Optimal Dynamic Coding-Window Selection for Serving Deadline-Constrained Traffic Over Time-Varying Channels

Ruogu Li, *Student Member, IEEE*, Harsha Gangammanavar, and Atilla Eryilmaz, *Member, IEEE*

***Abstract*—We formulate and solve the problem of optimal channel coding and flow-rate control for serving deadline-constrained traffic with average delivery ratio requirements (typical of multimedia streaming and interactive real-time applications) over time-varying channels. To that end, we first characterize the largest set of *arrival processes* (rather than rates) whose deadline and delivery ratio requirements can be satisfied. Then, we propose a dynamic (channel) coding algorithm that provably satisfies the requirements of any arrival process in this region. This optimal dynamic algorithm evolves through simple iterations to utilize a combination of *pricing* and finite-horizon dynamic programming operations. Next, we proposed two low-complexity approximations of the algorithm that has provable performance. We also extend the setup to allow for a flow controller that adjusts the incoming flow rates to satisfy their delivery ratio constraints when the arrival process is unknown but controllable. We propose a joint dynamic coding and a rate control algorithm to solve this problem, and prove its stability under the stochastic system operation. We also apply these general results to an important wireless down-link broadcast scenario with and without random network coding capabilities. Our theoretical work is supported by extensive numerical studies, which also reveal that our dynamic coding strategy outperforms any static coding strategy by opportunistically exploiting the statistical variations in the arrival and channel processes.**

***Index Terms*—Deadline-constrained throughput optimization, delay-aware dynamic coding, network coding, stochastic control.**

I. Introduction

While the traditional performance measure of a communication system is *throughput*, many real-world applications also have a range of delay-sensitivities and Quality-of-Service (QoS) requirements that are typically not accounted for. In particular, real-time media broadcasting or two-way voice/video communication applications possess requirements at different timescales: stringent deadline constraints in the short term and differing tolerance levels to long-term fraction of dropped bits. Such multitimescale requirements prevent the application of earlier approaches that are based on optimizing long-term average metrics. Moreover, different flows entering the system may have different degrees of importance, necessitating prioritization of certain flows over the others. Also, these flows may need to be transmitted over randomly changing channels, as is the case in wireless communications.

Information theory reveals that there is a fundamental relationship between the reliable transmission rate and the *coding block* (also called the *coding window*) *size* used to map messages into transmission signals. In particular, the reliable transmission rate may be increased toward the capacity of the channel by increasing the coding window size [4]. However, increasing the coding window size also causes larger delay and in the presence of the aforementioned deadline-constrained traffic it becomes unacceptable beyond a level. Thus, a radically different coding strategy must be employed by the transmitter to maximize the delay-sensitive applications' performance under the time-varying conditions of the channels. Since the channel and application characteristics are often stochastic, the solution must be able to adapt to their randomness.

Queueing systems under impatient customers have been studied in the literature (e.g., [2], [16], [23]) for various cases of preemption, arrival/service rate distributions, etc. Yet, these works do not model the priorities and tolerance levels of applications, and do not account for possible coding parameters, and hence are not applicable to our problem. Also, recent works [9], [10], [12] have studied the congestion-control and scheduling problem for similar deadline constrained traffic with reliability constraints. However, they also do not allow for coding flexibilities, which fundamentally changes the shape of the achievable rate region and calls for a dynamic strategy for optimizing over coding decisions. Other related works that deal with deadline-constrained traffic include [19], [17], which focus on single-flow scenarios and hence do not apply to the aforementioned multiflow scenario.

Motivated by these, in this paper, we study the basic scenario of a single transmitter serving multiple delay-sensitive flows under randomly varying channel conditions. Since the transmission time and rates are functions of the coding window size and content, they must be carefully chosen to meet the multitimescale QoS requirements of the flows. To capture the effects of channel randomness and coding decisions, we model the random transmission time of a coding window over the

Manuscript received December 02, 2010; revised December 08, 2011; accepted March 13, 2012. Date of publication June 12, 2012; date of current version September 11, 2012. R. Li was supported in part by the Qatar National Research Fund under Grant NPRP 09-1168-2-455 and in part by the Defense Threat Reduction Agency under Grant HDTRA 1-08-1-0016. A. Eryilmaz was supported by the National Science Foundation under Grants CAREER-CNS-0953515 and CCF-0916664. The material in this paper was presented at the 2010 IEEE International Symposium on Information Theory.

The authors are with The Ohio State University, OH 43210 USA (e-mail: lir@ece.osu.edu; gangammh@ece.osu.edu, eryilmaz@ece.osu.edu).

Communicated by R. Yates, Associate Editor for Communication Networks.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIT.2012.2204031

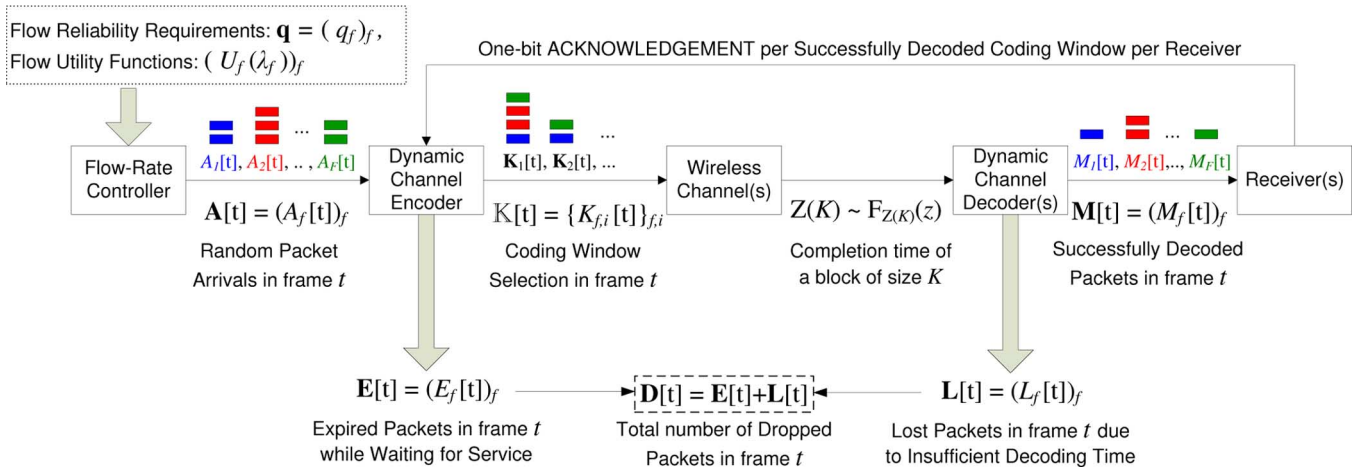

Fig. 1. Block diagram of the generic communication system for broadcasting deadline-constrained flows with varying reliability requirements and priorities.

fading channel by a completion time distribution.[1] In this setup, we aim to find an optimal rate-control and coding strategy for serving the aforementioned deadline-constrained flows with diverse reliability requirements (in terms of delivery ratios) and heterogenous priorities. A list of our contributions is as follows

1) We provide a generic communication system model (in Section II) with a multitimescale operation that accounts for the heterogenous priorities and the multitimescale delay sensitivities of the incoming traffic. This model is general enough to cover different block coding strategies and channel variation models.
2) We address the dynamic coding problem (DCP) for systems with fixed arrival statistics (in Section III). To that end, we first characterize the maximal set of arrival processes whose requirement can be satisfied by any stationary policy. Then, we propose and prove the optimality of a dynamic coding algorithm operating at different timescales: it uses dynamic pricing at a slow timescale to monitor the violation of the reliability requirements; and it uses finite-horizon dynamic programming at a fast timescale to determine the coding block size and its content.
3) We develop two low-complexity approximations to our dynamic coding algorithm (in Section IV-A). The first is based on the descretization of the state space, and has provably asymptotic optimality, while the second is a greedy algorithm. Both algorithms avoid solving the finite-horizon dynamic programming problem online and thus greatly reduce the computational complexity.
4) We extend our algorithm (in Section IV-B) to incorporate a flow controller that aims to satisfy their long-term reliability requirements when the arrival process is unknown. Our joint rate control and dynamic coding algorithm utilizes a primal-dual update rule rather than the more common dual update rule since the latter requires the solution of a large linear program in every iteration. We rigorously prove the stability of the joint algorithm for the stochastic system under appropriate step-size choices.
5) We apply (in Section V) the developed algorithm to an important application in cellular down-link scenario whereby a base station (BS) broadcasts multiple streaming deadline-constrained flows to N receivers over randomly varying erasure channels. We further study (in Section VI) the performance of different network coding strategies with our dynamic coding algorithm, and compare the performance of our dynamic coding algorithm to a static one to see strict improvements, even for small scenarios.

We note that this work extends our preliminary work [8] in various aspects: we characterize the set of arrival processes whose requirement can be satisfied by any stationary policy; we establish the optimality of our dynamic coding algorithm under the stochastic system operation rather than through a heuristic fluid-limit argument; we discuss and propose two low-complexity approximation algorithms of the dynamic coding algorithm; we extend the numerical results to study a larger and more realistic range of systems and requirement parameters for a deeper understanding.

II. System Model

We study the general communication system depicted in Fig. 1, whereby a transmitter serves a set $\mathcal{F}$ of flows, whose packets have a deadline of τ time slots after their arrival, over unreliable wireless channel(s). The arrivals of each flow f occur every τ time slots, and a fraction of $(1 - q_f)$ packets are required to be delivered within their deadlines. Our study concerns the optimal design of the flow-rate controller and the dynamic channel encoder–decoder pair that operate at different time scales. Next, we describe the system components, their operational constraints, and the application requirements in detail.

Multitimescale Operation: We set up a multitimescale system operation whereby the flow-rate controller is allowed to operate in the slower timescale of flow-level deadline constraints than the fast timescale of channel variations at which

[1]We note that this is an alternative description of the coding performance to the traditional one that describes the probability of decoding error for a fixed transmission duration. We find that this alternative model is more useful for our design and analysis.

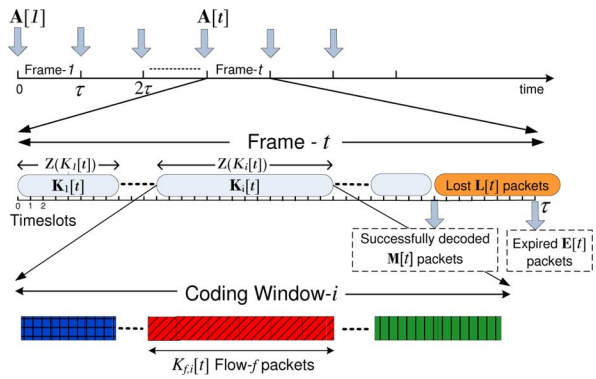

Fig. 2. Operation of the communication system of Fig. 1 over time.

the dynamic channel encoder/decoder operate. Accordingly, we use *time slots* as the smallest time unit in which channel variations occur, and in which the channel encoder/decoder operate. In comparison, the flow-level deadline constraints are at a slower time-scale which we call a *frame* of τ times-slots, within which each incoming packet is either successfully delivered to the destinations or otherwise dropped (see Fig. 2).

Arrival Process: We first assume that the arrival process is a fixed discrete stochastic process to focus on the design of the dynamic coding scheme in Section III, and then extend it to a dynamic controllable process for joint rate control and dynamic coding in Section IV-B.

The fixed arrival process is assumed to be stationary and ergodic. In particular, at the beginning of frame t, the number of arrivals for flow f is denoted by $A_f[t]$, where $A_f[t]$ is nonnegative integer-valued, independently, and identically distributed (i.i.d.) over time, and has a finite upper-bound $A_{\max}$ such that $P(A_f[t] \leq A_{\max}) = 1$ for all frames t and flows f. We use the vector $\boldsymbol{A}[t] \stackrel{\triangle}{=} \{A_f[t]\}_f$ to denote the arrival vector to all flows in frame t.

