Stochastic Differential Equation Modeling and Analysis of TCP-Windowsize Behavior

Vishal Misra† Wei-Bo Gong† Don Towsley‡

†Department of Electrical and Computer Enginnering
University of Massachusetts, Amherst, MA 01003
{vmisra,gong}@ecs.umass.edu

‡Department of Computer Science
University of Massachusetts, Amherst, MA 01003
{towsley}@cs.umass.edu

Abstract—**In this paper we present new modeling and analysis technique for characterizing the steady state performance of a TCP flow. We invert the loss process in our model, by treating loss events arriving at the source as a Poisson stream rather than packets going out on the network with some loss probability p. This enables us to model the window size behavior as a Poisson Counter driven Stochastic Differential Equation and perform analysis. We use the data collected in [1] to validate our modeling and analysis technique. Results indicate that our model is able to capture the behavior of TCP throughput quite accurately. Our technique enables simple fluid analysis of TCP and other TCP-like congestion control mechanism.**

I. Introduction

The Transmission Control Protocol (TCP) [2] carries a significant amount of today's Internet traffic. Applications like WWW (HTTP), email (SMTP), file transfer (FTP), remote access (Telnet) etc. use TCP as a means of reliable transport. The ubiquity of TCP necessitates that applications, which use some other transport protocol but share the same link, remain "TCP-friendly" in the sense that they should get a similar share of bandwidth as TCP connections under the same conditions of loss, round trip delay etc. Additionally, the stability of TCP is prompting designers of congestion control mechanisms for Multicast applications to remain TCP-friendly. Examples of these are given in [3], [4], [5]. Hence there is a great interest in analyzing the performance of TCP and TCP-like algorithms to quantify this notion of "TCP-friendliness". There have been a number of studies of TCP behavior, both via simulation as well as analysis [1], [6], [7], [8], [9]. Aside from that, there is interest in the formal analysis of TCP to study the effects of modifications like RED [10] and study the performance of TCP-like algorithms for other congestion control mechanisms. In this paper, we develop and analyze a new model for the TCP Congestion Avoidance algorithm. Our models differs from other models principally in the way the losses are modeled. Traditional modeling of losses in TCP analysis have been done from a source-centric point of view. Those models assume that packets go out on the network with some loss probability p which may be constant or depend upon factors like current window size etc. In our model, the *network* is the source of losses (congestion) and sources receive these signals (loss indications) as a Poisson process with some rate λ. We then model the window size of TCP as a fluid, having continuous increments as opposed to discrete ones. This model lends itself to a formulation of the window size behavior as a Poisson Counter driven Stochastic Differential Equation [11] (PCSDE). We analyze the PCSDE and obtain closed form solutions for TCP throughput. We consider the effects of different kinds of losses, viz. Timeout losses (TO) and Triple Duplicate Ack (TD) losses. Our closed form solution also explicitly accounts for the maximum window size limitation for TCP connections, which is an important parameter in characterizing the performance of TCP. We compare our results to other known results and formulae for TCP throughput. The remainder of the paper is organized as follows. We describe our model in Section 2, briefly reviewing the properties of Poisson Counters necessary for the development of the next section. We then cast the window size behavior as a Stochastic Differential Equation. In Section 3 we analyze our model and obtain closed form solution for TCP throughput by solving the SDE. We compare predictions of our formula with measurements done in [1] as well as formulas derived there in Section 4. We see that our formula does quite well, in regions of moderate to high throughput giving results comparable to and in some cases better than results derived in [1]. Finally in Section 5 we present our conclusions.

[1]This work is supported in part by the National Science Foundation under Grants CCR-9506764, CCR-9803727, EEC-9527422, NCR-9523807 and ANI-9809332, by the U.S. Army Research Office under Contracts DAAG55-98-1-0042, and by Air Force Rome Laboratory under Contract F30602-98-C-0263.

II. Modeling

A. Loss modeling

TCP implements the additive increase multiplicative decrease scheme to achieve a fair division of available bandwidth in a network amongst competing sources (users). It is a window based method in which, at any time, *window size* number of data packets are allowed in the network. This window size is additively increased roughly every *round trip time* (RTT) until congestion is detected whereupon it is multiplicatively decreased. The detection of congestion in the current implementations of TCP is implicit, i.e. congestion is only detected by the loss of packets. All flavors of TCP (TAHOE, VEGAS, RENO, SACK, FACK etc.) are successive refinements of the original attempt to implement ideal additive increase multiplicative decrease behavior. There have been proposals to make the congestion indication more explicit using some marking schemes [10], [12]. Recent analyses of TCP [1], [7], [8], [9] has been done from a source centric point of view. The assumption is that a source sends out packets in the network with an associated loss probability p, which could be identical for each packet (Bernoulli trials) or dependent on states like current window size etc. As earlier stated, we model the losses in a completely different and network centric way. We try to model losses keeping in mind one of the goals of TCP ,

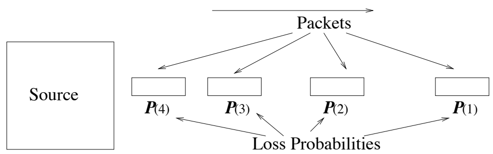

Fig. 1. Source Centric Loss Model

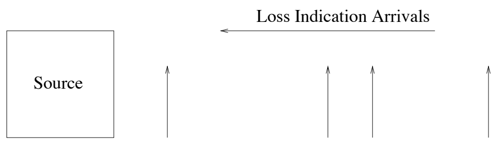

Fig. 2. Network Centric Loss Model

which is congestion avoidance. We assume that the network is a source of congestion, which the TCP flow tries to detect using losses. We assume that this "detection arrival" or loss-arrival is a Poisson process. The difference in the two loss models is exemplified by Figures 1 and 2. Figure 1 shows the source centric loss model, in which packets leave the source and go out on the network with an associated loss probability p_i, which may be constant or variable depending on factors like current window size etc. In contrast, in our network centric loss model, loss indications arrive at the source from the network at a certain rate (in the form of duplicate ACKs or gaps in sequence numbers). Specifically we model this arrival process as a Poisson process.

