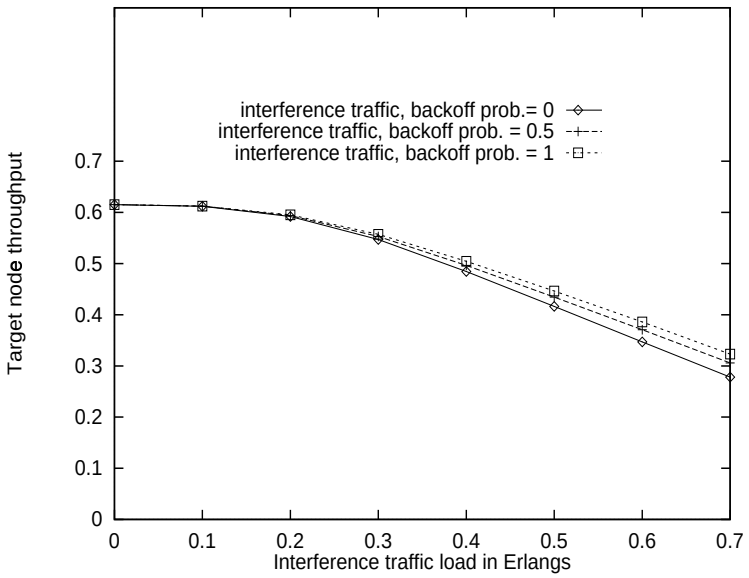

Figure 4: Target TCP source throughput: window size = 16, buffer size = 15, round trip delay = 25 and time-out interval = 75; reduced burstiness

window size to 8 packets. Although the achievable throughput in the absence of background traffic has now been reduced to one half, which is due to the limit imposed by the maximum window size, the achievable throughput under heavy background traffic is only slightly worse than the first case, i.e., with a maximum window size of 16. This is due to that fact that under such load conditions the router is congested, packets are dropped frequently, and the TCP source window size is very small. The mean window size under the 0.7 load level in both cases is comparable: 8.79 and 6.84 for the 16 and 8 maximum window sizes, respectively, and with $p_b = 1$ for the background traffic.

It is to be noted that we have investigated the effect of increasing the time-out interval on the throughput, and it was found that it has very little effect, even at heavy load, since most packet losses are detected using the duplicate acknowledgement method.

Next, we study window distribution functions under different conditions. In Figure 6 we show the probability mass function of the window size using the same scenario of Figure 3 when the buffer size is 15 packets. We consider two background traffic responsiveness levels, 0 and 1. It is clear that under light load the window is at its maximum size, 16, most of the time. Also, the window size almost never goes below 8 (half the maximum window size), which indicates that if a packet is dropped from the TCP source, this is done when the window size is 16. Under heavy load, the mass of the window probabilities are close to half the maximum window size. This is due to the fact that losses either take place in this region, or at higher window values, and then the window is halved. This is evident from Figure 7 which shows the window size probability mass function when a packet is discarded (for both 0-responsive and 1-responsive background traffic cases). Under heavy load, the probability that the window size is at its maximum when a loss a occurs

