Fixed Point Approximations for TCP Behavior in an AQM Network *

Tian Bu, Don Towsley

Department of Computer Science
University of Massachusetts
Amherst, MA 01003
September, 2000
{tbu,towsley}@cs.umass.edu

Abstract

In this paper, we explore the use of fixed point methods to model the behavior of a large population of TCP flows traversing a network of routers implementing active queue management (AQM) such as RED (random early detection). Both AQM routers that drop and that mark packets are considered along with infinite and finite duration TCP flows. In the case of finite duration flows, we restrict ourselves to networks containing one congested router. In all cases, we formulate a fixed point problem with the router average queue lengths as unknowns. Once the average queue lengths are obtained, other metrics such as router loss probability, TCP flow throughput, TCP flow end-to-end loss rates, average round trip time, and average session duration are easily obtained. Comparison with simulation for a variety of scenarios shows that the model is accurate in its predictions (mean errors less than 5%). Last, we establish monotonicity properties exhibited by the solution for a single congested router that explains several interesting observations, such as that TCP SACK suffers higher loss than TCP Reno.

1 Introduction

Networks and their workloads increase in size and complexity at an exponential rate. Thus there is an increasing need for performance evaluation methodologies that can keep up with this rapid rate in network size. In contrast it is interesting to note that TCP, the predominant Internet protocol, has changed little in the last ten years. This suggests that it should be possible to develop an analytically based methodology that can be used to evaluate the performance of large networks (regardless of technology) that support large populations of TCP flows. In this paper we address this problem when the network consists of AQM routers, i.e., those that implement active queue management (AQM) such as RED (random early detection). We formulate a fixed point problem with the average queue lengths at the routers as unknown quantities.

*This work was supported in part by DARPA under contract F30602-00-2-0554.

We consider both routers that drop packets and those that mark packets. Once these are obtained, other performance metrics such as router loss/marking probability and TCP flow throughput are easily obtained. Comparison with simulation for a single router, tandem routers and cyclic routers show high accuracy for all the metrics (mean errors less than 5%).

The modeling methodology reported here builds on the analyses of Mahdavi and Floyd [17], Ott, et al. [21], and Padhye et al. [22] of an individual TCP flow operating in a lossy environment. These analyses resulted in simple expressions of throughput as a function of round trip time and loss probability. They differ from each other primarily in that [22] accounts for TCP timeout behavior whereas [17], [21] do not. Our fixed point approach can be thought of as an extension of the approach presented in [18] from a single router to a network setting. Furthermore, that paper relied on the use of the "square root p" formula derived in [21]. We will observe that reliance on the "square root p" formula results in significantly less accurate loss rate predictions than achieved through reliance on the formula in [22]. One of the conclusions of our study is that the "square root p" formula can result in significant errors even when the loss rate is less than 5%. Some analysis has been done using the "square root p" for a specific, constrained, multiple congested router scenario in [6]

We provide a methodology that handles networks in which a subset of routers drop packets according to an AQM while another subset of routers mark packets. The ability to model routers that mark packets is important as this allows one to evaluate the performance of heterogeneous networks that support explicit congestion notification (ECN) [25] which is expected to become dominant in the future Internet. We consider the case where TCP flows are infinite as well as finite in duration. In the latter case, however, we are only able to handle the case when the network includes one congested router.

A number of papers have dealt with related topics. Closely related are the works of Misra et al. [18] and Firoiu et al. [10]. The work of Misra et al. considered a single congested router supporting a set of infinite duration flows. This work, however, is based on the "square root p" formula, which we will see can lead to inaccurate predictions in loss rate and average queue length. The work of Firoiu et al.[1] presents fixed point models of packet drop RED networks. Unlike Firoiu's model, ours handles networks that include both packet dropping and packet marking AQM routers. Furthermore, our models for finite TCP flows don't assume there are always fixed number of finite flows in network whereas the model in [10] does. Last, we study the sensitivity of the accuracy of the model to the TCP throughput formulas used. Also closely related are the works of Heyman, et al. [14], and Casetti, et al. [4], which consider a single congested router supporting a homogeneous set of finite duration flows. Unlike these latter works, we make no assumptions regarding the homogeneity of TCP flows.

There has been related work focusing on the development and solution of a set of differential equations describing the transient behavior of TCP flows and queue dynamics [20]. Our approach complements this

[1]This work was done independently of the work presented in this paper

approach. The fixed point approach is much more efficient computationally as the number of unknowns equals the number of links in the network whereas the DE approach requires the solution of a number of equations equal to the number of routers + number of TCP flows. On the other hand, the DE approach can be used to study transient behavior. Last, there is a considerable body of literature on the application of fixed point methods to estimating blocking probabilities in circuit switched networks, [26].

The paper is organized as follows. Section 2 introduces the model and the fixed point approximation for a single congested router. Section 3 extends the model to a network of multiple heterogeneous congested routers. Section 4 describes the validation of the approximation to *ns2*-based simulation. In the following section, we explain several interesting observations by establishing monotonicity properties characterizing the solution for a single congested router. Section 6 concludes the paper.

2 A single congested router

In this section, we model the behavior of a large population of TCP flows sharing a single congested router. We assume that the congested router implements active queue management which either drops or marks packets for the sake of congestion control. We propose models for two scenarios, one in which TCP flows are infinite in duration and the other in which TCP flows are finite in duration.

Consider a router v with transmission capacity C_v bits per second and buffer size B_v bits. Associated with router v is a probability discard/marking function $p_v(x_v)$ which takes as its argument x_v, the average queue length of router v. One popular AQM is RED [13]. The discard/marking function of the recently recommended "gentle_" variant of RED (G-RED) [12] is

$$p_v(x_v) = \begin{cases} 0, & : \quad 0 \leq x_v < t_v^{min} \\ \frac{x_v - t_v^{min}}{t_v^{max} - t_v^{min}} p_v^{max}, & : \quad t_v^{min} \leq x_v \leq t_v^{max} \\ p_v^{max} + \frac{x_v - t_v^{max}}{t_v^{max}}(1 - p_v^{max}), & : \quad t_v^{max} < x_v \leq 2t_v^{max} \\ 1, & : \quad t_v^{max} < x_v \leq B_v \end{cases} \tag{1}$$

where t_v^{min}, t_v^{max} and p_v^{max} are configurable G-RED parameters. We assume that it is well engineered so that the probability of a packet arriving to a full queue is close to zero and the probability that the router is idle when v is congested is also close to zero. Guidelines on how to do this for several AQM policies, including RED can be found in [15].

2.1 Infinite TCP flows

We first consider a workload of N infinite duration TCP flows, labelled $i = 1, \ldots, N$, that traverse router v. We (and others) have observed through measurements on the Internet and in numerous simulations that

- in the absence of a maximum rate constraint, each TCP flow traverses at least one congested router (here a congested router is one in which a flow suffers packet loss);

- each congested router is nearly fully utilized;
- each TCP flow exhibits a throughput that can be expressed as a function of the end-to-end packet loss/marking rate and average packet round trip time. Denote this by $T(q, R)$ where q denotes the end-end loss/marking rate and R the round trip time. For simplicity we assume that the TCP ACK packets will never be dropped. However our model can be extended to account for TCP ACK packet loss.