The basic reasoning behind our model is as follows: a typical TCP connection on the Internet traverses a number of hops before reaching the final destination. At each hop (router), the packet is enqueued or dropped if the buffer is full. This overflow process is the congestion in the network which TCP tries to detect. If the buffer overflow process at each queue is stationary and orderly, then the overall congestion process is a sum of these individual stationary, orderly processes. As the number of hops (with losses) increases, the congestion process starts approaching a Poisson process by Khinchine's theorem [13]. Another factor contributing to the Poisson like behavior is that the losses seen by a single flow would be a sampling of the buffer overflow process, much like probabilistic thinning. As the number of flows at each router starts increasing, then by Kallenberg's theorem [14] the thinned process starts approaching a Poisson process. Therefore the loss process seen by a single flow at a router is itself Poisson like already, making the aggregated loss process seen by the flow closer to Poisson according to Khinchine's theorem. We emphasize that our loss model is completely different from all other models analyzed elsewhere. This loss model is central to the analysis carried out in this paper. At the end of the paper we present statistical verification of our loss model in an appendix.

B. *Traffic model*

We model the traffic as a fluid. The increase in window size is considered continuous instead of step increase. It increases by 1 every round trip time (RTT) and hence the continuous increase is represented as dt/RTT. We assume idealized behavior, i.e. we model the losses as Poisson streams. We have two different kinds of losses, triple duplicate ack (TD) losses and time out losses (TO). The window size goes on increasing linearly with every RTT until a loss occurs. For a TD loss the window size is reduced to half it's current value whereas for a TO loss the window size is reduced to 1. In addition, for a TO loss the sender does an exponential backoff for a time T0, 2T0....64T0 depending upon the number of successive TO losses detected. We model the different kinds of losses as Poisson arrivals with individual rates λ_{TD},λ_{T0},$\lambda_{T1...}$ etc. Each process is represented by a Poisson counter N_{λ_i} where λ_i is the arrival rate of the underlying Poisson process.

C. *Poisson Counter Driven Stochastic Differential Equations*

We briefly review the properties of the Poisson counter [11] for the purpose of this paper. Consider a Poisson process N with rate λ. We have

$$\begin{aligned} dN &= \begin{cases} 1 & \text{at Poisson arrival} \\ 0 & \text{elsewhere} \end{cases} \\ E[dN] &= \lambda dt \end{aligned}$$

where λ is the arrival rate of the Poisson process. If we have a stochastic differential equation of the form

$$dx = f(x(t))dt + \sum_{i=1}^{m} g_i(x(t))dN_i \tag{1}$$

then using Ito's rule we can write the differential equation for $\psi(x(t))$(we'll drop the (t) part for simplicity) in the following form

$$d\psi = \frac{\partial \psi}{\partial x} f dt + \sum_{i=1}^{m} (\psi(x + g_i(x)) - \psi(x))dN_i \tag{2}$$

The terms after the first one account for the jump in the function $\psi(x)$ introduced by the jumps $g_i(x)$ in x due to the counters N_i. Finally, the independence of jumps property states that

$$E[g(x)dN] = E[g(x)]E[dN]$$

D. *Differential equation for the window size*

Let W be the window size. Then,

$$dW = \frac{dt}{RTT} + (-W/2)dN_{TD} + (1 - W)dN_{T0}. \tag{3}$$

The first term reflects the additive increase part of TCP, the second term reflects the multiplicative decrease and the third term reflects the timeout behavior. For simplicity we consider only single timeout losses. We have not modeled the slow start behavior of TCP since for typical window sizes seen in real traces, the slow start part of the process takes only a few round trip times and hence we neglect it for our analysis. It is possible to model the slowstart behavior in a differential equation of the following form. If T is the slowstart threshold, then

$$\begin{aligned} dW &= I_T(W)\frac{dt}{RTT} + (1 - I_T(W))W\frac{dt}{RTT} \\ &+(-W/2)dN_{TD} + (1 - W)dN_{T0}. \end{aligned}$$

where

$$I_T(W) = \begin{cases} 1, W > T \\ 0, W \le T \end{cases}$$

is an indicator function reflecting whether the window size is below the slowstart threshold or above it. However this complicates analysis and doesn't significantly affect the results (via examination of real measurements) and therefore we chose not to include it in our model.

