A Method for estimating the proportion of non-responsive traffic at a router

Zhili Zhao, Swaroop Darbha, A. L. Narasimha Reddy, *Senior Member, IEEE*

***Abstract*— In this paper, a scheme for estimating the proportion of the incoming traffic, that is not responsive to congestion at a router, is presented. The idea of the proposed scheme is that if the observed queue length and packet drop probability do not match the predictions from a model of responsive (TCP) traffic, then the error must come from non-responsive traffic; it can then be used for estimating the proportion of non-responsive traffic. The proposed scheme is based on the queue length history, packet drop history, expected TCP and queue dynamics. The effectiveness of the proposed scheme over a wide range of traffic scenarios is corroborated using ns-2 based simulations. Potential applications of the proposed algorithms in traffic engineering and control are discussed.**

***Index Terms*— Traffic modeling, control theory, estimation, non-responsive traffic.**

I. Introduction

MEASUREMENTS at various internet sites [1] suggest that 85-90% of current network traffic is based on TCP. However, with the emergence of new applications, the usage of protocols other than TCP is expected to increase. For instance, several multimedia applications rely on UDP to transport packets. Recently, there has been an increased interest in developing protocols that respond to congestion differently from TCP and provide smoother bandwidth to end applications [2], [3]. Furthermore, increases in bandwidth and computation power are expected to fuel the growth of multimedia applications that do not rely on TCP. These trends point to an increased diversity of network protocols and changes in the distribution of bandwidth among flows employing these diverse protocols in the future.

One can expect such new applications not to respond to network congestion the same way as TCP-based applications. The impact of non-responsiveness of applications on the bandwidth available for responsive applications and the overall goodput of network is of considerable importance and has attracted the attention of researchers [4]. This is especially so, because it is possible to stage denial-of-service(DoS) attacks on end hosts and the network by pumping large amounts of non-responsive flows into the network. Some of the recent DoS attacks have used such “UDP floods”. If the network could monitor and regulate the utilization of non-responsive traffic to a fraction of the link capacity, the impact of such attacks could be mitigated.

If all the non-responsive traffic used UDP for transport, a simple counter of UDP bytes at a link will provide an estimate of the proportion of non-responsive traffic. However, non-responsive traffic may use other protocols or use variants of TCP; some UDP applications, such as Real Audio/Video, actually respond to the congestion of network by adapting the sending rate. A simple counter of UDP bytes at a link/router, therefore, will not suffice for estimating the “apparent” proportion of non-responsive traffic. A protocol byte counter has other drawbacks: Consider a scenario where the arriving traffic consists of a large number of “small” bandwidth TCP flows, usually referred to as “web-mice”, which send out a small amount of bytes intermittently [5]. In this scenario, a counter for UDP bytes will yield a count of zero while packet drops do not reduce congestion significantly at the router. Although each flow employs TCP, the aggregate behavior of “web-mice” at the router cannot be differentiated from that of non-responsive traffic. A simple protocol counter is also easily defeated by fake protocol ids in packet headers by malicious users for avoiding this detection mechanism.

It is necessary to find mechanisms to estimate the amount of arriving traffic at a router that is not responding to congestion. Such an estimation can lead to a better control of heterogeneous network traffic through appropriate adaptation of traffic control algorithms at a router. For example, an estimate of the proportion of non-responsive traffic (PONRT) at a router can aid the choice of appropriate parameters for Active Queue Management (AQM). A need for such a tuning is indicated recently in [6]. With the knowledge of the PONRT at a router, appropriate traffic management and congestion control algorithms can be employed in different operating domains of network traffic [4], [7], [8].

Mathematical models for the dynamics of TCP flows and associated control schemes have been proposed [9–11]. These studies have led to better analysis of TCP behavior

This work was supported in part by an NSF Career Award and an NSF Grant ANI-0087372 and by grants from the State of Texas Higher Education Coordination Board and Texas Informatics and Telecommunications Task Force (TITF) and by a grant from Intel Corp.

Z. Zhao is with the Department of Electrical Engineering, Texas A&M University, 214 Zachry, College Station, TX 77843

S. Darbha is with the Department of Mechenical Engineering, Texas A&M University, 313 EPB, College Station, TX 77843

N. Reddy is with the Department of Electrical Engineering, Texas A&M University, 214 Zachry, College Station, TX 77843

and proposals for improved traffic controllers [12], [13].

In this paper, the model in [9] is extended to account for the effects of heterogeneity in traffic, by including the effects of non-responsive traffic into the model. Based on the extended model, a method is developed for estimating the PONRT at a router. The presented method employs a normalized gradient method for estimation and utilizes queue length history and packet drop history at the router, which are easily measurable at a router. The effectiveness of the proposed method is corroborated through ns-2 [14] based simulations.

The rest of the paper is organized as follows: In Section II, the development of extended traffic model is presented. In Section III, a basic estimation algorithm is developed based on the extended model in Section II. In Section IV, implementation details of and modifications to the basic algorithm are presented along with their corresponding ns-2 based simulation results and analyses. In Section V, the basic algorithm is extended to multi-hop topology. In section VI, limitations and potential applications of the presented method are discussed.

II. A MODEL OF THE AGGREGATE DYNAMICS OF HETEROGENEOUS TRAFFIC AT A ROUTER

The focus of this section is on the development of a dynamical model of heterogeneous traffic on a congested link in the network that is particularly well suited for estimating PONRT at a router. Traffic is assumed to consist of only two types of flows - TCP flows and Constant Bit Rate (CBR) flows. Without any loss of generality, it is assumed that TCP and CBR flows represent responsive and non-responsive flows respectively in the traffic. Following the fluid based models of TCP flows in [9], the extended model developed in this section is described in terms of three states - window size for responsive flows, sending rate for non-responsive flows and queue length at the router. The following are the underlying assumptions in developing this model:

- The effect of all non-responsive flows can be modeled by a number of equivalent "average flows". Similarly, all responsive flows can be modeled by the same window adaptation behavior and observe the same round trip time (RTT). Such assumptions are crude first approximations of the real-world traffic; nevertheless, they capture the average or macroscopic dynamics of the heterogeneous traffic at the router, especially when the number of flows is large. This assumption seems reasonable for applications where one is interested in the evolution of the queue length at time scales slower than the longest possible RTT of a responsive flow. The estimate of PONRT is a representative of the true time-averaged PONRT at this time scale. Such approximations are also used in modeling ground traffic flow, see [15].
- Queue length and window size of responsive flows change slowly within a single RTT of a responsive flow. Packet drop rate at the router changes slowly.

The impact of non-responsive traffic is modeled through its effect on the queue length and hence, on RTTs and packet drop probabilities, which, in turn, impact the window size of TCP flows; this interaction is captured by the presented model.

Dynamic models of homogeneous traffic with TCP flows have been proposed and studied in [9–11]. This model extends [9] by:

- introducing an evolution equation for the representative (or aggregate) sending rate of non-responsive flows as seen by the router, and by
- accounting for non-responsive flows in the evolution of queue in the buffer.

This model is described by the following set of differential equations:

$$\dot{X}_u = 0, \tag{1}$$

$$\dot{W}_s(t) = \frac{1}{R\big(q(t)\big)} - \frac{W_s(t)W_s\Big(t - R\big(q(t)\big)\Big)}{2R\big(q(t)\big)} \cdot p\Big(t - R\big(q(t)\big)\Big), \tag{2}$$

$$\dot{q}(t) = \frac{N_s W_s(t)}{R\big(q(t)\big)} + N_u X_u - C. \tag{3}$$

In eqn. 1, X_u is the sending rate of a representative non-responsive flow. In eqn. 2, $W_s(t)$ is the window size of a TCP flow; N_s is the number of incoming TCP flows; N_u is the number of incoming non-responsive flows; $R(t)$ is the Round Trip Delay, which is given by $\frac{q(t)}{C} + T_p$, where T_p is a fixed propagation delay; $p(t)$ is the packet drop probability. In eqn. 3, $q(t)$ is the queue length and C is the outgoing link capacity. Note that the impact of reverse path congestion is minimized by the implementation of TCP cumulative ACKs, and hence not considered in this model. Eqn. 1 describes the aggregate behavior of a CBR flow as a representative non-responsive flow. Eqn. 2 describes the behavior of a representative TCP flow in congestion control phase and is the same as in [9]. It indicates that the evolution of window size is related to the round trip delay, drop probability and to its history. Eqn. 3 represents the dynamics of queue length.

