2011 American Control Conference
on O'Farrell Street, San Francisco, CA, USA
June 29 - July 01, 2011

On a Rate Control Protocol for Networked Estimation

Vaibhav Katewa and Vijay Gupta

Abstract— **We study congestion control in a communication network that is supporting remote estimation of multiple processes. A stochastic rate control protocol is developed using the network utility maximization (NUM) framework. The decentralized protocol avoids congestion by regulating the transmission probabilities of the sources. The presence of estimation costs poses new challenges; however, for low congestion, the form of rate controller resembles that of the standard TCP rate controller.**

I. INTRODUCTION

Networked control systems (e.g., [2], [17] and the references therein) consist of spatially distributed components - processes, sensors, controllers, and actuators - that coordinate among each other through information exchange over a communication network to achieve a desired control objective. The communication network, thus, plays an important role in determining the stability and performance of a networked control system. Physical networks introduce various imperfections like packet drops, delays, scheduling constraints, and so on. The effect of these imperfections on the performance of such systems has been widely studied. For a survey of such results, see, e.g., [3], [9] and the references therein.

However, most of the research has focused on analyzing and designing a single networked control system in isolation. While this has led to important foundational results, it has ignored the new problems that may arise when multiple such systems operate over a common communication network. As an example, networked communication may give rise to congestion or MAC delays. Such effects will impact the performance of every networked control system and in fact, will couple their performance even though the systems may not be dynamically coupled. It is, thus, of interest to study the impact of communication network protocols on the performance of multiple networked control systems sharing a common network, and in fact, design network protocols that are suitable for estimation and control applications [7], [14].

In this paper, we focus on the rate control protocol suitable for an estimation oriented cost function. We consider multiple systems, each of which consists of an estimator that remotely estimates the state of an associated process. A sensor collocated with each process transmits information over a shared communication network to the estimator. The network has capacity constraints for every link which may result in congestion when the network load increases. Congestion results in packet losses and delays, which adversely affect the estimation performance. We show that traditional rate control protocols may not be suitable, and propose a distributed rate control protocol for optimizing the estimation performance.

Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN 46556. vkatewa,vgupta2@nd.edu Research funded in part by NSF awards 0834771 and 0846631.

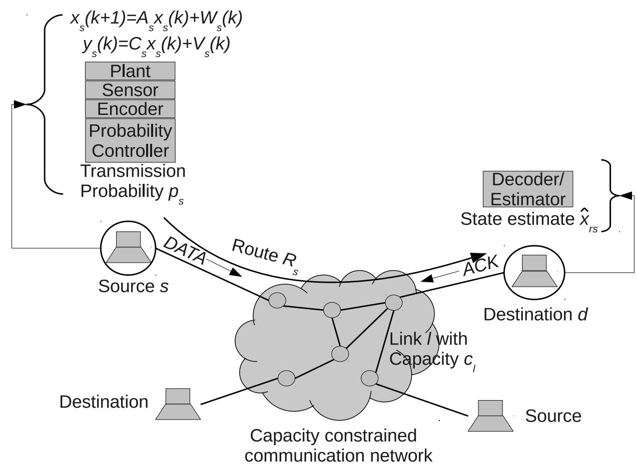

Fig. 1. The problem setup considered in the paper. Multiple processes are remotely estimated across a shared communication network.

The problem of congestion control [10] has been well studied for communication networks and TCP [16] is the most widely used congestion control protocol in the internet. While originally an engineering heuristic, TCP has been reverse engineered to show that it is a distributed solution that optimizes a particular utility function [11]. The chief tool in this regard is the Network Utility Maximization (NUM) framework [12]. We use a cost minimization framework that is analogous to the standard NUM framework. The total cost that the rate control protocol aims to minimize includes both an estimation performance cost and a congestion cost. The work closest to ours is that of [1] which presents a bandwidth allocation scheme by using a dual form of NUM problem. However, our solution is in the primal form and is similar to the structure of the standard TCP protocol. Moreover, we present a stochastic transmission scheme as opposed to the deterministic transmission scheme in [1].