III. Analysis

To calculate the expected value of W, we take expectations in the above differential equation, (for now we assume only TD and TO losses of the first order).

$$\begin{aligned} E[dW] &= E[\frac{dt}{RTT}] + (-\frac{E[W]}{2})E[dN_{TD}] \\ &+(1 - E[W])E[dN_{T0}]. \end{aligned}$$

Interchanging the operations of differentiation and expectation, we get

$$\begin{aligned} dE[W] &= \frac{dt}{RTT} + (-\frac{E[W]}{2})\lambda_{TD}dt + (1 - E[W])\lambda_{T0}dt \\ \frac{dE[W]}{dt} &= \frac{1}{RTT} + \frac{-E[W]}{2}\lambda_{TD} + \lambda_{T0} - E[W]\lambda_{T0} \\ &= (\frac{1}{RTT} + \lambda_{T0}) - (\frac{\lambda_{TD}}{2} + \lambda_{T0})E[W]. \end{aligned}$$

Solving the above for $E[W]$, we get

$$E[W](t) = \frac{\frac{1}{RTT} + \lambda_{T0}}{\frac{\lambda_{TD}}{2} + \lambda_{T0}} + Ce^{-(\frac{\lambda_{TD}}{2} + \lambda_{T0})t}.$$

Taking the steady state solution ($t \to \infty$), we get

$$E[W] = \frac{\frac{1}{RTT} + \lambda_{T0}}{\frac{\lambda_{TD}}{2} + \lambda_{T0}}.$$

We'll take into account the timeout backoff taken by TCP in a later section. The throughput (R) of the connection is obtained by dividing the expected window size by RTT.

$$R = \frac{1}{RTT}(\frac{\frac{1}{RTT} + \lambda_{T0}}{\frac{\lambda_{TD}}{2} + \lambda_{T0}}).$$

Until now we have not considered any limitation on the maximum window size. It is, in fact a very important parameter and affects the solution considerably. To account for the maximum window size, we modify (3) by multiplying the first term by an indicator function, i.e.,

$$dW = I_M(W)\frac{dt}{RTT} + (-W/2)dN_{TD} + (1 - W)dN_{T0}.$$

where

$$I_M(W) = \begin{cases} 1, W < M \\ 0, W = M. \end{cases}$$

This ensures that the window size doesn't grow once it has reached M. Again, taking expectations we get

$$\begin{aligned} E[dW] &= E[I_M(W)]\frac{dt}{RTT} + E[-W/2]\lambda_{TD}dt \\ &+(1 - E[W])\lambda_{TO}dt. \end{aligned}$$

Interchanging the operations of differentiation and integration, we get

$$\begin{aligned} \frac{dE[W]}{dt} &= E[I_M(W)] \cdot \frac{1}{RTT} + (-\frac{\lambda_{TD}}{2})E[W] \\ &+\lambda_{TO} - E[W]\lambda_{TO} \\ &= P[W < M].\frac{1}{RTT} + (-\frac{\lambda_{TD}}{2})E[W] \\ &+\lambda_{TO} - E[W]\lambda_{TO}. \end{aligned}$$

Using the same principles as before, we get the expected window size as

$$E[W] = \frac{(1 - P[W = M])\frac{1}{RTT} + \lambda_{T0}}{\frac{\lambda_{TD}}{2} + \lambda_{T0}}. \quad (4)$$

Using (2), we can get an equation for $E[W^2]$and by a similar procedure as above we obtain

$$E[W^2] = \frac{2MP[W = M] - 2E[W] - \lambda_{TO}}{(\lambda_{TO} + \frac{3}{4}\lambda_{TD})}.$$

The unknown factor here is $P[W = M]$. To solve for $P[W = M]$we look at the process

$$V = M - W$$

If Figure 3 represents the evolution of window size, then the evolution of V is represented by Figure 4. Since the loss arrivals are modeled as Poisson, V corresponds to the well known *Virtual Waiting Time* ($V(t)$) in an $M/G/1$ queue. This enables an alternative analysis of the process giving us a sufficient number of equations to account for all of the unknowns. We proceed with the analysis in the following manner: The *average* work in system $E(V)$ is defined as the limit

$$E(V) = \lim_{t\to\infty} \int_0^t V(u)du/t$$

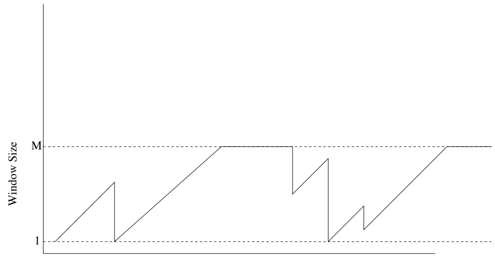

Fig. 3. Window size evolution for a TCP process

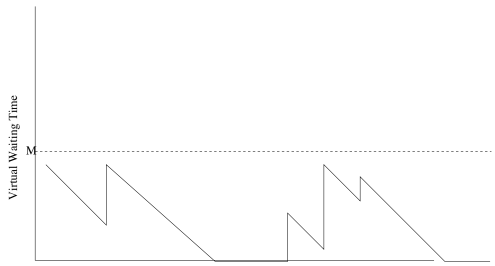

Fig. 4. Virtual Waiting Time in an $M/G/1$ queue

Every arrival brings in work which depends on the current workload in our system. The arrival of a TD loss brings in work equal to $(M - V)/2$ whereas the arrival of a TO loss brings in $(M - V - 1)$ amount of work, where V is the current workload. The corresponding quantities in the window size domain are $(W/2)$ and $(W - 1)$ respectively. Now let's assume that when a customer arrives in this hypothetical queue it is served to completion without interruption once it enters service. Let's look at the behavior for a two customer case in Figure 5. While an earlier customer is being serviced, the next (j^{th}) customer arrives. The arrival point on the time axis is marked by point e. The amount of work that the customer brings is the equal to the length of the line segment ad, which we denote by S_j. The prior customer gets serviced by the point marked f on the time axis. Thus the delay seen by the j^{th} customer is the length of the line segment ef, which we denote by D_j. Simple geometric considerations show that the contribution of the customer to the total area of the curve is the trapezoid $cbefg$, with the triangle cfg

