Automatica 49 (2013) 1310–1317

Contents lists available at SciVerse ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Brief paper

On a rate control protocol for networked estimation☆

Vaibhav Katewa [1], Vijay Gupta

Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN-46556, USA

ARTICLE INFO

Article history:
Received 8 November 2011
Received in revised form
21 November 2012
Accepted 29 November 2012
Available online 28 February 2013

Keywords:
Networked estimation
Congestion control
Rate control
Network utility maximization
Stability with delays

ABSTRACT

We study the problem of 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. This decentralized protocol avoids congestion by regulating the transmission probabilities of the sources. The presence of estimation costs poses new challenges; however, for low congestion levels, the form of rate controller resembles that of the standard TCP rate controller. Stability of the protocol is analyzed in the presence of fixed network delays.

1. Introduction

The architecture and protocols in a communication network should ideally depend on the objectives of the end users. Traditionally, such networks were used with the sole goal of reliable data transfer. More recently, such networks have been proposed to be used in control and estimation applications in the so-called Networked Control Systems (see, e.g., the special issue Antsaklis & Baillieul, 2007 and the references therein). In such applications, the performance metric is a complicated function of delay, throughput, and reliability; hence, traditional network protocols may not be suitable. For both the cases when the communication network is designed specifically for estimation or control, and when the communication network is shared with data unrelated to such applications, it is of interest to design network protocols that optimize the performance relevant to these applications.

However, most of the research in Networked Control Systems so far 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 control systems sharing a common network, and further, design network protocols more suitable for estimation and control (Garone et al., 2007; Schenato et al., 2007).

In this paper, we focus on a 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. Such a capacity constrained network 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 such as TCP may not be suitable for optimizing estimation performance, and propose a new distributed rate control protocol that can co-exist with existing rate control protocols.

The problem of congestion control has been well studied for communication networks (see, e.g., Jacobson, 1988). TCP (RFC, 1981) is the most widely used congestion control protocol on the Internet. While originally an engineering heuristic, TCP has now been reverse engineered to show that it is a distributed solution that optimizes a particular utility function (Kelly, 2001). The chief tool in this regard is the Network Utility Maximization (NUM) framework (Kelly et al., 1998) which transforms the end objective to an optimization problem with constraints. The communication protocols are the distributed solutions to these optimization problems (Chiang et al., 2007).

☆ Research was funded in part by NSF awards 0834771 and 0846631. The American Control Conference (ACC2011), June 29–July 1, 2011, San Francisco, California, USA. This paper was recommended for publication in revised form by Associate Editor Huijun Gao under the direction of Editor Ian R. Petersen.

E-mail addresses: vkatewa@nd.edu (V. Katewa), vgupta2@nd.edu (V. Gupta).

[1] Tel.: +1 5746311136; fax: +1 5746314393.

0005-1098/$ – see front matter © 2013 Elsevier Ltd. All rights reserved.
doi:10.1016/j.automatica.2013.01.050

The primary aim of traditional TCP is reliable transfer of data, even at the expense of delays. For estimation and control, it may be more useful to have a lower reliability, but a higher throughput. Moreover, not all processes need to transmit data at the same rate to achieve the same estimation error covariance. Thus, issues such as fairness relevant to traditional TCP may not be applicable. In fact, using TCP for estimation purposes may result in instability of the estimation error covariance. Because of these reasons, designing an estimation oriented rate control protocol is not simply a matter of substituting the estimation error covariance as a cost function instead of the throughput. Our proposed protocol, while sharing the formal structure of TCP protocols, considers these issues directly. The proposed protocol is implemented at the transport layer of the standard OSI layer stack, and thus, preserves the layered structure of the network.

To ensure that the proposed protocol can coexist with the standard TCP, 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 Al-Hammouri et al. (2006) 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 Al-Hammouri et al. (2006).

We also come up with conditions on network delay and system parameters for which the original protocol remains stable. The delays can be time varying in realistic networks. However, we analyze the stability of the system with fixed delays for tractability. Although it is a special case, fixed delay analysis is important and has a rich history for standard TCP (Chiang et al., 2007; Johari & Tan, 2001; Low & Lapsley, 1999; Vinnicombe, 2002).

The main contributions of the paper are as follows

- We propose a probabilistic rate control strategy and evaluate an estimation error measure.
- Using the NUM framework, we obtain a scalable rate control protocol that allocates rates optimally such that an estimation error metric is minimized.
- The protocol is developed in primal form and we show that under low network congestion, it resembles the structure of the standard TCP protocol.

The rest of the paper is organized as follows. In the next section, we describe the problem setting with random delays and formulate an optimization problem. In Section 3, we propose a distributed solution to the problem using the NUM framework and present our analysis results. In Section 4, we obtain conditions under which the network is stable for fixed delays and present simulation results. We conclude in Section 5.

2. Problem formulation

Network and process setting: Consider the problem set up shown in Fig. 1. Let all the sources form the source set $\mathcal{S}$. With every source $s \in \mathcal{S}$, associate a unique destination d and denote the destination set by $\mathcal{D}$. Let every source be connected to its corresponding destination through a shared capacity constrained network $\mathcal{N}$. We model the network as a graph, wherein the end-nodes are the sources and the destinations, the intermediate nodes are routers that forward packets and 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 average. Any individual link may be shared by one or more sources.

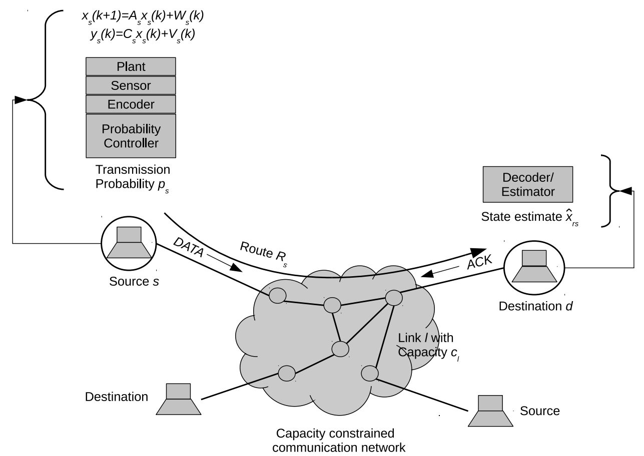


Fig. 1. The problem setup considered. Multiple processes are remotely estimated across a shared communication network.