II. PROBLEM FORMULATION

A. Network and Process Setting

Consider the problem set up shown in Figure 1. Let all the sources form a source set $\mathcal{S}$. With each source $s \in \mathcal{S}$, associate a unique destination d and denote the destination set by $\mathcal{D}$. Let each source be connected to its corresponding destination through the capacity constrained network $\mathcal{N}$. We model the network as a graph whose nodes correspond to sources, destinations and routers, while the edges correspond to the communication channels in the network. Let $\mathcal{L}$ be the set of links in the network and $L(s)$ be the set of links that are used by source s to communicate with its corresponding destination d. Further, denote the route between source s and destination d by R_s. Each link $l \in \mathcal{L}$ has a limited capacity c_l in terms of "packets per time slot" on an average. Any individual link may be shared by one or more sources.

Each source s comprises of a process P_s, a sensor SR_s, an encoder ENC_s and a rate controller PC_s. The process evolves according to the discrete-time linear model

$$P_s: \qquad x_s(k+1) = A_s x_s(k) + W_s(k), \qquad k \geq 0 \quad (1)$$

where $x_s(k) \in \mathbb{R}^{n_s}$ $(n_s \in \mathbb{N}_+)$ is the process state and $W_s(k)$ is the process noise. The initial condition $x_s(0)$ and the white process noise $W_s(k)$ are assumed to be Gaussian with zero mean and variance $X_s > 0$ and $Q_s > 0$, respectively. The output of the process P_s is sensed by the sensor SR_s which generates noisy measurements according to

$$SR_s: \qquad y_s(k) = C_s x_s(k) + V_s(k), \qquad k \geq 0 \quad (2)$$

where $y_s(k) \in \mathbb{R}^{m_s}$ $(m_s \in \mathbb{N}_+)$ is the process output, $V_s(k)$ is the measurement noise that is assumed to be white, Gaussian with zero mean and variance $\Sigma_s > 0$. The initial state and the noises $\{x_s(0), W_s(k), V_s(k)\}$ are assumed to be mutually independent $\forall\ s \in \mathcal{S}$ and $\forall\ k$. Further, these random variables are assumed to be mutually independent among all sources. Finally, we assume that each pair (A_s, C_s) is observable.

The encoder ENC_s uses the noisy measurements to generate data that is transmitted it to its corresponding destination using constant size packets. The packet size is assumed to be large enough to represent a real number with negligible quantization error. Therefore, we ignore any quantization effects in this study. Each destination comprises of a decoder DEC_d, which generates a state estimate using the received data that is optimal in the minimum mean squared error (MMSE) sense. We employ the encoder and decoder scheme described in [8]. At source s, denote the local estimate of state $x_s(k)$ given the measurements $\{y_s(j)\}_{j=0}^{k-1}$ by $\hat{x}_s(k)$. Further, denote the state estimate produced by DEC_d at the corresponding destination d, by $\hat{x}_{rs}(k)$. The encoder and the decoder are given by

- ENC_s:
  - At each time slot k, calculate $\hat{x}_s(k)$ using (say) a Kalman Filter.
- DEC_d:
  - If DEC_d receives a packet (local state estimate) in time slot k, $\hat{x}_{rs}(k) = \hat{x}_s(k)$.
  - If DEC_d does not receive a packet in time slot k, $\hat{x}_{rs}(k) = A_s \hat{x}_{rs}(k-1)$.

As discussed in [8], such an encoder-decoder structure is optimal amongst all other causal structures.

B. Communication Scheme

We assume a stop-and-wait type communication protocol. In time slot k, the source s transmits the local estimate $\hat{x}_s(k)$ to the corresponding destination d. The transmission is stochastic with transmission probability $p_s(k)$ at time slot k. Transmission events at different time slots are assumed to be mutually independent. The transmission probability $p_s(k)$ can be viewed as the effective transmit rate of the source s in terms of "packets per time slot" on average. Hence, the rate controller PC_s is implemented as a probability controller. We use the term 'rate' and 'transmission probability' interchangeably.

As the total rate on a link approaches the link capacity, congestion in the link increases, which may result in packets being dropped by routers in the network. The probability controllers control the congestion level by varying the transmission probabilities. Let $d_s(k)$ be the probability that a packet is dropped by the network on route R_s at time slot k. The packet drop events on route R_s at different time slots are assumed to be mutually independent. Further, the packet drop and packet transmission events on route R_s are assumed to be independent for every time slot.