Dynamic Channel Encoding/Decoding: The arrivals $\boldsymbol{A}[t]$ in frame t enter the dynamic channel encoder (see Fig. 1) that performs block coding for reliable transmission. In particular, the encoder collects the available packets into groups, called *coding windows* or simply *blocks*, for sequential transmission over the channel.

Let the column vector $\boldsymbol{K}_i[t] \stackrel{\triangle}{=} \{K_{f,i}[t]\}_f$ denote the composition of the ith coding window in frame t, whose indices $K_{f,i}[t]$ give the number of flow f packets in the ith coding window. Also, let $K_i[t] \stackrel{\triangle}{=} \sum_f K_{f,i}[t]$ denote the total number of packets in the ith block. The blocks are constructed and transmitted sequentially such that the $(i+1)$st block is constructed and transmitted only after the ith block is successfully decoded by all the intended receiver(s). The successful decoding of each block is indicated through a single bit ACK signal by the intended receiver(s) (see Fig. 1).

We emphasize that the amount of time required for the successful decoding of a coding block of size K, henceforth called the *completion time*, is a random variable that depends on K, the channel statistics, and the coding strategy employed by the encoder. It consists of the amount of time for all users to receive enough packets to decode a block, as well as the time for decoding that block.[2] Without restricting to any particular channel or scheme, we capture this randomness by letting $Z(K)$ denote the completion time (in time slots) of a block of size K that is generated independently according to a given distribution function $F_{Z(K)}(z)$. In Section V, we shall provide examples of such distribution functions in a down-link broadcast setup over erasure channels, but our current construction is applicable to any distribution.

We also note that the encoder only knows the *statistics* of $Z(K)$ at the outset of its block construction, and can deduce the realization of earlier block completion times through the acknowledgements it receives. To distinguish the two, we let $z(K)$ denote the realization of the random completion time $Z(K)$ with the convention that $z(K) = \infty$ if the frame ends before the block completion. Then, the time left in frame t at the beginning of the ith block transmission is denoted as $\tau_i \stackrel{\triangle}{=} \tau - \sum_{j=1}^{i-1} z(K_j[t])$. Similarly, the vector $\boldsymbol{A}_i[t] \stackrel{\triangle}{=} \{A_f[t] - \sum_{j=1}^{i-1} K_{f,j}[t]\}_f$ denotes the remaining packets of each flow at the beginning of the ith coding window construction within frame t (see Fig. 2). Thus, the construction of $\boldsymbol{K}_i[t]$ depends on the remaining time τ_i in the frame, the remaining packets $\boldsymbol{A}_i[t]$ awaiting service, and the distribution of $Z(K)$ that conveniently encapsulates the channel and coding capabilities in it.

For notational convenience, we use the matrix

$$\mathbb{K}[t] \stackrel{\triangle}{=} [\boldsymbol{K}_1[t], \boldsymbol{K}_2[t], \ldots, \boldsymbol{K}_b[t], \boldsymbol{E}[t]]$$

to compactly refer to the sequence of such coding window decisions made in frame t, where $\boldsymbol{E}[t] \stackrel{\triangle}{=} \{E_f[t]\}_f$ is the vector of number of packets of each flow that never get a chance to start their transmission before the end of frame t, and b is the number of blocks constructed in frame t. Note that since the construction of the blocks is limited by the available number of packets in that frame, we must have1.5pi

$$\sum_{i=1}^{b} \boldsymbol{K}_i[t] + \boldsymbol{E}[t] = \boldsymbol{A}[t].$$

We use $\mathcal{K}(\boldsymbol{A}[t])$ to denote the set of all *possible* matrices $\mathbb{K}[t]$ that satisfies the aforementioned equation when the arrival vector is $\boldsymbol{A}[t]$.

The process of block construction in frame t can be interpreted equivalently as choosing one of the *controls* $\mathbb{K}[t]$ from the set $\mathcal{K}(\boldsymbol{A}[t])$, regardless of how the actual system chooses the coding blocks. In this sense, we will refer to $\mathbb{K}[t]$ as a *control* or a coding block matrix decision interchangeably.

[2]We note that one can also model the feedback delay (which we assume to be 0) as part of the completion time (thus changing the completion time distribution). Our proposed algorithms operate unmodified under this new definition of the completion time. However, we can not claim optimality when there is feedback delay, since if we have some statistics of the feedback delay, it is possible to start the transmission of the next coding block before receiving all the feedbacks to improve performance.

We further denote the set of all possible coding block matrix choices as

$$\mathcal{K} \stackrel{\triangle}{=} \{\mathcal{K}(\boldsymbol{a}) : \boldsymbol{a} \text{ is a possible arrival vector}\}. \quad (1)$$

We note that since the arrival process $\boldsymbol{A}[t]$ has a finite support, $|\mathcal{K}(\boldsymbol{a})|$ is bounded for all $\boldsymbol{a}$, and hence $|\mathcal{K}|$ is also finite.

Measure of Transmission Success or Failure: If the frame ends before the completion of a block decoding, we consider all the packets in that block lost. The number of lost packets for each flow in frame t is captured by $\boldsymbol{L}[t] \stackrel{\triangle}{=} \{L_f[t]\}_f$. Hence, together with the expiries, the total number of dropped packets $\boldsymbol{D}[t] \stackrel{\triangle}{=} \{D_f[t]\}_f$ in frame t is equal to $(\boldsymbol{E}[t] + \boldsymbol{L}[t])$ (see Figs. 1 and 2).

We denote the number of successfully decoded packets in frame t by $\boldsymbol{M}[t] \stackrel{\triangle}{=} \{M_f[t]\}_f$, which is a function of the chosen coding block $\mathbb{K}[t] \in \mathcal{K}(\boldsymbol{A}[t])$ and realized completion times for those coding selections. Since the completion time is random, $\boldsymbol{M}[t]$ is also a random vector and can be described by

$$\boldsymbol{M}[t] = \sum_{i=1}^{b} K_{f,i}[t]\mathbf{1}\left(\sum_{j=1}^{i} Z(K_j[t]) \le \tau\right) \quad (2)$$

where $\mathbf{1}(\cdot)$ is the indicator function.[3] To emphasize that $\boldsymbol{M}[t]$ depends on both the coding block choices $\mathbb{K}[t]$ and the corresponding channel variation $\boldsymbol{Z}(\mathbb{K}[t])$, we may also write $\boldsymbol{M}[t] = \boldsymbol{M}(\mathbb{K}[t], \boldsymbol{Z})$.

Multitimescale Requirements of the Applications: Each flow f also imposes a long-term reliability requirement that on average at most $q_f \in (0,1)$ fraction of its packets are dropped. Alternatively, we call $(1 - q_f)$ the delivery ratio requirement for flow f. In other words, we must guarantee that

$$\liminf_{T\to\infty} \frac{\sum_{t=1}^{T} \mathbb{E}[M_f[t]]}{\sum_{t=1}^{T} \mathbb{E}[A_f[t]]} \ge (1 - q_f)\ \forall f. \quad (3)$$

We use $\boldsymbol{q} \stackrel{\triangle}{=} \{q_f\}_f$ to denote the vector of q_f, and refer to it as the *requirement vector*. The requirement gets more stringent as q_f decreases toward 0. Real-time applications such as voice/video transfers that can tolerate a certain fraction of packet losses typically have such delivery ratio requirements. This traffic modeling follows that of [9]–[12], and is attractive for both practical modeling and theoretical analysis purposes.

With the aforementioned system model, our paper studies two problems. First, in Section III, we characterize the maximal satisfiable requirement region achievable by any stationary policy for fixed arrival processes. Building on this characterization, we propose an optimal dynamic coding strategy that is guaranteed to support all arrival processes that lie strictly within the maximal requirement region. Then, in Section IV-A, we develop low-complexity approximations of the dynamic coding algorithm, and in Section IV-B, we extend our result to the case where a rate controller is implemented to control the arrival rate when the arrival process is unknown.

[3]Note that by definition, we have $\boldsymbol{A}[t] = \boldsymbol{M}[t] + \boldsymbol{D}[t]$ for each frame t.

III. Optimal Dynamic Coding Strategy

In this section, we assume fixed (uncontrollable) arrival processes associated with the flows and aim to design a dynamic coding strategy that guarantees the long-term reliability requirements imposed by the requirement vector $\boldsymbol{q}$ of the flows given in (3) for the packets that have a fixed deadline of τ time slots. To that end, we first formulate a stochastic control problem for a fixed arrival process in Section III-A, and propose a practical dynamic coding algorithm that utilizes a combination of iterative pricing and finite-horizon dynamic programming, and prove its stochastic optimality.

A. Problem Formulation

Our DCP is defined as the follows.

Definition 1 (∞-Horizon Dynamic Coding Problem):

$$\begin{aligned}(\text{DCP}):\ \underset{\{\boldsymbol{A}[t],\mathbb{K}[t]\}_{t\ge1}}{\text{Maximize}}\quad & 1 \\ \text{subject to}\quad & K_f[t] \le A_f[t]\ \forall f, \forall t \ge 1 \quad (4)\\ & \overline{\lambda}_f(1-q_f) \le \underline{\mu}_f\ \forall f \quad (5)\\ & \boldsymbol{M}[t] = \boldsymbol{M}(\mathbb{K}[t], \boldsymbol{Z})\ \forall t \ge 1 \quad (6)\end{aligned}$$

where

$$\overline{\lambda}_f = \limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}[A_f(t)]$$

$$\underline{\mu}_f = \liminf_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}[M_f(t)]$$

where the expectations are over the randomness of the arrival processes and the channel variation.

In the aforementioned DCP formulation, (4) assures that the coding is limited to packets available in that frame, (5) assures that long-term delivery ratio requirements are satisfied, and (6) indicates that the successful packet transmissions are random as a function of the completion-time distributions and coding decisions.

Note that the DCP is maximizing a constant subject to the long-term reliability requirement and the channel variation constraints. When solving DCP, we have no control on the arrival process, but the solution to this problem will satisfy these constraints. The problem which incorporates the rate control is discussed in Section IV-B.

Also, we emphasize that solving DCP through standard control theoretic methods is extremely difficult, and likely impossible. Not only it is an infinite-horizon problem, but it also contains instantaneous constraints (4) and channel randomness (6), as well as long-term average requirements (5) .

Instead, motivated by earlier works in stochastic control literature (see, for example, [14], [9], [11]), we introduce a time-varying price vector $\boldsymbol{X}[t] = \{X_f[t]\}_f$, where $X_f[t]$ evolves as

$$X_f[t+1] = (X_f[t] + \beta(D_f[t] - q_f A_f[t]))^+ \quad (7)$$

where $(y)^+ = \max(0, y)$, and $\beta > 0$ is a small step size. The value of $X_f[t]$ measures the experienced reliability requirement violation for flow f, and it can be viewed as a fictitious queue with the arrival $\beta D_f[t]$ and the service $\beta q_f A_f[t]$. It can be seen

that the evolution of $(\boldsymbol{A}[t], \boldsymbol{X}[t])$ forms a Markov chain, and that if $\boldsymbol{X}[t]$ is guaranteed to be stable, i.e., it is positive recurrent and the corresponding stationary distribution has $\mathbb{E}[X_f[t]] < \infty$ for all f, then all the long-term reliability requirements in (5) are met (see [13, Theorems 2.5 and 2.8]).

Before we can present a solution to DCP, we first need to characterize the region in which the requirement vector can be satisfied. This region is described next for the class of all stationary policies.