2.1.1 Packet dropping congested router

In this section we consider the scenario where the congested AQM router drops packets according to p_v. We consider the following two expressions for $T(q, R)$ that have appeared in the literature [21] [17] [22],

$$T_i(R_i, q_i) = M\frac{1.22(1-q_i)}{R_i\sqrt{q_i}}, \tag{2}$$

$$T_i(R_i, q_i) = M\frac{(1-q_i)((1-q_i)/q_i + W(q_i) + Q(q_i, W(q_i))/(1-q_i))}{R_i(W(q_i)/2+1) + Q(q_i, W(q_i))F(q_i)T_0/(1-q_i)} \tag{3}$$

where M is the packet size measured in bits and

$$\begin{aligned} W(q) &= 2/3 + 2\sqrt{(1-q)/(3q) + 1/9}, \\ Q(q, w) &= \min\{1, (1-(1-q)^3)(1+(1-q)^3(1-(1-q)^{w-3}))/(1-(1-q)^w)\}, \\ F(q) &= 1 + q + 2q^2 + 4q^3 + 8q^4 + 16q^5 + 32q^6. \end{aligned}$$

Here $W(q)$ is the expected window size at the time of a loss event; $Q(q, w)$ is the probability that a packet loss is detected by a timeout; T_0 is the latency of the first timeout without back-off; $F(q)/(1-q)$ is the expected number of times that T_0 is doubled. Henceforth, we refer to (2) and (3) as the *square root* and *PFTK* formula respectively.

The square root formula, (2), which ignores the effects of timeouts, has been derived in [21]. The second expression is the production of the TCP sending rate derived in [22] and the end to end success probability. This TCP sending rate expression was shown to accurately model the effects of timeouts. Expression (3) differs from the throughput expression derived in [23] for TCP-Reno operating over a drop tail queue. Our simulation results suggest that (3) is more accurate for AQM mechanisms such as RED. Note that TCP formulas (2) and (3) are asymptotically the same as $q \to 0$.

We further make the following assumptions:

- All of the TCP flows traversing the single congested router experience the same loss probability which is introduced by the AQM router discard function, i.e. $q_i = p_v(x_v), \quad i = 1, \ldots, N$.

- The expected round trip time of flow i, R_i can be expressed as $R_i = A_i + x_v/C_v$. Here A_i accounts for the two way propagation delay and the transmission times at router v. The second term corresponds to the expected queue delay through the congested router v.

Note that, as we have expressed q_i and R_i in terms of x_v, we can write the throughput of flow i as $T_i(x_v)$, $i = 1, \ldots N$.

We now focus on determining x_v. We have following equation,

$$\sum_{i=1}^{N} T_i(x_v) = C_v. \tag{4}$$

which can be solved to obtain x_v. Once obtain x_v, we can estimate the loss probability of a congested router and throughput of all TCP flows.

Theorem 1 *There exists an unique solution to equation (4) provided that*

1. $p_v(x_v)$ *is continuous and non-decreasing in* x_v.
2. $p_v(0) = 0$
3. $\sum_{i=1}^{N} T_i(B_v) < C_v$

Proof: Equations (2) and (3) coupled with the assumption that $p_v(x_v)$ is continuous and non-decreasing allows us to conclude that $T_i(x_v)$ is decreasing and continuous in x_v. Furthermore, $\lim_{x_v \to 0} T_i(x_v) = \infty$. These two properties coupled with the assumption that $\sum_{i=1}^{N} T_i(B_v) < C_v$ imply that there is one and exactly one solution to

$$\sum_{i=1}^{N} T_i(R_i(x_v), q_i(x_v)) = C_v. \tag{5}$$

■

2.1.2 Packet marking congested router

Throughput expressions (2) and (3) are valid only under the assumption that packets are dropped and have to be modified in order to model a packet marking router. Since a TCP flow experiences no loss when a packet is marked, there are no timeout events and the throughput is the same as the sending rate. As (2) does not account for timeout, it can simply be replaced by the sending rate expression,

$$T_i(R_i, q_i) = M \frac{1.22}{R_i \sqrt{q_i}}, \tag{6}$$

A modification of (3) requires the timeout probability to be set to zero. The result is

$$T_i(R_i, q_i) = M\frac{(1-q_i)/q_i + W(q_i)}{R_i(W(q_i)/2+1)} \tag{7}$$

We can now determine x_v by using either (6) or (7) coupled with equation (4). Theorem 1 holds also for the case that the congested router marks packets instead of dropping packets.

2.2 Finite TCP flows

We consider a scenario where there are N classes of finite duration TCP flows . TCP flows within the same class have the same characteristics, the same round trip time and duration distribution whereas TCP flows within different classes have different characteristics. We assume that all TCP flows traverse a single congested router v, i.e., $R_i(x) = A_i + x_v/C$, $i = 1, \ldots, N$. The TCP flows within class $i, i = 1, \ldots, N$, arrive according to a Poisson process with rate λ_i and the number of packets to be transferred by this flow is exponentially distributed with expected value $1/\mu_i$. For now, assume that a flow in class i achieves a throughput $T_i^{fin}(x_v, 1/\mu_i)$.

We digress for a moment and review a processor sharing model that was studied in [5]. Consider a single-server processor sharing system with N customer classes. Assume that the server has a processing rate of one. Associated with the classes are weights $\{g_i\}_{i=1}^N$. Let $L_i(t)$ denote the number of class i customers in the system at time t. Then, a class i customer receives service at rate $g_i/\sum_{k=1}^N g_k L_k(t)$ at time t.

Let $L_i = \lim_{t\to\infty} L_i(t)$. From the analysis in [5], when the customers within each class arrive according to a *Poisson process* with rate λ_i and require an amount of service that is exponentially distributed with expected value $1/\mu_i$, we have the following set of linear equations describing the expected values of L_i, $E[L_i]$

$$\frac{E[L_k]}{\lambda_k}\left[1 - \sum_{j=1}^N \frac{\lambda_j g_j}{\mu_j g_j + \mu_k g_k}\right] - \sum_{j=1}^N \frac{E[L_j] g_j}{\mu_j g_j + \mu_k g_k} = \frac{1}{\mu_k} \tag{8}$$

We can apply these results to our model by simply noting that, for any pair of TCP flows from classes i, j sharing a congested router, $g_i/g_j = T_i^{fin}(x, 1/\mu_i)/T_j^{fin}(x, 1/\mu_j)$. We then rewrite (8) as

$$\frac{E[L_k]}{\lambda_k}\left[1 - \sum_{j=1}^N \frac{\lambda_j}{\mu_j + \mu_k \frac{T_k^{fin}(x_v, 1/\mu_k)}{T_j^{fin}(x_v, 1/\mu_j)}}\right] - \sum_{j=1}^N \frac{E[L_j]}{\mu_j + \mu_k \frac{T_k^{fin}(x_v, 1/\mu_k)}{T_j^{fin}(x_v, 1/\mu_j)}} = \frac{1}{\mu_k} \tag{9}$$

We assume that the link is fully utilized as long as one or more TCP flows are active. We then have

$$\sum_{i=1}^N E[L_i] T_i^{fin}(x_v, 1/\mu_i) = M \sum_{i=1}^N \lambda_i/\mu_i \tag{10}$$

Combining (9) and (10), we have $N+1$ equations and $N+1$ unknowns, $E[L_i], i, \ldots, N$, x_v.