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 P_s evolves according to the discrete-time linear model

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

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 the relation

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

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 transmission data and sends 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. The data from ENC_s is received at the corresponding destination possibly with a stochastic delay τ_{sd} which models the transmission delay. Each destination comprises of a decoder DEC_d, that uses the received data to generate a state estimate that is optimal in the minimum mean squared error (MMSE) sense. We ignore any queuing delays in the network and assume the existence of a time stamp for every transmitted packet. When a destination receives a packet, it sends back an acknowledgment (ACK) to the corresponding source. We assume that $ACKs$ are never lost in the network.

We employ the encoder and decoder scheme described in Gupta et al. (2009). At source s, denote the local estimate of state $x_s(k)$ given the measurements $\{y_s(j)\}_{j=1}^{k}$ by $\hat{x}_s(k)$. Further, denote the remote 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.
  - · Transmit $\hat{x}_s(k)$ along with the time stamp k.
- DEC_d:
  - · If $k = 0$, set the stored time stamp $t_d = -1$.
  - · If DEC_d receives a packet in time slot k, extract the time stamp k' from the packet.

(1) If $k' \le t_d$, ignore the old packet and set $\hat{x}_{rs}(k) = A_s\hat{x}_{rs}(k-1)$.
(2) If $k' > t_d$, set $\hat{x}_{rs}(k) = A_s^{k-k'}\hat{x}_s(k')$ and set $t_d = k'$.

· If DEC_d does not receive a packet in time slot k, set $\hat{x}_{rs}(k) = A_s\hat{x}_{rs}(k-1)$.

As discussed in Gupta et al. (2009), this encoder–decoder structure is optimal amongst all causal structures.

Communication scheme: We consider a stop-and-wait type communication protocol. In any time slot k, the source s transmits the local estimate $\hat{x}_s(k)$ to the corresponding destination d. The transmission is stochastic with the transmission probability $p_s(k)$ at time slot k. The transmission events at different time slots are assumed to be 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, which controls the source rate. 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 the routers in the network. 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. In such a situation, all the packets are dropped by the router with a probability approaching 1. To avoid such instances, the routers use queue management protocols such as Random Early Detection (RED) protocol (Floyd & Jacobson, 1993). In RED, the 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 protocol, the link drop probability is a pre-specified increasing and convex function of the total rate (assuming, say, a M/M/1 queue model).

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 independent. Further, the packet drop and packet transmission events on route R_s are assumed to be independent for every time slot. Using the standard assumption (see e.g., Caceres et al., 1999) that the drop events on various links are independent, the drop probability $d_s(k)$ on route R_s as observed by the destination d can be expressed as

$$d_s(k) = 1 - \prod_{l\in L(s)} (1 - d_l(k-\tau_{ld})), \tag{3}$$

where τ_{ld} denotes the forward delay between link l and destination d. Thus, $d_s(k)$ depends on the rates of the sources that share the links with source s. This introduces a coupling to the problem. Note that d_s may not be a convex function of the source rates.

Remark 1 (*Stochastic Rate Control*)**.** The stochastic transmission scheme that we propose controls the source rate by varying the transmission probability. This is in contrast to deterministic schemes, wherein the sources send the information at deterministic instants and rate control is achieved by varying the time interval between the transmissions. A stochastic transmission scheme is a natural choice since a congested network drops packets stochastically. Therefore, the information reception process is inherently probabilistic. We superimpose an additional stochastic transmission process on the stochastic network, still resulting in a stochastic reception process.

Remark 2 (*Instantaneous Behavior*)**.** 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 average, the network will be utilized in an optimal manner.

For the 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 $F_s(k)$, respectively. 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. According to the decoder structure DEC_s, the remote estimation error covariance $F_s(k)$ evolves as

$$F_s(k) = \begin{cases} A_s^{\tau_{sd}} M_s(k-\tau_{sd}) A_s^{\tau_{sd},T} + \sum_{i=0}^{\tau_{sd}-1} A_s^i Q_s A_s^{i,T} \\ \quad \text{if a packet is received} \\ A_s F_s(k-1) A_s^T + Q_s, \quad \text{otherwise,} \end{cases}$$

where, $A^{\tau,T} \triangleq (A^\tau)^T$. Thus, $F_s(k)$ is a random variable. As a performance metric, we consider its expected value, that evolves as

$$\begin{aligned}\mathbb{E}[F_s(k)] = \mathbb{E}\Bigg[& p_s(k-\tau_{sd})(1-d_s(k)) \Bigg(A_s^{\tau_{sd}} M_s(k-\tau_{sd}) A_s^{\tau_{sd},T} \\ & + \sum_{i=0}^{\tau_{sd}-1} A_s^i Q_s A_s^{i,T} \Bigg) + (1 - p_s(k-\tau_{sd})(1-d_s(k))) \\ & \times (A_s \mathbb{E}[F_s(k-1)] A_s^T + Q_s) \Bigg]\end{aligned} \tag{4}$$

where $p_s(k-\tau_{sd})(1-d_s(k))$ is the packet reception probability and the expectation is taken with respect to the packet transmission process, packet drop process and delays in the network. Under the assumption that the system reaches a steady state, (4) can be written as

$$\begin{aligned}F_s(p_s, d_s) = {} & p_s(1-d_s) \left(\mathbb{E}\left[A_s^{\tau_{sd}} M_s A_s^{\tau_{sd},T} + \sum_{i=0}^{\tau_{sd}-1} A_s^i Q_s A_s^{i,T} \right] \right) \\ & + (1 - p_s(1-d_s)) \, (A_s F_s(p_s, d_s) A_s^T + Q_s),\end{aligned} \tag{5}$$

where p_s, d_s and $F_s(p_s, d_s)$ denote the steady state values of $p_s(k)$ and $d_s(k)$ and $\mathbb{E}[F_s(k)]$, respectively.