B. Satisfiable Requirement Region Characterization

As described in Section II, the process of choosing coding window size in a frame is equivalent to choosing a control $\mathbb{K}$ from the set $\mathcal{K}$ defined in (1) . Each control $\mathbb{K}$ will result in a random service $\boldsymbol{M}(\mathbb{K}, \boldsymbol{Z})$ due to channel variations. Then, we can characterize a necessary condition for the satisfiability of a given requirement vector $\boldsymbol{q}$ by any stationary policy as follows.

Lemma 1: Consider the class of stationary policies $\mathcal{G}$ that observe $(\boldsymbol{A}[t], \boldsymbol{X}[t])$ in each frame t and choose a control $\mathbb{K}_{j[t]} \in \mathcal{K}$. If there is a policy $G_0 \in \mathcal{G}$ that can stabilize the price vector $\boldsymbol{X}[t]$, then there exists $\{\alpha_k(\boldsymbol{a})\}_k$ such that

$$\alpha_k(\boldsymbol{a}) \geq 0, \ \forall k, \ \sum_{\mathbb{K}_k \in \mathcal{K}(\boldsymbol{a})} \alpha_k(\boldsymbol{a}) = 1 \ \forall \boldsymbol{a}, \tag{8}$$

$$\sum_{\boldsymbol{a}} P(\boldsymbol{a}) \sum_{\mathbb{K}_k \in \mathcal{K}(\boldsymbol{a})} \alpha_k(\boldsymbol{a}) \mathbb{E}[M_f(\mathbb{K}_k, \boldsymbol{Z})] > (1 - q_f)\lambda_f \quad \forall f \in \mathcal{F} \tag{9}$$

where $P(\boldsymbol{a}) = P\{\boldsymbol{A}[t] = \boldsymbol{a}\}$ is the probability of the arrival vector is $\boldsymbol{a}$.

Proof: The detailed proof is provided in Appendix A. ■

Lemma 1 characterizes the necessary condition for the arrival process $\boldsymbol{A}[t]$ to be feasible under the given requirement vector $\boldsymbol{q}$ and the underlying channel variation model $\boldsymbol{Z}(\mathbb{K})$. Note that this condition is on the whole distribution of the arrival process rather than its limited statistics, such as its mean and variance.

We define the set of requirement satisfying arrival processes of the system for a given requirement vector $\boldsymbol{q}$ as follows.

Definition 2: The requirement satisfying arrival processes $\mathcal{A}(\boldsymbol{q})$ for a requirement vector $\boldsymbol{q}$ is defined as

$$\mathcal{A}(\boldsymbol{q}) \stackrel{\triangle}{=} \left\{ \begin{array}{l} (\boldsymbol{A}) : \text{There exists } \{\alpha_k(\boldsymbol{a})\}_k, \\ s.t. \ (8) \text{ and } (9) \text{ are satisfied} \end{array} \right\}. \tag{10}$$

As a related concept, we define the satisfiable requirement region $\mathcal{Q}(\boldsymbol{A})$ for a given arrival process $\boldsymbol{A}[t]$ as

$$\mathcal{Q}(\boldsymbol{A}) \stackrel{\triangle}{=} \left\{ \begin{array}{l} (\boldsymbol{q}) : \text{There exists } \{\alpha_k(\boldsymbol{a})\}_k, \\ \text{s.t. } (8) \text{ and } (9) \text{ are satisfied} \end{array} \right\}. \tag{11}$$

According to the aforementioned definition, it follows that if $\boldsymbol{A}[t] \in \mathcal{A}(\boldsymbol{q})$ for some requirement vector $\boldsymbol{q}$, then $\boldsymbol{q} \in \mathcal{Q}(\boldsymbol{A}[t])$. Both sets characterize the complicated relationship between the arrival process $\boldsymbol{A}[t]$ and the requirement vector $\boldsymbol{q}$. Different from the similar concept of *capacity* or *stability region* in the previous works (see [21] and [15], for example), both the aforementioned sets depend on the *distribution* of the arrival process, rather than just the mean. An example of the satisfiable requirement region is shown in Section VI-A as part of our numerical results, showing the complex dependence of the regions on the distribution of the arrival processes.

C. Dynamic Coding Strategy

Based on the observation that if $\boldsymbol{X}[t]$ is guaranteed to be stable, then all the long-term reliability requirements are met, we propose our dynamic coding scheme which tries to stabilize the $\boldsymbol{X}[t]$, and use a finite-horizon dynamic programming strategy to solve the DCP. Our dynamic coding strategy uses $\boldsymbol{X}[t]$ to determine the composition $\mathbb{K}[t]$ of the coding window selection in frame t.

Definition 3 (Dynamic Coding Algorithm): For a given set $\mathcal{F}$ of τ-deadline-constrained flows and their requirement vector $\boldsymbol{q} = \{q_f\}_f$, the dynamic coding algorithm performs the following operations in each frame t.

1) *Price Update:* We maintain a price variable $\boldsymbol{X}[t] = (X_f[t])_f$, where $X_f[t]$ for each f is initiated at $X_f[0] = 0$ and is updated at each frame according to (7) . We recall that $A_f[t]$ and $D_f[t]$ denote the number of arrived and dropped flow-f packets in frame t, and hence are known at the beginning of frame $t+1$.
2) *Dynamic Coding Strategy:* The coding strategy is based on the following finite-horizon dynamic programming construction. For any nonnegative-valued price vector $\boldsymbol{X}$, we define the optimal reward-to-go function $J^{\star}_{\boldsymbol{X}}(\boldsymbol{B}, s)$ as the maximum value of the $\boldsymbol{X}$-weighted total mean success rates when there is a vector of $\boldsymbol{B} = \{B_f\}_f$ packets waiting for transmission and when there are $s \in \{0, \ldots, \tau\}$ slots left until the end of the frame. Then, $J^{\star}_{\boldsymbol{X}}(\boldsymbol{B}, s)$ satisfies Bellman's [1]

$$J^{\star}_{\boldsymbol{X}}(\boldsymbol{B}, s) = \max_{\{\boldsymbol{K}_1 : K_{f,1} \leq B_f, \forall f\}} \{\mathbb{E}[J^{\star}_{\boldsymbol{X}}((\boldsymbol{B} - \mathbb{K}_1), (s - Z(K_1))) + (\sum_f K_{f,1} X_f) \cdot \mathbf{1}(Z(K_1) \leq s)]\}.$$

This is solved through backward recursion with the initial conditions: $J^{\star}_{\boldsymbol{X}}(\boldsymbol{B}, s) = 0$, for all $\boldsymbol{B}$ and all $s \leq 0$.

Recall that $\boldsymbol{A}_i[t] = \{A_{f,i}[t]\}_f$ and $\tau_i[t]$, respectively, denote the vector of remaining packets and the number of remaining time slots in frame t at the beginning of the ith block construction (see Fig. 2). Then, the ith block of frame t for $i = 1, 2, \ldots$ is selected as follows until the frame ends:

$$\boldsymbol{K}_i[t] = \underset{\{\hat{\boldsymbol{K}}_i \geq \boldsymbol{0} : \hat{K}_{f,i} \leq A_{f,i}[t], \forall f\}}{\operatorname{argmax}} \mathbb{E}[J^{\star}_{\boldsymbol{X}[t]}(\boldsymbol{A}_i[t] - \hat{\boldsymbol{K}}_i, \tau_i - Z(\hat{K}_i)) + (\sum_f \hat{K}_{f,i} X_f[t]) \cdot \mathbf{1}(Z(\hat{K}_i) \leq \tau_i)]. \tag{12}$$

The aforementioned coding strategy in each frame weighs the successful service rates of flows with their existing prices, $\{X_f[t]\}_f$, therefore effectively prioritizing the service of those flows whose reliability requirement has, so far, been violated more severely.

Note that in the perspective of our block construction model, the choice of the control (or coding block matrix) $\mathbb{K}$ can be either done once at the beginning of the frame (e.g., fixed block size)

or dynamically chosen as in our proposed scheme. Either way, the controller will eventually choose some control (or coding block matrix) $\mathbb{K}$ in the set $\mathcal{K}$, and the number of served packets will be affected by the channel variation $\boldsymbol{Z}(\mathbb{K})$. Thus, we can equivalently express the dynamic coding part of our algorithm as follows: in each frame t, given $(\boldsymbol{X}[t], \boldsymbol{A}[t])$, the dynamic encoder chooses the control $\mathbb{K}_{j^*[t]} \in \mathcal{K}$ as

$$\mathbb{K}_{j^*[t]} \in \underset{\mathbb{K}_j \in \mathcal{K}}{\operatorname{argmax}} \sum_{f \in \mathcal{F}} X_f[t] \mathbb{E}[M_f(\mathbb{K}_j, \boldsymbol{Z})]. \tag{13}$$

With this equivalence in mind, we have the following lemma characterizing the performance of the dynamic coding algorithm.

Lemma 2: For any arrival process $\boldsymbol{A}[t]$ that lies strictly within $\mathcal{A}(\boldsymbol{q})$, the dynamic coding strategy satisfies the long-term delivery ratio requirements (3) of all flows.

Proof: The detailed proof is provided in Appendix B. ■

Combining the necessity and sufficiency results of Lemmas 1 and 2, respectively, we have the following optimality of the dynamic coding algorithm.

Proposition 1: The dynamic coding algorithm is optimal in the sense that if a reliability requirement vector $\boldsymbol{q}$ for a given arrival process vector $\boldsymbol{A}[t]$ can be satisfied by any stationary policy, then it can be satisfied by the dynamic coding algorithm. In other words, if the dynamic coding algorithm cannot satisfy the reliability requirement vector, then it is not satisfiable by any other stationary policy.

Next, we make a few observations on the aforementioned dynamic coding algorithm that will motivate the extensions in Section IV.

In the dynamic coding algorithm, the controller solves the maximization problem (12) using dynamic programming. In each frame t, the number of possible combinations for the coding block choice and the remaining time is $\tau A_{\max}^{|\mathcal{F}|}$. The value of the reward function $J_{\boldsymbol{X}}(\boldsymbol{A}[t], \tau)$ of all these combinations needs to be calculated and stored. This results in an order of $O(\tau A_{\max}^{|\mathcal{F}|})$ computational complexity in each frame, with a storage complexity of order $O(\tau A_{\max}^{|\mathcal{F}|})$. This computational overhead in each frame may become unacceptable when the number of flows increases, which motivates us to develop approximation algorithms with lower computational complexity in Section IV-A.

Proposition 1 shows that the dynamic coding algorithm can support all fixed arrival processes whose reliability requirements can be satisfied. However, in the likely case where the full distribution of arrival process is unknown but its mean may be adjustable via admission or congestion control, we would need a rate controller that adjusts the incoming rates so that the reliability requirements can be satisfied for all flows. This motivates an extension to our dynamic coding algorithm to include a rate controller, which we accomplish in Section IV-B.

IV. Extension of the Dynamic Coding Algorithm

In this section, we extend our dynamic coding algorithm in two important directions: we first provide two low-complexity approximations of the dynamic coding algorithm, where the computational complexity in each frame is greatly reduced; we then develop a rate controller for the flows so that the reliability requirements can be satisfied for all flows when the arrival process is unknown.

A. Low-Complexity Approximation Algorithms

In this section, we develop two approximation algorithms with significantly lower computational complexity. The first one is based on the discretization of the space where $\boldsymbol{X}[t]$ lies, and possesses asymptotical optimality characteristics; and the second one is a greedy algorithm with significantly less computational and storage complexity, which is still guaranteed to outperform any fixed coding window size choice.

Grid Approximation: In the operation of the dynamic coding algorithm, the controller needs to solve for the maximum value of the expected reward function $J^*_{\boldsymbol{X}}(\boldsymbol{A}[t], \tau)$, which is a function of the deficit counter values $\boldsymbol{X}[t]$. The solution for the past frames cannot be used in the current frame since $\boldsymbol{X}[t]$ lies in the space $\mathbb{R}^{|\mathcal{F}|}$ of uncountably many values, and hence they must be recomputed in each frame.