When a destination receives a packet, it sends back an acknowledgment (ACK) to the corresponding source. We assume that an ACK is received in the same time slot as the packet was transmitted. Further, we assume that $ACKs$ are never lost in the network. We also assume no retransmissions in case of packet loss. Retransmissions increase delay and delayed information is expected to be less useful to the estimator [13]. Finally, we assume synchronization between each source-destination pair.

In contrast to deterministic schemes, the proposed stochastic transmission scheme controls the source rate by varying the transmission probability. A stochastic transmission scheme is a natural choice since a congested network drops packets stochastically. Due to the stochastic rate control, there may be instants when many sources may not transmit resulting in instantaneous network underutilization or, many sources may transmit at the same time resulting in instantaneous increase in congestion. However, due to the feedback implicit in rate control, such instants will be few and on an average, the network is utilized in an optimal manner.

Let the packet drop probability on a link $l \in \mathcal{L}$ at time slot k be denoted by $d_l(k)$. The drop probability on a link depends on both the link capacity and the total rate on the link. As the total rate approaches link capacity, the queue in the corresponding router becomes full and the network becomes unstable. In such a situation, all the packets are dropped by the router with a probability approaching 1. To avoid such instances, routers use queue management protocols such as Random Early Detection (RED) [6]. In RED, routers increase the drop probability as the queue size increases. The packet drops serve as a feedback mechanism to rate control protocols such as TCP, which reduces the source rate in response to congestion. In standard RED, the link drop probability is a pre-specified increasing and convex function of the total rate (assuming, say, a M/M/1 queue model).

Using the standard assumption [4] that the drop events on various links are mutually independent, the drop probability $d_s(k)$ on route R_s can be expressed as

$$d_s(k) = 1 - \prod_{l \in L(s)} (1 - d_l(k)). \quad (3)$$

Thus, $d_s(k)$ depends on the rates of the sources that share the links with source s. This introduces a coupling to the networked estimation problem. Note that d_s may not be a convex function of the source rates.

For source s and its corresponding destination d, denote the estimation error covariances of the local estimate $\hat{x}_s(k)$ and the remote estimate $\hat{x}_{rs}(k)$ by $M_s(k)$ and $F_s(k)$. Thus,

$$\begin{aligned} M_s(k) &= \mathbb{E}\left[(x_s(k) - \hat{x}_s(k))(x_s(k) - \hat{x}_s(k))^T\right] \\ F_s(k) &= \mathbb{E}\left[(x_s(k) - \hat{x}_{rs}(k))(x_s(k) - \hat{x}_{rs}(k))^T\right], \end{aligned}$$

where the expectation is taken with respect to the process noise $\{W_s(k)\}$, the measurement noise $\{V_s(k)\}$ and the initial condition $x_s(0)$. Since the pair (A_s, C_s) is observable, the local estimation error covariance $M_s(k)$ converges to a steady state value, denoted by M_s with a slight abuse of notation. Further, according to the decoder structure DEC_s, the remote estimation error covariance $F_s(k)$ evolves as

$$F_s(k) = \begin{cases} M_s(k), & \text{if a packet is received} \\ A_s F_s(k-1) A_s^T + Q_s, & \text{otherwise.} \end{cases}$$

Thus, $F_s(k)$ is a random variable. As a performance metric, we consider its expected value, that evolves as

$$\begin{aligned} &\mathbb{E}[F_s(k)] = p_s(k)(1 - d_s(k)) M_s(k) \\ &+ (1 - p_s(k)(1 - d_s(k)))\left(A_s \mathbb{E}[F_s(k-1)] A_s^T + Q_s\right), \quad (4) \end{aligned}$$

where $p_s(k)(1 - d_s(k))$ is the packet reception probability and the expectation is further taken with respect to the packet transmission process and packet drop process. A necessary and sufficient condition for the convergence of $\mathbb{E}[F_s(k)]$ as (4) evolves is given by [8]

$$p_s(1 - d_s) \geq 1 - \frac{1}{\rho^2(A_s)} \triangleq p_s^{min}, \quad (5)$$

where p_s and d_s are the steady state values of $p_s(k)$ and $d_s(k)$, respectively, and $\rho(X)$ denotes the spectral radius of matrix X. Thus, the steady state remote estimation error covariance satisfies the discrete algebraic Lyapunov equation

$$\begin{aligned} F_s(p_s, d_s) &= p_s(1 - d_s) M_s \\ &\quad + (1 - p_s(1 - d_s))(A_s F_s(p_s, d_s) A_s^T + Q_s), \end{aligned}$$

where $F_s(p_s, d_s)$ denotes the steady state value of $\mathbb{E}[F_s(k)]$.