We then define $W_u(t) := X_u R\big(q(t)\big)$. As shown in Figure 1, the deviation of measured RTTs from the equilibrium state of the system is sufficiently small. For the simplicity of the model, on an average, $W_u(t)$ can be approximated not to change with time, i.e.,

$$\dot{W}_u(t) = 0.$$

By replacing $X_u R\big(q(t)\big)$ with W_u in eqn. 3, one gets:

$$\dot{q}(t) = \frac{N_s W_s(t) + N_u W_u}{R\big(q(t)\big)} - C. \tag{4}$$

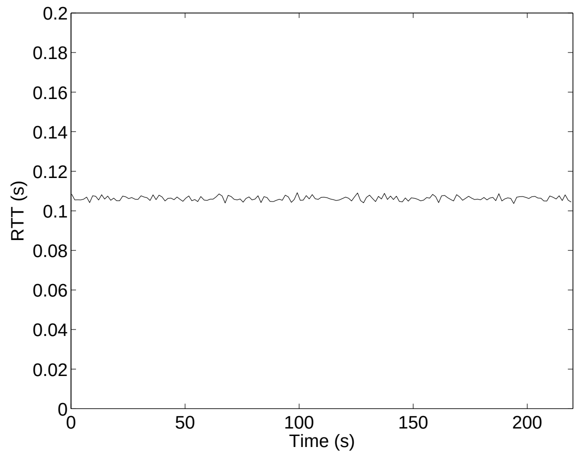


Fig. 1. Measured RTT under the Equilibrium State

Based on the model, an algorithm to determine the PONRT at a router is proposed in the next section. The focus is on estimating of PONRT on the aggregate as opposed to identifying individual flows that are non-responsive [4], [16], [17].

As with any model, the reasonableness of the proposed model depends on how well it predicts the proportion of non-responsive traffic. Simulation studies reported in the subsequent sections corroborate the suitability of the proposed model.

A model accounting for constant sending rate flows(CBR fows), instead of the approximation of using constant W_u flows, is in preparation [18]. The main idea of the estimation algorithm based on that model remains the same as the one presented in section III. The only difference is that there are three unknown quantities $(N_s, z(t), X_u N_u)$ to be estimated and they are related by one equation. In contrast, here are two unknowns$(N_s, z(t))$ in the model presented in this paper.

III. Basic Estimation Algorithm

In this section, an algorithm for estimating the PONRT will be developed based on the extended model presented in Section II. At first, a basic estimation algorithm will be presented in this section. This algorithm will then be modified to account for scenarios where no packets are dropped or where large number of incoming packets are dropped in Section IV. The basic algorithm, which considers a single bottleneck link, will be extended to the multi-hop topology in Section V. The simulations corresponding to the algorithms developed in this section are presented in Section IV.

For the purpose of developing an estimation algorithm, the dynamics of the traffic mix will be expressed in terms of the total responsive load $z(t)$ and the total non-responsive load D. The terms $z(t)$ and D are given by the following relationships:

$$z(t) := N_s W_s(t), \tag{5}$$

$$D := N_u W_u \tag{6}$$

The terms $z(t)$ and D are scaled loads and are respectively representative of the number of responsive and non-responsive packets seen by the router in a RTT. The quantity of D will be estimated in the algorithm.

In terms of $z(t)$ and D, the dynamics is given by:

$$\dot{z}(t) = \frac{N_s}{R(t)} - \frac{z(t)z\big(t-R(t)\big)}{2N_s R(t)} p\big(t-R(t)\big), \tag{7}$$

$$\dot{q}(t) = \frac{z(t)+D}{R(t)} - C, \tag{8}$$

$$R(t) = \frac{q(t)}{C} + T_p, \tag{9}$$

where $\dot{z}(t) = N_s \dot{W}_s(t)$. The underlying assumption in the estimation algorithm is that the number of TCP flows and non-responsive flows does not change or changes very slowly.

The packet drop probability, p and the queue length, q, are sampled at each *Sampling Interval* to estimate the desired fraction of non-responsive load, ψ:

$$\psi = 1 - \frac{z(t)}{D+z(t)}. \tag{10}$$

The term $D + z(t)$ represents the total number of incoming packets and can be counted at the ingress link of a router. If $z(t)$ can be estimated, ψ can be calculated with eqn. 10.

By taking the second derivative of q using eqn. 8, one gets

$$\ddot{q}(t) = \frac{\dot{z}(t)}{R(t)} - \frac{z(t)+D}{R^2(t)}\dot{R}(t). \tag{11}$$

Combining eqn. 11 with eqn. 7 yields

$$R(t)\ddot{q}(t) + (\frac{\dot{q}(t)}{C} + 1)\dot{q}(t) = \frac{N_s}{R(t)} - \frac{z(t)z\big(t-R(t)\big)}{2N_s R(t)} p\big(t-R(t)\big). \tag{12}$$

It is possible to show that the only physically realistic equilibrium of the dynamics for a fixed packet drop rate is stable, using standard linearized analysis of nonlinear differential equations [19–21]. The stability of equilibrium indicates that a “small signal” approximation of the above differential equation describes the evolution of solutions of the nonlinear differential equation reasonably accurately when the deviation from the equilbrium is sufficiently small.

The estimation algorithm is based on the small-signal behavior of the dynamics, i.e., $q(t) \approx q_0; R(t) \approx R_0 = \frac{q_0}{C} + T_p; z(t) \approx z_0 \approx z(t - R_0)$, where variables with subscript 0 correspond to their respective equilibrium values. The problem at hand is as follows: Given that the equilibrium is known partially in terms of q_0, can we describe the equilbrium completely (i.e., determine z_0, D) from the measurements of the packet drop rate, p and the queue length, q. The determination of equilbrium provides an estimate of PONRT using eqn. 10.

At this point, we may question why it cannot be assumed that the equilibrium value of packet drop rate, p_0, is known. The term p_0 is assumed small and in practice, the signal-to-noise ratio (SNR) of p_0 is small. We can develop a scheme for parameter identification based on the Jacobi linearization of eqn. 12; however, this would involve the computation of the deviation of the packet drop rate, p, from its equilibrium value, p_0. This was the scheme we first tried, but with little success owing to the small SNR of p_0. We can then think of treating p_0 as an unknown constant; however, this leads to an overparametrization with an equilibrium constraint $2N_s^2 = z_0^2 p_0$. In order to circumvent such difficulties, we obtain the following parametrization of the dynamics based on the practical small-signal approximations stated above:

$$R_0\ddot{q}(t) + (\frac{\dot{q}(t)}{C} + 1)\dot{q}(t) = \begin{bmatrix} \frac{1}{R_0} & -\frac{p\left(t-R_0\right)}{2R_0} \end{bmatrix} \begin{bmatrix} N_s \\ \frac{z_0^2}{N_s} \end{bmatrix} \qquad (13)$$

The resulting model is still nonlinear; however, it has the advantage that the unknown parameters are linearly parametrized in terms of the output, which is the left hand side of the equation.

We can define the following from eqn. 13:

$$\chi(t) := R(t)\ddot{q}(t) + (\frac{\dot{q}(t)}{C} + 1)\dot{q}(t) = \boldsymbol{W^T}(t) \begin{bmatrix} \boldsymbol{\beta_0^*} \\ \boldsymbol{\beta_1^*} \end{bmatrix} = \boldsymbol{W^T}(t)\boldsymbol{\beta^*}, \qquad (14)$$

where $\boldsymbol{W^T} = \begin{bmatrix} \frac{1}{R_0} & -\frac{p\left(t-R_0\right)}{2R_0} \end{bmatrix}$, and $\boldsymbol{\beta^*} = \begin{bmatrix} \boldsymbol{\beta_0^*} \\ \boldsymbol{\beta_1^*} \end{bmatrix} = \begin{bmatrix} N_s \\ \frac{z_0^2}{N_s} \end{bmatrix}$.