Note that the dynamic programming is essentially solving the maximization in (13); thus for the deficit counter values $\boldsymbol{X}[t]$ and $c\boldsymbol{X}[t]$, where c is any positive constant, the solutions are the same. Inspired by this observation, we use W discrete directions of $\{\tilde{\boldsymbol{X}}_1, \ldots, \tilde{\boldsymbol{X}}_W\}$ to approximate the directions that $\boldsymbol{X}[t]$ can take. Although there are many possible ways to select such set of directions, in the following algorithm we use a particular choice, namely choosing the points from an integer grid.

Definition 4 (Grid Approximation Algorithm): For a given set $\mathcal{F}$ of τ-deadline-constrained flows and their requirement vector $\boldsymbol{q} = \{q_f\}_f$, the grid approximation algorithm performs:

1) *Initialization and Storage:*
   a) Consider an $|\mathcal{F}|$-dimensional cube $[0, w]^{|\mathcal{F}|}$, where w is some positive integer. Choose all vectors with integer coordinates on all $|\mathcal{F}|$ surfaces that share the common vertex $(w, w, \ldots, w)$ of this cube. These vectors have at least one of their coordinates equals to w. Such vectors form the set of vectors $\{\tilde{\boldsymbol{X}}_1, \ldots, \tilde{\boldsymbol{X}}_W\}$, where $W = w^{|\mathcal{F}|-1}(w-1)$.
   b) For each $\tilde{\boldsymbol{X}}_w \in \{\tilde{\boldsymbol{X}}_1, \ldots, \tilde{\boldsymbol{X}}_W\}$, calculate and tabulate the optimal coding block choices $\boldsymbol{K}(\tilde{\boldsymbol{X}}_w, \boldsymbol{A}, t)$ for all possible arrival vector $\boldsymbol{A}$ and all possible remaining time $t = \{1, 2, \ldots, \tau\}$ using dynamic programming.
2) *In each frame t:*
   a) Approximate the direction of $\boldsymbol{X}[t]$ by $\tilde{\boldsymbol{X}}_{w^*}$, where w^* is chosen as

$$w^* = \underset{w \in \{1,2,\ldots,W\}}{\operatorname{argmax}} \frac{\langle \boldsymbol{X}[t], \tilde{\boldsymbol{X}}_w \rangle}{|\boldsymbol{X}[t]||\tilde{\boldsymbol{X}}_w|}$$

   where $\langle \cdot, \cdot \rangle$ is the inner product of two vectors.
   b) The ith coding block in this frame is chosen as $\boldsymbol{K}_i[t] = \boldsymbol{K}(\tilde{\boldsymbol{X}}_{w^*}, \boldsymbol{A}_i[t], \tau_i)$.

In this grid approximation algorithm, the computational heavy dynamic programming is done offline, but only for finitely many choices in the set of $\{\tilde{\boldsymbol{X}}_1, \ldots, \tilde{\boldsymbol{X}}_W\}$. In each frame t, the controller choose $\tilde{\boldsymbol{X}}_{w^*}$ which has the smallest angle with $\boldsymbol{X}[t]$ by performing inner product, and the choice of the coding block is done by lookup table. Thus, the computational complexity in a frame reduces to $O(W)$, and it requires a storage space of $O(\tau W A_{\max}^{|\mathcal{F}|})$. Thus, the grid approximation algorithm trades storage complexity with computational complexity.

We also note that as W goes to infinity, the set $\{\tilde{\boldsymbol{X}}_1, \ldots, \tilde{\boldsymbol{X}}_W\}$ provides better and better approximation to the direction of $\boldsymbol{X}[t]$. Based on the optimality of the dynamic coding algorithm, the performance of the grid approximation algorithm is asymptotically optimal as W goes to infinity in the sense that if a reliability requirement vector $\boldsymbol{q}$ for a given arrival process vector $\boldsymbol{A}[t]$ can be satisfied by any stationary policy, then it can be satisfied by the grid approximation algorithm.

Although the asymptotic optimality holds when $W \to \infty$, as we shall see in the numerical results presented in Section VI-B, even with a relative low value of W, the grid approximation algorithm can perform very close to the dynamic coding algorithm.

The grid approximation algorithm greatly reduces the computational complexity in each frame. However, when W is large, the space complexity may still become unacceptable. This motivates us to develop an even simpler algorithm that is guaranteed to outperform any static coding block size choice.

Greedy Algorithm: Both the original dynamic coding and the grid approximation strategies perform a joint optimization of the window size and content decisions. In the following approximation, we consider a decoupling of these decisions to first decide on the window size only based on the remaining time in the frame, and then fill in the selected window with the content of available flows with the largest deficit counter values. The details of this process are given next.

Definition 5 (Greedy Algorithm): For a given set $\mathcal{F}$ of τ-deadline-constrained flows and their requirement vector $\boldsymbol{q} = \{q_f\}_f$, the greedy algorithm operates as follows.

1) *Initialization:*
   a) For each possible remaining time $x \in \{1, 2, \ldots, \tau\}$, find and store
   $$K^*(x) = \operatorname*{argmax}_{K \geq 1} KP(Z(K) \leq x)$$
   which gives the block size achieving the maximum expected throughput when the remaining time is x.
2) *In each frame t:*
   a) Set size of the ith coding block in frame t as
   $$K_i[t] = K^*(\tau_i)$$
   where we recall that τ_i is the remaining time to the end of frame at the decision time of the ith block.
   b) The content of the ith coding block is chosen to be the solution of the following maximization:
   $$\max_{0 \leq K_{f,i} \leq A_{f,i}[t]} \sum_f X_f[t] K_{f,i}$$
   $$\text{s.t.} \qquad \sum_f K_{f,i} \leq K_i[t].$$

TABLE I
COMPARISON OF THE COMPLEXITY AND PERFORMANCE

	Computational complexity	Storage complexity	Optimality
DCA	$O(\tau A_{max}^{\lvert\mathcal{F}\rvert})$	$O(\tau A_{max}^{\lvert\mathcal{F}\rvert})$	Optimal
Grid	$O(W)$	$O(\tau W A_{max}^{\lvert\mathcal{F}\rvert})$	Asymptotically optimal as $W \to \infty$
Greedy	$O(1)$	$O(\tau)$	Suboptimal, but better than any static coding window size choice

The last maximization in the aforementioned greedy algorithm can be simply solved by assigning as much as possible packets for each flow without violating the constraints, in the order of decreasing $X_f[t]$. The greedy algorithm has $O(1)$ computational complexity and $O(\tau)$ space complexity. Moreover, its performance is at least as good as any static coding block choice since it chooses the block size that can achieve the highest expected throughput in the given remaining time (also see our numerical result in Section VI-B).

Comparison of the Complexity and Performance: In Table I, we summarize and compare the complexity and performance of the algorithms we proposed: the dynamic coding algorithm (DCA), the grid approximation algorithm (Grid), and the greedy algorithm (Greedy).

The tradeoffs of the algorithms can be observed from Table I. While the dynamic coding algorithm is optimal, its computational and storage complexity are relatively high. The grid approximation algorithm trades storage complexity for better computational complexity, and it is asymptotically optimal as the number of grids W goes to infinity. The greedy algorithm has the best computational and storage complexity at the cost of optimality. Yet, it is guaranteed to outperform any static coding window size choice (see Section VI-B for more discussion).

B. Joint Rate Control and Coding Algorithm

So far, our focus has been the optimal coding operation under a given arrival process vector that lies within the requirement satisfiable region. Yet, in may scenarios, it is more favorable to have a rate controller to stabilize the system when the arrival process is unknown. In this section, we consider such scenarios to extend the algorithm proposed in Section III to accommodate a flow controller to adjust the arrival rates for all flows to satisfy the reliability requirements. In particular, we formulate this problem in a form of utility maximization and assume that each flow f has a utility function $U_f(\lambda_f)$ associated with it, where λ_f is the controllable arrival rate of flow f.

As characterized in Definition 2 and illustrated by the numerical result in Fig. 5 in Section VI-A, the set of requirement satisfying arrival processes $\mathcal{A}(\boldsymbol{q})$ is tightly related to the distribution

of the arrival process and challenging to be precisely characterized. Thus, an ideal flow controller in this case may need to have full control of the distribution of the arrival process. However, such a flow controller is complicated to model and analyze, and can be unrealistic in practice. Thus, we consider a class of arrival process whose full distribution can be determined by its mean, and its realization lies in the interval $[A_{\min}, A_{\max}]$ with probability 1. Examples of such arrivals can be deterministic process, the sum of a deterministic process and a zero-mean random variable, etc. We aim to adjust its mean dynamically to guarantee that all the delivery ratio requirements are satisfied.

Problem Formulation: For the generic communication system of Fig. 1, we aim to design a joint rate controller and dynamic coding strategy that stabilizes the system. The associated stochastic optimization problem is provided next.

Definition 6 (∞-Horizon Utility Maximization Problem):

$$(\text{UMP}): \underset{\{\boldsymbol{A}[t],\mathbb{K}[t]\}_{t\geq 1}}{\text{Maximize}} \quad \sum\nolimits_f U_f(\lambda_f) \tag{14}$$

$$\text{subject to} \quad \text{constraints (4), (5), and (6)}$$
$$A_{min} \leq A_f[t] \leq A_{\max} \ \forall f, t \tag{15}$$

where we assume that $A_{\min} > 0$ is a lower bound for the number of arrivals in each frame; hence, it is a lower bound for arrival rate, and

$$\lambda_f = \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \lambda_f[t].$$

For feasibility, we assume that $A_{\min}$ is small enough such that when $\lambda_f[t] = A_{min}$ for all f and t, the corresponding arrival process lies within the set $\mathcal{A}(\boldsymbol{q})$.

We also impose the following assumptions on the utility functions $U_f(\cdot)$.

Assumption 1: $U_f(\cdot)$ is a twice differentiable, nondecreasing, strictly concave function of the flow rate λ_f.

Assumption 2: For all $0 < m < M < \infty$, there exist constants $0 < \tilde{c} < \tilde{C} < \infty$ such that

$$\tilde{c} \leq -\frac{1}{U_f''(x)} \leq \tilde{C}, \ \forall x \in [m, M]. \tag{16}$$

We note that these conditions are not restrictive, and they hold for the following class of utility functions:

$$U_f(x) = \beta_f \frac{x^{1-\alpha_f}}{1-\alpha_f}.$$

This type of utility functions is well known to characterize fairness concepts (refer to [18] and references therein).

Motivated by the dynamic coding algorithm, we solve (14) using a primal-dual algorithm described later.

Joint Rate Control and Dynamic Coding Algorithm: Similarly to Section III-C, we use the vector $\boldsymbol{X}[t]$ to measure the experienced reliability requirement violation for the flows. Yet, to capture the presence of the flow controller in this case, we will slightly change its evolution equation below. The dynamic rate controller uses $\boldsymbol{X}[t]$ to determine the arrival rate for each flow to satisfy the reliability constraint.

As revealed in Section III-C, the set of requirement satisfying arrival processes characterized in (10) imposes complicated linear constraints on the rate update rule, even in the case of a deterministic arrival. As a result, the dual algorithms which directly solve for the optimal rate (see [6] and [3], for example) involve solving a difficult linear program in each iteration no longer viable. This forces us to develop a primal-dual algorithm inspired in [7] for subsequent price and rate allocation updates. This algorithm will later be shown to slowly steer the rate vector to a requirement satisfying arrival rate.

Definition 7 (Joint Rate Control and Coding Algorithm): For a given set $\mathcal{F}$ of τ-deadline-constrained flows and their requirement vector $\boldsymbol{q} = \{q_f\}_f$, the joint dynamic algorithm performs the following operations in each frame t.