C. Problem Statement

Let $\mathbf{p}$ denote the vector of all steady state transmission probabilities, i.e. $\mathbf{p} = (p_1, p_2, \cdots, p_{|\mathcal{S}|})^T$, where $|\mathcal{S}|$ denotes the cardinality of set $\mathcal{S}$. We consider the estimation cost incurred for the source s as $c_s = tr(F_s(p_s, d_s))$, where tr denotes the trace. Lower value of this cost implies better estimation performance and vice-versa. Further, the total cost of the system $C_{sys}(\mathbf{p})$ is chosen to be the sum of individual costs. Other choices of individual and system costs are also possible. For ease of notation, we will denote $\{tr(F_s(p_s, d_s)), tr(M_s), tr(A_s A_s^T), tr(Q_s)\}$ by $\{f_s(p_s, d_s), m_s, a_s, q_s\}$, respectively. Thus,

$$\begin{aligned} C_{sys}(\mathbf{p}) &= \sum_{s \in \mathcal{S}} f_s(p_s, d_s) \\ f_s(p_s, d_s) &= p_s(1 - d_s) m_s \\ &\quad + (1 - p_s(1 - d_s))(tr(A_s F_s(p_s, d_s) A_s^T) + q_s). \quad (6) \end{aligned}$$

The problem is to find the optimal value of $\mathbf{p}$ which minimizes the cost function $C_{sys}(\mathbf{p})$. This problem can also be viewed as a resource (rate) allocation problem with an objective to minimize a system cost. We are particularly interested in decentralized solutions that scale to large networks.

III. Analysis and Results

A. Cost Function

The cost from equation (6) is not amenable to analysis. We begin by obtaining a closed form upper bound on it.

Lemma 3.1: The steady state value $f_s(p_s, d_s)$ is upper bounded by $f_s^u(p_s, d_s)$ where

$$f_s^u(p_s, d_s) \triangleq \frac{p_s(1 - d_s) m_s + (1 - p_s(1 - d_s)) q_s}{1 - tr(A_s A_s^T)(1 - p_s(1 - d_s))}. \quad (7)$$

In particular, for scalar processes (i.e., when the state $x_s(k)$ is a scalar), the upper bound in (7) is satisfied with equality. For analytical tractability, we replace f_s by its upper bound f_s^u in the system cost. Thus, we approximate

$$C_{sys}(\mathbf{p}) \approx C(\mathbf{p}) \triangleq \sum_{s \in \mathcal{S}} f_s^u(p_s, d_s),$$

where p_s is the transmission probability allotted to source s.

Thus, we have the following optimization problem

$$\begin{aligned} SYSTEM: \quad & \min_{\mathbf{p}} \sum_{s \in \mathcal{S}} f_s^u(p_s, d_s(\mathbf{p})), \\ & \text{s. t.} \sum_{s: l \in R_s} p_s \leq c_l, \forall\, l \in \mathcal{L}, \\ & p_s(1 - d_s(\mathbf{p})) \geq p_s^{min}, \ \forall\, s \in \mathcal{S}, \end{aligned}$$

where the notation $d_s(\mathbf{p})$ denotes the explicit relation between the drop probability and transmission probabilities. Assuming that a feasible region exists, we can use standard optimization techniques to obtain a globally optimal solution. However, there are many difficulties with this approach:

1) If the drop probability d_s is not a convex function of $\mathbf{p}$, then the system cost $C(\mathbf{p})$ may not be convex, thus making the problem difficult.
2) It is not scalable since each source requires information about the transmission probabilities and process parameters of all the other sources.
3) It requires the functional relation between $\{d_s : s \in (S)\}$ and $\{p_s : s \in (S)\}$, which may be unavailable in a practical scenario.

We now proceed to transform the problem into convex form and obtain a distributed solution.

B. Posing the Problem in the NUM Framework

To obtain a scalable solution in a distributed form, we employ a network cost minimization framework that is analogous to the primal formulation of the Network Utility Maximization (NUM) framework [15]. In the minimization framework, the transmission probabilities $\{p_s : s \in \mathcal{S}\}$ are regarded as resources and $\{f_s^u(p_s, d_s) : s \in \mathcal{S}\}$ are regarded as costs associated with utilizing the resources. Then, the optimization problem is to distribute the resources such that the sum of the costs is minimized.

