Burst-level Congestion Control Using Hindsight Optimization

Gang Wu, Edwin K. P. Chong†, and Robert Givan

Abstract— **We consider the burst-level congestion-control problem in a communication network with multiple traffic sources, each modeled as a fully-controllable stream of fluid traffic. The controlled traffic shares a common bottleneck node with high-priority cross traffic described by a Markov-modulated fluid (MMF). Each controlled source is assumed to have a unique round-trip delay. The goal is to maximize a linear combination of the throughput, delay, traffic loss rate, and a fairness metric at the bottleneck node. We introduce a simulation-based congestion-control scheme capable of performing effectively under rapidly-varying cross traffic by making use of the provided MMF model of that variation. The control problem is posed as a finite-horizon Markov decision process and is solved heuristically using a technique called Hindsight Optimization. We provide a detailed derivation of our congestion-control algorithm based on this technique. Our empirical study shows that the control scheme performs significantly better than the conventional proportional-derivative (PD) congestion-control method.**

Keywords— **Communication networks, congestion control, traffic models, Markov-modulated fluid, Markov decision processes, online simulation.**

I. Introduction

We study the rate-based congestion control of traffic in a network where a bottleneck node is shared by multiple best-effort traffic sources and other high-priority "cross-traffic" sources. We assume that the best-effort sources can be fully controlled, but that each such source originates at a unique distance from the bottleneck node, and thus has a unique control delay. Taking these unique delays into the account in decision making is a difficult problem. The objective of congestion control is to determine proper and fair transmission rates for the best-effort traffic sources to utilize the bandwidth available to them efficiently at the bottleneck node while achieving low average queuing delay and a low traffic loss rate.

† Correspondence author.

G. Wu is with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907, U.S.A. He can be reached by telephone at +1 (970) 491-3148 or by e-mail at gwu@ecn.purdue.edu.

E. K. P. Chong is with the Department of Electrical and Computer Engineering, Colorado State University, Fort Collins, CO 80523, U.S.A. He can be reached by telephone at +1 (970) 491-7858, by fax at +1 (970) 491-2249, or by e-mail at echong@engr.colostate.edu.

R. Givan is with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907, U.S.A. He can be reached by telephone at +1 (765) 494-9068, by fax at +1 (765) 494-6440, or by e-mail at givan@ecn.purdue.edu.

This research is supported by DARPA/ITO under contract F30602-00-2-0552, and by NSF under contracts 9977981-IIS, 0093100-IIS, and 0098089-ECS. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency, the National Science Foundation, or the U. S. Government.

Previous research on best-effort congestion control can be divided into rate-based and window-based approaches. Here we present a rate-based approach, controlling the rates of the best-effort sources rather than deciding window sizes to those sources. Previous rate-based work involves binary feedback [6] and proportional controllers [7], [8] for ATM (Asynchronous Transfer Mode) networks and linear-increase/exponential-decrease controllers for TCP/IP (Transmission Control Protocol/Internet Protocol) networks [15]. Recent approaches attempt to improve performance by incorporating control-theoretic techniques, including proportional-derivative (PD) controllers [9], [10], [17], [30], and those using optimal control and dynamic game techniques such as linear quadratic (LQ) team, H^∞, and noncooperative game controllers [12], [13], [11].

We motivate our work by noticing that most of the above control schemes are designed for constant or slowly-varying cross traffic (with the exception of the LQ team and the H^∞ controllers). We call these *connection-level* congestion controllers since they assume that the cross-traffic variation is caused primarily by the joining of new connections and the termination of existing ones. However, burstiness in cross traffic in real networks often occurs at small time scales, i.e., from several milliseconds up to a second [3]. Fast changes in the cross traffic, coupled with large bandwidth-delay products, often significantly degrade the performance of connection-level controllers.

We approach the congestion-control problem using an alternative paradigm that alleviates these drawbacks. Here we assume we are provided with a stochastic model of the cross traffic, and demonstrate a controller that achieves substantial benefits from exploiting this model. We call our approach *burst-level* congestion control. Our controller predicts future cross traffic using the stochastic cross-traffic model so that the controller can anticipate changes stochastically and act "before" the changes happen.

We model the cross traffic at the bottleneck node as a Markov-modulated fluid (MMF) [1], [2], [3]. We formulate our congestion-control problem as a discrete-time finite-horizon Markov decision process (MDP) [5]. The familiar congestion control goal—rapidly achieving queue stability in response to step changes in cross traffic—is not an appropriate measure of success here due to the inclusion of rapidly-varying cross traffic. Instead, we formulate a measure of performance over long traces of cross-traffic variation, balancing throughput, delay, loss, and fairness. We extend our previously proposed Hindsight Optimization (HO) technique [4] to provide a heuristic solution to the MDP problem—in particular, the HO technique has

never previously been used to address a problem with an infinite control action space.

The main contribution of this work is to demonstrate that a stochastic cross-traffic model can be effectively exploited in congestion control to achieve substantial performance benefits. A secondary contribution is to provide a specific means to obtain these benefits using a novel congestion-control framework based on online simulation. Our work provides both a strong motivation and a useful starting point for seeking a practically-realized scheme incorporating traffic models.

Although MMF models have been extensively employed in network performance analysis (e.g., [1], [2]), our work is the first to exploit such models for rate-based congestion control. Compared with the work of [12], [13], which models cross traffic by an auto-regressive moving-average process, MMF models have more structure, and better performance is therefore expected when such models are available. In [15], the authors incorporate a long-range-dependent model into the design of a linear-increase/exponential-decrease (LIED) controller. Our MMF model yields to a decision-theoretic analysis, as mentioned above, resulting in a controller that is not constrained to be LIED.

Previous work on congestion control using MDP formulation includes [33], [34], [35]. Our work differs from [33], [34], [35] in the sizes of action spaces, the generality of reward structures, and solution methods. Recent work on policy rollout algorithms (see, e.g., [36]) provides a means of using simulation to select "good" control actions heuristically, but requires starting with a good "base" heuristic policy. Policy rollout uses one-step look-ahead relative to the base policy value function (much like value iteration) to obtain an estimate of the so-called "Q-value"—this estimate must lower bound the true Q-value because the base policy value function lower bounds the true value function. The Q-value estimate used in the HO technique is an upper bound.

The remainder of the paper is organized as follows. In Section II, we describe our network model and define the congestion-control problem as an MDP. In Section III, we introduce the HO technique for heuristic MDP control, and present our gradient-based control algorithm. Section IV presents the simulation results of our controller and the PD controller to enable comparison. Section V concludes the paper.

II. System Model and Problem Description

A. System Model

We consider a network where a single bottleneck node is shared by multiple rate-controlled traffic sources and other high-priority "cross-traffic" sources. The controlled sources transmit at rates specified by a central controller residing at the bottleneck node. Associated with each source is a fixed round-trip delay. Without loss of generality, we assume that the round-trip delays are distinct from one another. We notate vectors and their components as follows: for vector $\vec{v}$, we write $v^{(i)}$ for the ith component of $\vec{v}$ when that component is scalar, and $\vec{v}^{(i)}$ for that component when it is itself a vector. We also notate the jth component of the ith component of $\vec{v}$ (when $\vec{v}$ is a vector of vectors) as $v^{(i,j)}$. Throughout this paper we use the notation E_X to denote expectation taken with respect to the random variable X.

We assume that time is discrete with small time increments δ. We now describe four essential components of our system: the controlled traffic sources, the cross traffic, the bottleneck node, and the congestion controller.

Controlled Sources. Denote the collection of controlled sources by the set $\mathbf{N}$ and let $N = |\mathbf{N}|$ be the cardinality of $\mathbf{N}$. We assume that in real time the round-trip delays are large compared with the time increment δ such that we can express the delay by integral multiples of δ with sufficient accuracy. Hence, in discrete time, we denote the round-trip delay of source i as $d^{(i)}$, a positive integer. We index the sources such that $0 < d^{(1)} < d^{(2)} < \cdots < d^{(N)}$. The sources constantly transmit at the controller-specified rates and respond to rate commands instantaneously upon their arrival. This model emulates controlled ABR (available bit rate) traffic in ATM networks and UDP (User Datagram Protocol) traffic in IP networks, which are suitable candidates for rate-based congestion-control schemes.

Cross Traffic. The high-priority cross traffic represents, for example, CBR and VBR traffic in ATM networks, or traffic in IP networks receiving high-priority service via the CBQ (class-based queuing) scheme [14]. This cross traffic determines the available bandwidth that the controlled traffic experiences at the bottleneck node.

The cross-traffic process can change at any (discrete) time. For convenience, instead of specifying the cross-traffic distribution, we specify the "service process," which is the difference between the rate of cross traffic and C, the constant bandwidth of the bottleneck node. This service process can also be characterized using an MMF model easily derived from the MMF model of the cross traffic. The service process is represented by a Markov chain with state space $\mathbf{S} = \{1, \ldots, m\}$, a transition probability matrix $\mathbf{M} : (\mathbf{S} \times \mathbf{S}) \mapsto [0, 1]$, and a set of distinct rate values $v_1, \ldots, v_m$ (i.e., m is the number of the values that the service rate can take) which are real numbers in the interval $[0, C]$. By "service rate" we mean the amount of fluid best-effort traffic that can be served in one time step. When in state s, the service rate is v_s. In this setting, measuring service rate suffices to determine the state of the service process.

Bottleneck Node. The bottleneck node has a buffer of finite size. The controlled best-effort traffic is buffered together, independently of any buffering needed for the high-priority cross traffic. We denote the size of the buffer by B. As defined above, we denote the bandwidth at the bottleneck node by C. We assume that the queue length at the bottleneck node is known at each time step. We also assume a first-in-first-out (FIFO) service discipline at the bottleneck node.

Congestion Controller. The controller, residing at the bottleneck node, makes control decisions at each time step. The congestion-control problem is to determine a rate

command $u_k^{(i)}$ to relay to source i, $i = 1, \ldots, N$, at time k to achieve some overall performance objective. Our objective is to balance throughput, delay, loss, and fair service to controlled sources, as described formally by the reward function below. The controller can use system observations and a model of the service process to compute rate commands. The rate command for a source at any given epoch impacts the bottleneck node arrivals after a time duration equal to one round-trip delay for that source. Therefore, at each decision-making epoch, the controller needs to compute an appropriate rate command for each source that takes into account the round-trip delays and anticipated service-rate variation. The order of event occurrence at the bottleneck is: decision making, MMF transition, traffic arrival and simultaneous traffic forwarding, and then check for buffer overflow/underflow.