1) *Price Update:* We maintain a price variable $\boldsymbol{X}[t] = (X_f[t])_f$, where $X_f[t]$ for each flow f is initiated at $X_f[0] = 0$ and is updated at each frame according to

$$X_f[t+1] = (X_f[t] + (1-q_f)A_f[t] - M_f[t])^+ \tag{17}$$

where $(y)^+ = \max(0, y)$. We recall that $A_f[t]$ and $M_f[t]$ denote the number of arrived and dropped flow-f packets in frame t, and hence are known at the beginning of frame $t+1$.

2) *Rate Control:* Given $\boldsymbol{X}[t]$, the rate controller updates the rate vector $\lambda[t] = \{\lambda_f[t]\}_f$ for each flow f in frame t as

$$\lambda_f[t+1] = \left[\lambda_f[t] + \alpha(RU_f'(\lambda_f[t]) - (1-q_f)X_f[t])\right]_{A_{\min}}^{A_{\max}} \tag{18}$$

where $[x]_{A_{\min}}^{A_{\max}}$ is the projection of x to the interval $[A_{\min}, A_{\max}]$, $\alpha > 0$ is a step-size parameter, and $R > 0$ is a design parameter. Then, the arrival vector $\boldsymbol{A}[t]$ is generated according to its mean $\lambda[t]$. Note that $A_f[t] \in [A_{\min}, A_{\max}]$ by our assumption on the arrival process.

3) *Dynamic Coding Strategy:* Given $\boldsymbol{X}[t]$, the dynamic coding is performed exactly as in Definition 3.

As will be revealed later, the choice of the design parameter R limits the selection of the step-size parameter, α, and determines the distance of the achieved average rates of the algorithm to the optimal solution of (14) .

Performance Analysis: The following result establishes the asymptotic boundedness of the vector $\boldsymbol{X}[t]$ under the stochastic operation of the system, which implies the requirement satisfying nature of the algorithm.

Proposition 2: There exists a constant $c(\alpha, R) < \infty$ which depends on the step-size α and the design parameter R such that

$$\limsup_{t\to\infty} \mathbb{E}\left[\sum_f X_f^2[t]\right] \leq c(\alpha, R).$$

Furthermore, when α is chosen to be $1/R^2$, $c(\alpha, R) = O(R^2)$, i.e., $\limsup_{R\to\infty} c(1/R^2, R)/R^2 = C < \infty$.

Proof: The detailed proof is provided in Appendix C. ■

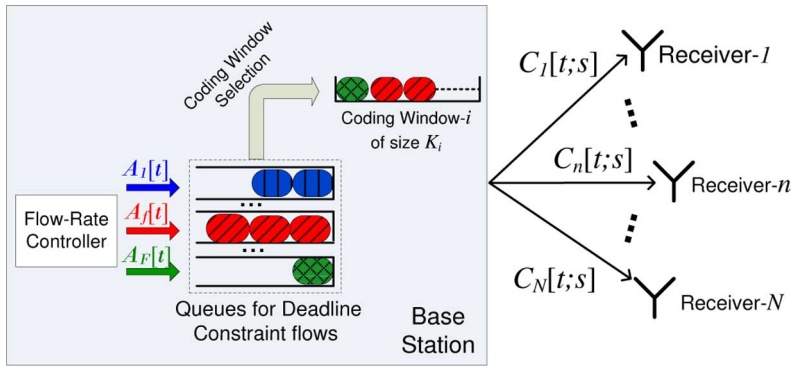

Fig. 3. Cellular down-link model for broadcasting deadline-constrained flows to N users over erasure channels.

V. Application: Cellular Down-Link Scenario

The generic model of the communication system can be used for lots of specific communication scenarios and coding strategies. In this section, we describe an important example of such scenarios whereby a BS is serving multiple flows by broadcasting their incoming packets to N receivers over time varying erasure channels (see Fig. 3). We assume that each packet is an element of a finite field $\mathbb{F}_d$ of size d.

The components of the particular system follow the descriptions in Section II. To precisely formulate the completion time $Z(K)$, we assume the channel state $C_n[t;s]$ of the nth receiver in the $s \in \{1,\cdots,\tau\}$ time slot of the tth frame to be an independent Bernoulli process with mean

$$c_n = P(C_n[t;s]=1) = 1 - P(C_n[t;s]=0).$$

We also assume that the channel variations are also independently but not necessarily identically distributed between different receivers. If $C_n[t;s]=1$, then the channel can transmit one packet successfully to receiver n in time slot s of frame t. The BS is assumed to know $\{c_n\}_n$ but not the realizations before its transmissions. Although not necessary for the theory, for ease of exposition, we assume in our following calculations that the channels are also identical, i.e., $c_n = c$ for all n. Also, we assume that a block is decoded immediately when the receiver receives enough packets, and an ACK is sent back to the transmitter immediately. With these assumptions, we can precisely characterize the distribution of the completion time $Z(K)$ for a coding block of size K for different coding strategies. In particular, we study the following two coding strategies as in [5]: randomized broadcast coding (RBC) and round robin scheduling (RR).

Definition 8 (RBC): A network coding strategy over a block of K packets where in a slot, say s, any linear combination of the K packets in the file can be transmitted. Specifically, if $\boldsymbol{P}(s)$ denotes the packet chosen for transmission in slot s, we have $\boldsymbol{P}[s] = \sum_{k=1}^{K} \alpha_k[s] P_k$, where $\{\alpha_k[s]\}_k$ chosen uniformly at random from the field $\mathbb{F}_d \backslash \{0\}$ for every time slot s. Each receiver sends an ACK back to the transmitter after it receives K linearly independent copies of the packets.

It has been shown in [5] that RBC is an optimal coding strategy as the field size $d \to \infty$. Since for RBC the transmission of a block of size K is completed when all users successfully receive K packets, the distribution of the completion time $F_{Z^{\mathrm{RBC}}(K)}(x) = P(Z^{\mathrm{RBC}}(K) \le x)$ is

$$F_{Z^{\mathrm{RBC}}(K)}(x) = \left(\sum_{n=K}^{x} \binom{n-1}{K-1} c^K (1-c)^{n-K} \right)^N$$

for $x \ge K$, and 0 otherwise.

Definition 9 (RR): For a given block of packets of size K, the BS at any given slot broadcasts a single packet from the current coding window. Thus, we have $\boldsymbol{P}(s) \in \{\boldsymbol{P}_k\}_{\{k=1,\ldots,K\}}$. Each receiver sends an ACK back to the transmitter after it receives the whole block. In the optimal RR (see [5] for the proof of optimality under channel symmetry), Packet k is transmitted in time slots $(rK+k)$ for $r = 0, 1, \ldots$ until all the receivers receive the whole block.[4]

The completion time distribution $F_{Z^{\mathrm{RR}}(K)}(x)$ for the RR can be expressed as follows, with $\bar{c} = (1-c)$:

$$F_{Z^{\mathrm{RR}}(K)}(x) = \sum_{y=0}^{x} \left(\left(1-\bar{c}^{r+1}\right)^{N(k-1)} \left(1-\bar{c}^{r}\right)^{N(K-k)} \cdot \sum_{n=1}^{N} \binom{N}{n} \left(1-\bar{c}^{r}\right)^{N-n} \left(\bar{c}^{r} c\right)^{n} \right)$$

where $r = \lfloor y/K \rfloor$, $k = y \bmod (K)$ if $y \bmod (K) \neq 0$, and $r = y/K - 1$, $k = K$ if $y \bmod (K) = 0$ such that $y = rK + k$ for $k \in \{1, 2, \ldots, K\}$. The derivation of this distribution is presented in Appendix D.

Fig. 4 illustrates the download completion time distributions of RBC and RR strategies. It can be observed that the completion time of RBC is more "concentrated" around its mean. It is known (see [5]) that the expected completion time for the RBC strategy is lower than that for the RR strategy and the difference grows as the number of receivers N and the block size K increase. Within our framework, our dynamic coding strategy and rate controller can be used together with these coding strategies to guarantee the delivery requirements (3) that are not considered for these coding strategies in the previous works.

[4]Note that for both RBC and RR coding schemes, one ACK is sent from each user for a coding block.

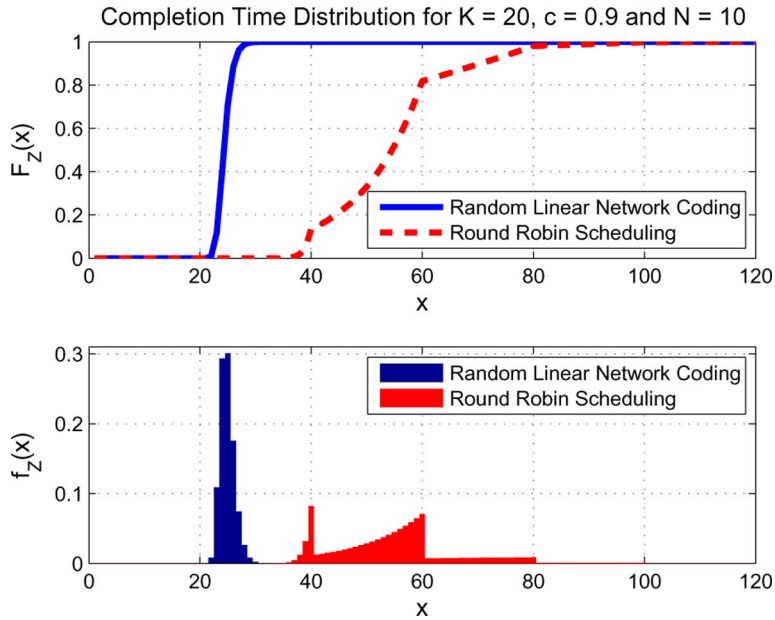

Fig. 4. Download completion time distribution for RBC and RR.

VI. Numerical Results

In this section, we provide numerical results to complement the analysis in the previous sections and to develop our intuition of the system operation under the multitimescale quality-of-service requirements of streaming applications. The simulations are presented for the cellular down-link network described in Section V, with our dynamic coding combined with RBC or RR coding strategies, unless otherwise stated. In all simulations, we assume that there are two flows in the system.

A. Example of the Satisfiable Requirement Region, $\mathcal{Q}(\boldsymbol{A})$

In this section, we illustrate the satisfiable requirement region defined in Definition 2. Unlike the set of requirement satisfying arrival processes $\mathcal{A}(\boldsymbol{q})$ which lies in the high-dimensional space of stochastic processes, we can plot the satisfiable requirement region which lies in $\mathbb{R}^{|\mathcal{F}|}$ that allows for better illustration.

We look at the satisfiable requirement region for two independent flows with identical arrival distributions. In an actual system such as the cellular down-link network, the size of the set $\mathcal{K}$ of coding matrices may grow exponentially with the number of possible arrivals. Instead, we choose to use a small set of $\mathcal{K}$ that does not strictly follow its definition in Section II, but still exhibits aspects of the satisfiable requirement region.