We now determine $T^{fin}()$ in both the packet dropping router and the packet marking networks.

2.2.1 Packet dropping congested router

In order to model finite duration flows, we need to account for the TCP startup phase. To do so, we rely on a model proposed in [2] to compute the TCP latency for transferring d data packets that accounts for this startup phase. Let D denote the time required to complete a TCP data transfer. The expected delay for transferring d data packets is,

$$E[D|d] = E[D_{ss}|d] + E[D_{loss}|d] + E[D_{ca}|d] + E[D_{delack}|d] \tag{11}$$

where $E[D_{ss}|d]$ is the expected duration of the slow start phase obtained from expression (15) in [2]; $E[D_{loss}|d]$ is the expected delay due to any retransmission timeouts or fast recovery that happens at the end of the initial slow start phase and is obtained from expression (20) in [2]; $E[D_{ca}|d]$ is the expected time required to send remaining data after slow start and loss recovery obtained from expression (24) in [2]; $E[D_{delack}|d]$ is the expected delay due to delayed acknowledgments and is determined by implementation details. Besides the number of packets d, two other parameters that affect expected TCP latency are average round trip time R and end-to-end loss probability q. Assume that TCP flow i has d_i packets to transfer and that its round trip R_i and loss probability q_i are known. Let D_i be the transfer time for TCP flow i; then $E[D_i|d_i]$ can be computed using (11). The throughput expression for finite TCP flow i can be written as $T_i^{fin}(R_i, q_i, d_i) = d_i/E[D_i|d_i]$. Since both R_i and q_i are functions of x_v as shown earlier, the throughput function for TCP flow i is expressed as $T_i^{fin}(x_v, d_i)$.

2.2.2 Packet marking congested router

We have to modify expression (11) to compute the expected delay for finite TCP sessions when the congested router marks packets. The term $E[D_{ss}|d]$ remains the same except that we use the marking probability instead of the loss probability as we apply expression (15) in [2]. The term $E[D_{loss}|d]$ computing the expected cost of the first loss disappears because the router never drops packets. $E[D_{delack}|d]$ is the same as in a packet dropping network when packets are dropped. The computation of $E[D_{ca}|d]$ relies on the TCP throughput function derived in [23] for infinite TCP flows. To account for packet marking we replace the infinite TCP throughput expression in expression (24), [2] with the infinite TCP throughput function (7) derived earlier for the packet marking router. Once the expected delay is available, the throughput can be obtained using the technique we presented for the packet dropping router.

2.2.3 General TCP session sizes

Until now we have assumed that the number of packets transferred in a finite TCP session is exponentially distributed. This is easily extended to the case where the number of packets to be transfered is described by a hyperexponential distribution, i.e., a distribution represented by parallel exponential stages. Assume that the transfer size of a class j TCP session is described by an s_j stage hyperexponential distribution. With probability $\alpha_{j,k}$ the session size is exponentially distributed with mean $1/\mu_{j,k}$. Note that $\sum_{k=1}^{N} \alpha_{j,k}/\mu_{j,k} = 1/\mu_j$ and $\sum_{k=1}^{N} \alpha_{j,k} = 1$. We divide the TCP sessions in class j into s_j subclasses $j_1, \ldots . j_{s_j}$ where $\lambda_{j,k} = \alpha_{j,k}\lambda_j$. This results in $\sum_{j=1}^{N} s_j$ subclasses of TCP sessions. A session in subclass (j,k) transfers a number of packets that is exponentially distributed with mean $1/\mu_{j,k}$. Furthermore, sessions from subclass (j,k) arrive according to a Poisson process with rate $\alpha_{j,k}\lambda_j$. Thus our earlier model applies here.

This extension is especially useful as there exist techniques for using a hyperexponential distribution to fit a heavy tailed transfer distribution [8].

3 A network of multiple congested heterogeneous routers

In the previous section, we proposed models for a network containing a single congested router for both infinite and finite duration TCP flows. In this section we first extend models for infinite TCP flows to a network of multiple congested AQM routers where some routers drop packets whereas the other routers mark packets for the sake of congestion control. How to model the finite TCP flows in a network containing more than one congested routers is a challenging problem that we have not solved.

Let V be a collection of AQM routers. Let V_d and V_m denote the set of packet dropping and the set of packet marking routers respectively. We have $V_d \cup V_m = V$. Each router $v \in V$ has a transmission capacity of C_v bits per second. In addition, router v can buffer up to B_v bits. Associated with each router $v \in V$ is a probability discard/mark function $p_v(x_v)$ which takes as its argument x_v, the average queue length of router v and is expressed in (1).

Let us consider a workload of N infinite duration TCP flows labelled $i = 1, \ldots, N$. Let $V_i = (j_{i,1}, j_{i,2}, \ldots, j_{i,n_i})$ be the ordered set of routers (i.e., route) taken by packets of flow i, where $j_{i,m} \in V$, $m = 1, \ldots, n_i$ and n_i is the route's length. From the perspective of a link $v \in V$, it is useful to introduce S_v as the set of TCP flows that traverse v. In addition, let $V_i(u) = (j_{i,g(i,u)}, \ldots, j_{i,n_i})$ be the portion of the path from router $g(i,u)$, the next router after u on the flow i's path, to the receiver, inclusive.

Let $\boldsymbol{x}$ denote the vector of x_v, $v \in V$. We make following assumptions

- The expected round trip time of flow i, $R_i(\boldsymbol{x})$ can be expressed as

$$R_i(\boldsymbol{x}) = A_i + \sum_{v \in V_i} x_v/C_v \tag{12}$$

Here A_i accounts for the two way propagation delay and the sum of the transmission times at all of the routers on the route of flow i. The second term corresponds to the average queue delay through the routers on the path.

- Drops and marks occurs as independent events. If we let $q_i^d(\boldsymbol{x})$ (resp. $q_i^m(\boldsymbol{x})$) denote the probabilities that a flow i packet is dropped (resp. marked) on its end-to-end path, then they are given by

$$q_i^d(\boldsymbol{x}) = 1 - \prod_{v \in V_i \cap V_d} (1 - p_v) \tag{13}$$

$$q_i^m(\boldsymbol{x}) = (1 - q_i^d(\boldsymbol{x}))(1 - \prod_{v \in V_i \cap V_m} (1 - p_v)) \tag{14}$$

  For simplicity, we also assume that TCP ACKs are neither marked nor dropped.