The term $\chi(t)$ may be thought of as an output which is linearly parametrized in terms of the unknown vector of parameters, $\boldsymbol{\beta^*}$. The term $\boldsymbol{\beta(t)}$ represents the estimate of the unknown vector of parameters at time t and $\chi_e(t)$ represents the predicted output with the current estimate of parameters. Then,

$$\chi_e(t) = \begin{bmatrix} \frac{1}{R_0} & -\frac{p\left(t-R_0\right)}{2R_0} \end{bmatrix} \begin{bmatrix} N_s \\ \frac{z_0^2}{N_s} \end{bmatrix} = \boldsymbol{W^T}(t) \begin{bmatrix} \boldsymbol{\beta_0} \\ \boldsymbol{\beta_1} \end{bmatrix} = \boldsymbol{W^T}(t)\boldsymbol{\beta} \qquad (15)$$

To develop a parameter adaptation algorithm, we require the knowledge/measurement of the output, $\chi(t)$, and the regressor, $\boldsymbol{W(t)}$. Since the parameter identification is expected to evolve at a time scale slower than an RTT, sampling of $q(t)$ must be made at least once in a RTT. If T is the time between two successive samplings of $q(t)$ (also called a *sampling interval*), it is required to be *smaller than* one RTT. While the regressor can be computed directly from the measurements of the packet drop rate and the queue length, the determination of $\chi(t)$ requires the measurements of $\dot{q}$ and $\ddot{q}$. The signal $q(t)$ is numerically differentiated in order to obtain $\dot{q}$ and $\ddot{q}$; specifically, they are computed as: $\dot{q}(t) = (q(t) - q(t-1))/T$, and $\ddot{q}(t) = (\dot{q}(t) - \dot{q}(t-1))/T$.

The error $e(t)$ between $\chi(t)$ and $\chi_e(t)$ is used to update $\boldsymbol{\beta}$ recursively. The error $\boldsymbol{e(t)}$ is given by:

$$\boldsymbol{e(t) = \chi(t) - \chi_e(t)}$$

While there are several recursive algorithms available for updating the parameters, Kaczmarz's projection algorithm [22–24] is employed due to its low computational complexity and quick convergence properties. Applying *normalized* Kaczmarz's projection algorithm to update $\boldsymbol{\beta(t)}$ yields:

$$\boldsymbol{\beta}(t+1) = \boldsymbol{\beta}(t) + \boldsymbol{e(t)}\frac{\boldsymbol{\gamma_2 W}(t)}{\boldsymbol{\gamma_1} + \boldsymbol{W^T}(t)\boldsymbol{W}(t)}, \qquad (16)$$
$$\text{where } \boldsymbol{\gamma_1 \geq 0} \text{ and } \boldsymbol{0 < \gamma_2 < 2}$$

$\boldsymbol{\gamma_1}$ and $\boldsymbol{\gamma_2}$ are user-defined tuning gains[1].

Once $\boldsymbol{\beta_0}$ and $\boldsymbol{\beta_1}$ are determined from eqn. 16, we can estimate the number of responsive flows, $\boldsymbol{N_s}$ and the scaled load of responsive flows, $\boldsymbol{z_0}$, as:

$$\boldsymbol{N_s = \beta_0}$$
$$\boldsymbol{z_0 = \sqrt{\beta_0 \beta_1}}$$

From the last equation for $\boldsymbol{z_0}$ and from eqn. 10, we can estimate PONRT. Note that, in eqn. 10, $\boldsymbol{z(t)+D}$ is the total number of packets coming at time $\boldsymbol{t}$ and can be counted at the ingress interface of the router in practice. It must be emphasized that the proposed algorithm measures this fraction relative to the arrival rate at the switch, and not relative to the capacity $\boldsymbol{C}$ of the outgoing link.

Note that the algorithm only depends on drop probability and queue length. Since the required number of samples of history information of $\boldsymbol{q(t)}$ and $\boldsymbol{p(t)}$ is relatively small, the use of memory resource is limited and the computation complexity is $\boldsymbol{O(1)}$.

IV. Implementation and Modifications of Basic Algorithm

The basic estimation algorithm developed in the earlier section was implemented in the RED module of the

[1] $\boldsymbol{\gamma_1 = 5, \gamma_2 = 0.45}$ for the simulations in Section IV and $\boldsymbol{\gamma_2 = 0.25}$ in Section V

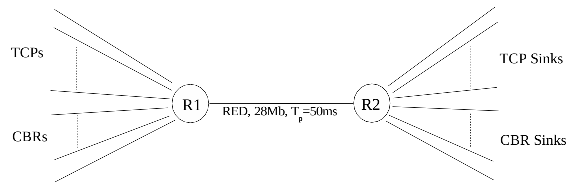


Fig. 2. Simulation Topology

Network Simulator (ns-2). The main issues in the implementation are the choice of the *Sampling Interval T* and the numerical differentiation/filtering of the signals used in estimation.

It is a common practice in control applications [25], [26] to numerically difference discrete inputs to obtain a differentiated value. The implementation of numerical differentiation of queue length($\dot{q}(t)$ and $\ddot{q}(t)$) was provided in Section III. In some cases, one further filters the difference to attenuate the high frequency content in the numerically differentiated value; this is referred to as a "dirty" derivative of the signal. The rationale behind filtering is to attenuate the high frequency noise content in the "dirty" derivatives as well as in the signal. The corner frequencies of the filters may be chosen so as to filter frequency components faster than one RTT. In the implementation, the drop probability and queue length are filtered according to the relation: *wp* = $\alpha \times wp_{old}$ + *(1 -* α*)* $\times$ *ap of current sampling interval* and *wq* = $\alpha \times wq_{old}$ + *(1 -* α*)* $\times$ *aq of current sampling interval*, where α is the forgetting factor. The average value is calculated by averaging all inputs over one *sampling interval*. A value of $\alpha = 0.4$ is chosen for filtering the queue length and a value of $\alpha = 0.6$ is chosen for filtering the drop probability. This choice of parameters results in forgetting the history information of p and q within a few *sampling intervals*(approximately one RTT).

A bottleneck link topology shown in Figure 2 was employed for simulations. A RED drop function with *(*min_{th}*,* max_{th}*,* p_{max}*) = (15, 45, 0.1)* is chosen for managing the queue. The bottleneck router has 60 buffers and a link capacity of 28Mb and a propagation delay of 50ms. CBR flows with a transmission rate of 1Mbps were employed to simulate non-responsive flows, and FTP flows were used to simulate long term responsive TCP flows. In the simulation, each packet has a size of 1000 bytes and the sampling interval, as discussed in Section III, is 33ms (corresponding to 1/3rd of RTT of TCP traffic).

Each simulation was run for 320s. The estimation algorithm was started after the first 100s of each simulation when the system stabilized. The estimation algorithm updated the unknown parameters every sampling interval. Several of these estimates were aggregated to produce a smoothed estimate over a larger time interval called an *estimation interval*. In this paper, the estimation interval was chosen to equal 20 sampling intervals and is approximately 700 ms with the choice of parameters made in this paper.

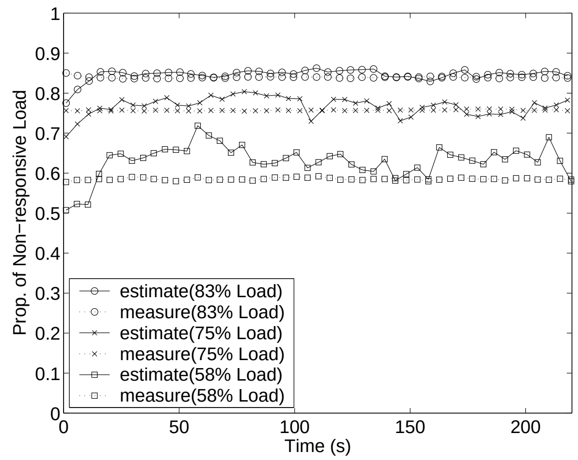


Fig. 3. High Non-responsive Load

In each estimation interval, the true value of PONRT is computed by counting the packets of non-responsive flows and dividing it by the total number of arrived packets. The estimate produced by the algorithm is then compared to their respective true values.