B. MDP Problem Formulation

We formulate the congestion-control problem as a Markov decision process (MDP). An MDP consists of an action space, a state space, a state-transition structure, and a reward structure. In the following, we describe each of these components for our problem.

Action Space. We assume that the transmission rates at the controlled sources are bounded from below by zero and from above by the common value C. We denote the action space by $\mathbf{U} = [0, C]^N$. For instance, at time k the control action is a vector $\vec{u}_k$ of the form $\vec{u}_k = [u_k^{(1)}, \ldots, u_k^{(N)}]$.

State Space. The system state has three components. The first is the state of the service process, taking values in $\mathbf{S}$. The state of the service process at each time step corresponds to the departure rate observed at the previous time step, and the MMF service model will transition before departures occur at the current time step. (This choice models that fact that we cannot know the current cross traffic precisely until after it has occurred and we have had the opportunity to measure it.) The second component is the current queue length l, taking values in $\mathbf{L} = [0, B]$. The third component consists of the control signals that have been issued in the past but whose impact has not yet been felt at the bottleneck node due to the round-trip delays. This control history $\vec{w}$ takes values in $\mathbf{U}^{d^{(N)}}$. For example, if the control actions selected over time are $\vec{u}_0, \vec{u}_1, \ldots$ then the control history $\vec{w}_k = (\vec{w}_k^{(1)}, \ldots, \vec{w}_k^{(d^{(N)})})$ at time k is such that $\vec{w}_k^{(i)} = \vec{u}_{k-i}$ and thus $w_k^{(i,j)} = u_{k-i}^{(j)}$. We note that this control history includes unnecessary information in the form of history beyond the round-trip delay for sources closer than delay $d^{(N)}$. This information is included to greatly simplify our notation throughout this paper, but is not truly needed in the intended model or for any of our methods. The complete state space is $\mathbf{X} = \mathbf{S} \times \mathbf{L} \times \mathbf{U}^{d^{(N)}}$.

State Transition. If the state is $\vec{x} = (s, l, \vec{w})$ where $\vec{w} = (\vec{w}^{\,(1)}, \ldots, \vec{w}^{\,(d^{(N)})})$ denotes the control history, and we apply a control $\vec{u}$, the system will make a transition to a new state $\vec{x}\,' = (s', l', \vec{w}')$. In the following, we specify how each component of $\vec{x}\,'$ depends on $\vec{x}$ and $\vec{u}$.

The service-process state makes a transition from s to s' with probability $P(s, s')$ specified in the given matrix $\mathbf{M}$—this transition is unaffected by the values of l and $\vec{w}$.

The queue-length component l' depends on $\vec{x}$ as follows. Let $a(\vec{x}) = \sum_{i=1}^{N} w^{(d^{(i)}, i)}$ be the aggregate fluid traffic that arrives during the transition from state x to state x' from all controlled sources—this traffic is due to rate commands that were issued to these sources in the past which are now recorded in the state component $\vec{w}$. Recalling that the service process consumes fluid traffic at rate $v_{s'}$ when in service process state s', and given the just computed fluid arrival, the queue-length component of the state changes according to the following difference equation, commonly called Lindley's equation:

$$l' = \max\{\min\{l + a(\vec{x}) - v_{s'}, B\}, 0\}.$$

The queue-length component l' does not depend on $\vec{u}$ due to non-zero round-trip delays.

Finally, the control history updates as follows:

$$\vec{w}\,'^{\,(1)} = \vec{u}, \qquad \vec{w}\,'^{\,(i+1)} = \vec{w}\,'^{\,(i)}, \qquad i = 1, \ldots, d^{(N)} - 1.$$

Reward Structure. We define the one-step reward at state $\vec{x}$ by

$$R(\vec{x}) = T(\vec{x}) - \alpha D(\vec{x}) - \beta F(\vec{x}) - \zeta L(\vec{x}), \tag{1}$$

where $0 < \beta < \min\{1, \alpha/2, \zeta\}$, $T(\vec{x})$ is the throughput received at one time step when the system is in state $\vec{x}$, $D(\vec{x})$ is the total queuing delay incurred at that time step, $F(\vec{x})$ is the sum of the absolute pairwise rate differences in arriving traffic from different controlled sources at that time step, and $L(\vec{x})$ is the fluid lost at the current time step due to buffer overflow (after "check for buffer overflow"). Note that these four quantities—throughput, delay, loss, and fairness—are independent metrics in evaluating control performance; in particular, specifying throughput and loss is not redundant, because the total arrival is not fixed or pre-determined.

The scaling factors α, β, and ζ reflect our tradeoff preference between throughput, delay, loss, and service fairness. Specifically, the restriction $0 < \beta < \min\{1, \alpha, \zeta\}$ represents our preference towards optimizing throughput, delay, and loss, with the fairness optimization somewhat subordinate. We are most interested in parameter values satisfying this restriction. The further restriction that $\beta < \alpha/2$, as we will explain after presenting Proposition 1, allows the analytical selection of a hindsight-optimal control sequence (defined later) more easily, and we do not consider the more difficult and less important case of parameter settings that violate this restriction. While it appears that, given the restrictions on β, the fairness term makes a negligible contribution to the reward function, this contribution is important, as we discuss later in Section IV-C.

The one-step reward $R(\vec{x})$ depends only on the state $\vec{x}$ and not explicitly on the control $\vec{u}$ because any rate command in $\vec{u}$ will not have impact on the bottleneck node until at least $d^{(1)}$ time units later, due to the non-zero round-trip delays $d^{\,(i)}$. Recall that $d^{(1)}$ is the smallest among all

$d^{(i)}$'s. We now provide formal expressions for $T(\vec{x})$, $D(\vec{x})$, $L(\vec{x})$, and $F(\vec{x})$ for completeness. The throughput, delay, loss, and unfairness penalty at a time step are given by:

$$T(\vec{x}) = \min\{l + a(\vec{x}), v_{s'}\} \quad (2)$$

$$D(\vec{x}) = \max\{\min\{l + a(\vec{x}) - v_{s'}, B\}, 0\} \quad (3)$$

$$L(\vec{x}) = \max\{l + a(\vec{x}) - v_{s'} - B, 0\} \quad (4)$$

$$F(\vec{x}) = \frac{1}{N-1} \sum_{i=1}^{N} \sum_{j=i+1}^{N} \left| w^{(d^{(i)},i)} - w^{(d^{(j)},j)} \right|, \quad (5)$$

where s' is the service-rate process state after MMF transition from state s. Note that the throughput, delay, and loss terms of the reward function (and thus the reward itself) are random variables due to their dependence on the random variable s'.

Optimization Goal. Based on the MDP model described above, we can state the congestion-control problem as follows. For a given initial state $\vec{x}_0 \in \mathbf{X}$, we apply a control $\vec{u}_0 \in \mathbf{U}$ to the system and receive a reward of $R(\vec{x}_0)$ by serving traffic at the bottleneck node. The system will then make a transition to a new state $\vec{x}_1$, stochastically according to the state-transition structure. We then apply a control $\vec{u}_1$, and so on. After a horizon of H steps, the cumulative reward received (a random variable) is given by

$$W_H(\vec{u}_0, \ldots, \vec{u}_{\tilde{H}-1}) \equiv \sum_{k=0}^{H-1} R(\vec{x}_k),$$

where $\tilde{H} = H - d^{(1)}$, and $\vec{u}_{\tilde{H}-1}$ is the latest control command that can impact the bottleneck node within the horizon H.

Our choice of $\vec{u}_k$ is based on $\vec{x}_k$; that is, we use a "state-feedback" map $\mu_k : \vec{x} \mapsto \vec{u}$ and apply $\vec{u}_k = \mu_k(\vec{x}_k)$. The sequence of maps $\pi = \{\mu_0, \mu_1, \mu_2, \ldots\}$ is called a *policy*. For a given initial state $\vec{x}_0$, the problem is to find a policy that maximizes the objective function

$$V_H^{\pi}(\vec{x}_0) = E\left(W_H(\mu_0(\vec{x}_0), \ldots, \mu_{\tilde{H}-1}(\vec{x}_{\tilde{H}-1}))\right).$$

Given a policy π or a fixed sequence of controls $\vec{u}_0, \vec{u}_1, \ldots$, we denote the state of the system at each time k in $0, 1, \ldots$ by the random variable X_k and the state of the service process at time k by the random variable S_k.

C. Optimal Solution

To describe our approach to the congestion-control problem, we first characterize the optimal congestion-control policy. For a given initial state $\vec{x}$, let

$$V_H^*(\vec{x}) = \max_{\pi} V_H^{\pi}(\vec{x}).$$

Following a standard approach to solving MDPs, we write

$$Q_k(\vec{x}, \vec{u}) = R(\vec{x}) + E(V_{k-1}^*(\vec{x}')), \qquad k = 1, \ldots, H,$$

where the expectation in the right-hand side is with respect to the next state $\vec{x}'$, and $V_{k-1}^*(\vec{x}')$ is the optimal cumulative reward over the $k-1$ time steps starting from the (random) state $\vec{x}'$. A key result in Markov decision theory [5] then states that

$$V_H^*(\vec{x}) = \max_{\vec{u} \in \mathbf{U}} Q_H(\vec{x}, \vec{u}),$$

and a policy $\pi^* = (\mu_0^*, \mu_1^*, \ldots)$ is an optimal policy if it satisfies for all k,

$$\mu_k^*(\vec{x}) = \arg\max_{\vec{u} \in \mathbf{U}} Q_{H-k}(\vec{x}, \vec{u}).$$

In particular, for a fixed horizon H, the control $\vec{u}^*$ is an optimal "current" action if it satisfies

$$\vec{u}^* = \mu_0^*(\vec{x}) = \arg\max_{\vec{u} \in \mathbf{U}} Q_H(\vec{x}, \vec{u}). \quad (6)$$

At each control epoch we apply the "current" control action $\vec{u}^*$ in (6). Each control epoch involves optimizing $Q_H(\vec{x}, \vec{u})$ with respect to $\vec{u}$ for a horizon of H into the future. This approach of applying a "moving-horizon" control solution in an online fashion is common in the optimal-control literature, for example in *receding-horizon control* (e.g., [19], [20]).