The expressions for TCP throughput in the literature so far are for either a pure packet dropping network or a pure packet marking network. In order to complete our model for a network containing both packet dropping and packet marking routers, we have to first derive TCP throughput formulas that can be expressed as functions of the end to end packet loss probability, the end to end packet marking probability and the average round trip time. These TCP throughput formulas are obtained by modifying either (2) or (3). Let define $q_i = q_i^m + q_i^d$. Since a TCP flow responds both to packet losses and packet marks, we modify (2) to

$$T_i(R_i, q_i^d, q_i^m) = M \frac{1.22(1 - q_i^d)}{R_i \sqrt{q_i}}, \tag{15}$$

Equation (3) accounts for timeouts. Since timeouts can only introduced by packet drops and not by packet marking we modify (3) to

$$T_i(R_i, q_i^d, q_i^m) = M \frac{(1 - q_i^d)((1 - q_i)/q_i + W(q_i) + Q(q_i^d, W(q_i))/(1 - q_i^d))}{R_i(W(q_i)/2 + 1) + Q(q_i^d, W(q_i))F(q_i^d)T_0/(1 - q_i^d)} \tag{16}$$

Note that we can express q_i^d, q_i^m and R_i in terms of $\boldsymbol{x}$; hence we can write the throughput of flow i as $T_i(\boldsymbol{x}), i = 1, \ldots, n$.

We now give the set of equations that describe the behavior of $\boldsymbol{x}$. Let $S \subset V$ denote the set of congested routers. Let $T_{j,v}(\boldsymbol{x}), j \in S_v$ be the rate at which packets from flow j leave congested router v. We have $\sum_{j \in S_v} T_{j,v}(\boldsymbol{x}) = C_v$ from the assumption that a congested router is fully utilized. We can relate $T_{j,v}(\boldsymbol{x})$ and $T_j(\boldsymbol{x})$ as follows. $T_j(\boldsymbol{x})$ is the rate at which packets for flow j arrive at its receiver after traversing the remaining routers on its path, $V_j(v)$ the fraction of packets belonging to flow j that leave v and are dropped is $\prod_{u \in V_j(v) \cap V^d}(1 - p_u(x_u))$. Thus, $T_j(\boldsymbol{x}) = T_{j,v}(\boldsymbol{x}) \prod_{u \in V_j(v) \cap V^d}(1 - p_u(x_u))$. Therefore, We have following set of equations, one for each congested router,

$$\sum_{j \in S_v} (T_j(\boldsymbol{x}) / \prod_{u \in V_j(v) \cap V^d} (1 - p_u(x_u))) = C_v, \quad v \in S. \tag{17}$$

Otherwise

$$\sum_{j \in S_v} (T_j(\boldsymbol{x}) / \prod_{u \in V_j(v) \cap V^d} (1 - p_u(x_u))) < C_v \qquad \text{and} \qquad x_v = 0, \qquad v \in V \setminus S.$$

where $T()$ is the throughput function in a network containing both packet dropping and packet marking routers which takes expression either (15) or (16).

This model can be used to model a pure packet dropping network or a pure packet marking network by letting $V_d = \phi$ or $V_m = \phi$ respectively.

We currently do not have any good results regarding the existence of solution except for some special cases. However our experiments have always produced "reasonable" solutions.

The model in this section can be extended to predict the behavior of a network that contains UDP flows as well as TCP flows. As UDP flows share the congested router with TCP, we subtract the average UDP throughput from the capacity of the congested router and assume that TCP flows take the rest of the router capacity. The detailed model and validation can be found in [1].

4 Model Validation

In order to understand the accuracy of the models proposed in the earlier sections, we performed a set of simulations under a variety of network conditions using the *ns2* simulator [16]. We varied the TCP version, network topology, G-RED configuration, propagation delay, traffic load, and traffic type. We simulated G-RED where the packet dropping/marking probability varies from p^{max} to 1 as the average queue size varies from t^{max} to twice t^{max} by setting the "gentle_" parameter to 1 as recommended in [12]. We also set the RED parameters in such a way that the average queue size lies between t^{min} and t^{max}. We believe a well engineered router should always have this property, although our solution algorithm does not rely on this limitation. To obtain the best behavior of RED, the RED routers in the simulation are all configured according to [11]. We simulated TCP Reno/SACK with a packet size 500 bytes. We first validate the model in the case of a single congested router where offered loads consisting of infinite and finite TCP flows. We then evaluate the model for tandem and cyclic networks containing multiple congested routers. For each test network, we estimate the network metrics using the models proposed in the preceding sections. We focus on the following metrics, average queue occupancy and drop rate within each RED router, throughput and end-to-end loss probability as seen by each TCP flow, and the average latency to transfer finite TCP sessions.

4.1 A single congested router: Infinite TCP flows

In this set of simulations, there are a total of N TCP flows sharing a common link with capacity C (Figure 1). Link 0-1 is the only congested link encountered by any TCP flow. (i.e. packet dropping/marking only

occurs on link 0-1).

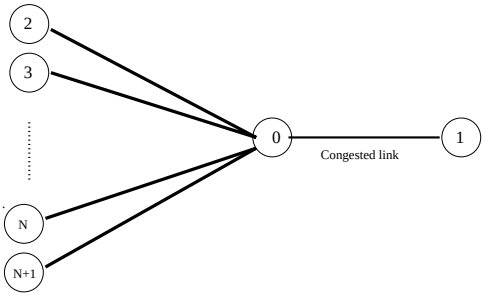

Figure 1: The n TCP flow share a single congested link

4.1.1 Packet dropping router

We set the G-RED control parameters of router 0 to be $p_0^{max} = 0.1$, $t_0^{min} = 30$, $t_0^{max} = 90$, $B_0 = 180$, and $C_0 = 4$Mbps. Since the G-RED control function is fixed, the average queue occupancy and drop rate at router 0 increase as the number of TCP flows increase. We consider a scenario where individual TCP flows exhibit different round trip times. This is done by varying the propagation delay. For TCP flow i, we let $A_i = (2i + 20)ms$, $i = 1, \ldots, N$. We run the simulations for $N = 10, 20, \ldots, 120$. In order to verify our model for different TCP versions, we run each simulation twice, one with TCP Reno and the other with TCP SACK.

We only present the results for TCP SACK from both the model and the simulation in Figure 2 because the model predictions for TCP Reno are only slightly better and TCP SACK is becoming more and more dominant in the Internet. Each graph is a scatter-plot of the different metrics, throughput of each TCP flow, average queue occupancy and loss probability within the G-RED routers. We plot the measured metrics along the x-axis and the estimated metrics along the y-axis. The PFTK model provides accurate estimates of all three metrics. On the other hand, the square root model only provides reasonable estimates of the throughput but significantly over-estimates the average router queue length and average router loss rate. The error in the loss rate estimate increases as the loss rate increases. This is due to the fact that the square root TCP formula doesn't account for timeouts and the timeout events become more frequent as the loss rate increases. In order to verify this last statement, we measured the the probability that a packet loss is detected by a timeout in our simulation and plot it as a function of measured loss rate for both TCP Reno and TCP SACK in Figure 3. We observe that even when the packet loss is as low as 4%, the probability that a packet loss detected by a timeout is about 25% for TCP Reno and 18% for TCP SACK. It is also observed that the timeout probability of TCP SACK is less than that of TCP Reno under same packet loss rate.