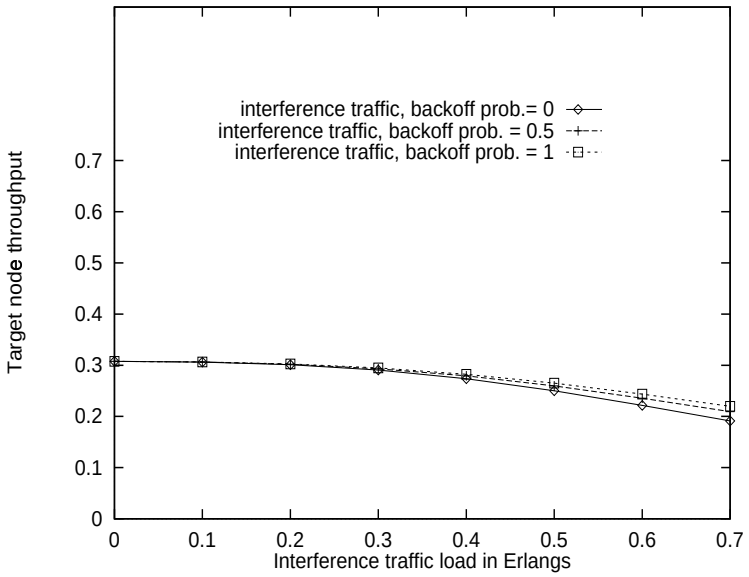

Figure 5: Target TCP source throughput: window size = 8, buffer size = 15, round trip delay = 25 and time-out interval = 75