In practice we do not have explicit knowledge of Q_H. Standard techniques can compute Q_H in time polynomial in the size of the state space. However, because we have an implicitly specified state space (specified component by component above), our actual state space is astronomical in size; as a result, these standard techniques cannot be applied in practice. Thus, (6) is not directly useful for determining the optimal policy. Our MDP problem does not yield to any other known analytical solution. Instead, we approach the problem by computing an upper bound estimate of Q_H. We stress that the non-negligible information delays in our system greatly complicate the problem. If the delays are not significant with respect to the dynamics of the Markov-modulated service process, we can greatly reduce the state space, and standard techniques are applicable to solving the resulting problem. In the next section, we describe a particular approach to solving our problem, based on evaluating candidate actions using such upper bound estimates of Q_H.

III. Congestion-control Algorithm Using Hindsight Optimization

A. The Hindsight Optimization Technique

In this subsection, we outline our solution approach, which extends a technique called *hindsight optimization*, first described in [4]. The overall control architecture is illustrated in Figure 1. The controller comprises three parts: a state observer, a traffic simulator, and a rate calculator. The state observer is responsible for obtaining the system state $\vec{x}$ by measuring the service rate at each time step (as well as observing the current queue length and storing the recent control history). Given our assumption that the MMF model state each determines a unique service rate, the system state is fully observable.

The traffic simulator takes the observed current state $\vec{x}$ and uses it as a starting state to generate a finite number

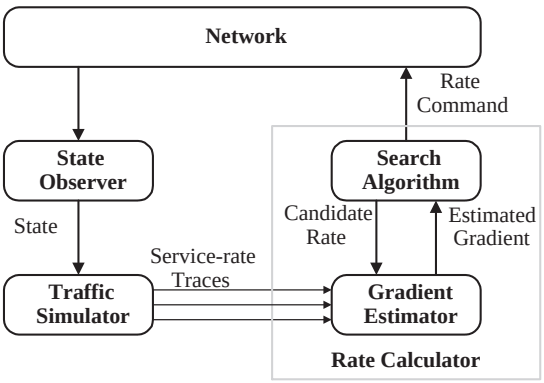


Fig. 1. Congestion-control architecture.

of possible service-rate sequences (traces) using our MMF model. The rate calculator takes these traces and computes a rate command vector $\vec{u}$. The calculation of the rate command vector is based on the following idea. Recall from equation (6) that at any given control epoch and any state $\vec{x}$, the optimal rate command is given by

$$\vec{u}^* = \arg\max_{\vec{u}\in\mathbf{U}} Q(\vec{x},\vec{u}) \tag{7}$$

(we omit the subscript H in $Q_H(\vec{x},\vec{u})$ for brevity). We rely on an estimate $\hat{Q}(\vec{x},\vec{u})$ of the $Q(\vec{x},\vec{u})$ to carry out the above maximization. This estimate is calculated as follows. For each service-rate trace t, we compute the cumulative reward by taking action $\vec{u}$ at state $\vec{x}$ followed by a *trace-optimal* sequence of actions $\vec{u}_1^t, \vec{u}_2^t, \ldots, \vec{u}_{\tilde{H}-1}^t$ for the remaining horizon of $\tilde{H}-1$ time steps. We say that the sequence $\vec{u}_1^t, \vec{u}_2^t, \ldots, \vec{u}_{\tilde{H}-1}^t$ is trace-optimal if this sequence achieves the largest possible cumulative reward under the assumption that the service rate does indeed vary according to the trace under consideration. We call any such trace-optimal sequence a *hindsight-optimal control sequence* and the optimal cumulative reward of any such sequence the *hindsight-optimal value* of the trace—computing such a sequence and its corresponding value is an offline optimization problem that is often considerably easier than finding the optimal stochastic control for the online problem. We compute the average of the hindsight-optimal values over the set of n generated traces for the specified initial control $\vec{u}$—this average is our estimate $\hat{Q}_n(\vec{x},\vec{u})$ of $Q(\vec{x},\vec{u})$. In other words, $\hat{Q}(\vec{x},\vec{u})$ and its sampled approximation $\hat{Q}_n(\vec{x},\vec{u})$ are given by

$$\hat{Q}_n(\vec{x},\vec{u}) = \frac{1}{n}\sum_{t=1}^{n} W_t^*(\vec{x},\vec{u}), \tag{8}$$

$$\hat{Q}(\vec{x},\vec{u}) = R(\vec{x}) + E_{S_1,\ldots,S_H}\max_{\vec{u}_1,\ldots,\vec{u}_{\bar{H}-1}} W_{H-1}(\vec{u}_1,\ldots,\vec{u}_{\bar{H}-1}), \tag{9}$$

where

$$W_{H-1}(\vec{u}_1,\ldots,\vec{u}_{\bar{H}-1}) \equiv \sum_{k=1}^{H-1} R(\vec{x}_k),$$

and $W_t^*(\vec{x},\vec{u})$ is the hindsight-optimal value for the trace t as a result of applying control $\vec{u}$ at state $\vec{x}$.

Given the estimate $\hat{Q}(\vec{x},\vec{u})$ of $Q(\vec{x},\vec{u})$ for each action, the hindsight optimization approach is to select and execute the action $\vec{u}$ that maximizes this estimate. Previous applications of the hindsight-optimization technique have all involved MDP problems with finite action spaces, unlike our congestion control problem. When the action space is finite, we can simply compute the estimate $\hat{Q}(\vec{x},\vec{u})$ for each action $\vec{u}$, and choose the candidate action associated with the largest such estimate for execution. Here, however, we have an uncountable continuous action space, and cannot compute this estimate for every action. Instead, we extend the hindsight optimization technique by finding a (locally) optimal action using a gradient ascent technique. Because we seek to use gradient ascent to solve this problem, we actually need to analyze traces to find the gradient of $\hat{Q}(\vec{x},\vec{u})$ relative to changes in $\vec{u}$ (rather than to find $\hat{Q}(\vec{x},\vec{u})$ itself). We discuss a method for estimating this gradient from traces below.

As explained in [4], $\hat{Q}(\vec{x},\vec{u})$ is an upper bound on $Q(\vec{x},\vec{u})$ if exactly computed (e.g., by infinite sampling). This upper bound can be arbitrarily loose. However, these estimates are only used to rank candidate actions, and thus it only matters whether or not these estimates preserve the relative values at different states. Our results below give evidence that for congestion control problems the ranking is preserved well enough to make $\hat{Q}(\vec{x},\vec{u})$ useful for selecting an effective control policy.

B. Hindsight-optimal Control Sequences

At each decision-making epoch, we wish to determine a rate vector $\vec{u}^*$ according to (7) using $\hat{Q}_n(\vec{x},\vec{u})$ in place of $Q(\vec{x},\vec{u})$. For a given $\vec{x}$, we wish to maximize $\hat{Q}_n(\vec{x},\vec{u})$ with respect to $\vec{u}$. Because the argument $\vec{u}$ is a vector of continuous variables, we can use a search algorithm based on the gradient of $\hat{Q}_n(\vec{x},\vec{u})$ with respect to $\vec{u}$, which we denote by $\nabla_{\vec{u}}\hat{Q}_n(\vec{x},\vec{u})$.

Note that from equation (8), we can write $\nabla_{\vec{u}}\hat{Q}_n(\vec{x},\vec{u})$ as

$$\nabla_{\vec{u}}\hat{Q}_n(\vec{x},\vec{u}) = \frac{1}{n}\sum_{t=1}^{n}\nabla_{\vec{u}} W_t^*(\vec{x},\vec{u}), \tag{10}$$

where $\nabla_{\vec{u}} W_t^*(\vec{x},\vec{u})$ is the gradient of $W_t^*(\vec{x},\vec{u})$ with respect to $\vec{u}$. Therefore, the calculation of $\nabla_{\vec{u}}\hat{Q}_n(\vec{x},\vec{u})$ reduces to calculating the gradients on a per-trace basis, i.e., the gradients of the $W_t^*(\vec{x},\vec{u})$. It turns out that these gradients can be computed analytically, as we will show later. Our gradient estimate above is akin to the idea of *infinitesimal perturbation analysis* [27].

Recall that $d^{(1)}$ is the smallest round-trip delay among all $d^{(i)}$'s. We note that if $d^{(1)} \geq H-1$, then no matter what control sequence we apply, by the end of the horizon H, no action will have impact on the bottleneck node. Therefore, we consider only the nontrivial case where $0 < d^{(1)} < H-1$.

Suppose that the current time is zero and the current state is $\vec{x}_0 = (s_0, l_0, \vec{w}_0)$. We wish to select as the current control the action $\vec{u}_0^*$ that maximizes $\hat{Q}(\vec{x}_0,\vec{u}_0^*)$. In the following, we first describe how we compute a hindsight-optimal control sequence for a given service-process trace.

Based on n such sequences, we then show how to obtain the gradient of $\hat{Q}_n(\vec{x}_0, \vec{u}_0)$, which together with a search algorithm forms our congestion-control algorithm. We now assume that we have generated a specific service-process trace $t = \{v_{s_1}, \ldots, v_{s_H}\}$.

Given the initial state $\vec{x}_0$ and action $\vec{u}_0$, define the trace-relative committed aggregate arrival rate (from all controlled sources due to rate commands in the control history as indicated in the $\vec{w}_0$ component of x_0) at times $k = 0, \ldots, H-1$ for initial control $\vec{u}_0$ by

$$a_k^t(\vec{u}_0) = \sum_{i=1}^{N} I_i(k) w_0^{(d^{(i)}-k,i)}, \text{ where } I_i = 1_{\{0,\ldots,d^{(i)}\}}$$

where we define $\vec{w}_0^{(0)}$ to be $\vec{u}_0$. The function $I_i(k)$ returns 1 when k is less than $d^{(i)}$, or equivalently, when the arrival from source i at time k is specified by $\vec{w}_0$. Define the trace-relative queue-length sequence $\{l_k^t,\ k = 0, \ldots, H\}$ (and the queue length $\{\tilde{l}_{k+1}^t,\ k = 1, \ldots, H\}$ before "checking for underflow") for initial control $\vec{u}_0$ by

$$\begin{aligned} l_{k+1}^t(\vec{u}_0) &= \max\{\tilde{l}_{k+1}^t(\vec{u}_0), 0\}, \text{ and} \\ \tilde{l}_{k+1}^t(\vec{u}_0) &= \min\{l_k^t(\vec{u}_0) + a_k^t(\vec{u}_0) - v_{s_{k+1}}, B\}, \end{aligned}$$

with $l_0^t(\vec{u}_0) = l_0$ the current queue size.