Problem statement: Let $\mathbf{p}$ denote the vector of all steady state transmission probabilities, i.e. $\mathbf{p} = (p_1, p_2, \ldots, p_{|\mathcal{S}|})^T$, where $|\mathcal{S}|$ denotes the cardinality of set $\mathcal{S}$. We consider the estimation cost incurred for the source s as $c_s = \mathrm{tr}(F_s(p_s, d_s))$, where tr denotes the trace. Further, the total cost of the system $C_{sys}(\mathbf{p})$ is chosen to be the sum of individual costs. For ease of notation, we will denote $\{\mathrm{tr}(F_s(p_s, d_s)), \mathrm{tr}(M_s), \mathrm{tr}(A_s A_s^T), \mathrm{tr}(Q_s)\}$ by $\{f_s(p_s, d_s), m_s, a_s, q_s\}$, respectively. Thus,

$$C_{sys}(\mathbf{p}) = \sum_{s\in\mathcal{S}} f_s(p_s, d_s)$$

$$\begin{aligned}f_s(p_s, d_s) = {} & p_s(1-d_s)\mathrm{tr}\left(\mathbb{E}\left[A_s^{\tau_{sd}} M_s A_s^{\tau_{sd},T} + \sum_{i=0}^{\tau_{sd}-1} A_s^i Q_s A_s^{i,T} \right] \right) \\ & + (1 - p_s(1-d_s))(\mathrm{tr}(A_s F_s(p_s, d_s) A_s^T) + q_s).\end{aligned} \tag{6}$$

The problem is to find the optimal value of $\mathbf{p}$ which minimizes the cost function $C_{sys}(\mathbf{p})$ under the rate constraints. 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 ensure that the solution is scalable for large networks.

3. Analysis and results

Cost function: The following upper and lower bounds for the cost follow from algebraic manipulations on (6).

Lemma 3. *The steady state value $f_s(p_s, d_s)$ satisfies $f_s^l(p_s, d_s) < f_s(p_s, d_s) < f_s^u(p_s, d_s)$, where*

$$f_s^u(p_s, d_s) \triangleq \frac{p_s(1-d_s)m_s^u + (1-p_s(1-d_s))q_s}{1-a_{s,\max}(1-p_s(1-d_s))} \quad (7)$$

$$f_s^l(p_s, d_s) \triangleq \frac{p_s(1-d_s)m_s^l + (1-p_s(1-d_s))q_s}{1-a_{s,\min}(1-p_s(1-d_s))} \quad (8)$$

$$m_s^u = \begin{cases} \left(m_s + \frac{q_s}{a_{s,\max}-1}\right)\mathbb{E}[a_{s,\max}^{\tau_{sd}}] - \frac{q_s}{a_{s,\max}-1} \\ \quad \text{if } a_{s,\max} \neq 1, \\ m_s + q_s\mathbb{E}[\tau_{sd}] \quad \text{otherwise,} \end{cases}$$

$$m_s^l = \begin{cases} \left(m_s + \frac{q_s}{a_{s,\min}-1}\right)\mathbb{E}[a_{s,\min}^{\tau_{sd}}] - \frac{q_s}{a_{s,\min}-1} \\ \quad \text{if } a_{s,\min} \neq 1, \\ m_s + q_s\mathbb{E}[\tau_{sd}] \quad \text{otherwise,} \end{cases}$$

where $\lambda(A)$ denotes the eigenvalues of A, $a_{s,\max} = \lambda_{\max}(A_sA_s^T)$ and $a_{s,\min} = \lambda_{\min}(A_sA_s^T)$.

Proof. From (6), we have,

$$\begin{aligned} f_s(p_s, d_s) &= p_s(1-d_s)\left(\mathbb{E}\left[\operatorname{tr}(A_s^{\tau_{sd},T}A_s^{\tau_{sd}}M_s)\sum_{i=0}^{\tau_{sd}-1}\operatorname{tr}(A_s^{i,T}A_s^iQ_s)\right]\right) \\ &\quad + (1-p_s(1-d_s))(\operatorname{tr}(A_s^TA_sF_s(p_s, d_s)) + q_s) \\ &\leq p_s(1-d_s)\left(\mathbb{E}\left[m_sa_{s,\max}^{\tau_{sd}} + q_s\sum_{i=0}^{\tau_{sd}-1}a_{s,\max}^i\right]\right) \\ &\quad + (1-p_s(1-d_s))(a_{s,\max}f_s(p_s, d_s) + q_s) \end{aligned}$$

where we have used the following trace identities:

(1) $\operatorname{tr}(ABC) = \operatorname{tr}(CAB)$,
(2) $\operatorname{tr}(\mathbb{E}[X]) = \mathbb{E}[\operatorname{tr}(X)]$, and
(3) $\operatorname{tr}(M)\lambda_{\min}^k(AA^T) \leq \operatorname{tr}(M)\lambda_{\min}(A^kA^{k,T}) \leq \operatorname{tr}(A^kA^{k,T}M) \leq \operatorname{tr}(M)\lambda_{\max}(A^kA^{k,T}) \leq \operatorname{tr}(M)\lambda_{\max}^k(AA^T)$, for any positive semi definite matrix M.

Simplifying and rearranging the last inequality, we get the desired upper bound. The lower bound can be obtained in a similar way, thus completing the proof. □

In particular, for scalar processes, the upper and lower bounds in (7) and (8) are satisfied with equality. For analytical tractability, we replace f_s by 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 under the vector $\mathbf{p}$.