Remark 3.2 (Advantage of the primal form): Although the NUM framework can be applied in both primal and dual forms, we use the primal form due to the following reasons:

- Since the communication network may be used for data unrelated to estimation / control, the dynamics of the distributed solution should be at the sources and not at the links. This is important especially in heterogeneous networks, where different sources may have different interpretations of link prices. Thus, a single link price controller may not be suitable for all the sources.
- The primal solution requires changes to the standard TCP only at sources and not in the network. Thus, our solution is practically useful since implementation of the rate controllers can be done at the source node.

The NUM framework imposes some requirements on the costs. First, the costs should be separable among various sources. In other words, the cost associated with source s should depend only on the resource p_s. Second, the cost should be positive, monotonically decreasing and convex. To satisfy these requirements, we proceed as follows.

The costs $\{f_s^u(p_s, d_s) : s \in \mathcal{S}\}$ in (7) are coupled through the drop probabilities d_s and hence are non-convex. Therefore, we eliminate d_s from the costs to make them separable and convex. Denote by $f_s^u(p_s, 0)$ the separable cost, i.e.

$$f_s^u(p_s, 0) = f_s^u(p_s | d_s = 0).$$

To include the effect of the drop probabilities, we define a barrier of the form $B_l\Big(\sum\limits_{s:l\in R_s} p_s\Big)$ corresponding to each link l, and add it to the total cost. The barrier maps the congestion level in the link to an additional additive cost to the system. We obtain following relaxation of the $SYSTEM$ problem

$$\begin{aligned} USER: \qquad & \min_{\mathbf{p}} \sum_{s\in\mathcal{S}} f_s^u(p_s, 0) + \sum_{l\in\mathcal{L}} B_l\Big(\sum_{s:l\in R_s} p_s\Big), \\ \text{s.t.} \qquad & p_s \geq p_s^{min}, \ \forall\, s \in \mathcal{S}. \end{aligned}$$

The choice of the barrier function requires some care. It should be a monotonically increasing function of the total rate on a link. This ensures that as the congestion increases, the total system cost also increases. Thus, congestion control can be achieved by minimizing the system cost. By ensuring a steep increase in the barrier function as the rates approach capacity of the links, the capacity constraints can be explicitly incorporated in the system cost. Note that the different choices of the barrier function may change the way in which congestion control is handled. For example, in a conservative approach, the barrier may be high for low link rates. Thus, the system cost is high for even low levels of congestion.

Once we have satisfied the separability requirement, we can prove that the cost used in the $USER$ problem satisfies the remaining constraints. There are two terms in the cost function, that we consider one by one.

Proposition 3.3: The cost function $f_s^u(p_s, 0)$ is positive, monotonically decreasing and convex in p_s.

To ensure the convexity of the barrier function, we assume that B_l is differentiable and denote

$$B_l\Big(\sum_{s:l\in R_s} p_s\Big) \triangleq \int_0^{\sum_{s:l\in R_s} p_s} t_l(x)dx,$$

where t_l is the penalty function corresponding to link l. If t_l is a monotonically increasing function of the total rate on the link l, then B_l is convex. We will ensure this by choosing an appropriate penalty function in Section III-D. Finally, we note that the constraints on minimum transmission rate are automatically satisfied.

Lemma 3.4: The cost used in the problem $USER$ implicitly guarantees the constraints $p_s > 1 - \frac{1}{\rho^2(A_s)}$.

Proof: This result is true since the cost $f_s^u(p_s, 0)$ becomes infinite when p_s approaches p_s^{min}. This can be seen as follows. By examining the denominator of cost $f_s^u(p_s, 0)$, we can see that it is positive and finite iff $p_s > 1 - \frac{1}{a_s}$. But the quantity $a_s = tr(A_s A_s^T)$ is simply the square of the Frobenius norm of A_s, which is greater that the square of spectral norm (spectral radius) of A_s, i.e. $a_s \geq \rho^2(A_s)$. Thus, $f_s^u(p_s, 0)$ is positive and finite only for $p_s > 1 - \frac{1}{a_s} \geq 1 - \frac{1}{\rho^2(A_s)}$. ■