Notice $u_0^{(i)}$ is the control decision we are going to make at the current time and has not been determined yet; however, in the following proposition, $u_0^{(i)}$ is assumed given because we will assign it candidate values to determine the associated $\hat{Q}$ values. For a service-rate trace $t = \{v_{s_1}, \ldots, v_{s_H}\}$, we wish to compute a hindsight-optimal control sequence $\vec{u}_1^t, \ldots, \vec{u}_{\tilde{H}-1}^t$, where $\tilde{H} = H - d^{(1)}$ as before. In choosing such a control sequence, we will need notation for the number of sources that can affect a given time (i.e., sources such that the round-trip delay is less than the specified time)—we write $\tilde{\mathbf{N}}_k$ for the set of sources j such that $k \geq d^{(j)}$ and let $\tilde{N}_k = |\tilde{\mathbf{N}}_k|$. The following proposition gives the means to compute the unique hindsight-optimal control sequence analytically. (For the proof, see Appendix I.)

Proposition 1: Given system state $\vec{x}_0 = (s_0, l_0, \vec{w}_0)$, action $\vec{u}_0$, and a trace of future service rates $v_{s_1}, \ldots, v_{s_H}$, the sequence $\{\vec{u}_k^t,\ k = 1, \ldots, \tilde{H}-1\}$ with $(u_k^t)^{(i)}$ for source i specified as below is the unique hindsight-optimal control sequence.

$$(u_k^t)^{(i)} = \begin{cases} 0 & \text{when } \tilde{l}_{k+d^{(i)}+1}^t \geq 0, \\ -\tilde{l}_{k+d^{(i)}+1}^t / \tilde{N}_{k+d^{(i)}} & \text{otherwise,} \end{cases}$$

where $\tilde{l}_{k+d^{(i)}+1}^t = \tilde{l}_{k+d^{(i)}+1}^t(\vec{u}_0)$ is a function of $\vec{u}_0$.

The hindsight-optimal control sequence given in Proposition 1 is justified as follows. We first note that $\tilde{l}_{k+d^{(i)}+1}^t(\vec{u}_0)$ is the first value of the queue length "before checking for underflow" that can be influenced by the control $(u_k^t)^{(i)}$, because control sent to source i experiences a delay of $d^{(i)}$, and then the arrivals at time $k + d^{(i)}$ have their first effect on queue length at the next time step $k + d^{(i)} + 1$. The first case of the proposition corresponds to the situation where the predicted cross traffic leaves insufficient service bandwidth over the interval $0, \ldots, k + d^{(i)}$ to empty the queue at time step $k + d^{(i)} + 1$ (before new arrivals at that time). In this case, the full available service at time $k + d^{(i)}$ is utilized to serve fluid already requested by controls we are committed to (i.e., either $\vec{u}_0$ or controls that are in the history at time 0, that is, fluid in the "pipeline") so that any further fluid arriving at time $k + d^{(i)}$ will only be delayed until at least the next time step (incurring delay without improving throughput). Therefore, in the first case, it is optimal to add no further requests for arrivals at time k by setting $(u_k^t)^{(i)}$ to zero (given that delay and dropping are penalized more heavily than fairness according to the above assumptions on the parameters α, β, and ζ).

In the second case, we have $\tilde{l}_{k+d^{(i)}+1}^t(\vec{u}_0) < 0$, indicating that the bandwidth of the bottleneck node is not fully used without additional traffic requests, leading to loss in throughput. Hence, we need to request some additional traffic to maintain throughput but not so much that the queue fails to empty during traffic forwarding (incurring delay penalties without gain in throughput). We also need to divide this new traffic among all eligible $(\tilde{N}_{k+d^{(i)}})$ sources to be fair. The expression given in case 2 for $(u_k^t)^{(i)}$ selects exactly this amount of new traffic from source i.

The above arguments hold when $0 < \beta < \min\{1, \alpha/2, \zeta\}$, as we have assumed in our choice of reward function. By noticing that one infinitesimal increase in arrival (given that the bottleneck is under utilized) will result in one such increment in throughput but at most one such increment in the unfairness penalty, it is immediate that the restriction $\beta < 1$ will prioritize utilization of the bottleneck bandwidth over any reduction in fairness penalty. Any strictly positive value of α penalizes any traffic backlogging which only causes delay but does not improve throughput. Furthermore, setting $\beta < \min\{\alpha/2, \zeta\}$ makes it sub-optimal to increase arrival from any source(s) to decrease unfairness penalty without gain in throughput because this will result in more penalty from delay and/or loss. To see why we need $\beta < \alpha/2$, suppose an increase in arrival will remain in the buffer for $K > 1$ time steps before it is drained completely. Thus, the maximum possible reduction in fairness penalty is βK, and the delay penalty due to this traffic is $\alpha(K-1)$. To maintain our preferred priority of reducing delay over reducing unfairness, we require $\beta K < \alpha(K-1)$, or equivalently $\beta < (K-1)\alpha/K$. Since $(K-1)/K < K/(K+1)$, we set $\beta < \alpha/2$ (for $K = 2$) to satisfy this inequality for all $K \geq 2$.

C. Search Algorithm

Recall that at each time step, we wish to determine and relay to the sources the control action $\vec{u}_0^*$ that yields the largest estimate $\hat{Q}(\vec{x}_0, \vec{u}_0)$. Here, we use a simple search algorithm that uses only the gradient of $\hat{Q}$. Let $\nabla_{\vec{u}} \hat{Q}_n(\vec{x}_0, \vec{u}_0)$ represent the gradient of $\hat{Q}_n(\vec{x}_0, \vec{u}_0)$ (with respect to the control action $\vec{u}_0$). The search algorithm is of the form (see, e.g., [16])

$$\vec{u}(k+1) = \vec{u}(k) + \gamma(k) \nabla_{\vec{u}} \hat{Q}_n(\vec{x}_0, \vec{u}(k)), \qquad (11)$$

where $\gamma(k)$ is a positive step size, the iterate $\vec{u}(k)$ is an estimate of $\vec{u}_0^*$, and $\nabla_{\vec{u}} \hat{Q}_n(\vec{x}_0, \vec{u}_0)$ is given by equation (10) with the argument $\vec{u}$ replaced by $\vec{u}_0$.

The result of Proposition 2 below can be used to compute the gradient $\nabla_{\vec{u}} W_t^*(\vec{x}_0, \vec{u}_0)$ (and hence $\nabla_{\vec{u}} \hat{Q}_n(\vec{x}_0, \vec{u}_0)$). Combining this result with the algorithm (11), we now have an iterative procedure to compute $\vec{u}_0^*$. In practice, we terminate the algorithm (11) when the gradient $\nabla_{\vec{u}} \hat{Q}_n(\vec{x}_0, \vec{u}(k))$ is sufficiently close to $\vec{0}$. Note that we also need a value $\vec{u}_0(0)$ to initialize the algorithm (in our experiments, we set $\vec{u}_0(0) = \vec{0}$). We note that using a reasonable guess of the value of $\vec{u}_0(0)$ instead of $\vec{0}$ could speed up convergence; for instance, we can use the historical average of $\vec{u}_0(0)$ as the initial value when invoking our gradient search algorithm. For the step size sequence $\{\gamma(k)\}$, a typical and simple choice is to set $\gamma(k)$ to be a small positive constant.

We summarize the search procedure in the following routine. Let $\mathbf{Tr}$ be a given set of future bandwidth traces, $n = |\mathbf{Tr}|$ the cardinality of $\mathbf{Tr}$ as defined before, and $\nabla_{\vec{u}} W_t^*(\vec{x}_0, \vec{u}(k))$ the gradient of the hindsight-optimal value for trace t in $\mathbf{Tr}$, as given by Proposition 2. We also let $\vec{d}$ be a vector whose ith entry is $d^{(i)}$. We assume a given initial state $\vec{x}_0$. The search routine is as follows.

grad-search$(\mathbf{Tr}, \vec{d})$
1. Initialize $\vec{u}(0)$.
2. For $k = 1, 2, \ldots,$ do
   $\nabla_{\vec{u}} \hat{Q}_n(\vec{x}_0, \vec{u}(k)) = (1/n) \sum_{t=1}^{n} \nabla_{\vec{u}} W_t^*(\vec{x}_0, \vec{u}(k))$
   $\vec{u}(k+1) = \vec{u}(k) + \gamma(k) \nabla_{\vec{u}} \hat{Q}_n(\vec{x}_0, \vec{u}(k))$
   until $|\nabla_{\vec{u}} \hat{Q}_n(\vec{x}_0, \vec{u}(k))| \leq \varepsilon$.
3. Output $\vec{u}(k)$.

The set of traces $\mathbf{Tr}$ and delay parameter vector $\vec{d}$ are listed as arguments of the routine because both are needed in the calculation of $\nabla_{\vec{u}} W_t^*(\vec{x}_0, \vec{u}(k))$.

There are points where the trace-relative hindsight-optimal value is not differentiable. The use of gradient ascent methods with functions that are not everywhere differentiable has been studied before (e.g., [32]). In practice, we have found that the non-differentiable points in our objective function do not impact the efficacy of the gradient ascent algorithm. Hence, we do not delve further into this issue.

The search algorithm is in fact only a local-search method; the solution obtained depends on the initial condition, and may not be globally optimal. To search for a globally optimal solution, we could employ other familiar search techniques such as simulated annealing, but we have found satisfactory empirical results using only this local search.

D. The Gradient of the Hindsight-optimal Value

This section gives a technical account of our efficient means of computing the gradient of the hindsight-optimal value for the purposes of controlling the step direction in our search algorithm. Let $k^{t,i}(\vec{u}_0)$ be the first buffer-underflow or buffer-full time after $d^{(i)} - 1$ when encountering trace t with no additional flow requested after $\vec{u}_0$, given by

$$
\begin{aligned}
k_{\downarrow}^{t,i}(\vec{u}_0) &= \min\Big\{k : \Big(l_{k+1}^t = 0 \text{ and } d^{(i)} \leq k < H-1\Big) \\
&\qquad \text{or } k = H-1\Big\}, \\
k_{\uparrow}^{t,i}(\vec{u}_0) &= \min\Big\{k : \Big(l_{k+1}^t = B \text{ and } d^{(i)} \leq k < H-1\Big) \\
&\qquad \text{or } k = H-1\Big\}, \text{ and} \\
k^{t,i}(\vec{u}_0) &= \min\left\{k_{\downarrow}^{t,i}, k_{\uparrow}^{t,i}\right\}.
\end{aligned}
$$

We define $\mathbf{N}_i$ to be the set of all sources j in $\mathbf{N}$ with round-trip delays greater than $d^{(i)}$, so by our ordering of the sources we have $\mathbf{N}_i = \{j \in \mathbf{N} : j > i\}$. Sources with longer round-trip delays than $d^{(i)}$ have fluid arrivals for time $d^{(i)}$ specified in the control history of $\vec{w}_0$, and no hindsight-optimal control sequence can change the arrivals that will occur at time $d^{(i)}$ from those sources.