Lemma 4. *A sufficient condition for the convergence of $\mathbb{E}[F_s(k)]$ as (4) evolves is given by*

$$p_s(1-d_s) \geq \left(1 - \frac{1}{\rho^2(A_s)}\right)^+ \triangleq p_s^{\min}, \quad (9)$$

$$\mathbb{E}[a_{s,\max}^{\tau_{sd}}] < \infty \quad \text{if } a_{s,\max} \neq 1, \quad (10)$$
$$\mathbb{E}[\tau_{sd}] < \infty \quad \text{otherwise},$$

where $\rho(X)$ denotes the spectral radius of matrix X.

Proof. See Gupta et al. (2009) for condition (9). Condition (10) can be obtained from (7) in a straightforward manner. □

Thus, we have the following optimization problem

$$SYSTEM : \min_{\mathbf{p}} \sum_{s\in\mathcal{S}} f_s^u(p_s, d_s(\mathbf{p})),$$

$$\text{s.t.} \sum_{s:l\in R_s} p_s \leq c_l, \quad \forall l \in \mathcal{L},$$

$$p_s(1-d_s(\mathbf{p})) \geq p_s^{\min}, \quad \forall s \in \mathcal{S},$$
$$p_s \geq 0, \qquad p_s \leq 1 \quad \forall s \in \mathcal{S},$$

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, this approach is not desirable for many reasons:

(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) The method 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 $\{d_s : s \in (\mathcal{S})\}$ and $\{p_s : s \in (\mathcal{S})\}$, which may be unavailable in a practical scenario.

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

Posing the problem in the NUM framework: To obtain a scalable and distributed solution, we employ a network cost minimization framework that is analogous to the primal formulation of the Network Utility Maximization framework (Shakkottai & Srikant, 2007).

Remark 5 (*Advantage of the Primal Form*)**.** Since the communication network may also 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 needs to be done only at the source node, which is aware of estimation application.

The NUM framework imposes some requirements on the costs. The costs should be separable among the sources. In other words, the cost associated with source s should depend only on the resource p_s. Moreover, the cost should be positive, monotonically decreasing and convex. However, the costs $\{f_s^u(p_s, d_s) : s \in \mathcal{S}\}$ in (7) are coupled among each other through the drop probabilities d_s and hence are neither separable nor convex. Therefore, we eliminate d_s from the costs and let this modified separable cost be denoted by $f_s^u(p_s, 0)$. To include the effect of the drop probabilities, we define a barrier of the form $B_l\left(\sum_{s:l\in R_s} p_s\right)$ corresponding to each link l, and add it to the total cost. The barrier maps the congestion level in the link to an additive cost to the system. Thus, we obtain the following relaxation of the *SYSTEM* problem

$$USER : \min_{\mathbf{p}} \sum_{s\in\mathcal{S}} f_s^u(p_s, 0) + \sum_{l\in\mathcal{L}} B_l\left(\sum_{s:l\in R_s} p_s\right),$$

$$\text{s.t.} \sum_{s:l\in R_s} p_s \leq c_l, \quad \forall l \in \mathcal{L}, \quad (11)$$

$$p_s \geq p_s^{\min} \geq 0, \quad \forall s \in \mathcal{S}, \quad (12)$$

$$p_s \leq 1 \quad \forall s \in \mathcal{S}. \quad (13)$$

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. 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 6. *The cost function $f_s^u(p_s, 0)$ is positive, monotonically decreasing and convex for $p_s > 1 - \frac{1}{a_{s,\max}}$.*

Proof. The proof follows by differentiating $f_s^u(p_s, 0) = \frac{p_s m_s^u + (1-p_s) q_s}{1 - a_{s,\max}(1-p_s)}$ twice and verifying that the terms in numerator and denominator are of appropriate signs. □

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

$$B_l\left(\sum_{s:l\in R_s} p_s\right) \triangleq \int_0^{\sum_{s:l\in R_s} p_s} t_l(x)dx, \tag{14}$$

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 (17). Finally, we have the following result.

Lemma 7. *The cost used in the problem USER implicitly guarantees the constraints (11) and (12).*

Proof. The cost $f_s^u(p_s, 0)$ is positive and finite iff $p_s > 1 - \frac{1}{a_{s,\max}}$. Since $a_{s,\max} = \lambda_{\max}(A_s A_s^T) \geq \rho^2(A_s)$, $f_s^u(p_s, 0)$ is positive and finite only for $p_s > 1 - \frac{1}{a_{s,\max}} \geq 1 - \frac{1}{\rho^2(A_s)}$. Thus, the cost $f_s^u(p_s, 0)$ becomes infinite when p_s approaches $p_s^{\min}$. Further, the barrier function B_l on link l rapidly increases as the total rate on the link approaches the link capacity, thereby increasing the cost function. Thus, both (11) and (12) are satisfied. □

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 controller of the form

$$PC_s : p_s(k+1) = p_s(k) - k_s\left(\frac{d}{dp_s} f_s^u(p_s, 0) + \sum_{l:l\in L(s)} t_l\left(\sum_{s:l\in R_s} p_s\right)\right), \tag{15}$$

with $k_s > 0$ being sufficiently small step size. The quantity

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

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 8 (*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 provided implicitly or explicitly by the network through *ACKs* from the destination to the source.

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*. We observe here that the congestion in the network affects the system performance through the drop probabilities. Since drop probabilities have a direct effect on the system performance, we choose a penalty function that depends on the 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

$$t_l\left(\sum_{s:l\in R_s} p_s\right) = -\log\left(1 - d_l\left(\sum_{s:l\in R_s} p_s\right)\right), \tag{16}$$

wherein the link drop probability depends on the total rate on the link. Note that t_l is positive and monotonically increases to infinity as the total rate on the corresponding link approaches its capacity; thus the barrier function is indeed convex as required. In fact, the barrier function is

$$B_l\left(\sum_{s:l\in R_s} p_s\right) = \int_0^{\sum_{s:l\in R_s} p_s} -\log(1 - d_l(x))dx. \tag{17}$$

Also, the route price is given by

$$q_{R_s} = \sum_{l:l\in L(s)} -\log(1 - d_l) = -\log\left(\prod_{l:l\in L(s)} (1 - d_l)\right) = -\log(1 - d_s).$$

Remark 9 (*Estimating the Route Price*)**.** The advantage of choosing a logarithmic penalty function is apparent 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 merely based on whether *ACKs* are received or not.