C. Solution of the Optimization Problem

We have shown that if we choose the penalty function appropriately, then the total system cost in the $USER$ problem is positive and convex. Moreover, the problem constraints are implicitly included in the system cost. Thus, a gradient descent algorithm can be used to minimize the total system cost. We propose a rate (probability) controller of the form

$$\begin{aligned} PC_s: \qquad & p_s(k+1) = p_s(k) \\ & - K\Big(\frac{d}{dp_s} f_s^u(p_s, 0) + \sum_{l:l\in L(s)} t_l\Big(\sum_{s:l\in R_s} p_s\Big)\Big), \end{aligned} \tag{8}$$

where $K > 0$ is a sufficiently small step size. The quantity

$$q_{R_s} \triangleq \sum_{l:l\in L(s)} t_l\Big(\sum_{s:l\in R_s} p_s\Big)$$

can be viewed as the price of using the route R_s, which is the aggregate of prices of all the links on the route.

Remark 3.5 (Scalability): The proposed rate control protocol is scalable to large networks. The values of process parameters and transmission probabilities of other sources are not required to implement the algorithm. The only information that a source needs is the route price. This can be implicitly or explicitly provided by the network through the $ACKs$ from the destination to the source.

Proposition 3.6: Let $\mathbf{p}^*$ be the unique optimal point of the unconstrained strictly convex optimization problem $USER$. Then $\mathbf{p}^*$ is an asymptotically stable equilibrium point of the probability controllers $\{PC_s : s \in \mathcal{S}\}$.

D. Penalty Function

Besides being monotone increasing in the rates, the penalty functions t_l should be chosen such that the problem $USER$ closely approximates the problem $SYSTEM$. The congestion in the network affects the system performance through the drop probabilities. Accordingly, we choose a penalty function that depends on drop probabilities. In turn, since the drop probability d_l on a link l depends on the total rate on the link, the penalty function also depends on the total rate on the link, as required by the optimization framework. In particular, we choose

$$t_l\Big(\sum_{s:l\in R_s} p_s\Big) = -\log\Big(1 - d_l\Big(\sum_{s:l\in R_s} p_s\Big)\Big), \qquad (9)$$

where the notation $d_l\Big(\sum_{s:l\in R_s} p_s\Big)$ explicitly denotes that link drop probability depends on the total rate on the link. Note that t_l is positive and monotonically increases to infinity; thus the barrier function is indeed convex as required. In fact, the barrier function is given by

$$B_l\Big(\sum_{s:l\in R_s} p_s\Big) = \int_0^{\sum_{s:l\in R_s} p_s} -\log(1 - d_l(x))dx.$$

Also, the route price is given by

$$\begin{aligned} q_{R_s} &= \sum_{l:l\in L(s)} -\log(1 - d_l), \\ &= -\log\Big(\prod_{l:l\in L(s)} (1 - d_l)\Big), \\ &= -\log(1 - d_s). \end{aligned}$$

Remark 3.7 (Estimating the route price): The advantage of choosing a logarithmic penalty function is visible from the preceding calculation. To calculate the route price, the probability controllers PC_s require only the route drop probability d_s. They do not require the prices of individual links along the route. Thus, no explicit field in the $ACKs$ is required to collect price information from the links. The route drop probability can be estimated just based on whether $ACKs$ are received or not.

The barrier B_l is the integral of a logarithmic function between the interval [0,1]. Therefore, it does not diverge as the congestion increases. Ideally, when the network congestion is large, the barrier should be large as compared to the estimation cost. Thus, we scale down the cost (analogous to increasing the barrier function) $f_s^u(p_s, 0)$ by a constant β_s to satisfy this property. We choose $\beta_s = N_s(q_s - m_s(1 - a_s))$, where N_s is a large positive constant. The constant β_s is large when the process is more unstable or the process and measurement noises are large. Thus, it acts like a normalization factor to the estimation error covariance.