We now introduce notation that divides the sources in $\mathbf{N}_i$ according to whether they have arrival rates at time $d^{(i)}$ that are higher or lower than any particular rate r. For each source $i \in \mathbf{N}$, we partition the set $\mathbf{N}_i$ of "uncontrollable" sources into two subsets, $\mathbf{N}_<^i(r)$ and $\mathbf{N}_>^i(r)$, according to how the arrival rates from those sources at time $d^{(i)}$ compare to the rate r, as follows:

$$\mathbf{N}_<^i(r) = \{j \in \mathbf{N}_i : w_0^{(d^{(j)} - d^{(i)}, j)} < r\}, \;\; N_<^i(r) = |\mathbf{N}_<^i(r)|,$$

$$\mathbf{N}_>^i(r) = \{j \in \mathbf{N}_i : w_0^{(d^{(j)} - d^{(i)}, j)} > r\}, \;\; N_>^i(r) = |\mathbf{N}_>^i(r)|.$$

For a given state $\vec{x}_0$ and action $\vec{u}_0$, the gradient $\nabla_{\vec{u}} W^*(\vec{x}_0, \vec{u}_0)$ of the hindsight-optimal value can be computed analytically as characterized in the following proposition. (For the proof, see Appendix II.)

Proposition 2: Given state $\vec{x}_0$, candidate initial control $\vec{u}_0$, and service-rate trace $t = \{v_{s_1}, \ldots, v_{s_H}\}$, the source i component (for any i) of $\nabla_{\vec{u}} W^*(\vec{x}_0, \vec{u}_0)$, where it exists, is given by the weighted sum of the following four terms (with the weights 1, $-\alpha$, $-\beta$, and $-\zeta$, respectively), representing the throughput, delay, fairness, and loss components of the change in total reward:

$$
\begin{aligned}
\nabla_{\vec{u}} T(\vec{x}_0, \vec{u}_0)^{(i)} &= \begin{cases} 1 & \text{if } i = 1 \text{ and } k_{\downarrow}^{t,1}(\vec{u}_0) = d^{(1)} \\ 0 & \text{otherwise} \end{cases} \\
\nabla_{\vec{u}} D(\vec{x}_0, \vec{u}_0)^{(i)} &= k^{t,i} - d^{(i)} \\
\nabla_{\vec{u}} F(\vec{x}_0, \vec{u}_0)^{(i)} &= N_<^i(u_0^{(i)}) - N_>^i(u_0^{(i)}) + \nabla_{\vec{u}} F_1^{(i)} \\
\nabla_{\vec{u}} L(\vec{x}_0, \vec{u}_0)^{(i)} &= \begin{cases} 1 & \text{if } k_{\uparrow}^{t,i} < k_{\downarrow}^{t,i} \\ 0 & \text{otherwise,} \end{cases}
\end{aligned}
$$

where

$$
\nabla_{\vec{u}} F_1^{(i)} = \begin{cases} 0 & \text{if } i = 1 \\ i - 1 & \text{else if } \tilde{l}_{d^{(i)}+1}^t > 0 \\ i + N_>^i(r) - N_<^i(r) & \text{else if } r < u_0^{(i)} \\ -i + N_>^i(r) - N_<^i(r) & \text{else if } r > u_0^{(i)} \end{cases},
$$

$$r = -\tilde{l}_{d^{(i)}+1}^t(u_0^{(i)})/(i-1).$$

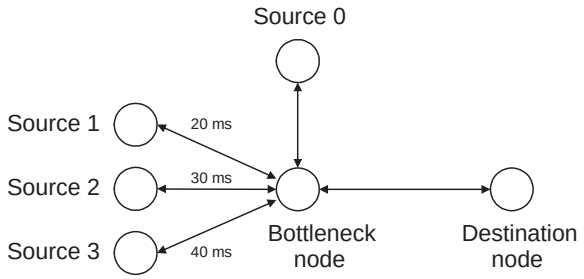


Fig. 2. Network configuration for empirical study.

E. The Congestion-control Algorithm

We conclude this section by summarizing the congestion-control algorithm as follows. At each control epoch, we perform these steps:

1. Observe current system state $\vec{x}_0$;
2. Generate a set **Tr** of future service-rate traces;
3. Compute $\vec{u}_0^* = \textbf{grad-search}(\textbf{Tr}, \vec{d})$;
4. Transmit rate command $\vec{u}_0^*$ to sources.

Our control approach exploits two structural properties of our problem. The first is that the stochasticity in our problem is "exogenous"; i.e., the randomness in the service process is independent of the control we choose. This property greatly eases the application of the hindsight-optimization framework, resulting in more natural offline optimization subproblems that are typically more easily solved. The second property is the piecewise linearity of the reward function, which leads to the tractability of our per-trace analysis, in particular, the piecewise derivation of the gradient of the hindsight-optimal value.

IV. Empirical Results

A. Evaluation Setup

We use the network simulator *ns*2 to carry out simulation. We have modified *ns* to implement our congestion-control algorithm over UDP to emulate an ATM network, and we set the packet size to be 53 bytes, the size of a standard ATM cell.

Figure 2 illustrates our simulated network. The traffic sent from four sources, indexed by 0 through 3, shares a common bottleneck node. All links between the sources and the bottleneck node have bandwidth of 155 Mbps, while the bottleneck link is of only 55 Mbps. The size of the buffer at the bottleneck node is 150 cells. Source 0 represents the source for high-priority cross traffic. Sources 1, 2, and 3 are controlled best-effort traffic sources, which send traffic at the rates determined by the controller residing at the bottleneck node. These three sources are associated with round-trip delays of 20, 30, and 40 ms, respectively.

Our empirical study consists of two parts. In the first part, the cross-traffic source 0 is composed of 10 identical connections, each generating fluid traffic according to a two-state ON-OFF MMF model. We will be varying the two transmission rates corresponding to the ON and the OFF states, to let the aggregated cross traffic rate have different variances (ranging from 0 to 72.6 Mbps^2) but the same mean (22 Mbps). This allows us to study the impact of the variance of the cross traffic on system performance. In the two-state MMF model, the expected lengths of the ON and the OFF periods are 400 and 600 ms, respectively; these values were chosen to be the same as in the second part of our empirical study.

In the second part, source 0 consists of 1000 independent and identically-distributed voice connections, reflecting a typical scenario arising in networks with mixed voice and data traffic. The dynamics of voice connections are well captured by MMF models [1], [18]. We model a single voice connection by a two-state ON-OFF MMF model, with the expected ON and OFF periods being 400 and 600 ms, respectively. Since a standard voice connection consumes 64 Kbps bandwidth, we set the rate of each voice connection to 70.7 Kbps by considering that the actual payload in a 53-byte ATM cell is only 48 bytes.

B. Comparison Metrics

We will compare the performance of the controller described in this paper, called the "HO controller" hereafter, with the well-known PD controller. Our comparison metrics are utilization, average queuing delay, cell loss rate, and cumulative fairness, which are of direct interest to network users. Utilization is defined as the ratio of the total number of cells forwarded to the total available service "volume" (the sum of the service rates) over the simulation period at the bottleneck node. The average queuing delay is the total amount of time that all the cells spend waiting in the queue at the bottleneck node divided by the total number of cells forwarded. The cell loss rate is defined as the number of cells (from the controlled sources) lost due to buffer overflow divided by the total number of cells that arrive at the bottleneck node (from the controlled sources) over the simulation period. To define cumulative fairness, let T_i denote the total number of cells that arrive from source i. Then, cumulative fairness is defined as the standard deviation of the T_i's divided by the mean of the T_i's.

In most previous papers, e.g., [9], [10], the test metric is the controller's ability to maintain a target queue size. However, by design the HO controller does not aim to maintain a fixed queue size. Thus, we do not evaluate the HO controller's ability in this respect.

C. Impact of Cross-traffic Variance

In this subsection, we investigate the impact of the variance in cross traffic on the performance of the HO and PD controllers. PD-type congestion controllers have been shown to be generally effective. PD controllers adjust the transmission rate based on the deviation of the queue size from a target value. We tested the PD controller with different target queue sizes. The target queue size reflects the network administrator's tradeoff among utilization, delay, and cell loss rate. A larger target value indicates the desire for higher utilization at the expense of higher delay and cell loss rate. To maintain fairness, the PD controller issues the same rate command to every controlled source at each decision epoch.

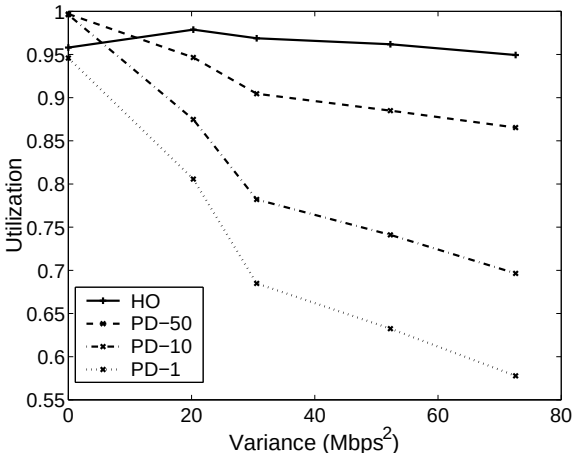


Fig. 3. Plots of utilizations.

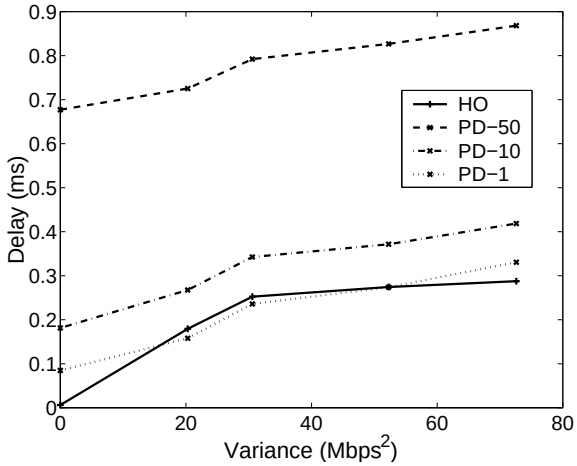


Fig. 4. Plots of average delays.