Mean Square Error(MSE) and Relative Error(RE) are chosen as a metric for the accuracy of estimation. Mean square error is computed as $\sum_{i=1}^{n}(estimate_i - actual_i)^2/n$ and relative error is computed as $\sum_{i=1}^{n}(estimate_i - actual_i)/\sum_{i=1}^{n} actual_i$, where n is the total number of estimations per simulation.

A. High Non-responsive Load

Three simulations were set up to examine the effectiveness of the proposed algorithm under different non-responsive load conditions. There were 35 responsive flows in each simulation; three simulations correspond to 16, 22 and 25 CBR flows respectively. Each CBR flow sends packets at the rate of 1Mbps. The PONRT corresponding to 25 CBR flows is larger than PONRT corresponding to 16 or 22 flows. Shown in Figure 3 are the estimated and measured values of the PONRT obtained with the basic algorithm.

The PONRT is computed relative to the arrival rate at the switch and hence, it varies over time as the responsive traffic arrival rate changes over time.

The estimates of PONRT in Figure 3 fluctuate around their true values in a very small band. This indicates that the accuracy of estimating PONRT with the basic algorithm is high when the non-responsive load is high.

B. Persistency of Excitation and Modification of Basic Algorithm

Persistency of Excitation (PE) is an important issue in the convergence of parameters to their true values. For the parameter estimates to converge to their true values through the known regressor $\boldsymbol{W(t)}$, it is required that $\boldsymbol{W(t)}$ be persistently exciting. The rate of convergence, in general, depends on the strength of the reference signal, as can be inferred from the proofs of convergence [27].

Since the "small signal" behavior of the nonlinear dynamics of heterogeneous traffic is approximated with a static linear parametrization in terms of unknown parameters, the allowable strength of the reference signals will necessarily be limited by the region of validity of this approximation. Such an analysis is out of the scope of this paper; as such, this contribution is focused on and can only be viewed in the engineering design context.

Since there are only two unknown parameters with the parametrization chosen in this paper, it follows that there must necessarily be a non-zero frequency component in the regressor for the parameter estimates to converge to their true values. When the load of non-responsive arriving traffic is high, continued packet drops and fluctuations in queue length provide the necessary persistence of excitation. It is for this reason, the basic algorithm performs well under these conditions, as can be noticed from Figure 3.

In a real scenario, packet drops and variations in queue length occur persistently except when the buffer is empty. The first modification to the basic algorithm is specifically meant to address this shortcoming of the basic algorithm when there are no packet drops or queue variations.

When the buffer is empty, zero packet drop rate corresponds to the additive increase of the window size of TCP flows. For this reason, by applying $p(t - R_0) = 0$ to eqn. 7 and eqn. 12, we get:

$$\dot{z}(t) = \frac{N_s}{R(t)} = R(t)\ddot{q}(t) + (\frac{\dot{q}(t)}{C} + 1)\dot{q}(t) \quad (17)$$

Note that $\dot{z}(t)$ is reflective of the load change and can be calculated with known parameters/measurements ($R(t)$, $q(t)$ and C) according to eqn. 17. If one knows $z(t_0)$, which is the last estimate of $z(t)$ prior to having no packet drops, then $z(t_x)$ $(x = 1, 2, \cdots, n-1)$ can be computed recursively in the following way when $p(t - R_0) = 0$:

$$\begin{aligned} z(t_1) &= z(t_0) + \dot{z}(t_1)(t_1 - t_0), \\ z(t_2) &= z(t_1) + \dot{z}(t_2)(t_2 - t_1), \\ &\cdots \\ z(t_{n-1}) &= z(t_{n-2}) + \dot{z}(t_{n-1})(t_{n-1} - t_{n-2}). \end{aligned}$$

This algorithm is employed when no packets are dropped. As soon as packets are dropped, the basic algorithm developed in the earlier subsection is used. In Figure 4, the MSEs of basic and modified algorithms under different non-responsive loads are compared. As can be seen from this figure, the modified algorithm is more accurate than the basic algorithm in terms of MSE when non-responsive load is below 60%. Therefore, this modified estimation algorithm will be used in the rest of the simulations.

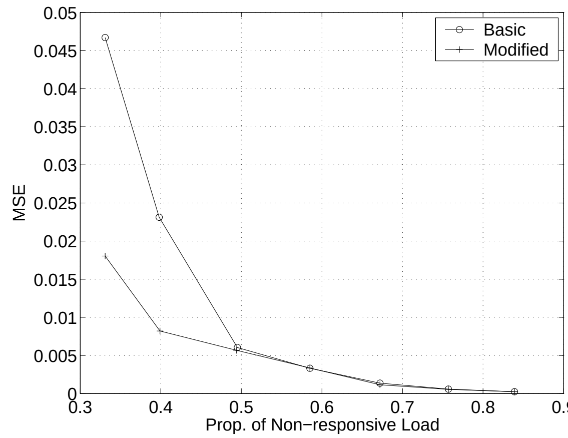


Fig. 4. Comparison of Basic and Modified Algorithm Accounting for p=0 Periods

C. Effectiveness of the Estimation Algorithm with time varying non-responsive loads

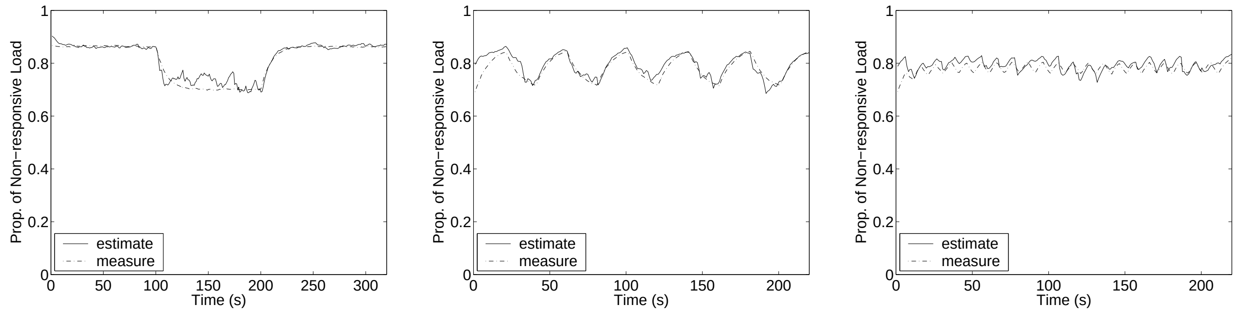
To examine the effectiveness of the proposed algorithm under time varying non-responsive loads, 35 responsive flows, and 26 non-responsive flows were considered; of the non-responsive flows, 6 were ON/OFF type flows. The ON/OFF flows were ON for "x" number of seconds and OFF for the next "x" number of seconds. As a result, the non-responsive load has the shape of a square wave with a period of "2x" seconds. Different sets of simulations were performed corresponding to three different values of x: x = 100 seconds, 20 seconds, and 5 seconds. The results from these simulations are shown in Figure 5.

From Figure 5, one can observe that the algorithm can estimate the PONRT fairly well even when the non-responsive load is varying with time. It is possible to estimate faster varying non-responsive loads by choosing an estimation interval smaller than 700ms, which is used in the above simulations.

D. Mixed Traffic

To test the effectiveness of the proposed algorithm under a more realistic traffic scenario, mixed traffic consisting of short-term TCP flows, long-term TCP flows and a number of non-responsive flows was simulated. In the simulations, only long-term TCP flows and non-responsive flows populate the traffic initially. At 100s, 300 short-term TCP flows were introduced into the traffic. Each short-term flow sends 20 packets randomly five times in a 50-second time period.