With this relaxation, the optimization problem becomes

$$USER: \qquad \min_{\mathbf{p}} \sum_{s\in\mathcal{S}} \frac{1}{\beta_s} f_s^u(p_s, 0) + \sum_{l\in\mathcal{L}} B_l\Big(\sum_{s:l\in R_s} p_s\Big),$$

and the probability controller becomes

$$\begin{aligned} PC_s: \qquad & p_s(k+1) = p_s(k) \\ & + K\left(\frac{1}{(1 - a_s(1 - p_s(k)))^2} + N_s \log(1 - d_s(k))\right). \qquad (10) \end{aligned}$$

E. Modified TCP-like Probability Controller

As a final step, we show how to implement the probability controller structure in (10) using a TCP-like structure under low network congestion conditions. In this regime, the route drop probabilities are also low, $\{d_s \ll 1, s \in \mathcal{S}\}$. In this case, $-\log(1 - d_s) \approx d_s$ and the route drop probability becomes the route price. Under this condition, the probability controller in (10) becomes

$$\begin{aligned} PC_s: \qquad & p_s(k+1) = p_s(k) \\ & + K\left(\frac{1}{(1 - a_s(1 - p_s(k)))^2} - N_s d_s(k)\right). \qquad (11) \end{aligned}$$

We propose a stochastic counterpart of (11) that varies the transmission probability based on whether $ACKs$ are received or not. Consider the following TCP-like probability controller, denoted by PC_s^{TCP}

- If a packet is not transmitted in time slot k, then set
$$p_s(k+1) = p_s(k).$$
- If a packet is transmitted and ACK is received, then set
$$p_s(k+1) = p_s(k) + K.$$
- If a packet is transmitted and ACK is not received, then set
$$p_s(k+1) = p_s(k) - K(N_s(1 - a_s(1 - p_s(k)))^2 - 1).$$

The actions of the probability controller PC_s^{TCP} are intuitive. The transmission probability is varied only if a packet is sent. If a packet is successfully transmitted (i.e an ACK received), then the transmission probability is increased by a small constant amount. However, if the transmission fails (no ACK is received), this indicates congestion and the probability is reduced. Note that no explicit estimation of the drop probability is required.

Proposition 3.8: The mean rate achieved by the TCP-like probability controller PC_s^{TCP} is upper bounded by the steady state rate of probability controller PC_s in (11).

The modified probability controller is similar in structure to the standard TCP controller, which also regulates the rate based on the received $ACKs$. For rate control, the TCP

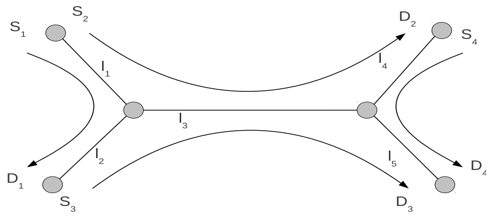

Fig. 2. The network model used for simulation.

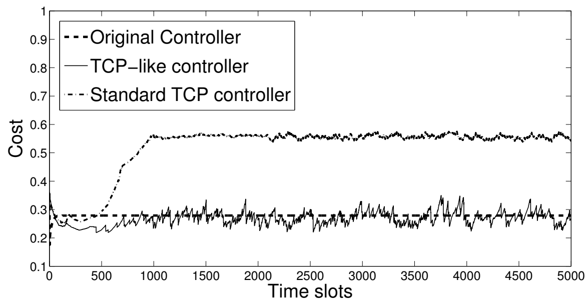

Fig. 3. Estimation costs achieved by various rate controllers.

controller changes the window size whereas the proposed probability controller changes transmission probabilities. The proposed probability controller performs an additive increase in the probability similar to the additive increase of window size in the congestion avoidance phase of TCP. The rate decrease schemes on detecting a packet loss are different in the two cases. The TCP-like form of the probability controller can be easily implemented in current networks due to its resemblance to the TCP controller.

A key difference between the two rate controllers is that TCP involves retransmissions as opposed to no retransmissions in the proposed protocol. This can be attributed to the different end-objectives, i.e. reliability for TCP and estimation performance for the proposed probability controller. Nevertheless, both work in a distributed manner to solve an overall network optimization problem.

F. Simulation Results

Simulations were performed in Matlab to test the protocol performance for the network shown in Fig. 2. There are four source destination pairs and five links in the network. Vector processes evolve at sources S_1 and S_2 and scalar processes evolve at sources S_3 and S_4.

For simulating the packet drops, we use a crude form of the standard RED [6] protocol. In RED protocol, the drop probability on a link is a linear function of the queue size, which depends on the link utilization factor. We assume a M/M/1 queuing model to calculate the queue size.