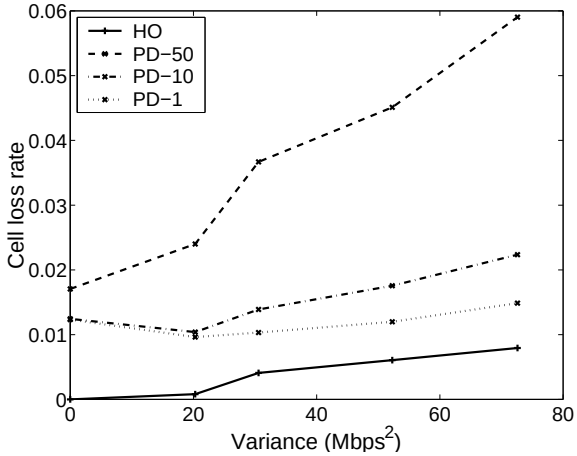


Fig. 5. Plots of cell loss rates.

For the HO controller, we generate 200 service-rate traces at each decision epoch using the cross-traffic model. Each trace is of length $H = 50$ time intervals. We chose the duration δ of each time interval to be 1 ms. The value of δ was chosen on one hand to be small enough to capture the fast variation in service rate and on the other hand large enough for affordable computation. In addition, 1 ms is a value of δ small enough to express typical round-trip delays as integral multiples of δ. We set $\alpha = 1000$, $\beta = 2/3$, and $\zeta = 1$. These values satisfy the restrictions on the values of α, β, and ζ that we have given in defining our objective function, in equation (1).

Figure 3 shows the utilization values achieved by the competing controllers. The PD controller with a target queue size of 50 cells is denoted by PD-50, and similarly for PD-10 and PD-1. The horizontal axis is the rate variance of the cross traffic. We can see that all controllers achieve high utilization, when constant-rate cross traffic is present. As the cross traffic becomes more variable, the utilization values achieved by the PD controllers decrease at rates much faster than that of the HO controller. The reason is that when the cross traffic is highly variable, the PD controllers cannot maintain a stable queue size to ensure satisfactory utilization. For example, sudden increases in the service rate can drain the cells waiting in the buffer completely, leading to loss in utilization. In contrast, by making use of the service-rate model, the HO controller can stochastically "anticipate" changes in service rate and can in turn respond to these changes beforehand. Figure 3 demonstrates the effectiveness of the HO controller in an environment with a highly-variable service rate, a condition which is often found in practical networks.

Figure 4 shows plots of the average queuing delays. The HO controller achieves much smaller queuing delays than those of PD-50 and PD-10. PD-1 has queuing delays close to those of HO; however, it does so at the significant cost of much smaller utilizations (see Figure 3). Compared with HO, PD-50 has much larger delay and less utilization; it has more than four times the delay and 8% less utilization in the most variable service rate case, i.e., the rightmost points in the figures shown. Figures 4 and 3 suggest that the HO controller can achieve higher utilization with smaller delay compared with the PD controllers.

Figure 5 shows plots of cell loss rates (CLRs). The CLR for the HO controller is the smallest for all the experiments we carried out due to the fact that the implicit goal of a hindsight-optimal control sequence is to keep a zero queue length, and hence it leaves most of the buffer ready to absorb bursts of incoming traffic. The PD-50 controller, which achieves the closest utilization values to HO among the PD controllers, has a CLR that is about an order of magnitude larger than that of the HO controller in all our experiments.

Our simulation shows that the HO controller is less fair than the PD controllers: HO's cumulative fairness values are under 0.08 while those of PD's are negligible. (We omit the fairness plots for brevity.) While the magnitude of the fairness term is quite small, it is still important—in fact, it is possible to show that, for a wide class of reasonable traffic models, when $\beta = 0$ the HO controller will only request traffic from the closest source, starving all more distant sources.

As expected, the HO controller achieves the highest cu-

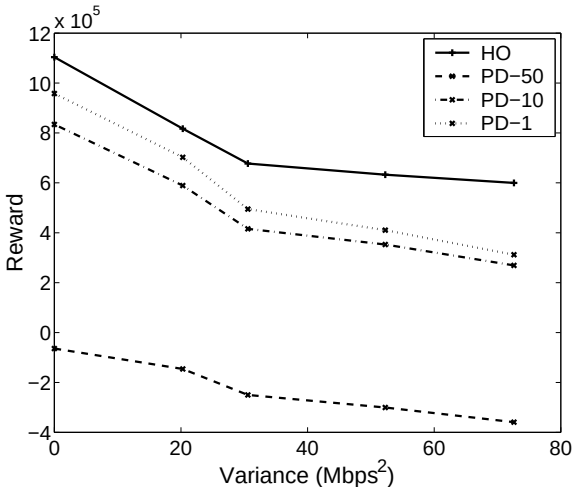


Fig. 6. Plots of cumulative reward.

mulative reward in all the experiments we conducted, as shown in Figure 6.

D. Voice Connections As Cross Traffic

In this subsection, source 0 consists of 1000 identical voice connections. The parameters of the HO and PD controllers are the same as in the first part of our empirical study. Table 1 summarizes the performance comparison between the HO and PD controllers. In this experiment, the rate variance of the cross traffic is relatively small (1.2 Mbps^2), and therefore the PD controllers achieve good utilizations. However, the HO controller, while maintaining high utilization value, enjoys much smaller queuing delay and cell loss rate.

Table 1: Performance comparison using 1000 voice connections as cross traffic

Controller	Util	Delay (ms)	CLR	UF
HO	0.995	0.123	0.00	6.79e-2
PD-50	0.998	0.505	4.12e-3	8.88e-4
PD-10	0.957	0.213	5.78e-4	6.87e-4
PD-1	0.925	0.106	1.78e-4	9.14e-4

Util = Utilization CLR = Cell Loss Rate UF = Unfairness

V. Conclusions

We have introduced a simulation-based burst-level congestion controller for a generic network to regulate best-effort traffic to achieve high network efficiency while maintaining fairness, represented by high utilization, traffic loss, and low queuing delay. We have demonstrated that exploiting the structure of service-rate models can result in significantly improved network performance.

Our empirical study demonstrates the effectiveness of the HO controller with rapidly-varying cross traffic. In such situations, conventional PD controllers lose their ability to stablize the queue size, leading to decreased throughput, increased queuing delay, and increased cell loss rate. The use of a cross-traffic model provides a clear advantage to the HO controller, enabling prediction and reaction to likely traffic variations. Moreover, the HO controller does not require the tuning of controller gains, a nontrivial task in PD controller design.

While the proposed control scheme is promising, two main issues remain to be addressed:
1) Our hindsight-optimization framework is founded on a crisp and powerful decision-theoretic formulation, but little is understood on general conditions under which the technique works well.
2) To incorporate a long-range-dependence traffic model into our control scheme is an interesting direction worth pursuing. Such a model can be made Markovian but with a potentially large state space. Managing the size of the state space but still capturing the long-range dependence is important for our approach to apply in this case.

Appendices

I. Proof of Proposition 1

Before presenting the proof of Proposition 1, we give two lemmas needed for the proof.

Lemma 1: Consider the function $f(w_1,\ldots,w_N) = \frac{1}{N-1}\sum_{i=1}^{N}\sum_{j=i+1}^{N}|w_i - w_j|$. Then we have:

i. $f(w'_1,\ldots,w'_N) \geq f(w_1,\ldots,w_N) - \sum_{i=1}^{N}|w_i - w'_i|$.

ii. $f(w'_1,\ldots,w'_N) \geq f(w_1,\ldots,w_N)$ holds when $\sum_{i=1}^{N} w'_i = \sum_{i=1}^{N} w_i$ and $w'_i \neq w_i$ imply $w_i = c$ for all i, where c is a constant.

The lemma can be proven with ease using elementary algebra, which we omit here.

Lemma 2: For a given service-rate trace t, starting state $\vec{x}_0$, and initial control $\vec{u}_0$, the hindsight-optimal value, $W_t^*(\vec{x}_0,\vec{u}_0)$, can be written as follows:

$$W_t^*(\vec{x}_0,\vec{u}_0) = T_t^*(\vec{x}_0,\vec{u}_0) - \alpha D_t^*(\vec{x}_0,\vec{u}_0) - \zeta L_t^*(\vec{x}_0,\vec{u}_0) - \sum_{k=0}^{H-1}\beta F_t^*(\vec{x}_0,\vec{u}_0,k),$$

where $T_t^*(\vec{x}_0,\vec{u}_0)$, $D_t^*(\vec{x}_0,\vec{u}_0)$, and $L_t^*(\vec{x}_0,\vec{u}_0)$ are the maximal possible throughput, the minimum possible delay, and the minimum possible loss, respectively, achievable from $\vec{x}_0$ taking $\vec{u}_0$ and encountering trace t. The term $F_t^*(\vec{x}_0,\vec{u}_0,k)$ is the minimum unfairness penalties at time k subject to optimizing $T_t^*(\vec{x}_0,\vec{u}_0)$, $D_t^*(\vec{x}_0,\vec{u}_0)$, and $L_t^*(\vec{x}_0,\vec{u}_0)$.

Proof: Let t, $\vec{x}_0$, and $\vec{u}_0$ be given. The cumulative reward, $W_t(\vec{x}_0,\vec{u}_0)$, is a continuous function on $\mathbf{L}\times\mathbf{U}^{d^{(N)}}$, which is a compact set. By the Weierstrass theorem [16], there is a point in $\mathbf{L}\times\mathbf{U}^{d^{(N)}}$ such that at the point $W_t(\vec{x}_0,\vec{u}_0)$ achieves its maximum, $W_t^*(\vec{x}_0,\vec{u}_0)$. For a given initial condition, the queue length, taking values in $\mathbf{L}$, is completely determined by the control in $\mathbf{U}^{d^{(N)}}$. Hence, we conclude that there is a control sequence in $\mathbf{U}^{d^{(N)}}$ that achieves $W_t^*(\vec{x}_0,\vec{u}_0)$. We refer to any such control sequence an *optimal control*.