Fig. 5 shows the satisfiable requirement pair (q_1, q_2) for different arrival distributions. For this simulation, we use five different arrival statistics to demonstrate the effect of the whole distribution on the satisfiable requirement:

1) a deterministic arrival of four packets per frame, i.e., an integer-valued uniform distribution with mean 4 and variance 0 (Uniform(4,0));
2) an integer-valued uniform distribution with mean 4 and variance 2 (Uniform(4,2));
3) an integer-valued uniform distribution with mean 4 and variance 4 (Uniform(4,4));
4) a Poisson distribution with rate 4 thus variance 4 (Poisson(4,4));

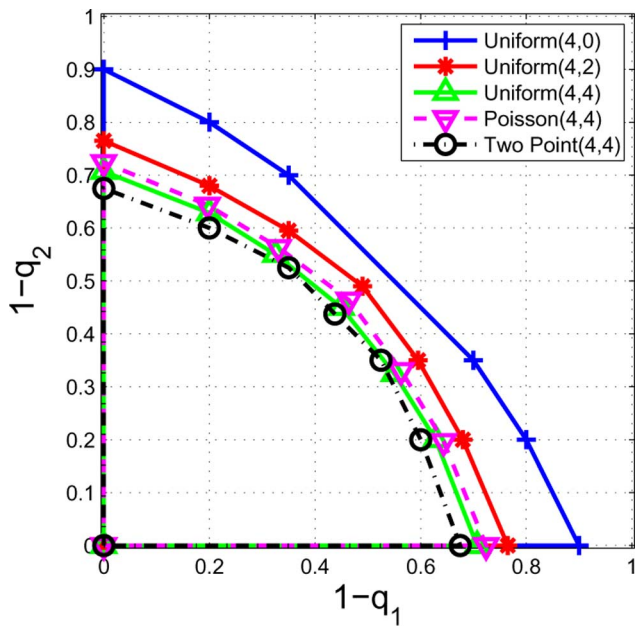

Fig. 5. Satisfiable requirement pair (q_1, q_2) for different arrival processes.

5) a two-point distribution with $P(A = 2) = P(A = 6) = 0.5$, which has a mean of 4 and a variance of 4 (Two Point(4,4)).

It can be observed in Fig. 5 that the first three arrival processes which share the same mean but have increasing variance have gradually shrinking satisfiable requirement regions. This implies that increasing variance hurts the supportable delivery ratio requirements in a nontrivial manner. Also, Uniform(4,4), Poisson(4,4), and two-point(4,4) distributions that have the same mean and variance but different overall distributions achieve different satisfiable requirement regions. This observation confirms that the satisfiable requirement region is tightly related to the whole distribution of the arrival process, rather than its limited

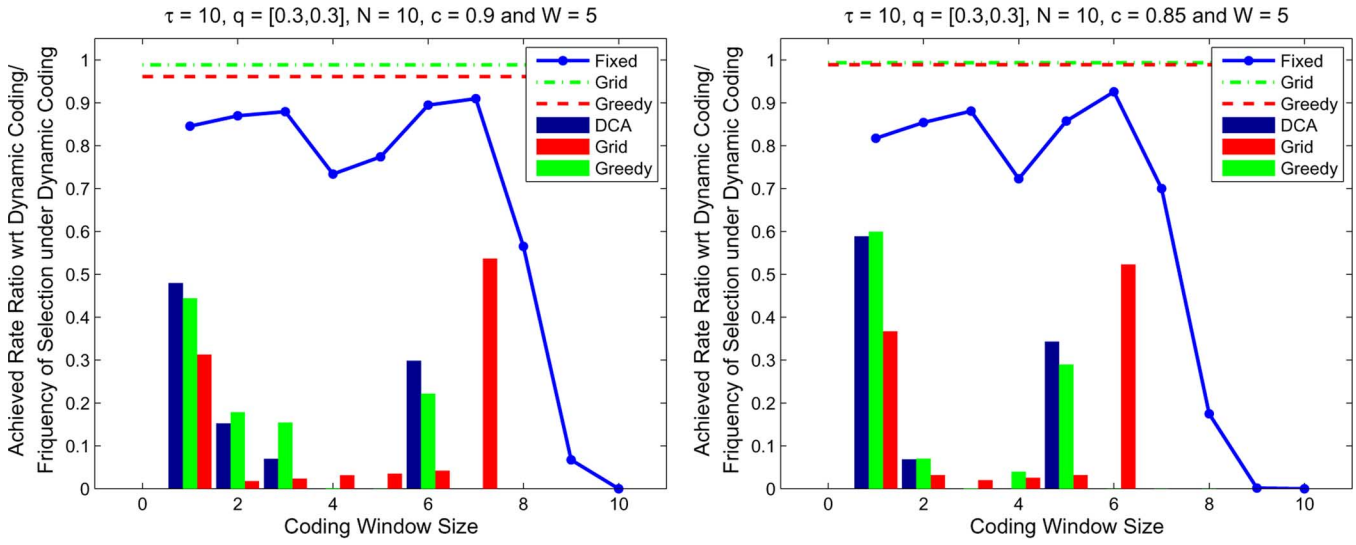

Fig. 6. Performance for fixed coding window compared to dynamic coding algorithm and its approximations. The lines represent the relative achieved throughput by different algorithms with respect to the dynamic coding algorithm. The bars show the fraction of time that a particular block size is chosen by the algorithms.

statistics, such as its mean and variance. Also, these suggest that when the arrival process is deterministic, we can achieve better performance than when the arrival is random.

B. Performance Comparison of Proposed Algorithms

In this section, we study the effect of using dynamic block sizes over fixed ones that do not vary over time. To that end, we compare the achieved delivery rate of the RBC strategy without dynamic coding to the performance achieved by the dynamic coding algorithm together with its low-complexity approximations. The arrival process is assumed to be an integer-valued uniform distribution in $[0, 2\lambda_f]$ where $\lambda_1 = \lambda_2 = 5$, and the other parameters are shown in Fig. 6. For a fair comparison, we let the coding strategy with fixed block size choose the content of the coding block of length k to maximize

$$\sum_{f\in\mathcal{F}} X_f[t]\mathbb{E}[M_f|\boldsymbol{X}[t], \boldsymbol{A}[t], K = k]$$

by using a similar dynamic programming technique as in our dynamic coding algorithm.

Fig. 6(a) and (b) shows the comparison of the performance of RBC with fixed block size to RBC with dynamic coding algorithm and its low-complexity approximations in different scenarios. The curves show the relative delivery rate ratio achieved by the different algorithms with respect to the dynamic coding algorithm. It can be observed that the dynamic coding algorithm, together with its both low-complexity variants, outperforms all fixed window size strategies by a nonnegligible fraction. Also, it is observable that the optimal choice of the fixed window size is highly nontrivial, and hence practically infeasible. Instead, the dynamic policy automatically adapts to the conditions to achieve the optimal performance. In Fig. 6(a) and (b), the channel reliability, thus the completion time distribution, is different. It can be observed that in both cases the grid approximation algorithm performs almost the same as the dynamic coding algorithm, but the performance of the greedy algorithm varies. Also, we note that the number of discrete directions for the grid approximation algorithm we used in these simulations is $W = 5$, which shows even with a relatively small W value that the grid approximation algorithm can perform close to the dynamic coding algorithm.

The dynamic nature of our coding algorithm is observable from the bars in Fig. 6, where the blue bars show the fraction of time that the dynamic coding algorithm chooses a particular block size, the block size of 1 and 6 being chosen more frequently. These more favorable block sizes have relatively better performance in the case of fixed block sizes. Yet, no fixed choice can achieve the performance of the dynamic coding algorithm that can adaptively choose a larger block size (such as 6 or 5) when there is sufficient time-to-go in the frame, and a smaller block size (such as 1 or 2) if the time to the end-of-frame is short at the decision time. This allows the dynamic algorithm to better utilize the remaining time slots in the frame which are otherwise underutilized under a small coding window size selection, or wasted under a large coding window size selection by the fixed block size strategy. Similar observations can be made for its two low-complexity approximation algorithms.

C. Set of Requirement Satisfying Arrival Processes, $\mathcal{A}(\boldsymbol{q})$

While it is impossible to plot the set of arrival distributions for more general cases, by restricting the arrival process to be deterministic, we can illustrate the requirement satisfying arrival rate region by using our joint rate control and dynamic coding algorithm.

Fig. 7 shows the achieved delivery rate pairs (μ_1, μ_2) for two flows under the cellular system model introduced in Section V. The deterministic nature of the arrival processes results in these triangle-shaped rate regions. We observed that the requirement satisfying rate region is smaller when using a less reliable channel, or choosing the RR coding strategy, since both of these actions result a larger mean completion time. We can

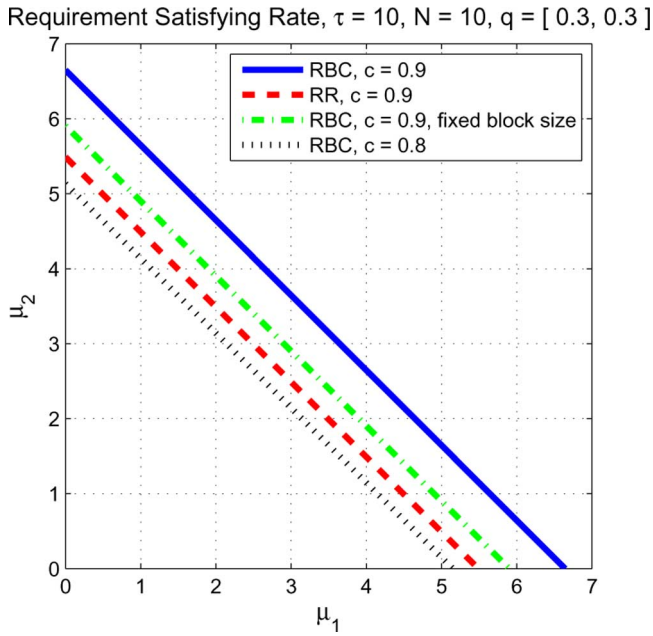

Fig. 7. Requirement satisfiable rate pair (μ_1, μ_2) under deterministic arrivals.

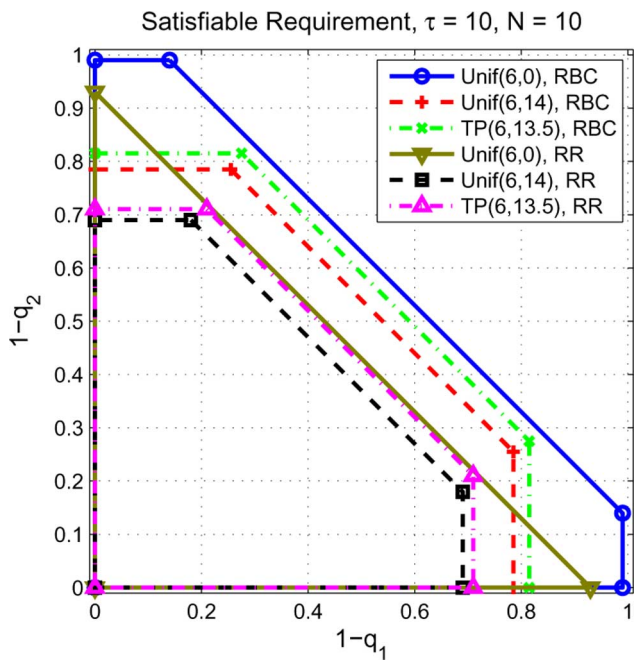

Fig. 8. Satisfiable requirement region under different arrival processes.

also observe that RBC using a fixed block size of six packets achieves a smaller region than RBC with dynamic coding algorithm, clearly showing the advantage of the dynamic coding algorithm.

D. Satisfiable Requirement Region for Fixed Arrivals

By slightly modifying our joint rate control and dynamic coding algorithm, we can dynamically update the requirement vector $\boldsymbol{q}$ which enables us to get the maximum satisfiable requirement region for a fixed arrival process.

Fig. 8 illustrates the satisfiable requirement region for different arrival processes and coding strategies in the cellular system. All arrival processes have their means equal to 6, but with different distributions:

1) a deterministic arrival of six packets per frame, i.e., an integer-valued uniform distribution with mean 6 and variance 0 (Unif(6,0));
2) an uniform distribution on integers $\{0, 1, \ldots, 12\}$, which has a mean of 6 and variance of 14 (Unif(6,14));
3) a two-point distribution with $P(A = 3) = P(A = 9) = 0.5$, which has a mean of 6 and a variance of 13.5 (TP(6,13.5)).