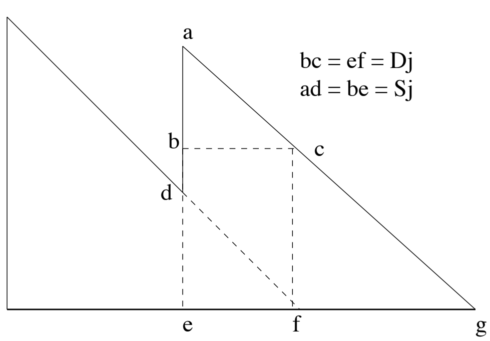

Fig. 5. Contribution to $V(t)$ by each customer

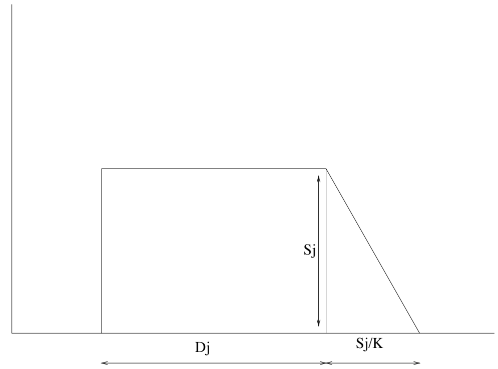

Fig. 6. Workload brought by each customer

contributing the part when the j^{th} customer is in service. We'll denote that by C_j. Then looking at Figure 6 the total amount of work contributed by this customer is $D_j S_j + C_j$. While in queue, the customer contributes $D_j S_j$to the system while when in service it contributes the remainder of S_j, i.e. C_j to the system. Thus the total contribution of the customer to the area of the curve is given by the trapezoid in Figure 6. The rectangle is the contribution of a customer while it is waiting for service, whereas the triangle is the contribution when it is in service. Let K be the service rate ($1/RTT$ in our case). Thus, if $\Lambda(t)$ arrivals have occurred in the interval $[0, t]$, then

$$\int_0^t V(u)du = \sum_{j=1}^{\Lambda(t)} (D_j S_j + S_j^2/2K) + \text{an error term}$$

with the error term occurring because of customers left unserviced by t. Dividing by t and taking the limit $t \to \infty$ we obtain an expression

$$E[V] = \lambda(E[SD] + E[S^2]/2K) \qquad (5)$$

where

$$E[S^2] = \lim_{n \to \infty} \sum_{j=1}^{n} S_j^2/n$$

and

$$E[SD] = \lim_{n \to \infty} \sum_{j=1}^{n} S_j D_j/n$$

The work brought by the customer depends on the type of loss arrival, as well as the current workload of the system. For notational convenience, let's denote λ_{TO} by $\lambda_1$and λ_{TD} by λ_2. Now, $E[S^2]$ is given by

$$\begin{aligned} S^2 &= (\lambda_1(W-1)^2 + \lambda_2(W^2/4)).\frac{1}{\lambda_{1+}\lambda_2} \\ E[S^2] &= (\lambda_1(E[W^2] - 2E[W] + 1) \\ &\quad +\lambda_2(E[W^2]/4).\frac{1}{\lambda_1 + \lambda_2} \end{aligned}$$

D, the delay the customer sees before service is $(M - W)/K$, where $W$was the window size at the time of arrival. Thus $E[SD]$ is given by

$$
\begin{aligned}
E[SD] &= E[(M-W).(\lambda_1(W-1) \\
&\quad +\lambda_2(W/2)).\frac{1}{\lambda_1+\lambda_2}].\frac{1}{K} \\
&= (M\lambda_1 E[W] - M\lambda_1 + M\lambda_2 E[W]/2 \\
&\quad -\lambda_1 E[W^2] + \lambda_1 E[W] - \lambda_2 E[W^2]/2) \\
&\quad .\frac{1}{\lambda_1+\lambda_2}\frac{1}{K}.
\end{aligned}
$$

Combining (5) with expressions for $E[S^2]$ and $E[SD]$ just obtained, we get

$$M - E[W] = (\lambda_1+\lambda_2)(E[SD] + E[S^2]/2K). \qquad (6)$$

We already have equations for $E[W]$ and $E[W^2]$ in terms of $P[W=M]$and combining that with (6), we solve for $P[W=M]$to obtain

$$P[W=M] = \frac{(2\lambda_1^2 + 2\lambda_1 + \lambda_1\lambda_2 + 2\lambda_1 K + 2K^2 + 2K)}{(K+1)(2M\lambda_1 + M\lambda_2 + 2K)}.$$

Substituting the value of$P[W=M]$ back into the equation (4), we get the value of throughput as a function of $M, RTT(1/K), \lambda_{TD}(\lambda_2)$ and $\lambda_{TO}(\lambda_1)$.