We now argue that any optimal control must optimize throughput, delay, and loss *independently*, in the following sense. The trace-relative queue-size trajectory $\{l_k^t$,

$k = 0, \ldots, H\}$ is the trajectory the system will follow if no further fluid is requested by controls sent after time 0. The loss and delay experienced under this trajectory clearly lower bound those achieved by requesting further fluid—more fluid requested leads to potentially more loss and more delay, but no less. These loss and delay lower bounds are $D_t^*(\vec{x}_0, \vec{u}_0)$ and $L_t^*(\vec{x}_0, \vec{u}_0)$, respectively. The lemma claims that any optimal control must result in delay of $D_t^*(\vec{x}_0, \vec{u}_0)$ and loss of $L_t^*(\vec{x}_0, \vec{u}_0)$. Recall that $d^{(1)}$ is the earliest time any control can affect the bottleneck node. Then, it is apparent that $T_t^*(\vec{x}_0, \vec{u}_0)$ is the sum of three terms: the throughput received from time 0 to $d^{(1)}-1$ (affected only by the control history recorded in $\vec{w}_0$), the throughput at time $d^{(1)}$ (determined by $\vec{u}_0$), and the sum of service rates after $d^{(1)}$ (determined by a particular control sequence after time 0). The lemma claims that any optimal control must also fully use the service rate after time $d^{(1)}$, thus achieving $T_t^*(\vec{x}_0, \vec{u}_0)$.

We prove these claims by showing in the following that any control that does not optimize throughput, delay, and loss independently can be improved. If a control does not fully use the service rate at time k (after time $d^{(1)}$), as indicated by a negative $\tilde{l}_{k+1}^t$, we can increase the arrival at time k to obtain a larger throughput without incurring any delay and loss until $\tilde{l}_{k+1}^t$ reaches zero from below; $\tilde{l}_{k+1}^t = 0$ indicates that the service rate is fully used. Furthermore, the potential fairness penalty due to the increased arrival will always be less than the gain in throughput because $\beta < 1$. Hence, an optimal control must maximize throughput. If a control causes more delay than the delay lower bound at time k (indicated by a positive $\tilde{l}_{k+1}^t$), we can defer the arrival of a sufficiently small amount of fluid unserved at time k until time $k+1$ to decrease delay while keeping throughput fully utilized. In the case of excess loss, a similar change can be made by cancelling the arrival of a sufficiently small amount of fluid unserved at time k. Moreover, considering part i of Lemma 1 and the restrictions on α, β, and ζ, it is easy to show that the fairness penalty from this deferred or reduced arrival will always be smaller than the gain from the reduced delay and/or loss. Thus, an optimal control must minimize delay and loss.

The argument above implies that the optional traffic arrival at any time k ($> d^{(1)}$) from an optimal control will not affect any other time step (because no delay other than D_t^* is experienced), and thus we conclude that an optimal control maximizes reward received at time k independently of optimization at any other time steps. In particular, an optimal control optimizes the fairness component of the reward at a time independently of fairness optimization at any other times (where each fairness optimization is done subject to the constraint that throughput, delay, and loss are already optimized independently). Then, a control that optimizes throughput, delay, and loss independently, and, subject to this, optimizes fairness at each time is an optimal control. This completes the proof. ■

Now, we present the proof of Proposition 1.

Proof: The control given in the proposition is easily seen to independently and simultaneously optimize throughput, delay, and loss; i.e., the control achieves $T_t^*(\vec{x}_0, \vec{u}_0)$, $D_t^*(\vec{x}_0, \vec{u}_0)$, and $L_t^*(\vec{x}_0, \vec{u}_0)$. Furthermore, subject to optimizing $T_t^*(\vec{x}_0, \vec{u}_0)$, $D_t^*(\vec{x}_0, \vec{u}_0)$, and $L_t^*(\vec{x}_0, \vec{u}_0)$, the control minimizes fairness at each time step, by Lemma 1 part ii. Therefore, by Lemma 2 the desired result follows. ■

II. Proof of Proposition 2

Proof: Note that by Lemma 2 the hindsight-optimal value can be decomposed into the throughput, delay, loss, and fairness terms. We analyze the gradient of $W_t^*(\vec{x}_0, \vec{u}_0)$ with respect to $\vec{u}_0$ for each term separately in the following.

Throughput, Delay, and Loss. The hindsight-optimal control achieves full utilization of link bandwidth after time $d^{(1)}$ without incurring more delay or loss than the control history specified by $\vec{w}_0$. Hence, regardless of what value we set for $\vec{u}_0$, the hindsight-optimal throughput, loss, and delay will be the same for a given trace, except possibly for throughput and loss experienced at time step $d^{(i)}$, and for delay experienced by flow arriving at time step $d^{(i)}$ (and possibly any time after $d^{(i)}$ for delay) for any value of i.

An infinitesimal increase in arrivals will result in an increase of the same amount in the hindsight-optimal throughput if and only if the source in question is source one (i.e., $i = 1$) and there is an underflow at time $d^{(1)}$ under the current $\vec{u}_0$ (as indicated by finding $k_{\downarrow}^{t,1}(\vec{u}_0) = d^{(1)}$). Changes in $\vec{u}_0$ for sources other than source one impact the trajectory after $d^{(1)}$ where the hindsight-optimal throughput is already maximal and cannot be increased by further arrivals (these arrivals simply have the effect of leaving less bandwidth for exploitation by hindsight-optimal controls selected for later times for closer sources). Thus, we have

$$\nabla_{\vec{u}} T(\vec{x}_0, \vec{u}_0)^{(i)} = \begin{cases} 1 & \text{if } i = 1 \text{ and } k_{\downarrow}^{t,1}(\vec{u}_0) = d^{(1)}, \\ 0 & \text{otherwise.} \end{cases}$$

An increase in the arrivals for source i can increase delay at time $d^{(i)}$ and for arbitrarily many time steps thereafter—until such time that the controls in the pipeline (along with $\vec{u}_0$) allow some leftover service bandwidth (assuming no further fluid is requested after time zero) or until the incremental arrival is dropped due to buffer overflow. The delay suffered by the incremental arrival will equal the incremental arrival multiplied by $k^{t,i} - d^{(i)}$. Thus, we have

$$\nabla_{\vec{u}} D(\vec{x}_0, \vec{u}_0)^{(i)} = k^{t,i} - d^{(i)}.$$

The increase in arrivals at time $d^{(i)}$ will result in increased dropping in the hindsight-optimal case if and only if the increase eventually results in buffer overflow, i.e., if the next buffer-full time (after $d^{(i)}$) occurs before any buffer underflow (unused service rate). This condition can be detected by comparing the next buffer-full time to the next buffer-underflow time. Hence, we have

$$\nabla_{\vec{u}} L(\vec{x}_0, \vec{u}_0)^{(i)} = \begin{cases} 1 & \text{if } k_{\uparrow}^{t,i} < k_{\downarrow}^{t,i}, \\ 0 & \text{otherwise.} \end{cases}$$

If $l^t_{k+1}(\vec{u}_0) = 0$ or $l^t_k(\vec{u}_0) + a^t_k(\vec{u}_0) - v_{s_{k+1}} = B$ in calculating $k^{t,i}_\uparrow$ or $k^{t,i}_\downarrow$, then $T(\vec{x}_0, \vec{u}_0)^{(i)}$, $D(\vec{x}_0, \vec{u}_0)^{(i)} = k^{t,i} - d^{(i)}$, and $L(\vec{x}_0, \vec{u}_0)^{(i)}$ are non-differentiable due to piecewise linearity of these functions.

Fairness. The gradient of the fairness term involves two parts. The first part is the change in fairness penalty caused by an infinitesimal change in $u_0^{(i)}$ by comparing $u_0^{(i)}$ with the sources in $\mathbf{N}^{(i)}$. Increases in $u_0^{(i)}$ will decrease the fairness penalty terms comparing source i with sources in $\mathbf{N}^i_>(u_0^{(i)})$ and increase the fairness penalty terms comparing source i to sources in $\mathbf{N}^i_<(u_0^{(i)})$, so that the total change in the fairness penalty terms comparing source i with sources in $\mathbf{N}_i$ will be $(N^i_<(u_0^{(i)}) - N^i_>(u_0^{(i)}))/(N-1)$. Note this part of the gradient does not exist when $u_0^{(i)} = w_0^{(d^{(j)} - d^{(i)}, j)}$ for any $j > i$(in particular, the effects of positive and negative infinitesimal perturbations of $u_0^{(i)}$ do not sum to zero).

The second part is the changes in fairness penalty terms involving sources not in $\{i\} \cup \mathbf{N}_i$ caused by an infinitesimal increase in $u_0^{(i)}$, and this part can be divided into four cases:
Case 1. When $i = 1$ there is no change in this portion of the fairness penalty.
Case 2. Otherwise, when $\tilde{l}^t_{d^{(i)}+1}(\vec{u}_0) > 0$, sources numbered less than i will be controlled to send no fluid to arrive at time $d^{(i)}$ in the hindsight-optimal control sequence for the candidate action $\vec{u}_0$ with or without the infinitesimal increase in source i; as a result the only change in the fairness penalty terms comparing source i with sources closer than i will be due to the increase in source i, moving the source i arrival rate (at time $d^{(i)}$) further from zero. These changes amount to one $(N-1)$th of one (infinitesimal) step penalty for the term for each source less than i, or a total additional penalty of $(i-1)/(N-1)$.
Case 3. Otherwise, if $r < u_0^{(i)}$, where $r = -\tilde{l}^t_{d^{(i)}+1}(u_0^{(i)})/(i-1)$ as given in the proposition, then the $i-1$ closest sources decrease by $1/(i-1)$ of the infinitesimal increase and the resulting fairness term changes show an increase in penalty equal to $(i + N^i_>(r) - N^i_<(r))/(N-1)$, with the first term being from the increased distance between the arrival rates from closer sources and the arrival rate from source i, and the last two terms being from the change in comparisons between the closest $i-1$ source arrivals and the arrivals from sources further away than source i.
Case 4. Otherwise, if $r > u_0^{(i)}$ then the decrease in hindsight-optimal arrivals at time $d^{(i)}$ for the closest $i-1$ sources actually improves fairness in the terms comparing to source i, giving the resulting change in fairness of $(-i + N^i_>(r) - N^i_<(r))/(N-1)$.

The second part of the gradient does not exist when $r = u_0^{(i)}$ or $r = w_0^{(d^{(j)} - d^{(i)}, j)}$ for any $j > i$ due to piecewise linearity in the fairness term.