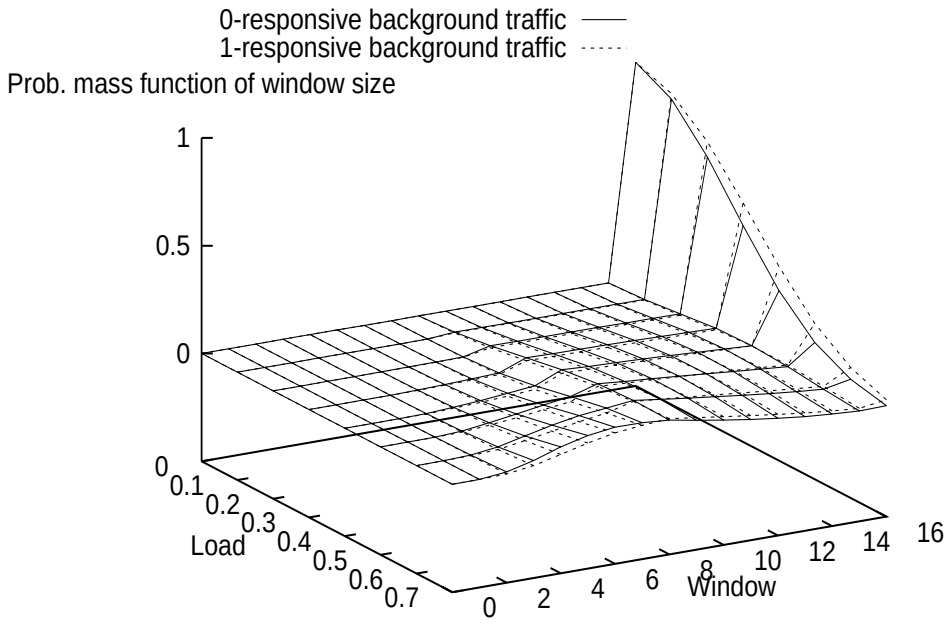

Figure 6: Window size probability mass function

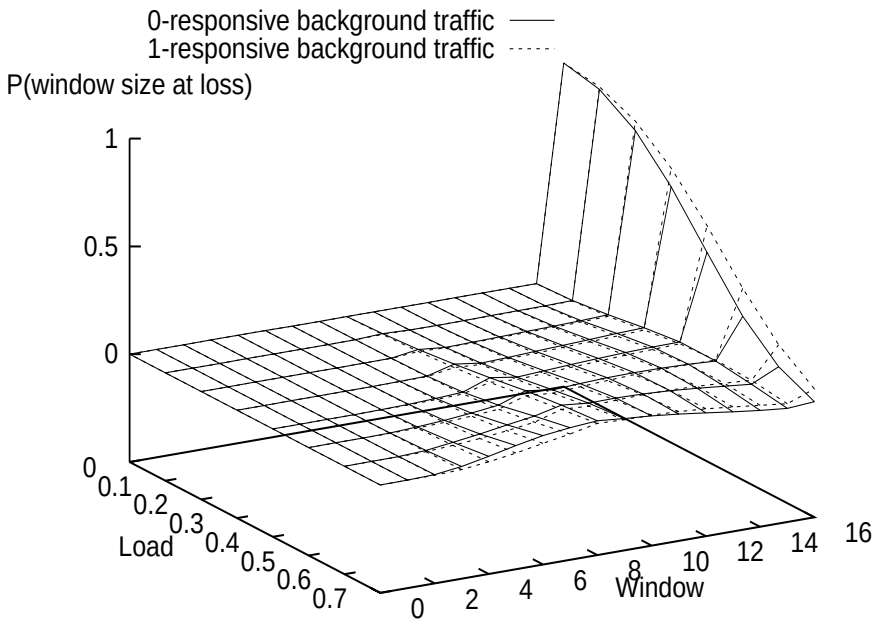

Figure 7: Probability mass function of window size when a loss occurs

is relatively large (around 0.1), especially with the 1-responsive traffic. This is due to the fact that this value is a boundary where the window cannot increase further. This contributes to the window distribution being centered around half the maximum window size. It is to be also noted that when the background traffic is responsive, the window tends to be slightly larger. For example, under the 0.7 load case, the mean window size is 7.67 with the 0-responsive traffic, while it is 8.79 with the 1-responsive traffic.

If we compare the probability mass functions of the unconditional window size (Figure 6) and the window size when a loss occurs (Figure 7) we observe that smaller window sizes have slightly larger probabilities in the former figure for the case 0-responsive traffic. This is due to the fact that an unconditional window size probability takes into account reaching this window size, which occurs if loss occurs at this, or larger window size values.

We finally explore the probability of a packet being discarded in a certain position, given that loss has already occurred in the window by discarding an earlier packet. This is shown in Figure 8, which depicts the distribution of the distance to the following loss positions, given the position of the first loss. The figure is for the case in which the window size is 12. It is shown that once a loss occurs, the following packet in the same window is discarded with a relatively large probability. The following packets in the same window are also discarded with a smaller, but also relatively large probabilities[5]. Such losses from the background traffic

[5]In the case of 1-responsive traffic, the probability of loss drops significantly due to backing-off of the background traffic.

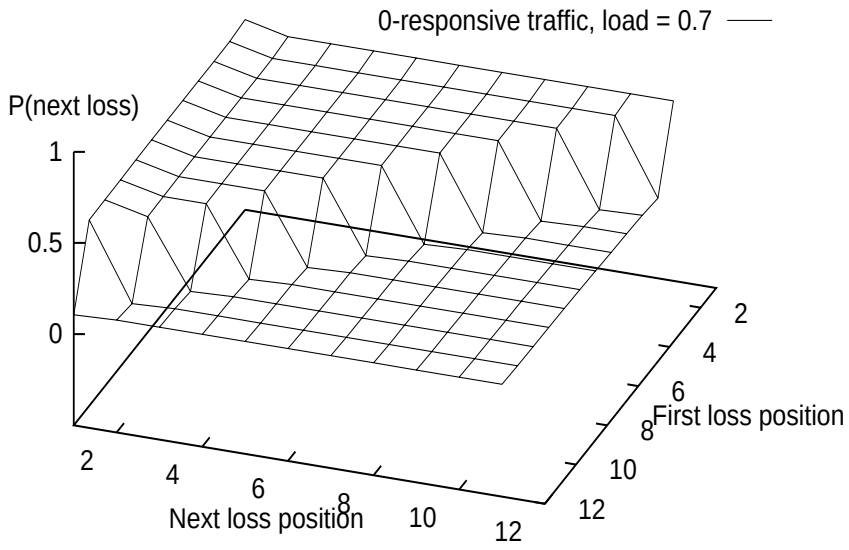

Figure 8: Probability mass function of successive loss positions, given the position of the first loss

occur very close to the point where the first packet is lost from the TCP source. However, the probability of discarding packets from the following partial window (distances greater than the window size - the first loss position) is very small, and almost negligible.

These observations reveal that the assumptions made in [8, 9] in which all packets following the first discarded packet in a window, hold in an approximate manner only, and under heavy load. However, such assumptions are not expected to impact the throughput calculations under light load significantly, since packet losses are already very rare.