A. The timeout backoff

We can modify the preceding analysis to take into account the backoff $T0$ taken by TCP after a timeout occurs. The effect of the backoff is that the window size doesn't grow for a period of $T0$ seconds, after which it starts growing at the normal rate. Thus, we need to suppress the dt term for a period of $T0$ seconds after a timeout occurs. We can achieve that by multiplying the indicator function I_M with another indicator function I_{TO} which is 0 for a period $T0$ following a timeout and is 1 elsewhere. Thus, we have an event $\{W=M\}$ and another event $\{W \in TO\}$ representing the max-window size and timeout suppressions of dt respectively. The complements of the two indicator functions represent mutually exclusive events. Hence, the expectation operation would give the probability of the union of the two events, i.e.

$$E[I_M I_{T0}] = 1 - P[\{W=M\} \cup \{W \in TO\}]$$

We can carry out the entire analysis in the same manner replacing $P[W=M]$ with $P[\{W=M\} \cup \{W \in TO\}]$. The only change required is in the step to calculate the virtual waiting time. Every timeout arrival will bring in an additional workload of $(W-1)$ with a waiting time of $T0$ as explained by Figure 7. The dashed line indicates the additional waiting time $T0$ before the window starts growing again. Solving all the equations with the additional term, yields

$$
\begin{aligned}
&P[\{W=M\} \cup \{W \in TO\}] = \\
&\qquad P[W=M] + \frac{\lambda_1 K(2\lambda_1+\lambda_2)(M-1)T0}{(K+1)(2K+2M\lambda_1+M\lambda_2)}.
\end{aligned}
\qquad (7)
$$

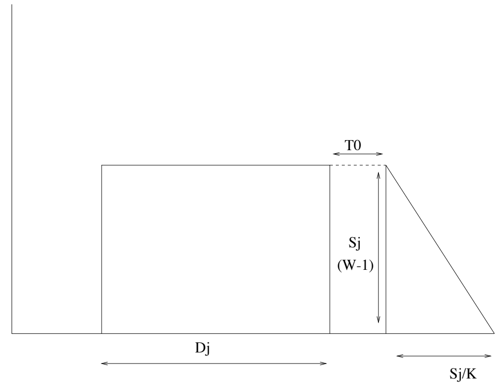

Fig. 7. Additional workload introduced by the timeout backoff

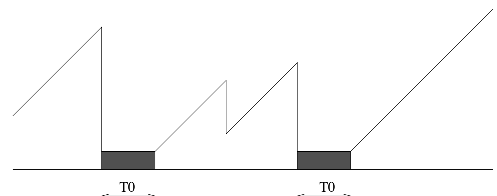

Fig. 8. Window size evolution with silence (backoff) periods

The presence of timeout backoff necessitates two corrections to be applied to the throughput formula to get correct results. If we look at Figure 8, the shaded portion refers to the backoff or "silence" mode. During that time TCP is not transmitting and we need to correct for that. The expression we obtained for $E[W]$ is for the entire time axis, we need an expression for $E[\hat{W}]$ where $\hat{W}$ is the average value over the active periods of TCP. We obtain that by the following: Let A be the total area of the curve and $\hat{A}$ the total "active" area. Let T be the total time TCP transmitted and λ_1 the arrival rate of the TO losses (we use $\lambda_1 \cdot T0$ as an estimate of the average time spent in the timeout backoff mode). The height of the inactive area is 1. Then

$$
\begin{aligned}
\hat{A} &= E[\hat{W}](T - \lambda_1 \cdot T0 \cdot T) \\
&= A - \lambda_1 \cdot T0 \cdot T \\
E[\hat{W}] &= \frac{\hat{A}}{T - \lambda_1 \cdot T \cdot T0} \\
&= \frac{E[W]T - \lambda_1 \cdot T0 \cdot T}{T - \lambda_1 \cdot T \cdot T0}.
\end{aligned}
$$

This is the average value transmitted per RTT. To get the number of rounds, we again have to look at the active period of TCP. Thus, we have, as our final throughput formula

$$
\begin{aligned}
R &= \frac{E[W] - \lambda_1 \cdot T0}{1 - \lambda_1 \cdot T0} \cdot \frac{T - \lambda_1 \cdot T0 \cdot T}{RTT} \cdot \frac{1}{T} \\
&= \frac{E[W] - \lambda_1 \cdot T0}{RTT}.
\end{aligned}
$$

where

$$E[W] = \frac{(1 - P[\{W=M\} \cup \{W \in TO\}])\frac{1}{RTT} + \lambda_1}{\frac{\lambda_2}{2} + \lambda_1}.$$

and $P[\{W = M\} \cup \{W \in TO\}]$ is given by (8).