Summarizing, we have:

$$\nabla_{\vec{u}} F(\vec{x}_0, \vec{u}_0)^{(i)} = \frac{1}{N-1}(N^i_<(u_0^{(i)}) - N^i_>(u_0^{(i)}) + \nabla_{\vec{u}} F_1^{(i)}),$$

where

$$\nabla_{\vec{u}} F_1^{(i)} = \begin{cases} 0 & \text{if } i = 1 \\ i-1 & \text{else if } \tilde{l}^t_{d^{(i)}+1}(u_0^{(i)}) > 0 \\ i + N^i_>(r) - N^i_<(r) & \text{else if } r < u_0^{(i)} \\ -i + N^i_>(r) - N^i_<(r) & \text{else if } r > u_0^{(i)}. \end{cases}$$

Combining the above arguments, we have the desired result. ■

References

[1] N. B. Shroff and M. Schwartz, "Improved loss calculations at an ATM multiplexer," *IEEE/ACM Trans. Networking*, vol. 6, no. 4, pp. 411–421, Aug. 1998.

[2] L. A. Kulkarni and S.-Q. Li, "Performance analysis of a rate-based feedback control scheme," *IEEE/ACM Trans. Networking*, vol. 6, no. 6, pp. 797–810, Dec. 1998.

[3] R. Pazhyannur and R. Agrawal, "Feedback-based flow control of B-ISDN/ATM networks," *IEEE J. Select. Areas Commun.*, vol. 13, no. 7, pp. 1252–1266, Sep. 1995.

[4] E. K. P. Chong, R. L. Givan, and H. S. Chang, "A framework for simulation-based network control via Hindsight Optimization," in *Proc. 39th IEEE Conf. on Decision and Control*, Sydney, Australia, Dec. 2000, pp. 1433–1438.

[5] D. P. Bertsekas, *Dynamic Programming and Optimal Control, Volumes 1 and 2*, Athena Scientific, 1995.

[6] K. Ramakrishnan and R. Jain, "A binary feedback scheme for congestion avoidance in computer networks," *ACM Trans. Comput. Syst.*, vol. 8, pp. 158-181, 1990.

[7] L. Roberts, "Enhanced proportional rate control algorithm (PRCA)," in ATM Forum/9494-0735R1, Aug. 1994.

[8] A. W. Barnhart, "Explicit rate performance evaluations," in ATM Forum/94-0983, Sep. 1994.

[9] L. Benmohamed and S. M. Meerkov, "Feedback control of congestion in packet switching networks: The case of a single congested node," *IEEE/ACM Trans. Networking*, vol. 1, no. 6, pp. 693–707, Dec. 1993.

[10] A. Kolarov and G. Ramamurthy, "A control-theoretic approach to the design of an explicit rate controller for ABR service," *IEEE/ACM Trans. Networking*, vol. 7, no. 5, pp. 741–753, Oct. 1999.

[11] E. Altman and T. Basar, "Multiuser rate-based flow control," *IEEE Trans. Commun.*, vol. 46, no. 7, pp. 940–949, Jul. 1998.

[12] E. Altman, T. Basar, and R. Srikant, "Robust rate control for ABR services," in *Proc. IEEE INFOCOM*, Mar. 1998, San Francisco, CA, pp. 166–173.

[13] Z. Pan, E. Altman, and T. Basar, "Robust adaptive flow control in high speed telecommunications networks," in *Proc. 35th IEEE Conf. on Decision and Control*, Dec. 1996, Kobe, Japan, pp. 1341–1346.

[14] S. Floyd and V. Jacobson, "Link-sharing and resource management models for packet networks," *IEEE/ACM Trans. Networking*, vol. 3, no. 4, pp. 365–386, Aug. 1995.

[15] T. Tuan and K. Park, "Multiple time scale congestion control for self-similar network traffic," *Performance Evaluation*, vol. 36-37, pp. 359–386, Aug. 1999.

[16] E. K. P. Chong and S. H. Zak, *An Introduction to Optimization*, John Wiley and Sons, New York, 1996.

[17] S. C. Liew and D. C. Tse, "A control-theoretic approach to adapting VBR compressed video for transport over a CBR communication channel," *IEEE/ACM Trans. Networking*, vol. 6, no. 1, pp. 42–55, Feb. 1998.

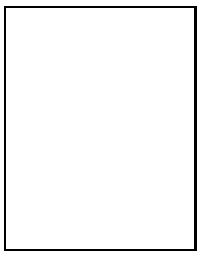
[18] I. Rubin and K. D. Lin, "A burst-level adaptive input-rate flow control scheme for ATM networks," in *Proc. IEEE INFOCOM*, 1993, San Francisco, CA, vol. 2, pp. 386–394.

[19] D. Q. Mayne and H. Michalska, "Receding horizon control of nonlinear systems," *IEEE Trans. Auto. Contr.*, vol. 35, no. 7, pp. 814–824, Jul. 1990.

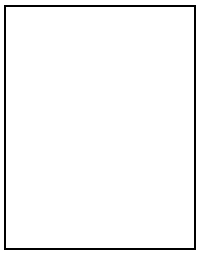
[20] J. B. Rawlings and K. R. Muske, "The stability of constrained receding horizon control," *IEEE Trans. Auto. Contr.*, vol. 38, no. 10, pp. 1512–1516, Oct. 1993.

[21] J. M. Jaffe, "Bottleneck flow control," *IEEE Trans. Commun.*, vol. 29, no. 7, pp. 954–962, Jul. 1981.

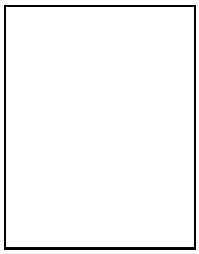
[22] F. Bonomi and K. Fendick, "The rate-based flow control framework for the Available Bit Rate ATM service," *IEEE Network*, vol. 9, no. 2, pp. 25-39, Mar./Apr. 1995.
[23] K. Bharath-Kumar and J. M. Jaffe, "A new approach to performance-oriented flow," *IEEE Trans. Commun.*, vol. 29, no. 4, pp. 427–435, Apr. 1981.
[24] V. M. Misra and W. B. Gong, "A hierarchical model for teletraffic," in *Proc. 37th IEEE Conf. Decision and Control*, 1998, Tampa, FL, vol. 2, pp. 1674–1679.
[25] R. Jain, "Congestion control in computer networks: Issues and trends," *IEEE Network*, vol. 4, no. 3, pp. 24–30, May 1990.
[26] R. Pazhyannur and R. Agrawal, "Feedback based flow control in ATM networks with multiple propagation delays," in *Proc. IEEE INFOCOM*, 1996, San Francisco, CA, vol. 2, pp. 585–593.
[27] Y.-C. Ho and X.-R. Cao, *Perturbation Analysis of Discrete Event Dynamic Systems*, Kluwer Academic Publishers, Boston, 1991.
[28] L. R. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, *Proc. of the IEEE*, vol. 77, no. 2, pp. 257–285, Feb. 1989.
[29] M. L. Littman, *Algorithms for Sequential Decision Making*, Ph.D. Thesis, Department of Computer Science, Brown University, 1996.
[30] I. Lengliz and F. Kamoun "A rate-based flow control method for ABR service in ATM networks," *Computer Networks*, vol. 34, no. 1, pp. 129–138, Jul. 2000.
[31] H. Zhang, "Service Disciplines For Guaranteed Performance Service in Packet-Switching Networks," *Proc. of the IEEE*, vol. 83, no. 10, pp. 1374-1396, Oct. 1995.
[32] E. K. P. Chong, S. Hui, and S. H. Żak, "An Analysis of a Class of Neural Networks for Solving Linear Programming Problems," *IEEE Trans. Auto. Contr.*, Special Section on *Neural Networks in Control, Identification, and Decision Making*, vol. 44, no. 11, pp. 1995–2006, Nov. 1999.
[33] P. Kermani and L. Kleinrock, "Dynamic flow control in store-and-forward computer networks," *IEEE Trans. Commun.*, vol. COM-28, no. 2, pp. 263–271, Feb. 1980.
[34] E. Altman and P. Nain, "Closed-loop control with delayed information," *Performance Evaluation Review*, vol. 20, no. 1, pp. 193–204, Jun. 1992.
[35] J. Kuri and A. Kumar, "Optimal control of arrivals to queues with delayed queue length information," *IEEE Trans. Auto. Contr.*, vol. 40, no. 8, pp. 1444–1450, Aug. 1995.
[36] D. P. Bertsekas and D. A. Castanon, "Rollout algorithms for stochastic scheduling problems," *Journal of Heuristics* , vol. 5, pp. 89–108, 1999.



Gang Wu received the B.E. degree in automatic control from Shanghai Jiao Tong University, China, in 1993; and the M.Sc. degree in industrial engineering from the University of Manitoba, Canada, in 1997. He is currently a Ph.D. candidate in the School of Electrical and Computer Engineering at Purdue University. His research interests are in developing new rate control methods to substantially improve performance of communication networks using stochastic traffic models. He was awarded a Canadian Ontario Graduate Scholarship, and he was also a recipient of the Canadian National Science and Engineering Research Council Postgraduate Scholarship, both in 1998.



Edwin K. P. Chong received the B.E.(Hons.) degree with First Class Honors from the University of Adelaide, South Australia, in 1987; and the M.A. and Ph.D. degrees in 1989 and 1991, respectively, both from Princeton University, where he held an IBM Fellowship. He joined the School of Electrical and Computer Engineering at Purdue University in 1991, where he was named a University Faculty Scholar in 1999, and was promoted to Professor in 2001. Since August 2001, he has been a Professor of Electrical and Computer Engineering and a Professor of Mathematics at Colorado State University. His current interests are in communication networks and optimization methods. He coauthored the recent book, An Introduction to Optimization, 2nd Edition, Wiley-Interscience, 2001. He was on the editorial board of the IEEE Transactions on Automatic Control, and is currently an editor for Computer Networks. He is an IEEE Control Systems Society Distinguished Lecturer. He received the NSF CAREER Award in 1995 and the ASEE Frederick Emmons Terman Award in 1998.



Robert Givan received the B.S. degree with distinction in Mathematics and Biology and the M.S. Degree in Computer Science from Stanford University in 1987; and the Ph.D. degree in 1996 from the Massachusetts Institute of Technology, where he held both National Science Foundation and Fannie and John Hertz Foundation fellowships. After a one year post-doctoral position at Brown University, he joined the School of Electrical and Computer Engineering at Purdue University in August, 1997. His interests are in machine learning, planning, representation, and reasoning in the field of artificial intelligence. He received the NSF CAREER Award in February, 2001.