From Figure 6, it is clear that the estimation algorithm treats short-term TCP flows as a part of the non-responsive load. These flows do not persist in the network long enough



(a) One-time Change (x=100s)

(b) Continuous Change (x=20s)

(c) Continuous Change (x=5s)

Fig. 5. Dynamic Response to Change of Traffic

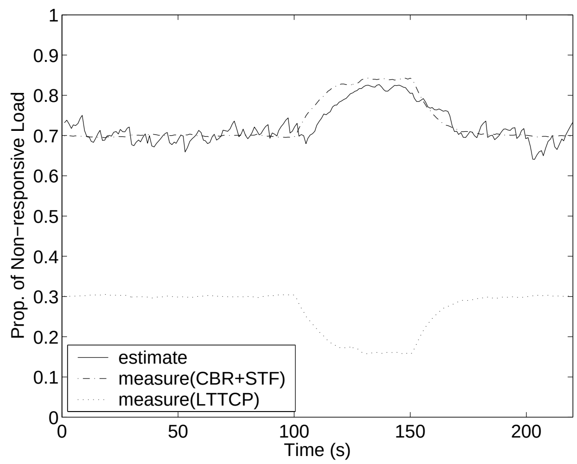


Fig. 6. Traffic Mix – Short-term TCP, Long-term TCP and Non-responsive Load

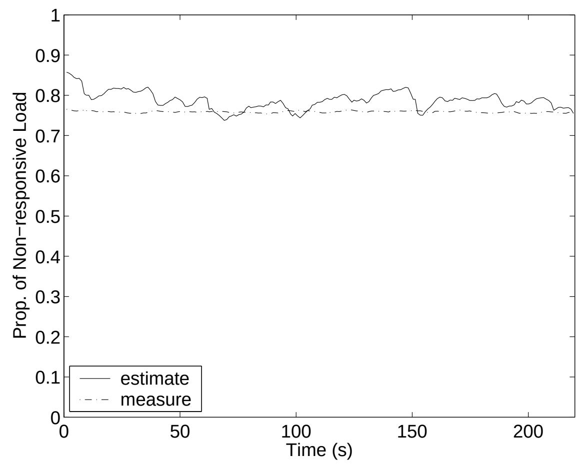


Fig. 7. TCPs with Different RTTs

to experience significant number of packet drops and the congestion response of a short-term TCP flow only results in an insignificant difference in the amount of traffic at the router. Moreover, the response of a single short-term TCP flow may be replaced by the arrival of another flow. As a result, these flows on an aggregate appear to be non-responsive. Similar observations about short-term flows have been made in a number of recent studies [5], [28].

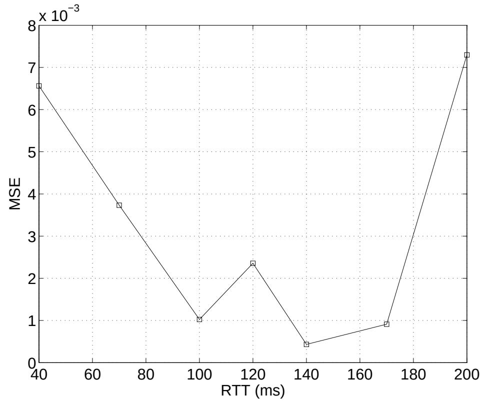
E. Impact of RTTs

Another set of simulations was conducted to study the impact of RTTs. In these simulations, 35 TCP flows with different RTTs ranging from 24.4ms to 175.5ms were considered. The parameter R0 in the algorithm was set to be the average value of the range (90ms). The result of simulations is shown in Figure 7.

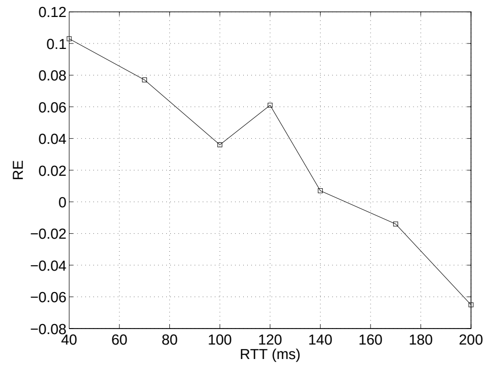
From Figure 7, the estimation algorithm is still effective, although the algorithm over-estimates the non-responsive traffic by a small amount. This small discrepancy can be attributed to the assumption that all TCP flows have the same RTT in the traffic model. Nevertheless, the results here show that as long as we employ a reasonable average RTT in the estimation algorithm, the different RTTs of different flows do not impact the accuracy of estimation significantly. Recent studies based on wavelets provide a convenient way to estimate the range of RTTs of flows passing through a router [29] and further help us to set a reasonable average value.

In order to further study the impact of RTT on the estimation, a set of simulations, where R0 in the algorithm was fixed to be 120ms, were conducted. The RTTs of all TCP flows in one simulation were the same, but different between simulations, varying from 40ms to 200ms. MSEs and REs from the set of simulations were collected and presented in Figure 8.

From Figure 8, we can notice that when RTT of the flows is below R0, the algorithm overestimates the non-responsive traffic (since it underestimates the responsive TCP traffic). When RTT of the flows is higher than R0,



(a) MSE vs. RTT of Flows (R0=120ms)



(b) RE vs. RTT of Flows (R0=120ms)

Fig. 8. Accuracy vs. RTT of Flows

the algorithm underestimates the non-responsive traffic. However, it is observed the relative errors are within 10% even over such a wide range of RTTs.

F. Impact of Variable Bit Rate(VBR) Traffic

The estimation algorithm is based on the model, in which non-responsive flows were represented by CBR flows. Since not all non-responsive flows are CBR flows, the effectiveness of the estimation algorithm in realistic scenarios will depend on its ability to estimate the PONRT in the presence of other non-responsive flows.

To address this issue, a set of simulations was set up with 16 VBR flows and 35 TCP flows. Each VBR flow is a non-responsive flow and changes its sending rate randomly selected in the range of 0.5Mbps and 1.5Mbps. A time interval between 2 different sending rates is also randomly selected within a given interval range. Each VBR flow changed its sending rate at the end of each interval till the simulation finished.

Figure 9 shows the simulation the proportion of VBR traffic with 2 different interval ranges, [1 20]s and [0.1 1]s. The larger the interval is, the less frequently the non-responsive flow changes its sending rate. So a VBR flow with intervals in [0.1 1]s changes its sending rate faster than one with intervals in [1 20]s. X axis shows the simulation period of 220s. It is noticed that the estimate of PONRT is still accurate, although the non-responsive load changes randomly. The MSEs (0.003534 and 0.002445) of this set of simulations compares well with MSE(0.003353) of the simulation with 16 CBR flows.

G. Responsive Protocols other than TCP

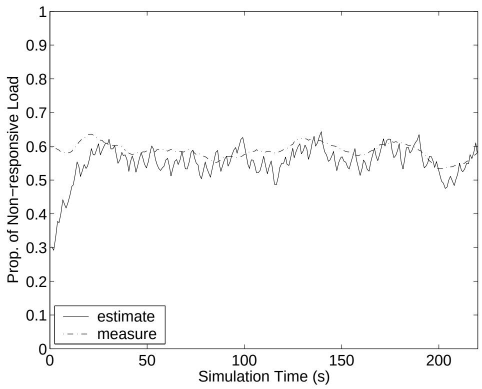
New applications, such as multimedia applications, require smooth bandwidth adaption in order to deliver quality service over internet. As a result, variants of TCP congestion control have been proposed and studied. They tend to provide much smoother sending rate than TCP does [2], [3] and still be fair to TCP over a longer time scale. In [3], the authors propose IIAD and SQRT binomial algorithms and claim that they are TCP-friendly using AQM schemes, such as RED.

TABLE I
COMPARISON OF MSES UNDER DIFFERENT MIXTURE

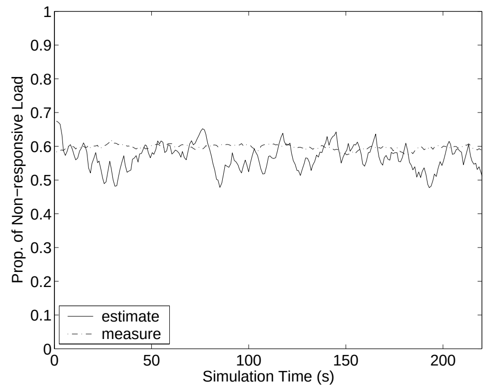
Prop. of binomial flows	**TCP est. model**	
	IIAD	**SQRT**
0%	0.000550	0.000550
50%	0.000584	0.001206
100%	0.002199	0.000451

Since TCP flows are one type of responsive flows, the effectiveness of the proposed estimation algorithm can be checked against other types of responsive flows. If, indeed, the other variants of TCP employing IIAD and SQRT binomial algorithms were responsive to congestion at the time scale of estimation, then the estimate of PONRT using the proposed algorithm should be accurate, provided the algorithm is effective.