Similarly as in Fig. 5, the satisfiable requirement region varies for different arrival processes. It also shows that deterministic arrivals achieve the largest satisfiable requirement region. It can also be observed that the RBC achieves better performance than the RR, due to its shorter average completion time.

VII. Conclusion

In this paper, we studied the problem of serving deadline-constrained traffic with reliability requirements over time varying wireless channels. We used a general model that captures the multi-timescale QoS requirements of the flows and the operation of the system. We first developed a dynamic coding algorithm that adaptively determines the coding block sizes and contents for fixed arrival stochastic based on a pricing and finite-horizon dynamic programming mechanism. We proved the optimality of this algorithm in the sense that it satisfies the requirements of all arrivals that can be satisfied by any stationary policy. Then, we extended this algorithm in two directions: first, we developed two low-complexity approximation algorithms to reduce the computational complexity of the dynamic coding algorithm; then, we added a flow controller to adjust the rate of all flows to guarantee the reliability requirements. The stochastic stability of our joint rate control and dynamic coding algorithm is established. Also, we applied our theoretical results to the important scenario of a cellular down-link network that serves multiple streaming flows over fading broadcast channels. Finally, we provided extensive numerical results to corroborate our analytical results, and observed the advantage of dynamic channel coding over any static choice for serving such deadline-constrained traffic.

Appendix A
Proof of Lemma 1

We note the similarity of the proof technique to the one in [20] and [22]. However, those works focus on characterizing the set of mean arrival rates to achieve long-term network stability under fading conditions. In comparison, the short-term nature of the deadline constraints in our setup requires us to characterize the set of arrival processes to guarantee the satisfaction of the traffic requirements.

We look at the steady-state operation of policy G_0, i.e., the Markov chain $(\boldsymbol{A}[t], \boldsymbol{X}[t])$ is in its steady state. Let $\mathbb{K}_{j[t]}$ be the control selected by G_0, i.e., $j[t] : (\boldsymbol{A}[t], \boldsymbol{X}[t]) \longmapsto \mathcal{K}$. Then the expected drop rate is given by $\mathbb{E}[\boldsymbol{A}[t] - \boldsymbol{M}(\mathbb{K}_{j[t]}, \boldsymbol{Z})]$. Since the system is stabilized by G_0, we must have for any $f \in \mathcal{F}$, the arrival rate to the price $X_f[t]$ is less than the service rate, i.e.,

$$\mathbb{E}[A_f[t] - M_f(\mathbb{K}_{j[t]}, \boldsymbol{Z})] < q_f \mathbb{E}[A_f[t]] \quad \forall f \in \mathcal{F}$$

which can be further simplified as

$$\mathbb{E}[M_f(\mathbb{K}_{j[t]}, \boldsymbol{Z})] > (1 - q_f)\lambda_f \ \forall f \in \mathcal{F}. \tag{19}$$

In the following equations, the time index t is omitted for brevity. The left-hand side of (19) can be calculated as

$$\begin{aligned}&\mathbb{E}[M_f(\mathbb{K}_{j[t]}, \boldsymbol{Z})]\\&= \mathbb{E}[\mathbb{E}[M_f(\mathbb{K}_{j(\boldsymbol{X},\boldsymbol{A})}, \boldsymbol{Z})|\boldsymbol{X}, \boldsymbol{A}]\\&= \sum_{\boldsymbol{x},\boldsymbol{a}} P(\boldsymbol{X}=\boldsymbol{x}, \boldsymbol{A}=\boldsymbol{a})\mathbb{E}[M_f(\mathbb{K}_{j(\boldsymbol{x},\boldsymbol{a})}, \boldsymbol{Z})]. \quad (20)\end{aligned}$$

Note that the joint distribution of $\boldsymbol{X}[t]$ and $\boldsymbol{A}[t]$ in (20) is well defined since $\boldsymbol{X}[t]$ has a steady-state distribution and $\boldsymbol{A}[t]$ has a finite support

$$\begin{aligned}(20) &= \sum_{\boldsymbol{a}} P(\boldsymbol{A}=\boldsymbol{a}) \sum_{\boldsymbol{x}} P(\boldsymbol{X}=\boldsymbol{x}|\boldsymbol{A}=\boldsymbol{a})\\&\quad \mathbb{E}[M_f(\mathbb{K}_{j(\boldsymbol{x},\boldsymbol{a})}, \boldsymbol{Z})\\&= \sum_{\boldsymbol{a}} P(\boldsymbol{A}=\boldsymbol{a}) \sum_{\boldsymbol{x}} P(\boldsymbol{X}=\boldsymbol{x}|\boldsymbol{A}=\boldsymbol{a})\\&\quad \sum_{\mathbb{K}_k\in\mathcal{K}(\boldsymbol{a})} \mathbf{1}\{j(\boldsymbol{a},\boldsymbol{x})=k\}\mathbb{E}[M_f(\mathbb{K}_k, \boldsymbol{Z})] \quad (21)\end{aligned}$$

where $\mathbf{1}(\cdot)$ is the indicator function. By switching the order of summation, we have

$$\begin{aligned}(21) &= \sum_{\boldsymbol{a}} P(\boldsymbol{A}=\boldsymbol{a}) \sum_{\mathbb{K}_k\in\mathcal{K}(\boldsymbol{a})} \mathbb{E}[M_f(\mathbb{K}_k, \boldsymbol{Z})]\\&\quad \sum_{\boldsymbol{x}} P(\boldsymbol{X}=\boldsymbol{x}|\boldsymbol{A}=\boldsymbol{a})\mathbf{1}\{j(\boldsymbol{a},\boldsymbol{x})=k\}\\&\stackrel{\triangle}{=} \sum_{\boldsymbol{a}} P(\boldsymbol{A}=\boldsymbol{a}) \sum_{\mathbb{K}_k\in\mathcal{K}(\boldsymbol{a})} \alpha_k(\boldsymbol{a})\mathbb{E}[M_f(\mathbb{K}_k, \boldsymbol{Z})]\end{aligned}$$

where $\alpha_k(\boldsymbol{a})\stackrel{\triangle}{=}\sum_{\boldsymbol{x}} P(\boldsymbol{X}=\boldsymbol{x}|\boldsymbol{A}=\boldsymbol{a})\mathbf{1}\{j(\boldsymbol{a},\boldsymbol{x})=k\}$. Combined with (19), the lemma is proved.

Appendix B
Proof of Lemma 2

Note that the evolution of $\boldsymbol{X}[t]$ forms a Markov chain. We proof this result by defining a Lyapunov function of the form

$$V(\boldsymbol{X}) = \frac{1}{2\beta}\sum_{f\in\mathcal{F}} X_f^2$$

and study its expected drift. The time index t is omitted in the following equations for brevity when there is no ambiguity:

$$\begin{aligned}&\triangle V(\boldsymbol{x})\\&= \frac{1}{2\beta}\sum_{f\in\mathcal{F}} \mathbb{E}\left[X_f^2[t+1] - X_f^2[t]|\boldsymbol{X}[t]=\boldsymbol{x}\right]\\&\le \frac{1}{2\beta}\sum_{f\in\mathcal{F}} \mathbb{E}\Bigg[\mathbb{E}\Bigg[\Big(x_f + \beta(A_f - M_f(\mathbb{K}_{j^*(\boldsymbol{A})}, \boldsymbol{Z}))\\&\quad - q_f A_f\Big)^2 - x_f^2\Big|\boldsymbol{A}=\boldsymbol{a}\Bigg]\Bigg]\\&= \sum_{f\in\mathcal{F}}\sum_{\boldsymbol{a}} P(\boldsymbol{a})\Big((1-q_f)a_f - \mathbb{E}[M_f(\mathbb{K}_{j^*(\boldsymbol{a})}, \boldsymbol{Z})]\Big)x_f\\&\quad + \frac{\beta}{2}\sum_{f\in\mathcal{F}} \mathbb{E}\left[\Big((1-q_f)A_f - \mathbb{E}[M_f(\mathbb{K}_{j^*(\boldsymbol{A})}, \boldsymbol{Z})]\Big)^2\right] \quad (22)\end{aligned}$$

where $\mathbb{K}_{j^*(\boldsymbol{a})}$ denotes the control that our policy chooses when the arrival vector is given by $\boldsymbol{a}$. Note that the second-order expectation (22) is finite since $A_f[t]$ has finite support and the fact that the service $M_f[t]$ is no more than the arrival $A_f[t]$. Hence, (22) can be bounded by some positive constant B. Also note that the arrivals $\boldsymbol{A}$ is independent of $\boldsymbol{x}$; thus, we have

$$\begin{aligned}&\triangle V[t]\\&\le \sum_{f\in\mathcal{F}}(1-q_f)\lambda_f x_f \quad (23)\\&\quad - \sum_{\boldsymbol{a}} P(\boldsymbol{a})\sum_{f\in\mathcal{F}} x_f\mathbb{E}[M_f(\mathbb{K}_{j^*(\boldsymbol{a})}, \boldsymbol{Z})] + B. \quad (24)\end{aligned}$$

Since the arrival process lies strictly within $\mathcal{A}(\boldsymbol{q})$ as defined in (10), there exists $\epsilon > 0$, independent of $\boldsymbol{x}$, such that for each $f\in\mathcal{F}$, we have

$$(1-q_f)\lambda_f \le \sum_{\boldsymbol{a}} P(\boldsymbol{a})\sum_{\mathbb{K}_j\in\mathcal{K}} \alpha_j(\boldsymbol{a})\mathbb{E}[M_f(\mathbb{K}_j, \boldsymbol{Z})] - \epsilon$$

for some $\{\alpha_j(\boldsymbol{a})\}_j$. Substitute into (24)

$$\begin{aligned}&\triangle V(\boldsymbol{x})\\&\le \sum_{f\in\mathcal{F}}\sum_{\boldsymbol{a}} P(\boldsymbol{a})\sum_{\mathbb{K}_j\in\mathcal{K}} \alpha_j(\boldsymbol{a})\mathbb{E}[M_f(\mathbb{K}_{j(\boldsymbol{a})}, \boldsymbol{Z})]x_f\\&\quad - \sum_{\boldsymbol{a}} P(\boldsymbol{a})\sum_{f\in\mathcal{F}} x_f\mathbb{E}[M_f(\mathbb{K}_{j^*(\boldsymbol{a})}, \boldsymbol{Z})]\\&\quad + B - \epsilon\sum_{f\in\mathcal{F}} x_f\\&= \sum_{\boldsymbol{a}} P(\boldsymbol{a})\sum_{\mathbb{K}_j\in\mathcal{K}} \alpha_j(\boldsymbol{a})\sum_{f\in\mathcal{F}} x_f\mathbb{E}[M_f(\mathbb{K}_{j(\boldsymbol{a})}, \boldsymbol{Z})]\\&\quad - \sum_{\boldsymbol{a}} P(\boldsymbol{a})\sum_{\mathbb{K}_j\in\mathcal{K}} \alpha_j(\boldsymbol{a})\sum_{f\in\mathcal{F}} x_f\mathbb{E}[M_f(\mathbb{K}_{j^*(\boldsymbol{a})}, \boldsymbol{Z})]\\&\quad + B - \epsilon\sum_{f\in\mathcal{F}} x_f.\end{aligned}$$

Our dynamic coding strategy chooses the control $\mathbb{K}_{j[t]}\in\mathcal{K}$ as in (13); thus

$$\begin{aligned}&\sum_{\boldsymbol{a}} P(\boldsymbol{a})\sum_{\mathbb{K}_j\in\mathcal{K}} \alpha_j(\boldsymbol{a})\sum_{f\in\mathcal{F}} x_f\mathbb{E}[M_f(\mathbb{K}_{j(\boldsymbol{a})}, \boldsymbol{Z})]\\&\le \sum_{\boldsymbol{a}} P(\boldsymbol{a})\sum_{\mathbb{K}_j\in\mathcal{K}} \alpha_j(\boldsymbol{a})\sum_{f\in\mathcal{F}} x_f\mathbb{E}[M_f(\mathbb{K}_{j^*(\boldsymbol{a})}, \boldsymbol{Z})].\end{aligned}$$

Thus, we have

$$\triangle V(\boldsymbol{x}) \le B - \epsilon\sum_{f\in\mathcal{F}} x_f.$$

Taking the expectation over $\boldsymbol{X}$, and sum the expected drift over $t=0$ through $T-1$, we have

$$\frac{1}{2\beta}\sum_f \mathbb{E}\left[X_f^2[T] - X_f^2[0]\right] \le TB - \epsilon\sum_{t=0}^{T}\sum_f \mathbb{E}\left[X_f[t]\right].$$

Thus, by rearranging terms, we get

$$\frac{1}{T}\sum_{t=0}^{T}\sum_f \mathbb{E}\left[X_f[t]\right] \le \frac{B}{\epsilon} + \frac{1}{2\epsilon\beta T}\sum_f \mathbb{E}\left[X_f^2[0]\right].$$

Taking the limit as T goes to infinity yields

$$\limsup_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T}\sum_{f}\mathbb{E}\left[X_f[t]\right]\leq\frac{B}{\epsilon}.$$

Since $X_f[t]$ only takes on nonnegative values, thus the aforementioned equation implies for each f, we have

$$\limsup_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T}\mathbb{E}\left[X_f[t]\right]\leq\frac{B}{\epsilon}$$

i.e., all deficit counters are stable.