B. Special cases and comparison with other models

Traditional analysis of TCP has been done using a packet loss model [1], [7], [8], [9]. Some involve considering timeouts while others don't. We can transform our formula to ones involving a packet loss probability by invoking the following argument. Let p be the loss probability, R the expected throughput and λ_1 and λ_2 the arrival rates in our loss model. Then

$$\begin{aligned} loss/sec &= \lambda_1 + \lambda_2 \\ packets/sec &= R \\ loss/packet(p) &= (loss/sec)/(packet/sec) \\ &= (\lambda_1 + \lambda_2)/R \end{aligned}$$

TCP has frequently been analyzed under the assumption of no timeouts ($\lambda_1 = 0$ in our model) [1]. Under that assumption, we have the following relations:

$$\begin{aligned} P[W = M] &= \frac{2K}{M\lambda_2 + 2K} \\ p &= \lambda_2/R \\ R &= \frac{2MK^2}{M\lambda_2 + 2K} \\ &= 2\frac{MK}{1 + \sqrt{1 + 2M^2 p}} \\ &= 2\frac{K}{1/M + \sqrt{1/M^2 + 2p}}. \end{aligned}$$

Most analyses of TCP are done with the assumption that there is no limit on the window size. That can be accommodated in the above formula by letting $M \to \infty$. Then the above formula reduces to

$$R = \frac{\sqrt{2}K}{\sqrt{p}}.$$

which is identical (up-to an empirical multiplicative constant) derived in [8] and [7]. If we use delayed ACKS, then our multiplicative constant reduces by a factor of $\sqrt{2}$ consistent with results shown elsewhere.

IV. RESULTS

We used the datasets collected in [1] to verify our results. In particular, we used the 100 second traces collected in that paper as well as the estimated values of RTT and $T0$ as given in the paper. To get an estimate of λ_{TD} and λ_{TO}, we simply used the number of loss arrivals of each kind in a a particular 100 second interval and divided that by 100 to get an average arrival rate. We compare the throughput predicted by our formula with that of the actual throughput (as well as throughput predicted by the formula given in [1] (we use only the exact formula derived in

[1]this assumption is likely to be more valid with the implementation of TCP SACK