Fig. 3 shows the temporal variation of the total cost $C(\mathbf{p}(k))$ for the original probability controller PC_s, the TCP-like probability controller PC_s^{TCP} and the standard TCP rate controller. We observe that original controller achieves a steady state minimum cost. The cost achieved by PC_s^{TCP} fluctuates with its mean coinciding with the steady state cost of the original controller. Further, there a big performance margin between probability controllers and the TCP controller. For many choices of system parameters, it was observed that the TCP controller results in an unstable system and the cost becomes infinite while our controller maintained stability. Thus, standard TCP protocol is not well suited for networked estimation and the proposed controller achieves a better performance.

IV. CONCLUSIONS

We studied the problem of rate control for networked estimation in presence of congestion. A stochastic rate control protocol was proposed that optimizes the estimation performance of the network by varying the source transmission probabilities. The protocol was developed using a minimization framework analogous to NUM framework and is scalable for large networks. An approximated controller analogous to the standard TCP controller was also developed.

REFERENCES

[1] A. T. Al-Hammouri, M. S. Branicky, V. Liberatore, and S. M. Phillips, "Decentralized and Dynamic Bandwidth Allocation in Networked Control Systems," IPDPS, 2006.

[2] P. Antsaklis and J. Baillieul (Editors), "Special Issue on Networked Control Systems," IEEE Trans. Automatic Control, 49(9), Sep 2004.

[3] P. Antsaklis and J. Baillieul (Editors), "Special Issue on Networked Control Systems," Proc. of the IEEE, 95(1), Jan 2007.

[4] R. Caceres, N. G. Duffield, J. Horowitz, and D. Towsley, "Multicast-Based Inference of Network-Internal Loss Characteristics," IEEE Transactions on Information Theory, vol. 45, pp. 2462-2480, Nov. 1999.

[5] M. Chiang, S. H. Low, A. R. Calderbank, and J. C. Doyle, "Layering as Optimization Decomposition: A Mathematical Theory of Network Architectures," Proceedings of IEEE, 95(1), Jan. 2007.

[6] S. Floyd, and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance," IEEE/ACM Transactions on Networking, pp. 397-413, Aug. 1993.

[7] E. Garone, B. Sinopoli, and A. Casavola, "Communication Protocols for Optimal Control Over Lossy Networks," Annual Allerton Conference on Communication, Control and Computing, Sept. 2007.

[8] V. Gupta, A. F. Dana, J. Hespanha, R. M. Murray, and B. Hassibi, "Data Transmission over Networks for Estimation and Control," IEEE Transactions on Automatic Control, 54(8):1807-1819, Aug. 2009.

[9] J. P. Hespanha, P. Naghshtabrizi, and Y. Xu, "A Survey of Recent Results in Networked Control Systems," Proc. of IEEE, 95(1):138-162, Jan. 2007.

[10] V. Jacobson, "Congestion avoidance and control," ACM Computer Communication Review, 18:314-329, Aug. 1988.

[11] F. P. Kelly, "Mathematical modeling of the Internet," in Mathematics Unlimited-2001 and Beyond (Editors:B. Engquist and W. Schmid), Berlin:Springer-Verlag, pp. 685-702, 2001.

[12] F. P. Kelly, A. Maulloo, and D. Tan, "Rate control in Communication Networks: Shadow Prices, Proportional Fairness and Stability," Journal of the Operational Research Society, 49:237-252, 1998.

[13] A.S.C. Leong, S. Dey and J.S. Evans, "On Kalman Smoothing with Random Packet Loss," IEEE Transactions on Signal Processing, vol. 56, no. 7, pp. 3346-3351, July 2008.

[14] L. Schenato, B. Sinopoli, M. Franceschetti, K. Poolla, and S. S. Sastry, "Foundations of Control and Estimation Over Lossy Networks." Proceedings of the IEEE, Special Issue on Networked Control Systems, vol. 95, Issue 1, Jan. 2007.

[15] S. Shakkottai, and R. Srikant, "Network Optimization and Control," Foundations and Trends in Networking, 2(3):271-379, Jan. 2007.

[16] "Transmission Control Protocol", RFC-793, Sept. 1981.

[17] W. Zhang, M. S. Branicky, and S. M. Phillips, "Stability of Networked Control Systems," IEEE Control Systems Magazine, Feb. 2001.