Note that the different choices of the penalty/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. We do not claim that the particular choice we have proposed provides the best performance in all the cases. Other choices may be beneficial depending on the system and application.

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 $f_s^u(p_s, 0)$ (analogous to increasing the barrier function) by a constant β_s to satisfy this property. We choose $\beta_s = N_s(q_s - m_s^u(1 - a_{s,\max}))$, 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 and delays are large. Thus, it acts like a normalization factor to the estimation error covariance. With this relaxation, the optimization problem becomes

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

and the probability controller becomes

$$PC_s : p_s(k+1) = p_s(k) + k_s'\left(\frac{1}{(1 - a_{s,\max}(1 - p_s(k)))^2} + N_s \log(1 - d_s(k))\right), \tag{18}$$

where $k_s' = \frac{k_s}{N_s}$.

Modified TCP-like probability controller: The probability controller structure in (18) can be implemented using a TCP-like structure under low network congestion conditions. In this regime, the route drop probabilities are also low, $\{d_s \ll 1, s \in \mathcal{S}\}$ which implies that $-\log(1 - d_s) \approx d_s$. Thus, (18) becomes

$$PC_s : p_s(k+1) = p_s(k) + k_s'\left(\frac{1}{(1 - a_{s,\max}(1 - p_s(k)))^2} - N_s d_s(k)\right). \tag{19}$$

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 $p_s(k+1) = p_s(k) + k'_s$.
- If a packet is transmitted and *ACK* is not received, then set $p_s(k+1) = p_s(k) - k'_s(N_s(1 - a_{s,\max}(1 - p_s(k)))^2 - 1)$.

Proposition 10. *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 (19).*

Proof. The mean rate achieved by the TCP-like probability controller PC_s^{TCP} (where the expectation is taken with respect to the transmission and drop processes) is given by

$$\begin{aligned}\mathbb{E}[p_s(k+1)] &= \mathbb{E}[p_s(k)] \\ &\quad + \lambda k'_s\left(\frac{1}{\mathbb{E}\left[(1 - a_{s,\max}(1 - p_s(k)))^2\right]} - N_s d_s(k)\right) \\ &\le \mathbb{E}[p_s(k)] \\ &\quad + \lambda k'_s\left(\frac{1}{\left(1 - a_{s,\max}(1 - \mathbb{E}[p_s(k)])\right)^2} - N_s d_s(k)\right). \qquad (20)\end{aligned}$$

The mean rate obtained by PC_s^{TCP} is thus upper-bounded by a probability controller similar to that in (19), except the scaling factor $\lambda > 0$. □

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 controller changes the window size whereas the proposed probability controller changes transmission probabilities. Thus, the proposed 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 protocols solve an overall network optimization problem in a distributed manner.

4. Stability with delays in the network

We now consider the effect of network delays on the stability of the proposed probability controllers. For tractability, we assume that the delays are constant. Let the delay in the forward direction between source s and link l be denoted by τ_{sl}^f. Further, let the delay in backward direction between link l and source s via the corresponding destination d be denoted by τ_{sl}^b. Both the forward and backward delays are assumed to be positive integers. We assume that the total round trip time w.r.t. link l is constant for every link in the route, i.e. $\tau_s = \tau_{sl}^f + \tau_{sl}^b\ \forall l$. Let $R = [r_{ij}]$ denote the $|\mathcal{L}| \times |\mathcal{S}|$ routing matrix, where

$$r_{ij} = \begin{cases}1 & \text{if source } j \text{ uses link } i (i \in R_j)\\ 0 & \text{otherwise.}\end{cases}$$

Further, let y_l denote the aggregate rate on link l

$$y_l(k) = \sum_{s: l \in R_s} p_s(k - \tau_{sl}^f) = \sum_s r_{ls} p_s(k - \tau_{sl}^f). \qquad (21)$$

The penalty function t_l at link l depends on y_l through the relation

$$t_l(k) = h_l(y_l(k)), \qquad (22)$$

where t_l is the penalty function as denoted in (14) and h_l is a positive non-decreasing function. The route price q_s associated with route R_s can be written as

$$q_s(k) = \sum_{l: l \in L(s)} t_l(k - \tau_{sl}^b) = \sum_l r_{ls} t_l(k - \tau_{sl}^b). \qquad (23)$$

At the source s, probability controller updates $p_s(k)$ using the relation

$$p_s(k+1) = g_s(p_s(k), q_s(k)), \qquad (24)$$

where g_s is the nonlinear function as described in (18). The Eqs. (21)–(24) form a nonlinear feedback system. The presence of delays can make the network unstable. We wish to characterize the local asymptotic stability of the network around the equilibrium point and obtain conditions under which stability is guaranteed. We proceed by linearizing the system of equations around the equilibrium point. Denote the vector of source rates by p, the vector of aggregate rates of all links by y, the vector of link penalties by t and the vector of route prices by q. Let $\{p_s, y_l, t_l, q_s\}$ denote the equilibrium values for $\{p_s(k), y_l(k), t_l(k), q_s(k)\}$. Further, let $p_s(k) = p_s + \delta p_s(k), y_l(k) = y_l + \delta y_l(k), t_l(k) = t_l + \delta t_l(k), q_s(k) = q_s + \delta q_s(k)$ be small perturbations around the equilibrium point. Linearizing (21)–(24), we obtain

$$\delta y_l(k) = \sum_s r_{ls}\delta p_s(k - \tau_{sl}^f), \qquad (25a)$$

$$\delta t_l(k) = h'_l(y_l)\delta y_l(k), \qquad (25b)$$

$$\delta q_s(k) = \sum_l r_{ls}\delta t_l(k - \tau_{sl}^b), \qquad (25c)$$

$$\delta p_s(k+1) = \alpha_s \delta p_s(k) + \beta_s \delta q_s(k), \quad \text{where} \qquad (25d)$$

$$\alpha_s = \frac{\partial}{\partial p} g_s(p, q)|_{p_s, q_s}, \qquad \beta_s = \frac{\partial}{\partial q} g_s(p, q)|_{p_s, q_s}.$$