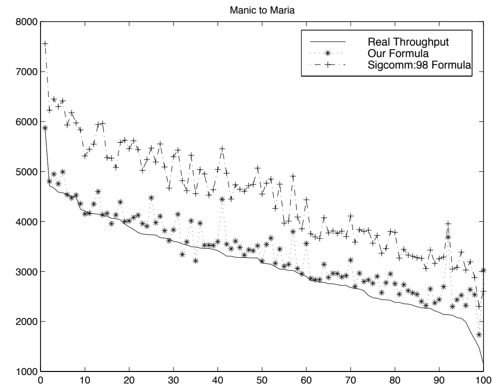

Fig. 9. Manic to Maria

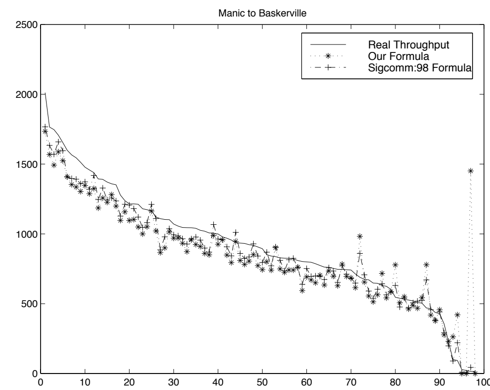

Fig. 10. Manic to Baskerville

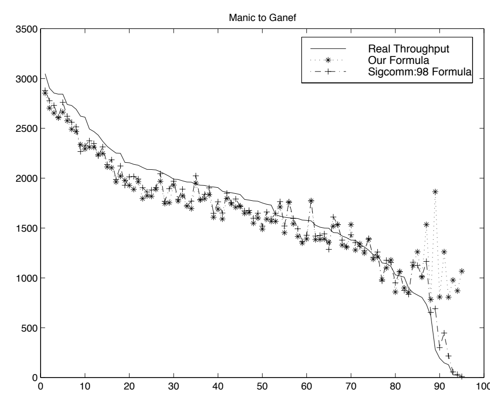

Fig. 11. Manic to Ganef

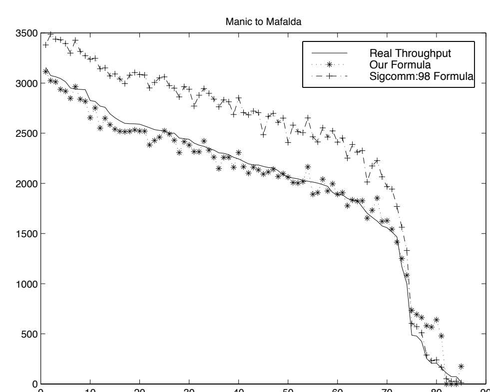

Fig. 12. Manic to Mafalda

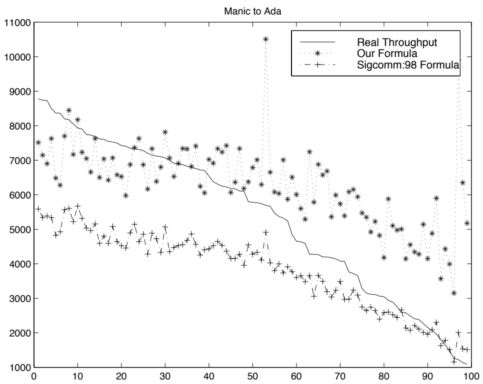

Fig. 13. Manic to Ada

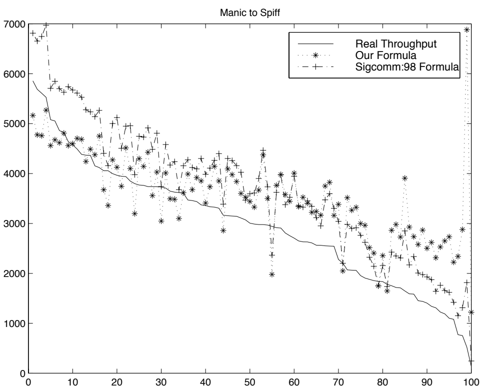

Fig. 14. Manic to Spiff

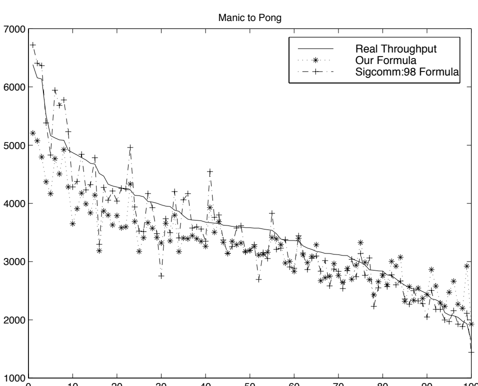

Fig. 15. Manic to Pong

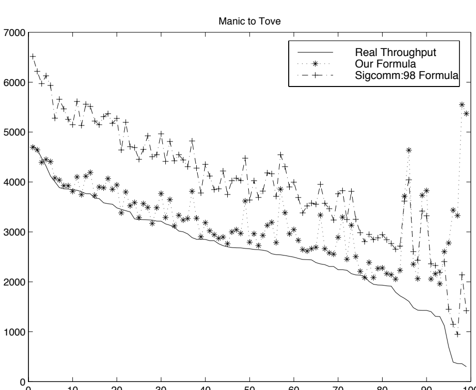

Fig. 16. Manic to Tove

that paper as it performs significantly better than others). We took the 100 traces for each experiment and plotted the actual throughput in descending order, i.e. the leftmost point of the curve represents the trace which gave the maximum throughput in a 100 second interval of the experiment and the rightmost gave the lowest throughput. For each of those points, we also plot the throughput predicted by the two formulas at the same x location. As can be seen, our formula does quite well in regions of moderately low to high throughput. In the case of very low throughput however our formula doesn't do as well. The reason for that being that in that regime TCP goes into multiple timeouts. Thus firstly our assumption of only a single timeout becomes invalid as is also our estimate of λ_{TO} (since only very few packets get transmitted, there are correspondingly only very few loss indications, thereby artificially introducing a low loss arrival rate). In addition, those areas correspond to loss rates of nearly 60-80 $\%$, which make our assumption of Poisson loss arrivals unrealistic.

V. Conclusion

In this paper we have developed a new model for TCP and TCP-like protocols. The key to our analysis is a completely different loss model. The loss model enables us to cast the TCP window size behavior as a Stochastic Differential Equation and obtain closed form solution for it's throughput via simple analysis. Although we ignore details like fast recovery, fast retransmit, slow start and make a fluid (continuous) approximation of the window size, our formula is able to predict real life measurements with a great deal of accuracy indicating the power of our model. Asymptotically our results reduce to results derived elsewhere. Our model reflects reality when a single flow doesn't significantly affect loss rates, as in the case of dedicated buffers. The simplicity of analysis and the accuracy of results suggests that our model could be a useful tool in doing fluid analysis of transport protocols. We are currently investigating application of our techniques in analyzing multicast congestion protocols.

VI. Acknowledgements

We'd like to express our sincere thanks to Jitendra Padhye of the Department of Computer Science, University of Massachusetts, Amherst for providing us with the datasets and data analysis tools used in this paper. We'd also like to sincerely thank Dan Rubenstien of the Department of Computer Science, University of Massachusetts, Amherst for giving useful comments and suggestions on earlier drafts of the paper.

Appendix

I. Statistical Tests

To test for Poisson arrivals, we have to test two different things: (a) independent inter-arrival times and (b) exponentially distributed inter-arrival times. Note that (b) by itself is not enough, as arrivals could happen deterministically to give an empirical distribution function which looks exponential. We use the test of renewal hypothesis suggested by Lewis and Robinson [15]. To test for exponentiality we use the Anderson-Darling [16] test.

A few sample plots for the data collected in [1] are presented (Figures 17-20). We treat the two types of loss indications sep-