To test this hypothesis, a simulation was set with up 22 CBR flows representing non-responsive traffic and 35 responsive flows represented by a mixture of TCP and IIAD/SQRT flows. The same topology and RED configurations, as in previous simulations, was used. A comparison of MSEs with different proportions of IIAD and SQRT binomial flows is shown in Table I. If there is 0% of IIAD/SQRT, it means that all the 35 responsive flows are TCP flows. If there is 100% of IIAD/SQRT, it means that no TCP flow is among the 35 responsive flows. From the simulation results, it can be observed that the difference among MSEs is very small. This result corroborates the



(a) Slowly Changing VBR Flows - Period in [1 20]s (MSE=0.003534)



(b) Fast Changing VBR Flows - Period in [0.1 1]s (MSE=0.002445)

Fig. 9. Estimation under VBR and TCP Traffic

effectiveness of the proposed scheme with other responsive flows.

H. Modification of Basic Algorithm for High Packet Drop Rate

The model, on which the estimation algorithm is based, assumes that the packet drop rate is small enough to affect the queue dynamics (see eqn. 8). However, packet drop rates can be significant when the queue lengths are close to the buffer capacity or to the maximum threshold of a RED router. In order to account for such high drop rates, the estimation algorithm is improved by considering packet drops in the queue dynamics of the model.

Specifically, the queue dynamics can be modeled as:

$$\dot{q}(t) = \frac{z(t) + D - l(t)}{R(t)} - C, \tag{18}$$

where $l(t)$ is the number of packet drops at time t and is known at the router.

Following the same procedure presented in section III, but replacing eqn. 8 with eqn. 18, one gets:

$$R(t)\ddot{q}(t) + \dot{l}(t) + (\frac{\dot{q}(t)}{C} + 1)\dot{q}(t) = \frac{N_s}{R(t)} - \frac{z(t)z\big(t - R(t)\big)}{2N_sR(t)}p\big(t - R(t)\big) \tag{19}$$

The left hand side of eqn. 19 is either known or can be easily calculated by employing a numerical differentiation scheme: $\dot{l}(t) \approx \big(l(t) - l(t-1)\big)/T$. The calculation of $\dot{q}(t)$ and $\ddot{q}(t)$ was given in section III. The left hand side of this equation can be thought of as the modified $\chi_{mod}(t)$, while the right hand side is the same as that for the basic algorithm developed in an earlier subsection. Eqn. 19 provides a linear parametrization of the modified output, $\chi_{mod}(t)$, with respect to the set of unknown parameters, β^*; following the same procedure as in III, the normalized Kaczmarz's projection algorithm is employed to update the unknown parameter vector recursively.

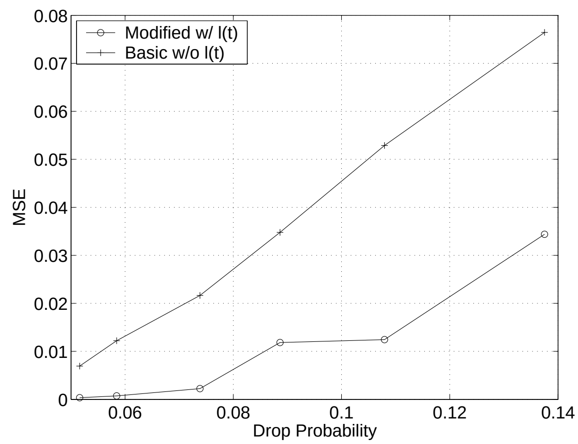


Fig. 10. Comparison of Basic and Modified Algorithm w/ l(t)

To corroborate the effectiveness of the modified algorithm when the drop probability is high, a simulation with 35 TCP flows and 22 CBR flows was set up. In order to increase drop probability $p(t)$, the minimum threshold of RED was increased. In Figure 10, MSEs of algorithm with and without $l(t)$ are compared. It is noticed that, with the increase of drop probability(large number of packet drops), the algorithm *that accounts for* $l(t)$ yields more accurate estimates of PONRT.

V. Estimation Algorithm for Multi-hop Topology

The traffic model developed in Section 2 for one-hop network topology is extended to a multi-hop network topology in this section. Correspondingly, the estimation algorithm is modified. Under a multi-hop topology, the drop probability observed by each TCP flow reflects all the packet drops along the path from its source to its destination. Let P_T^i denote the total drop probability of TCP flow i along its path. In particular, P_T^i = *total number of packet drops/total number of packets sent by TCP flow* i.

The total drop probability P_T^i of TCP flow i can be further decomposed as $p_e(t) + p^i_{\sum}(t)$. The term $p_e(t)$ represents the drop probability seen by the flow at the router employing the proposed estimation algorithm and $p^i_{\sum}(t)$ is the sum of all drop probability encountered by TCP flow i along its path, excluding the drop probability $p_e(t)$. This decomposition is based on the fact that RED routers operate in the linear region of the drop function under recommended configuration so that the drop rate is small enough. Then the approximation rule of $(1-p_a)(1-p_b) \approx 1 - p_a - p_b$ can be applied to decompose P_T. It is noted that $p_e(t)$ is known by the algorithm, while the measurement of $p^i_{\sum}(t)$ is not available and must be determined or taken into account by the estimation algorithm.

Under a multi-hop topology, the window dynamics of TCP flow i is as follows:

$$\dot{W}_s^i(t) = \frac{1}{R(q(t))} - \frac{W_s^i(t)W_s^i\Big(t-R(q(t))\Big)}{2R(q(t))} \cdot\Big(p_e\Big(t-R(q(t))\Big) + p^i_{\sum}\Big(t-R(q(t))\Big)\Big) \quad (20)$$

The aggregated TCP traffic dynamics $\dot{z}(t)$ at the ingress interface of the estimation router is defined as $\dot{z}(t) = \sum_{i=1}^{N_s} \dot{W}_s^i(t)$, where N_s is the total number of TCP flows. Applying eqn. 20 to the definition of $\dot{z}(t)$, we have:

$$\dot{z}(t) = \frac{N_s}{R(t)} - \frac{z(t)z(t-R(t))}{2N_s^2R(t)} \cdot \Big(N_s p_e(t-R(t)) + S_P(t-R(t))\Big) \quad (21)$$

where $S_P(t-R(t)) = \sum_{i=1}^{N_s} p^i_{\sum}(t-R(t))$.

Following the same procedure presented in section III and accounting packet drops $l(t)$ in section IV-H, we get the following equation using eqn. 21:

$$R(t)\ddot{q}(t)+\dot{l}(t) + (\frac{\dot{q}(t)}{C}+1)\dot{q}(t) = \frac{N_s}{R(t)} - \frac{z(t)z(t-R(t))}{2N_s^2R(t)} \cdot \Big(N_s p_e(t-R(t)) + S_P(t-R(t))\Big) \quad (22)$$

An observation from ns-2 simulations is that $S_P(t-R(t))$ changes very slowly or is constant within each sampling interval T. So are $z(t)$ and N_s. We can then parametrize the output linearly in terms of the unknown parameters, analogous to eqn. 22:

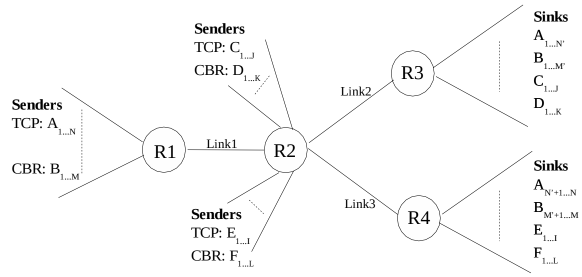


Fig. 11. Multi-hop Simulation Topology with Cross Traffic

$$\begin{aligned} R(t)\ddot{q}(t) + \dot{l}(t) + (\frac{\dot{q}(t)}{C}+1)\dot{q}(t) &= \begin{bmatrix} \frac{1}{R(t)} & -\frac{p(t-R(t))}{2R(t)} & -\frac{1}{2R(t)} \end{bmatrix} \cdot \begin{bmatrix} N_s \\ \frac{z(t)z(t-R(t))}{N_s} \\ \frac{z(t)z(t-R(t))S_P(t-R(t))}{N_s^2} \end{bmatrix} \\ &\approx \begin{bmatrix} \frac{1}{R_0} & -\frac{p(t-R_0)}{2R_0} & -\frac{1}{2R_0} \end{bmatrix} \cdot \begin{bmatrix} N_s \\ \frac{z_0^2}{N_s} \\ \frac{z_0^2 S_P(t-R(t))}{N_s^2} \end{bmatrix} \\ &= W^T(t)\beta \end{aligned} \quad (23)$$

With normalized Kaczmarz's projection algorithm, the unknown vector β can be estimated by utilizing the error $e(t)$ between measured and estimated values. The term $e(t)$ has the same import as in in section III.