For the source law in (18), we have

$$\alpha_s = 1 - \frac{2k'_s a_{s,\max}}{[1 - (a_{s,\max}(1 - p_s))]^3}, \quad \text{and} \quad \beta_s = -k'_s N_s.$$

Denote by $\{y, t, q, p\}$ the vectors of aggregate rates, penalty functions, route prices and transmission probabilities. Taking z transform of (25) and combining the variables in vector form we obtain

$$\begin{aligned}&\delta y(z) = R^f \delta p(z), \qquad \delta t(z) = F\delta y(z),\\ &\delta q(z) = R^b \delta t(z), \qquad z\delta p(z) = \alpha\delta p(z) + \beta \delta q(z), \quad \text{where}\\ &R^f(z) = [r_{ij}^f(z)], \qquad r_{ij}^f(z) = r_{ij} z^{-\tau_{ji}^f},\\ &R^b(z) = [r_{ij}^b(z)], \qquad r_{ij}^b(z) = r_{ji} z^{-\tau_{ij}^b},\\ &F = \text{diag}(h'_l(y_l)), \qquad \alpha = \text{diag}(\alpha_s), \qquad \beta = \text{diag}(\beta_s).\end{aligned}$$

Thus, the overall return ratio of the linearized system as seen by the sources becomes

$$\begin{aligned}&T(z) = (zI - \alpha)^{-1}\beta R^b F R^f = [T_{ij}(z)], \qquad (26)\\ &T_{ij}(z) = \beta_i(z - \alpha_i)^{-1}\sum_l r_{li} r_{lj} h'_l z^{-(\tau_{jl}^f + \tau_{il}^b)}.\end{aligned}$$

Theorem 11. *The system described by Eqs. (21)–(24) is locally asymptotically stable if the following conditions are satisfied*

$$k'_s < \min\left\{\frac{[1 - (a_{s,\max}(1 - p_s))]^3}{a_{s,\max}}, \frac{2\sin\left(\frac{\pi}{2(2\tau_s+1)}\right)}{N_s} \times \sum_j \sum_k r_{ki} r_{kj} h'_k(y_k)\right\} \quad \forall s \in \mathcal{S}, \text{ and}$$

$$-1 \notin Co\left(\left\{2\sin\left(\frac{\pi}{2(2\tau_s + 1)}\right)(e^{j\omega} - \alpha_s)^{-1} e^{-j\omega\tau_s}\right\}\right),$$

where Co denotes the convex hull.

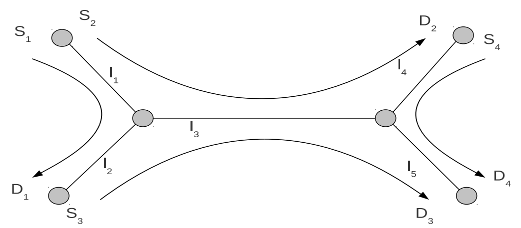


Fig. 2. The network model used for simulation.

Proof. Denote the spectrum of a square matrix Z by $\sigma(Z)$. Using $R^b(z) = \text{diag}\{z^{-\tau_s}\}R^{f,T}(z^{-1})$ and the properties of similar and diagonal matrices, we can show $\sigma(T(z)) = \sigma(A(z)B(z))$, where,

$$A(z) = \text{diag}\left\{\sqrt{\frac{\beta_s}{w_s}}\right\} R^{f,T}(z^{-1})FR^f(z)\text{diag}\left\{\sqrt{\frac{\beta_s}{w_s}}\right\},$$
$$B(z) = \text{diag}\{w_s(z-\alpha_s)^{-1}z^{-\tau_s}\},$$

and $w_s = 2\sin\left(\frac{\pi}{2(2\tau_s+1)}\right) > 0$. Assuming that the system is open loop stable and using the generalized Nyquist stability criterion (Desoer & Wang, 1980), the system is stable if the eigenloci of $T(z = e^{j\omega}), \omega \in [0, \pi]$ do not cross the real axis to the left of -1. For open loop stability we should have $|\alpha_s| < 1$, which (since $\alpha_s < 1$) is equivalent to

$$k_s < \frac{[1-(a_{s,\max}(1-p_s))]^3}{a_{s,\max}}. \tag{27}$$

Assume that λ is an eigenvalue and v is the corresponding normalized eigenvector of $A(e^{j\omega})B(e^{j\omega})$. Then, $\lambda v = A(e^{j\omega})B(e^{j\omega})v$ or $\lambda = v^*A(e^{j\omega})B(e^{j\omega})v$, where v^* denotes the conjugate transpose of v. Since $A(e^{j\omega}) = A^T(e^{-j\omega}) > 0$, we have (Vinnicombe, 2002)

$$\lambda \subset \rho(A(e^{j\omega}))\, Co\left(\left\{w_s(e^{j\omega}-\alpha_s)^{-1}e^{-j\omega\tau_s}\right\}\right), \tag{28}$$

where ρ is the spectral radius. Since the spectral radius is upper bounded by the maximum absolute row sum,

$$\rho(A(e^{j\omega})) \le \max_{s\in\mathcal{S}} \sum_j \left\| \sum_k r_{ki}r_{kj}h'_k(y_k)\left(\frac{\beta_s}{w_s}\right)e^{-j\omega(\tau^f_{jk}-\tau^f_{ik})}\right\|$$
$$\le \max_{s\in\mathcal{S}} \frac{k'_sN_s}{w_s}\sum_j\sum_k r_{ki}r_{kj}h'_k(y_k) \overset{(a)}{\le} 1, \tag{29}$$

where (a) follows from the theorem statement. The result follows from (27) to (29). □

Simulation results: Simulations were performed in Matlab to test the protocol performance. Consider 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.