We find from our simulations that the TCP SACK flows suffered higher a verage queue latencies and loss rates than TCP Reno flows . We will return to this in Section 5.

We have examined the accuracy of the model predictions as we increase the number of TCP flows but fix the G-RED parameters. We also performed a set of simulations on the same topology but scaled the G-RED parameter of an AQM router with the number of TCP flows sharing the router. Assume there are a total of N TCP flows, the G-RED control parameter of the congested router are configured to be $p_0^{max} = 0.1$, $t^{min} = 3N/2$, $t_0^{max} = 9N/2$, $B_0 = 9N$, $C_0 = N/10$Mbps. The conclusion from this set of simulations is that the model predictions become more accurate as the network scales up.

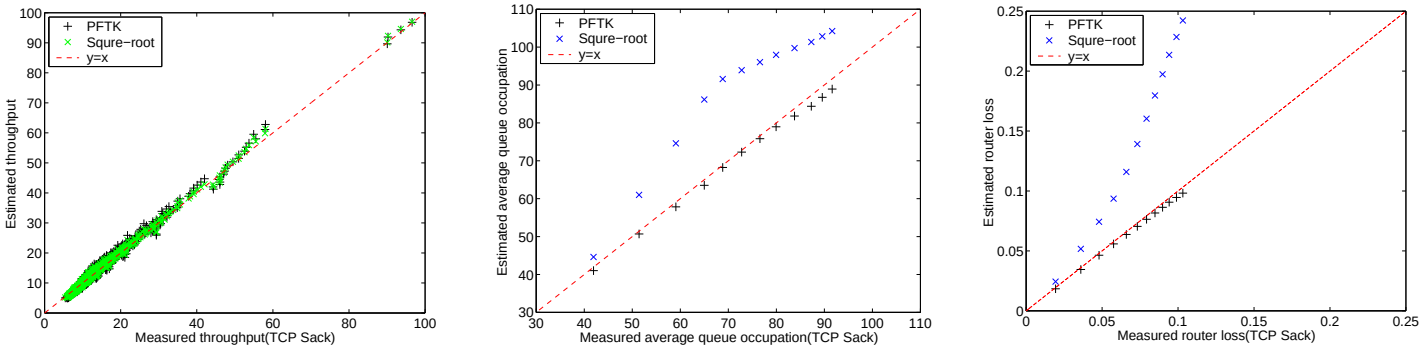

Figure 2: Estimate vs. Simulation: summarized results for the single congested packet dropping router

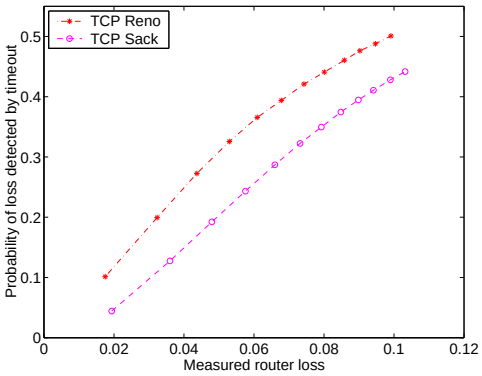

Figure 3: The significance of timeout

4.1.2 Packet marking router

We ran the same set of simulations as for the packet dropping router except that the G-RED routers to mark packets when they experience congestion and enable TCP to respond to the marked packets. We also only simulate TCP SACK as it is recommended as the proper TCP version to couple with packet marking routers.

Figure 4 depicts the results of these experiments. Each graph is a scatter-plot of the different metrics, throughput of each TCP flow, average queue occupancy and loss probability within the G-RED routers. We plot the measured metrics along the x-axis and the estimated metrics along the y-axis. The results from square root and PFTK model are very close for predicting packet marking network and both of them provide

good estimates of all metrics.

We also find from simulations that the packet marking router is characterized by a higher average queue length than the packet dropping router. We'll address this in section 5

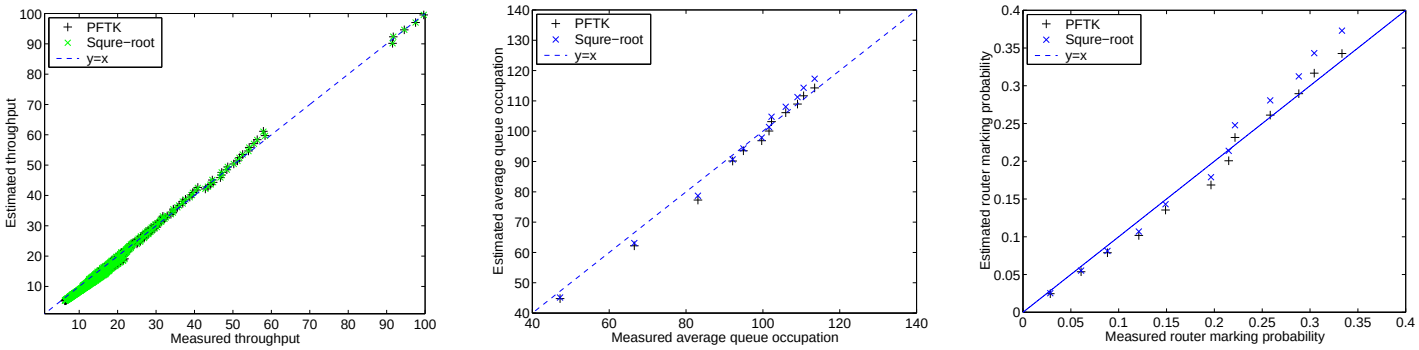

Figure 4: Estimate vs. Simulation: summarized results for single packet marking congested router.

4.2 A single congested router: Finite TCP flows

In order to evaluate the model for finite TCP flows, we simulated the network shown in Figure 1 when supporting several classes of TCP flows. TCP flows within each class arrive according to a Poisson process. We first simulate a scenario where the number of packets to be transfered by a flow is exponentially distributed. We then simulate the scenario where the number of packets to be transfered by TCP flows within each class is described by a hyperexponential distribution.