VII Conclusions

This paper has presented a discrete time model of the TCP Reno protocol in the presence of interference from background traffic. The target TCP session was modeled to capture all the basic features of the TCP Reno protocol. The background traffic was modeled as a general modified D-BMAP process. Two levels of modeling were used, one at the packet transmission time level (microscopic model), and the other at the window evolution level (macroscopic model).. Several performance measures were derived, including window size distributions. Numerical examples were presented, and several of the protocol features were discussed. It was shown through comparison to simulation that the model is highly accurate.

References

[1] M. Allman, V. Paxson, and W. Stevens, "Tcp congestion control." Network Working Group Request for Comment, RFC 2581, Apr. 1999.

[2] W. R. Stevens, *TCP/IP Illustrated, Volume 1: The Protocols.* Addison-Wesley, 1994.

[3] C. Blondia and O. Casals, "Performance analysis of statistical multiplexing of vbr sources," in *Proceedings of the IEEE INFOCOM*, pp. 828–83, 1992.

[4] G. R. Wright and W. R. Stevens, *TCP/IP Illustrated, Volume 2: The Implementation.* Addison-Wesley, 1995.

[5] S. Floyd, "Connections with multiple congested gateways in packet-switched networks, part 1: One-way traffic," *ACM Computer Communication Review*, pp. 30–47, Oct. 1991.

[6] T. V. Lakshman and U. Madhow, "The performance of tcp/ip for networks with high bandwidth-delay products and random loss," *IEEE/ACM Transactions on Networking*, vol. 5, pp. 336–350, June 1997.

[7] A. Kumar, "Comparative performance analysis of versions of tcp in a local network with a lossy link," *IEEE/ACM Transactions on Networking*, vol. 6, pp. 485–498, Aug. 1998.

[8] J. Padhye, V. Firoiu, d. Towsley, and J. Kurose, "Modeling tcp throughput: A simple model and its empirical validation," in *Proceedings of ACM SIGCOMM Symposium*, pp. 303–314, 1998.

[9] J. Padhye, V. Firoiu, d. Towsley, and J. Kurose, "Modeling tcp reno performance throughput: A simple model and its empirical validation," *IEEE/ACM Transactions on Networking*, vol. 8, pp. 133–145, Apr. 2000.

[10] N. Cardwell, S. Savage, and T. Anderson, "Modeling tcp latency," in *Proceedings of the IEEE INFOCOM*, 2000.

[11] M. A. Marsan, E. de Souza e Silva, R. L. Cigno, and M. Meo, "A markovian model for tcp over atm," *Telecommunication Systems Journal*, vol. 12, pp. 341–368, 1999.

[12] P. Brown, "Resource sharing of tcp connections with different round trip times," in *Proceedings of the IEEE INFOCOM*, 2000.

[13] C. Casetti and M. Meo, "A new approach to model the stationary behavior of tcp connections," in *Proceedings of the IEEE INFOCOM*, 2000.

[14] A. Abouzeid, S. Roy, and M. Azizoglu, "Stochastic modeling of tcp over lossy links," in *Proceedings of the IEEE INFOCOM*, 2000.

[15] A. Veres and M. Boda, "The chaotic nature of tcp congestion control," in *Proceedings of the IEEE INFOCOM*, 2000.

[16] E. Altman, K. Avrachenkov, and C. Barakat, "Tcp in the presence of bursty losses," *Performance Evaluation*, vol. 42, pp. 129–147, 2000.

[17] E. Altman, K. Avrachenkov, and C. Barakat, "A stochastic model of tcp/ip with stationary random losses," in *Proceedings of ACM SIGCOMM Symposium*, 2000.

[18] F. Baccelli and D. Hong, "Tcp is max-plus linear and what it tells us on its throughput," in *Proceedings of ACM SIGCOMM Symposium*, pp. 219–230, 2000.

[19] A. E. Kamal, "Efficient solution of multiple server queues with applications to the modeling of atm concentrators," in *Proceedings of the IEEE INFOCOM*, pp. 248–254, 1996.

[20] "http://www.isi.edu/nsnam/ns."