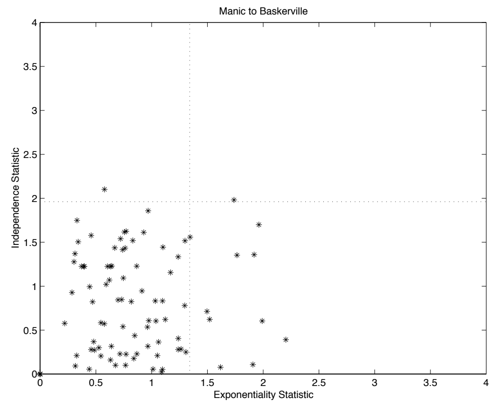

Fig. 17. Manic to Baskerville

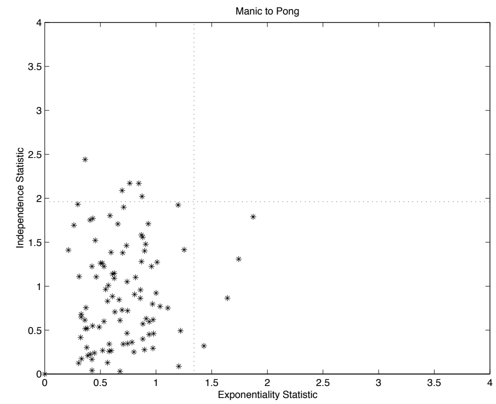

Fig. 19. Manic to Pong

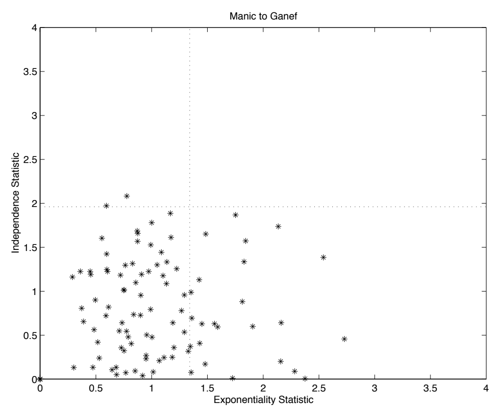

Fig. 18. Manic to Ganef

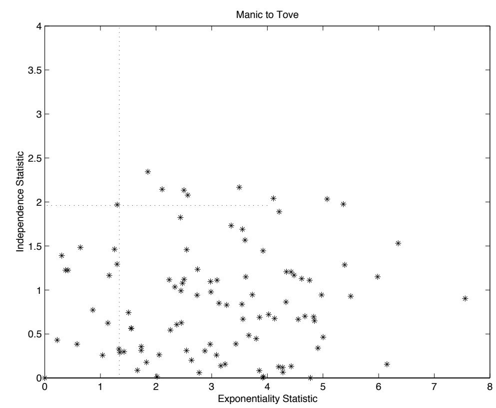

Fig. 20. Manic to Tove

arately, i.e. triple duplicate and time out losses are analyzed separately. We analyze 100 traces of length 100 seconds each for the plots presented. The renewal statistic is plotted on the y-axis whereas the exponentiality statistic is plotted on the x-axis along with 95 % confidence lines for both. Points to the left of and below the dashed lines satisfy the two criterion. As can be seen, the assumption of Poisson arrivals is not very unrealistic. The assumption about independent loss arrivals sees to be much more valid than the exponential distribution assumption. The renewal test was passed consistently by over 90% of the traces. The exponentiality test ranged in the 60-80% success range for most cases. Interestingly, even when the exponential distribution assumption fails badly, as in Figure 20, if we look at the throughput plots for that corresponding trace, our formula does quite well in predicting the performance, indicating the robustness of the technique. As the number of hops and (more importantly) the number of users sharing lossy links becomes higher and higher, we expect our model to come closer and closer to reality.

REFERENCES

[1] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP throughput: A simple model and its empirical validation," in *Proceedings of SIGCOMM'98*, 1998.

[2] V. Jacobson, "Congestion avoidance and control," *Computer Communications Review*, vol. 18, Aug. 1988.

[3] L. Vicisano, L. Rizzo, and J. Crowcroft, "TCP-like congestion control for layered multicast data transfer," in *Proceedings of INFOCOMM'98*, 1998.

[4] D. Sisalem and H. Schulzrinne, "The Loss-Delay Based Adjustment Algorithm: A TCP-Friendly Adaption Scheme," in *Proceedings of NOSSDAV'98*, 1998.

[5] J. Padhye, J. Kurose, D. Towsley, and R. Koodli, "TCP-Friendly Rate Adjustment Protocol for Continuous Media Flows over Best Effort Networks," 1999. UMASS CMPSCI Technical Report.

[6] T. Lakshman and U. Madhow, "The performance of TCP/IP for networks with high bandwidth-delay products," *IEEE/ACM Transactions on Networking*, vol. 5, June 1997.

[7] J. Mahdavi and S. Floyd, "TCP-friendly unicast rate-based flow control." Note sent to end2end-interest mailing list, Jan 1997.

[8] M. Mathis, J. Semke, J. Mahdavi, and T. Ott, "The macroscopic behavior of the TCP congestion avoidance algorithm," *Computer Communication Review*, vol. 27, July 1997.

[9] T. Ott, J. Kemperman, and M. Mathis, "The stationary behavior of ideal TCP congestion avoidance." ftp://ftp.bellcore.com/pub/tjo/TCPwindow.ps.

[10] S. Floyd and V. Jacobson, "Random Early Detection gateways for congestion avoidance," *IEEE/ACM Transactions on Networking*, vol. 1, August 1997.

[11] R. Brockett, "Stochastic control." Lecture Notes, Harvard University.

[12] F. Kelly, A. Maulloo, and D. Tan, "Rate control in communication networks: shadow prices, proportional fairness and stability," *Journal of the Operational Research Society*, 1998.

[13] A. Khinchine, *Mathematical Methods in the Theory of Queuing*. Hafner Publishing Co., New York, 1960.

[14] O. Kallenberg, "Limits of compound and thinned point processes," *Journal of Applied Probability*, vol. 12, pp. 269–278, 1975.

[15] P. Lewis and D. Robinson, *Reliability and Biometry*, ch. Testing for a monotone trend in a modulated renewal analysis. SIAM, Philadelphia, Pennsylvania, 1974.

[16] M. Stephens, "Edf statistics for goodness of fit and some comparisons," *Journal of American Statistics Association*, no. 69, pp. 730–737, 1974.