4.2.1 Exponentially distributed finite TCP sessions size

We simulate 5 TCP classes sharing a single congested router as in Figure 1. TCP flows within each class arrive according to a Poisson process and the number of packets to transfer is exponentially distributed. In the simulation, $\lambda_i = 10$ flows/sec, for $i = 1, \ldots, 5$. The average number of packets to transfer is 19. (i.e., $1/\mu_i = 19$, $i = 1, \ldots, 5$). The two way propagation delay for a class i flow is $(20 + 10i)ms$. Router 0 is configured with $p_0^{max} = 0.1$, $t_0^{min} = 15$ packets, $t_0^{max} = 3t_0^{min}$ packets, $B_0 = 2t_0^{max}$ packets and $C_0 = 4$Mbps. We first configure the congested router to drop packets according to the discard function and simulate TCP Reno and TCP SACK. We then set the congested router to mark packets and simulate TCP SACK. Finally we evaluate the models we proposed earlier for both a packet dropping and a packet marking congested router. The average latency for each TCP class from both models and simulations are plotted in Figure 5 and the average queue length and router loss probability are recorded in Table 1. In Figure 5, we plot the average latency for each TCP class as a function of their two way propagation delay. Table 1 records the loss/mark probability and average queue occupancy from the simulations and from models. We observe that the simulated finite TCP Reno and TCP SACK flows in the packet dropping router shows

Metric	Packet dropping router			Packet marking router	
	Model	SACK	Reno	Model	SACK
Loss/Mark	0.029	0.034	0.034	0.043	0.04
Queue	23.6	24.7	25	28	26

Table 1: Estimates vs. Simulations for finite TCP: Loss(marking) probability and average queue occupancy

slightly different behavior and both of them agree with the model reasonably well. The accuracy of the model for a packet marking router is slightly better than that of the model for a packet dropping router.

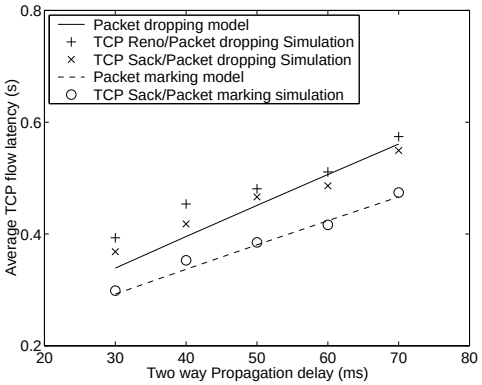

Figure 5: Average TCP latency: exponentially distributed TCP session size

4.2.2 Hyperexponentially distributed finite TCP sessions size

We now evaluate the quality of our model in predicting performance when TCP session sizes come from a hyperexponential distribution. The scenario is the same as that used in the preceding section except for the change in the TCP session size. We consider flows operating with TCP SACK over a packet dropping router. Each finite TCP flow draws its session size from an exponential distribution having either mean 24 packets or mean 14 packets with equal probabilities. We plot the average latency for each TCP class as predicted by our model and as measured from the simulation in Figure 6 and observe good agreement. In addition, the average queue length was estimated to be 23.6 packets from the model and 25 packets from the simulation. The loss probability is estimated to be 0.029 from the model and 0.032 from the simulation.

4.3 A network of multiple congested routers

In order to understand how well the models predict behavior in a multiple bottleneck network, we set up a network composed of a core tandem network and some exterior nodes (Figure 7). We denote the set of routers within the core network as $S = \{1, \ldots, M\}$. The network is configured in such a way that all routers in S are congested. The G-RED control parameters are chosen as follows, t_i^{min} is uniformly

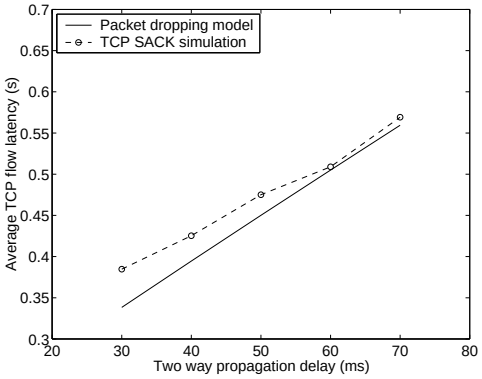

Figure 6: Average TCP latency: hyperexponentially distributed TCP session size

distributed in $[30, 90]$ packets, and $t_i^{max} = 3t_i^{min}$, $B_i = 2t_i^{max}$, C_i is uniformly distributed in $[2, 6]$Mbps, $i = 1, \ldots, M$. The propagation delays of all the links along the core network are uniformly distributed in $[5, 10]ms$. By varying M from 5 to 10 and randomizing the G-RED control parameters, we created 8 core tandem networks. For each of these we generate two simulation instances by adding different TCP flows to each core tandem network. All the TCP flows start from an exterior node and end at a node in the core tandem network. For each router i, we attach E_i exterior nodes $u_{ij}, j = 1, \ldots, E_i$ to it. E_i is chosen uniformly from the range $[3, 6]$. The link propagation delays are randomly chosen from the range $[5, 10]ms$. For each node u_{ij}, a TCP flow f_{ij} starts from u_{ij} and traverses L_{ij} congested routers via router i. L_{ij} is chosen uniformly from $[2, 5]$. If f_{ij} reaches the end of tandem network before it traverse L_{ij} congested links, it stops at that end. In order to make router $i, i = 1, \ldots, M$ congested, for each i, we set up 1 to 4 TCP flows that traverse only router i in S. For a TCP flow j, A_j is in the range of $[20, 120]ms$. For each of the 8 core network created earlier, we first generate two groups of TCP flows using different random seeds. We thus have 16 scenarios.

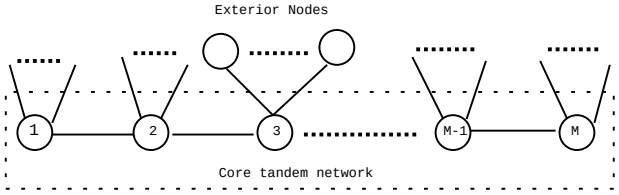

Figure 7: A tandem network with exterior nodes

We configure the congested RED routers to drop packets. In order to evaluate the models for different versions of TCP, we ran each of 16 scenarios twice, one with TCP Reno, the other with TCP SACK. The model provides more accurate predictions of TCP Reno performance than TCP SACK performance and we only present the results for TCP SACK. Figure 8 depicts the results of these experiments. Each graph is a scatter-plot of the different metrics, throughput and end-to-end loss rate of each TCP flow, average queue

occupancy and loss probability within the G-RED routers. We plot the measured metrics along the x-axis and the estimated metrics along the y-axis. As we can observe from Figure 8, the PFTK model provides accurate estimates of all metrics, whereas the square root model can only predict throughput well. The square root model overestimates both router loss and TCP flows end to end loss more and more as the loss probability increases because it doesn't account for timeouts and timeouts become more significant as the loss increase. To summarize, we compute the mean percentage errors in the prediction made by the PFTK model for router loss rate, flow end-to-end loss, flow throughput, and router average queue length. They were 1.9%, 2%, 4.1% and 3.5% and we never saw errors greater than 8%, 8%, 12% and 10% respectively.

We also observed from our experiments that TCP SACK produces higher average queue lengths, router loss rates and end to end loss than TCP Reno. We will address this in the next section.

All simulations performed so far are on feed forward networks. We also simulated cyclic networks obtained by connecting two ends of the core tandem networks. The results from simulations are presented in [1]. The model accuracy for cyclic networks is close to that for tandem networks.