Figure 11 shows the simulation topology with multiple hops and congestion links. The capacity of each link is 28Mb. Router R1 employs the extended estimation algorithm on Link 1. Router R2 employs RED queue management scheme on its outgoing Links 2 and 3. Let *TCP flows* be assigned to the flow set of A, C and E and *CBR flows* to the flow set of B, D and F. The terms N, M, J, K, I and L represent the number of flows in their corresponding flow set. The proposed multi-hop estimation algorithm is corroborated using two sets of simulations.

First, the number of CBR flows M of flow set B was changed to be 25, 22 and 16. The number of TCP flows was set to 35 for flow set $A(N=35)$, 17 for flow set $C(J=17)$ and 18 for flow set $E(I=18)$. The number of CBR flows was set to 11 for flow set $D(K=11)$ and 6 for flow set $F(L=6)$. Half of the flows in $A(N'=N/2)$ and in $B(M'=M/2)$ went through router R3. Rest of the flows in A and B went through router R4. Drop probabilities of each link are shown in Table II. The simulation results are shown in Figure 12.

TABLE II

DROP PROB. UNDER DIFFERENT NON-RESPONSIVE LOADS

# of CBRs (M)	Link1	Link2	Link3
25	3%	2.7%	1.6%
22	1.1%	2.8%	1.7%
16	0.3%	1.8%	1.4%

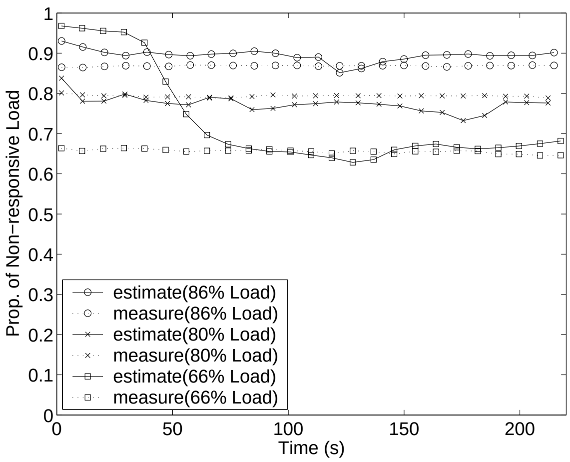


Fig. 12. Estimation of Non-responsive Load under Multi-hop Topology at Router R1

It is observed that even though the drop probability at the router under consideration is equal to or smaller than drop probabilities over other links, the extended estimation algorithm still can produce accurate estimates. In case of 16 CBRs, since most of drops happen at other links, the drop probability at the router under consideration was very low($p_e = 0.3\%$). The level of excitation is low, so the convergence is slow (Notice that estimates converge to measures after 70s.) Here, the modification to the basic algorithm for low packet drop rate is employed to ensure accuracy of estimation.

Second, instead of evenly distributing M to either branch of sinks, 6 flows were assigned to the top branch and 16 to the bottom branch. So in the case, $M = 22$ and $M' = 6$. The number of flows for other flow sets was the same as it was in the previous simulation. The simulation result is shown in Figure 13. It is observed from this simulation that the extended estimation algorithm can produce accurate estimates, independent of the distribution of CBR flows.

VI. DISCUSSION AND FUTURE WORK

In this paper, an estimation algorithm to estimate the fraction of incoming traffic that is non-responsive to congestion signals (packet drops) is presented. This method relies on the evolution of queue length and packet drop rate.

When the arrival rate at the router is low, the queue lengths tend to remain low and so is the dropping probability. In this case, the level of excitation used in this method is low. As a result, the presented method tends to be more accurate in higher load situations. It is at these times of higher loads that traffic engineering decisions or potential attack detections need to be made. Hence, the proposed method seems well suited for such situations.

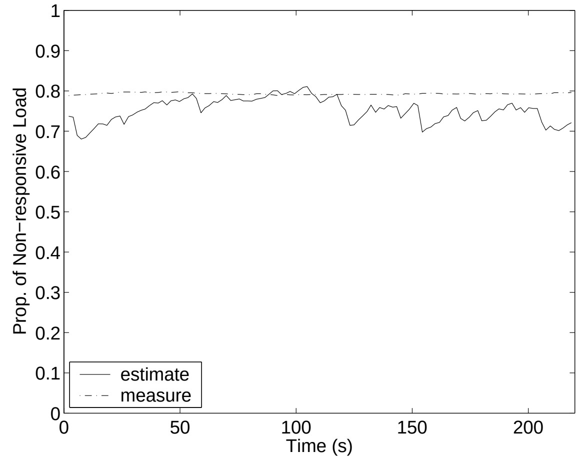


Fig. 13. Estimation under Multi-hop Topology with Unevenly Distributed Flows at Router R1

The focus of the presented method is on estimating the aggregate amount of non-responsive traffic at the router; this is in contrast to the earlier work suggesting the implementation of checks to see if individual flows are responding to congestion [4].

The presented work is motivated by traffic engineering concerns. It is expected that the estimation of non-responsive traffic would lead to the following applications: (a) providing a means to controlling non-responsive traffic; it is possible that an attack detection mechanism could be developed based on a robust estimation of PONRT. One may probably declare that an “attack” is in progress if the PONRT exceeds certain acceptable threshold. (b) providing a mechanism for tuning traffic control algorithms at the time of congestion. In [7], it is shown that in the presence of high non-responsive loads, drop tail buffer management may be better than RED style active queue management. The presented method could possibly be used to make such decisions at the times of congestion. To illustrate this, a simulation was set up with 22 CBR flows and 35 TCP flows competing over one 28Mb bottleneck link. In first two rows of Table III, the ability of Droptail and RED(15/45/0.1) routers to handle non-responsive flows and protect responsive flows is compared. One can observe that Droptail outperforms RED in protecting TCP flows from non-responsive CBR flows, since TCP still consumes approximately 70% of link bandwidth using Droptail, compared to 25% of bandwidth using RED. But, as shown in the third row of Table III, if one sets min_{th} = max_{th} = *buffer size of RED*, RED performs better to protect TCP flows than RED with (15/45/0.1) parameters. So it may be

TABLE III
PERFORMANCE OF DROPTAIL AND RED

Queue Mngt.	% of TCP	% of CBR	Drop Rate
Droptail	71.5%	28.4%	34.2%
RED (15/45/0.1)	24.3%	75.6%	5.64%
RED (min_{th}=max_{th}=*buf. size*)	61.3%	38.6%	29.3%

possible to tune the configurations of RED to suit different workloads. (c) providing a better control for enforcing service differentiation. Earlier work [30] on analyzing assured forwarding in differentiated services has shown that non-responsive traffic may disrupt service for responsive sources even when traffic is marked differently at the edge. With the presented estimation method, it is possible to adapt the traffic control parameters to provide better service. These applications are currently being researched by the authors.