The process parameters are chosen arbitrarily as $\{A_1, C_1, Q_1, R_1\} = \left\{\begin{bmatrix}0.5 & 0.6\\1.1 & 0.1\end{bmatrix}, [1 \quad 1], \begin{bmatrix}1 & 0\\0 & 2\end{bmatrix}, 3\right\}$, $\{A_2, C_2, Q_2, R_2\} = \left\{\begin{bmatrix}1 & 0.5\\0.7 & 1\end{bmatrix}, [1 \quad 1], \begin{bmatrix}2.5 & 0\\0 & 1.5\end{bmatrix}, 2\right\}$, $\{A_3, C_3, Q_3, R_3\} = \{1.2, 1, 3.5, 3\}$, and $\{A_4, C_4, Q_4, R_4\} = \{1.1, 1, 2.5, 1.5\}$. The link capacities are $\{c_1, c_2, c_3, c_4, c_5\} = \{1.5, 1.6, 1.8, 1.7, 1.4\}$, the step size $k'_s = 0.001$ and $N_s = 100$. The delays on the links are $\{d_1, d_2, d_3, d_4, d_5\} = \{1, 2, 2, 3, 4\}$.

For simulating the packet drops, we use a crude form of the standard RED protocol. Let μ be the link utilization factor, which is the ratio of total rate on an link to the link capacity. In the RED protocol, the drop probability on a link is a linear function of the queue size, which depends on the link utilization factor.

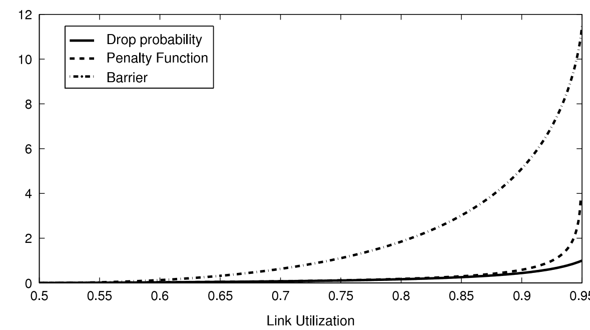


Fig. 3. Link drop probability, penalty function and barrier for the RED scheme.

We assume a M/M/1 queuing model to calculate the queue size. Let $\{\mu_{\min}, \mu_{\max}\}$ denote the link utilization extremes and let $\{N_{\min}, N_{\max}\}$ denote the corresponding queue sizes. Then the link drop probability varies as

$$d_l = \begin{cases} 0 & \text{if } N < N_{\min}, \\ \dfrac{N-N_{\min}}{N_{\max}-N_{\min}} & \text{if } N_{\min} \le N \le N_{\max}, \\ 1 & \text{if } N > N_{\max}, \end{cases}$$

where $N = \frac{\mu}{1-\mu}$ is the queue size. The values of $\{\mu_{\min}, \mu_{\max}\}$ are $\{0.5, 0.95\}$. We assume that the route drop probabilities are known to the sources.

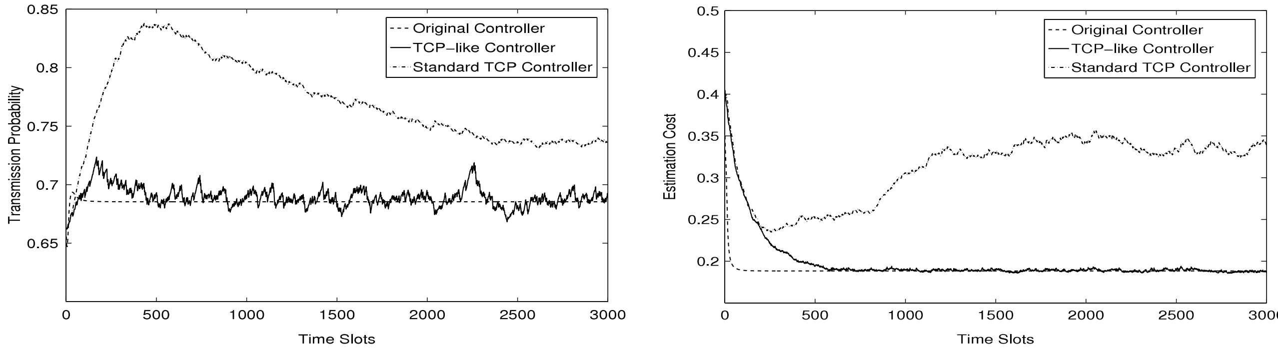
Fig. 3 shows the link drop probability d_l, penalty function t_l and the scaled barrier β_sB_l as a function of link utilization factor μ for a link that implements the RED algorithm. For low rates, there are no drops. As the drop probability increases from 0 to 1, the penalty function becomes infinite. The barrier is scaled so that the congestion cost becomes large for large μ. All the three curves are positive, monotonically increasing and convex. Further, the penalty function is approximately equal to the drop probability for low values of link utilization.

Fig. 4 shows the temporal variation of the transmission probability of the second source ($p_2(k)$) and the *USER* cost $C(\mathbf{p}(k))$ for the original probability controller PC_s (18), the TCP-like probability controller PC_s^{TCP} and the standard TCP rate controller. The transmission probabilities of the other sources also vary is a similar way and are omitted for clarity. We observe that PC_s achieves a steady state minimum cost of 0.1884 for the optimal transmission probability vector $\mathbf{p}_{\mathbf{USER}} = [0.51, 0.69, 0.49, 0.43]$. It can be verified that the system parameters satisfy the conditions of Theorem 11 and hence the overall system is stable. We also performed an exhaustive numerical search over the variable $\mathbf{p}$ to find the solution to the *USER* problem. This exhaustive search yields the minimum value of the cost as 0.1863 which is quite close to the cost achieved by PC_s.