We performed a similar validation for packet marking networks. The predictions from the square root model and the PFTK model are quite close and both models provide accurate estimates of all the metrics. The simulations results can be found in [1].

4.4 A discussion of the model and the simulation results

The simulations have demonstrated that our models provide fairly accurate predictions for both TCP and router metrics. But we also observe the existence of minor divergence between our models and simulation. We conjecture that the prediction error of the TCP throughput formula could contribute to this divergence. Our method is based on combining analyses of individual TCP flows with a network model. The PFTK and square root TCP throughput expressions are both derived for a variety of simplified assumptions. We already know that square root ignores the timeout events which are significant even when the loss rate is less than 5%. The PFTK formula provides better predictions than the square root formula due to the fact that it accounts for timeouts. But it is also pointed out in [24] that there is prediction error in PFTK formula in some cases. Since our models include two components, it is straightforward to replace the PFTK and square root expressions with more accurate expressions as they are produced. In addition, expressions based on measured throughput/loss/RTT profiles can be used as well.

5 Some monotonicity properties and applications

During the process of validating the fixed point methodology, we observed that an infinite duration TCP SACK session incurs higher loss rate and average round trip time than an infinite duration TCP Reno session when traversing a single congested router. and that a marking router exhibits a higher average queue length than a router that drops packets when offered a fixed number of infinite duration TCP SACK (or TCP Reno)

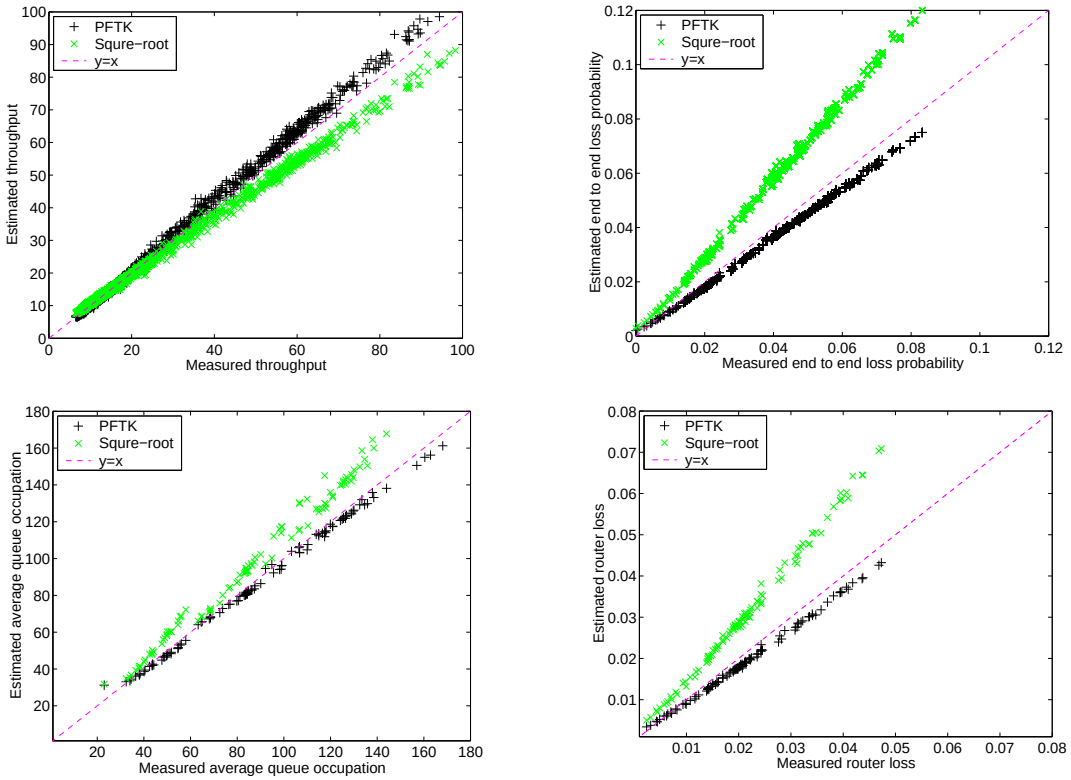

Figure 8: Estimate vs. Simulation: summarized results for tandem packet dropping network

sessions. These observations are easily explained by an important and fundamental monotonicity property exhibited by congested AQM routers as stated next.

Consider two classes of flows, $i = 1, 2$, whose throughput, $T^1(p, R)$ and $T^2(p, R)$ are functions of loss probability, p, and round trip time, R. We have the following result for a single congested router.

Theorem 2 *Let x^i denote the average queue length of a congested AQM router supporting N class i flows, $i = 1, 2$ where $T^1(p, R) \geq T^2(p, R)\ \forall p, R$. If $p()$ is continuous and non-decreasing, then $x^1 \geq x^2$. Furthermore, if $T^1(p, R) > T^2(p, R)\ \forall p, R$, then $x^1 > x^2$.*

Proof. Let x^i denote the average queue length incurred by a set of N infinite duration class i flows passing through the congested router. We can express the throughput of the j-th flow as $T_j^i(x)$, $j = 1, \ldots, N; i = 1, 2$. By the statement of the theorem, $T_j^1(x) \geq T_j^2(x)$, $\forall x$ Let the AQM router have capacity C. It follows then that

$$\sum_{i=1}^{N} T_i^1(x^1) = \sum_{i=1}^{N} T_i^2(x^2) = C. \tag{18}$$

Since both the round trip time and loss rate are increasing functions of average queue length, it follows that the solutions to the above equations must satisfy $x^1 \geq x^2$. If $T_j^1(x) > T_j^2(x) \forall j, x$, then $x^1 > x^2$.

This can be used to explain the difference between TCP Reno and TCP SACK that has been observed in simulation. Let $T^{Reno}(p, R)$ and $T^{SACK}(p, R)$ denote the throughput of a TCP Reno session and a TCP SACK session respectively. The simulation study in [7] suggests that a TCP SACK session can achieve a higher throughput than a TCP Reno session for a given end-to-end loss rate and round trip time, i.e., $T^{Reno}(p, R) < T^{SACK}(p, R)$. Application of Theorem 2 predicts that $x^{SACK} > x^{Reno}$ and, as a consequence, $q_i^{SACK} > q_i^{Reno}$, $i = 1, \ldots, N$. Although TCP SACK suffers a higher loss rate than TCP Reno, it may still exhibit a higher application-level throughput. This is because TCP/SACK is more efficient in its retransmissions than TCP/Reno, i.e., it is less likely to retransmit a packet that has been already successfully received by the receiver.

Theorem 2 can also be used to explain why the same set of TCP flows see a higher average queue length if they share a packet marking congested router instead of a packet dropping congested router.

More generally, Theorem 2 states an invariance that must be heeded by a protocol designer. It is easy to design a protocol that provides higher throughput for a given packet loss rate than TCP, i.e., $T^{new}(p, R) > T^{tcp}(p, R)$. However, according to Theorem 2 this protocol necessarily generates higher average delays and loss rates than TCP when passed through two identically configured AQM drop routers.