Initial results based on the presented estimation method are promising. The general problem of building a robust estimation tool, however, is more challenging. In order to make the tool more robust, the following extensions are planned: (a) incorporating a mechanism to differentiate short-term web flows from non-responsive attack flows in order to provide a better attack detection mechanism; this can be achieved through a filtering mechanism based on partial state [16] that allows the algorithm to be applied to only long-term flows, (b) using a more detailed model of TCP behavior that includes timeouts to extend the useful operating range of the estimation algorithm, (c) identifying techniques to systematically choose proper parameters in the algorithm for different applications, (d) corroborating the presented analytical and simulation results by implementing the proposed method on a Linux-based router, and (e) studying the potential applications of the developed model in traffic engineering and control.

VII. CONCLUSION

In this paper, a mathematical model describing the aggregate dynamics of heterogeneous traffic at a router is presented. The model is used in deriving an algorithm for estimating the fraction of the incoming traffic that is non-responsive to congestion. The effectiveness of the proposed algorithm, over a wide range of traffic conditions, is corroborated using ns-2 based simulations. Possible future applications of the proposed algorithm are discussed.

ACKNOWLEDGMENTS

Jayesh Ametha contributed to our initial work. A short version of this paper appears in ACM Sigmetrics 2002 conference. The comments of conference reviewers and reviewers from ACM/IEEE Transactions on Networking have helped us in improving the paper.

REFERENCES

[1] N. L. for Applied Network Research, “Network traffic packet header traces,” *http://moat.nlar.net/Traces/*, 2000.

[2] S. Floyd, M. Handley, J. Padhye, and J. Widmer, “Equation-based congestion control for unicast applications,” *Proc. of ACM SIGCOMM*, Aug. 2000.

[3] D. Bansal and H. Balakrishnan, “TCP-friendly congestion control for real-time streaming applications,” *Proc. of Infocom*, Apr. 2001.

[4] S. Floyd and K. Fall, “Promoting the use of end-to-end congestion control in the internet,” *ACM/IEEE Trans. on Networking*, Aug. 1999.

[5] E. to-end discussion group, “Web mice versus elephants,” *Mailing list discussion*, 2000-2001.

[6] M. Christiansen, K. Jeffay, D. Ott, and F. D. Smith, “Tuning RED for web traffic,” *Proc. of SIGCOMM’00*, Aug. 2000.

[7] C. Diot, G. Iannaccone, and M. May, “Aggregate Traffic Performance with Active Queue Management and Drop from Tail,” *Sprint ATL Technical Report TR01-ATL-012501*, 2001.

[8] P. Inc., “Packetshaper: Opening a world of intelligent bandwidth management services,” *Technology white paper, http://www.packeteer.com/*, 1998.

[9] V. Misra, W. B. Gong, and D. Towsley, “A fluid based analysis of a network of AQM routers supporting TCP flows with an application to RED,” *Proc. of ACM SIGCOMM*, Aug. 2000.

[10] C. V. Hollot, V. Mishra, D. Towsley, and W. Gong, “A control theoretic analysis of RED,” *Proc. of Infocom*, Apr. 2001.

[11] E. Altman, K. Avrachenkov, and C. Barakat, “Stochastic Model of TCP/IP with Stationary Random Losses,” *Proc. of SIGCOMM’00*, Aug. 2000.

[12] S. Kuniyur and R. Srikant, “Analysis and design of an adaptive virtual queue algorithm for active queue management,” *Proc. of ACM SIGCOMM’01*, Aug. 2001.

[13] C. V. Hollot, V. Mishra, D. Towsley, and W. Gong, “On designing improved controllers for AQM routers supporting TCP flows,” *Proc. of Infocom*, Apr. 2001.

[14] U. of California at Berkeley, “Network simulator (ns),” *Available via http://www-nrg.ee.lbl.gov/ns/*, 1997.

[15] S. Darbha and K. R. Rajagopal, “Limit of a collection of dynamical systems: An application to modeling the flow of traffic,” *Mathematical Modeling and Methods in Applied Sciences*, vol. 12, no. 10, pp. 1381–1399, Oct. 2002.

[16] Smitha and A. L. N. Reddy, “LRU-RED: An active queue management scheme to contain high bandwidth flows at congested routers,” *Proc. of Globecomm*, Nov. 2001.

[17] D. Tong and A. L. N. Reddy, “QoS enhancement with partial state,” *Proc. of IWQoS*, Jun. 1999.

[18] Z. Zhao, “Modeling and estimation techniques for understanding heterogeneous traffic behavior,” Ph.D. dissertation, Texas A&M University, in preparation.

[19] M. Vidyasagar, *Nonlinear Systems Analysis*, 2nd ed. SIAM, 2002.

[20] A. M. Lyapunov and A. T. Fuller, *The General Problem of the Stability of Motion.* Taylor and Francis, Inc., 1992.

[21] K. L. Cooke and W. Huang, “On the problem of linearization for state-dependent delay differential equations,” *Proc. Amer. Math. Soc.*, vol. 124, no. 5, pp. 1417–1426, May 1996.

[22] K. J. Åström and B. Wittenmark, *Adaptive Control.* Reading, MA: Addison Wesley, 1995.

[23] L. Ljung and E. J. Ljung, *System Identification: Theory for the User*, 2nd ed. Pearson Education, 1999.

[24] G. C. Goodwin and K. S. Sin, *Adaptive Filtering, Prediction and Control.* Prentice Hall, 1984.

[25] J. J. Craig, *Introduction to Robotics: Mechanics and Control*, 2nd ed. Addison-Wesley, 1989, p.289.

[26] J. Green and K. Hedrick, “Nonlinear speed control of engines,” *Proc. of the American Control Conference, San Diego*, 1990.

[27] S. Sastry and M. Bodson, *Adaptive Control: Stability, Convergence, and Robustness.* Prentice Hall, 1989.

[28] N. Caldwell, S. Savage, and T. Anderson, “Modeling tcp latency,” *Proc. of IEEE Infocom*, March. 2000.

[29] P. Huang, A. Feldmann, and W. Willinger, “A non-intrusive, wavelet-based approach to detecting network performance problems,” *Proc. of Internet Measurement Workshop*, Nov. 2001.

[30] I. Yeom and A. L. N. Reddy, "Realizing throughput guarantees in a differentiated services network," *Proc. of IEEE Conf. on Multimedia Computing and Systems*, June 1999.

Zhili Zhao Zhili Zhao received his B.S. in Electronic Technology from East China Normal University, Shanghai, China, in July 1994 and M.S. in Optoelectronics from Shanghai Institute of Technical Physics, Academia Sinica, Shanghai, in April 1997. He is currently a Ph.D. candidate in Computer Engineering, Department of Electrical Engineering, Texas A & M University. His current research interest is internet traffic estimation and regulation.

Swaroop Darbha Swaroop Darbha received his B. Tech degree in Mechanical Engineering from the Indian Institute of Technology at Madras in 1989, and earned M. S. and Ph. D. degrees in Mechanical Engineering from the University of California at Berkeley in 1992 and 1994 respectively. He is currently an Associate Professor in Mechanical Engineering at Texas A&M University, where he conducts research in the areas of modeling the dynamics of the flow of ground traffic, and in the development of control, diagnostic algorithms for ground vehicles. His e-mail address is: dswaroop@mengr.tamu.edu.

A. L. Narasimha Reddy A. L. Narasimha Reddy received his B.Tech. degree in Electronics and Electrical Engineering from the Indian Institute of Technology, Kharagpur, India, in August 1985, and the M.S. and Ph.D degrees in Computer Engineering from the University of Illinois at Urbana-Champaign in May 1987 and August 1990 respectively. At Illinois, he was supported by an IBM Fellowship.

He is currently an Associate Professor in the department of Electrical Engineering at Texas A & M University. He was a Research Staff Member at IBM Almaden Research Center in San Jose from Aug. 1990- Aug. 1995.

Reddy's research interests are in Storage Systems, Multimedia, Network and Computer Architecture. Currently, he is leading projects on building network-based storage systems and partial-state based network elements. While at IBM, he coarchitected and designed a topology-independent routing chip operating at 100 MB/sec, designed a hierarchical storage management system and participated in the design of video servers and disk arrays.

Reddy is a member of ACM SIGARCH and is a senior member of IEEE Computer Society. He has received an NSF CAREER award in 1996. He recieved an Outstanding Professor award at Texas A & M during 1997-98.