Similarly, an exhaustive numerical search over the variable $\mathbf{p}$ to find the solution of the *SYSTEM* problem yields the minimum value of the cost as 0.41 which is achieved by $\mathbf{p}_{\mathbf{SYS}} = [0.58, 0.62, 0.51, 0.46]$. We compare this with the *SYSTEM* cost achieved by $\mathbf{p}_{\mathbf{USER}}$, which is 0.47. We can observe that the solutions achieved by the proposed protocol for the *USER* problem is close to the optimal solution of the *SYSTEM* problem, thus verifying the approximation of the latter by the former.

Further, we observe from Fig. 4 that the cost achieved by PC_s^{TCP} fluctuates slightly with time due to its structure. More importantly, we can notice that the mean cost and the transmission probability achieved by PC_s^{TCP} coincides with the steady state cost and transmission probability of the original controller. This provides empirical evidence that under low network congestion conditions the TCP-like probability controller well approximates the original controller.

Moreover, as seen in Fig. 4, there is a big performance margin between the proposed controller and the TCP controller. This is



(a) Transmission probabilities.

(b) Estimation cost.

Fig. 4. Transmission probability and estimation costs achieved by various rate controllers.

because the TCP controller is not suitable for estimation oriented applications since it minimizes a cost that is different from the estimation cost considered in this paper. To emphasize this point further, we present a simple scenario in which the TCP controller results in an unstable system and the cost becomes infinite while the proposed controller maintains stability. Assume that the network consists of a single link with capacity $c_1 = 1.3$ shared by two sources. The process parameters are $\{A_1, C_1, Q_1, R_1\} = \{1.15, 1, 2, 2\}$ and $\{A_2, C_2, Q_2, R_2\} = \{2, 1, 2, 2\}$. The rest of the system parameters are same as before and we ignore delays. The minimum value of the transmission probabilities required to stabilize the estimation error covariance (Lemma 4) are $\{p_1^{\min}, p_2^{\min}\} = \{0.24, 0.75\}$. The TCP controller distributes the rate approximately equally among the two sources as $\{0.65, 0.65\}$. We can see that although the estimation error of the first process is stable, the estimation error of the second process becomes unbounded since the transmission probability is less than $p_2^{\min}$. On the other hand, the optimal transmission probabilities achieved by the proposed controller is $\{0.36, 0.79\}$ and the error covariances remain stable. Thus, we can conclude that the standard TCP protocol may not be suitable for an estimation oriented application since it caters to a different set of applications, for example applications which require a notion of proportional fairness among the users.

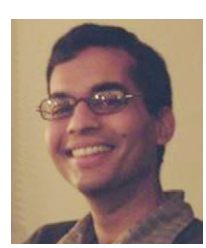
5. Conclusion

We studied the problem of rate control for networked estimation in the 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. The stability of the network was analyzed in the presence of delays.

References

Al-Hammouri, A., et al. (2006). Decentralized and dynamic bandwidth allocation in networked control systems. WPDRTS.
Antsaklis, P., & Baillieul, J. (Eds.) (2007). Special issue on networked control systems. *Proceedings of the IEEE*, *95*(1).
Caceres, R., et al. (1999). Multicast-based inference of network-internal loss characteristics. *IEEE Transactions on Information Theory*, *45*, 2462–2480.
Chiang, M., et al. (2007). Layering as optimization decomposition: a mathematical theory of network architectures. *Proceedings of IEEE*, *95*(1).
Desoer, C. A., & Wang, Y. T. (1980). On the generalized Nyquist stability criterion. *IEEE Transactions on Automatic Control*, *25*, 187–196.
Floyd, S., & Jacobson, V. (1993). Random early detection gateways for congestion avoidance. *IEEE/ACM Transactions on Networking*, (397–413).
Garone, E., et al. (2007). Communication protocols for optimal control over lossy networks. In *Allerton conference on communication, control and computing*.
Gupta, V., et al. (2009). Data transmission over networks for estimation and control. *IEEE Transactions on Automatic Control*, *54*(8), 1807–1819.
Jacobson, V. (1988). Congestion avoidance and control. *ACM Computer Communication Review*, *18*, 314–329.
Johari, R., & Tan, D. K. H. (2001). End-to-end congestion control for the internet: delays and stability. *IEEE/ACM Transactions on Networking*, *9*(6), 818–832.
Kelly, F. (2001). Mathematical modeling of the internet. In B. Engquist, & W. Schmid (Eds.), *Mathematics unlimited-2001 and beyond* (pp. 685–702).
Kelly, F., et al. (1998). Rate control in communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research Society*, *49*, 237–252.
Low, S. H., & Lapsley, D. E. (1999). Optimization flow control-I: basic algorithm and convergence. *IEEE/ACM Transactions on Networking*, *7*(6), 861–874.
RFC, (1981). Transmission Control Protocol, RFC-793.
Schenato, L., et al. (2007). Foundations of control and estimation over lossy networks. *Proceedings of IEEE: Special Issue on Networked Control Systems*, *95*(1).
Shakkottai, S., & Srikant, R. (2007). Network optimization and control. *Foundations and Trends in Networking*, *2*(3), 271–379.
Vinnicombe, G. (2002). On the stability of networks operating TCP-like congestion control. In *Proceedings of the 15th IFAC world congress on automatic control*.

Vaibhav Katewa is a Ph.D. candidate in the Department of Electrical Engineering at the University of Notre Dame. He received his B.Tech. degree from the Indian Institute of Technology, Kanpur and M.S. degree from the University of Notre Dame, both in Electrical Engineering. His research interests include decentralized control and protocol design for networked control systems.



Vijay Gupta is an Assistant Professor in the Department of Electrical Engineering at the University of Notre Dame. He received his B. Tech degree from the Indian Institute of Technology, Delhi and the M.S. and Ph.D. degrees from the California Institute of Technology, all in Electrical Engineering. He has served as a research associate in the Institute for Systems Research at the University of Maryland, College Park, and as a consultant to the Systems Group at the United Technology Research Center, Hartford, CT. His research interests include various topics at the interaction of communication, computation and control. He received the NSF Career award in 2009.