Our model also can be used to establish a second monotonicity property which we state without proof.

Theorem 3 *Consider N infinite duration TCP sessions characterized by throughput function $T_i(p, R)$, $i = 1, \ldots, N$, that traverse a single AQM router with link capacity C. Let $x(C)$ denote the average queue length of the router as a function of C. Under the assumption that the probability discard function $p(x)$ is continuous and nondecreasing, the loss probability incurred by the i-th TCP session, $q_i(C)$, is a non-increasing function of the link capacity.*

This property has the following interesting implication. Suppose that we add packet-level forward error correction (FEC) to a TCP protocols such as TCP/SACK. This will have the effect of reducing the packet loss rate seen by the receiver in a TCP session (after correcting for losses). At first glance, this appears to be beneficial as the throughput of an infinite duration TCP session increases as the end-to-end loss rate decreases. Suppose that all of the TCP sessions traversing a congested router introduce FEC. This necessarily has the effect of reducing the effective capacity of the link as some of the capacity will be used to transmit unnecessary parity packets. Thus, according to the preceding theorem, the net effect will be to *increase* the packet loss rate seen by the session after it corrects for losses over the packet loss rate seen by a session when no FEC is used. Consequently, FEC cannot be effectively used by long-lived flows to deal with congestion.

Last, we conjecture that these properties hold in a more general network setting. This conjecture is supported by the simulations reported in the preceding section.

6 Conclusion

We have developed and studied a fixed point method for analyzing the behavior of a large population of TCP flows traversing a network of routers with active queue management. We considered both routers that drop and mark packets serving infinite duration flows as well as finite duration flows. We also developed algorithms for a single congested router supporting finite duration TCP flows. Last, we considered models that account for timeouts as well as models that do not. Our experience indicates that these methods can be extremely accurate in predicting metrics such as average queue length, loss probability, and throughput.

Last, we presented some useful monotonicity properties that explain certain types of behavior observed in simulations, e.g., that TCP SACK produces higher loss probabilities than TCP Reno.

We are currently pursuing the following topics:

- more thorough validation of the methodology,
- establishment of existence and uniqueness properties for the methods,
- relaxation of the need for feedback to be reliable and timely,
- extension of the finite duration flow model to the network setting.

References

[1] T.Bu, D.Towsley. "Fixed Point Approximations for TCP behavior in an AQM Network", *UMass CMP-SCI Technical Report 00-44*

[2] N. Cardwell, S. Savage, and T. Anderson. " Modeling TCP Latency " *Proc. of the 2000 IEEE Infocom*, Mar. 2000

[3] M.E. Crovella, A. Bestavros. "Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes," *IEEE/ACM Transactions on Networking*, 5(6):835–846, December 1997.

[4] C. Casetti, M. Meo. "A New Approach to Model the Stationary Behavior of TCP Connections " *Proc. of the 2000 IEEE Infocom*, Mar. 2000

[5] G. Fayolle, I. Mitrani and R. Iasnogorodski. " Sharing a Processor Among Many Job Classes" *Journal of the ACM* July 1980.

[6] S. Floyd, "Connection with Multiple Congested gateways in packet-Switched Networks Part 1: One-way Traffic" *Computer Communication Review V.21 N.5* October 1991

[7] K. Fall and S. Floyd. "Simulation-based Comparison of Tahoe,Reno, and SACK TCP" *Computer Communication Review V. 26 N. 3* July 1996

[8] A. Feldmann and W. Whitt. "Fitting Mixtures of exponentials to long-tail distributions to analyze network performance models" *Performance evaluations* 31, 1998

[9] V. Firoiu and M. Borden. "A study of Active Queue Management for Congestion Control" *Proc. of the 2000 IEEE Infocom*, Mar. 2000

[10] Victor Firoiu, Ikjun Yeom and Xiaohui Zhang. “A Framework for Practical Performance Evaluation and Traffic Engineering in IP Networks”*Nortel Networks Technique Report,* 2000

[11] S. Floyd. “Notes on RED configuration,” http://www.aciri.org/floyd/red.html

[12] S. Floyd. “Recommendation on using the “gentle_” variant of RED,” http://www.aciri.org/floyd/red/gentle.html Mar. 2000.

[13] S. Floyd, V. Jacobson. “Random early detection gateways for congestion avoidance,” *IEEE/ACM Trans. on Networking,* **1**(4), Aug. 1997.

[14] D.P. Heyman, T.V. Lakshman and A. L. Neidhardt “A new method for analysing feedback-based protocols with applications to engineering Web traffic over the Internet,” *Proc. of the 1997 ACM SIGMETRICS*.

[15] C.V. Hollot, V. Misra, D. Towlsey, W. Gong. “On designing improved controllers for AQM routers supporting TCP flows” To appear in *Proc. of the 2001 IEEE Infocom*

[16] S. MCanne and S. Floyd. ns-LBL network simulator,1997. obtain via http://mash.cs.berkeley.edu/ns/ns.html

[17] J.Mahdavi and S. Floyd. “TCP-Friendly Unicast Rate-Based Flow control,” http://www.psc.edu/networking/papers/tcp_friendly.html

[18] A. Misra, T. Ott, J. Baras. “The window distribution of multiple TCPs with random loss queues,” *Proc. of Globecom’99,* Dec. 1999.

[19] V. Misra, W. Gong, D. Towsley. “Stochastic differential equation modeling and analysis of TCP window size behavior,” Technical Report ECE-TR-CCS-99-10-01, Dept. of Electrical and Computer Engineering, Univ. of Massachusetts, Oct. 1999.

[20] V. Misra, W. Gong, D. Towsley. ”A Fluid-based Analysis of a Network of AQM Routers Supporting TCP Flows with an Application to RED” *Proc. of ACM SIGCOMM’00,* (Stockholm, Sweden, September 2000).

[21] T. Ott, J. Kemperman, M. Mathis. “The stationary behavior of the ideal TCP congestion avoidance,” ftp://ftp.telcordia.com/pub/tjo/TCPwindow.ps

[22] J. Padhye, V. Firoiu, D. Towsley, J. Kurose “Modeling TCP Throughput: A Simple Model and its Empirical Validation” *Proc. ACM SIGCOMM’98,* (Vancouver, CA, September 1998). Also appeared in *IEEE/ACM Transactions on Networking,* **8**(2), April 2000.

[23] J. Padhye, V. Firoiu and D. Towsley “A Stochastic Model of TCP Reno Congestion Avoidance and Control.” UMASS CMPSCI Technical Report 99-02, Feb 1999

[24] L. Qiu, Y. Zhang, and S. Keshav “On Individual and Aggregate TCP Performance” *Proceedings of 7th International Conference on Network Protocols (ICNP’99),* Toronto, Canada

[25] K.K. Ramakrishnan, S. Floyd. “A proposal to add Explicit Congestion Notification (ECN) to IP,” RFC 2481, January 1999.

[26] K.W. Ross. *Multiservice loss networks for broadband telecommunication networks*, Springer Verlag, 1995.