Appendix C
Proof of Proposition 2

We define the Lyapunov function

$$L(\boldsymbol{X})=\frac{1}{2}\sum_{f}X_f^2$$

and study its expected drift given by

$$\begin{aligned}&\triangle L(\boldsymbol{X},t)\\&=\mathbb{E}\left[L(\boldsymbol{X}[t+1])-L(\boldsymbol{X}[t])|\boldsymbol{X}[t]=\boldsymbol{x},\lambda[t]=\lambda\right]\\&\leq\mathbb{E}\left[\sum_{f}\Big(X_f[t]+(1-q_f)A_f[t]-M_f[t]\Big)^2\right.\\&\quad\left.-\sum_{f}X_f^2[t]\Big|\boldsymbol{X}[t],\lambda[t]\right].\end{aligned}$$

In the following derivation, we omit the frame index t for brevity. Note that $X_f[t]$ and $\lambda_f[t]$ can be pulled out of the conditional expectation

$$\begin{aligned}&\triangle L(\boldsymbol{X},t)\\&\leq\sum_{f}X_f\big((1-q_f)\lambda_f-\mathbb{E}[M_f|\boldsymbol{X},\lambda]\big)\\&\quad+\frac{1}{2}\sum_{f}\big((1-q_f)\lambda_f-\mathbb{E}[M_f|\boldsymbol{X},\lambda]\big)^2\\&\leq\sum_{f}X_f\big((1-q_f)\lambda_f-\mathbb{E}[M_f|\boldsymbol{X},\lambda]\big)+B_1\end{aligned}$$

where $B_1<\infty$ is a finite constant that is a function of $A_{\max}$.

Note that by our feasibility assumption, the arrival process $\boldsymbol{A}[t]=\{A_{\min}\}_f$ lies within the set $\mathcal{A}(\boldsymbol{q})$, i.e., there exists a stationary policy such that all deficit counters can be stabilized. We assume that such a stationary policy under the arrival vector $\boldsymbol{A}[t]=\{A_{min}\}_f$ yields an average service vector $\tilde{\mu}=\{\tilde{\mu}\}_f$, such that $\tilde{\mu}_f>(1-p_f)A_{\min}$ for all flows. By adding and subtracting the term $\sum_f X_f\tilde{\mu}_f$, we get

$$\begin{aligned}&\triangle L(\boldsymbol{X},t)\\&\leq B_1+\sum_{f}[(1-q_f)\lambda_f-\tilde{\mu}_f]X_f\\&\quad+\sum_{f}X_f\tilde{\mu}_f-\sum_{f}X_f\mathbb{E}[M_f|\boldsymbol{X},\lambda].\end{aligned}\tag{25}$$

We have $(25)\leq 0$ since when the arrival vector $\lambda[t]=\{A_{\min}\}_f$, our policy maximizes $\sum_f X_f\mathbb{E}[M_f|\boldsymbol{X},\{A_{\min}\}_f]$; thus, $\sum_f X_f\tilde{\mu}_f\leq\sum_f X_f\mathbb{E}[M_f|\boldsymbol{X},\{A_{\min}\}_f]$. When the arrival vector has $\lambda_f[t]>A_{\min}$ for any f, $\sum_f X_f\mathbb{E}[M_f|\boldsymbol{X},\lambda]$ is at least $\sum_f X_f\mathbb{E}[M_f|\boldsymbol{X},\{A_{\min}\}_f]$. Thus, $\sum_f X_f\tilde{\mu}_f\leq\sum_f X_f\mathbb{E}[M_f|\boldsymbol{X},\lambda]$ for all arrival vectors. Thus, we have

$$\triangle L(\boldsymbol{X},t)\leq B_1+\sum_{f}[(1-q_f)\lambda_f-\tilde{\mu}_f]X_f.\tag{26}$$

To see the boundedness of $\sum_f[(1-q_f)\lambda_f-\tilde{\mu}_f]X_f$, we first show the following lemma.

Lemma 3: Let $q_{\max}=\max_f q_f$. Define

$$r\stackrel{\triangle}{=}\frac{1}{1-q_{\max}}\left(\frac{A_{\max}}{\alpha R^2}+\max_f U_f'(A_{\min})\right).$$

If $X_f[t]>R(r+A_{\max})$ for any f and $t\geq R$, then $\lambda_f[t]=A_{\min}$.

Proof: Since the arrival in a frame is upper-bounded by $A_{\max}$, we must have $X_f[t]\leq X_f[t-1]+A_{\max}$. Thus, $X_f[t]>R(r+A_{\max})$ implies $X_f[i]>Rr$ for all $i\in\{t-R+1,\ldots,t\}$. Therefore, for all flow f and each frame $i\in\{t-R+1,\ldots,t\}$, we have

$$\begin{aligned}&\lambda_f[i]-\lambda_f[i-1]\\&\leq\alpha(RU_f'(\lambda_f[i-1])-(1-q_f)X_f[i-1])\\&\stackrel{(a)}{\leq}\alpha RU_f'(A_{\min})-\alpha(1-q_{\max})rR\\&=\alpha RU_f'(A_{\min})-\alpha R\left(\frac{A_{\max}}{\alpha R^2}+\max_f U_f'(A_{\min})\right)\\&\leq-\frac{A_{\max}}{R}\end{aligned}$$

where (a) holds because $U_f'(\cdot)$ is a decreasing function and $\lambda_f[\cdot]\geq A_{\min}$. This implies that for each $i\in\{t-R+1,\ldots,t\}$, the rate $\lambda_f[i]$ will decrease by at least $A_{\max}/R$ in each frame until it hits its minimum possible value of $A_{\min}$, and stays at $A_{\min}$ until frame t. Thus, even if $\lambda_f[t-R]=A_{\max}$, at the end of the subsequent R frames, the flow rate will for sure decrease to $\lambda_f[t]=A_{\min}$. ■

Based on Lemma 3, we let

$$g(\alpha,R)\stackrel{\triangle}{=}R(r+A_{\max}).$$

Then we have

$$\begin{aligned}&\sum_{f}[(1-q_f)\lambda_f[t]-\tilde{\mu}_f]X_f[t]\\&=\sum_{X_f[t]>g(\alpha,R)}[(1-q_f)\lambda_f[t]-\tilde{\mu}_f]X_f[t]\\&\quad+\sum_{X_f[t]\leq g(\alpha,R)}[(1-q_f)\lambda_f[t]-\tilde{\mu}_f]X_f[t]\\&\leq\sum_{X_f[t]>g(\alpha,R)}[(1-q_f)A_{\min}-\tilde{\mu}_f]X_f[t]\\&\quad+|\mathcal{F}|g(\alpha,R)A_{\max}.\end{aligned}$$

To bound the remaining term, note that we have $(1-q_f)A_{\min} - \tilde{\mu}_f \leq -\epsilon$ for some $\epsilon > 0$ for all flows f by our feasibility assumption. Then, we have

$$\begin{aligned}
&\sum_f [(1-q_f)\lambda_f[t] - \tilde{\mu}_f]X_f[t] \\
&\quad \leq -\epsilon \left[\sum_f X_f[t] - \sum_{X_f[t] \leq g(\alpha,R)} X_f[t] \right] \\
&\qquad + |\mathcal{F}| g(\alpha,R) A_{\max} \\
&\quad \leq -\epsilon \sum_f X_f[t] + |\mathcal{F}| g(\alpha,R)(A_{\max} + \epsilon).
\end{aligned}$$

We define $B_2(\alpha,R) = |\mathcal{F}| g(\alpha,R)(A_{\max}+\epsilon)$. Substitute the above into (26), we have

$$\triangle L(\boldsymbol{X},t) \leq -\epsilon \sum_f X_f[t] + B_1 + B_2(\alpha,R).$$

This implies that if $\sum_f X_f[t] \geq (B_1 + B_2(\alpha,R) + \delta)/\epsilon$ for some $\delta > 0$, then $\triangle L(\boldsymbol{X},t) \leq -\delta$.

Also note that we have

$$\left(\sum_f X_f[t] \right)^2 \geq \sum_f X_f^2[t] = 2L(\boldsymbol{X},t).$$

Thus, if $L(\boldsymbol{X},t) \geq \frac{1}{2}\left[(B_1 + B_2(\alpha,R) + \delta)/\epsilon\right]^2$, then $\triangle L(\boldsymbol{X},t) \leq -\delta$. Further, $\triangle L(\boldsymbol{X},t) \leq B_1 + B_2(\alpha,R)$ otherwise. These facts imply that as $t \to \infty$

$$\mathbb{E}[L(\boldsymbol{X},t)] \leq \left(\frac{B_1 + B_2(\alpha,R) + \delta}{\sqrt{2}\epsilon} \right)^2 + B_1 + B_2(\alpha,R).$$

Defining the right-hand side of the aforementioned inequality to be $c(\alpha,R)$ gives the desired result.

Appendix D
Derivation of $F_{Z^{\mathrm{RR}}(K)}(x)$

We first derive the probability $P(Z^{\mathrm{RR}}(K) = y)$ that the transmission to N users finishes in exactly $y = rK + k$ time slots, where $r = \lfloor y/K \rfloor$, $k = y \bmod (K)$ if $y \bmod (K) \neq 0$, and $r = y/K - 1$, $k = K$ if $y \bmod (K) = 0$ such that $y = rK + k$ for $k \in \{1,2,\ldots,K\}$.

Note that for packets $\{1,2,\ldots,k\}$, they get a total of $r+1$ transmission opportunities up to the yth slot, while for the rest of the packets $\{k+1,k+2,\ldots,K\}$, they receive r transmission opportunities in the total y time slots. Also, the packet k is being transmitted by the BS in time slot y. Thus, the event of $\{Z^{\mathrm{RR}}(K) = y\} = E_1 \cap E_2 \cap E_3$, where E_1, E_2, and E_3 are independent events as described next.

1) $E_1 = \{$The packets $\{1,\ldots,k-1\}$ are received successfully by all N users in the previous $r+1$ transmission opportunities$\}$. $P(E_1) = \left(1 - \bar{c}^{r+1}\right)^{N(k-1)}$.
2) $E_2 = \{$The packets $\{k+1,\ldots,K\}$ are received successfully by all N users in the previous r transmission opportunities$\}$. $P(E_2) = \left(1 - \bar{c}^{r}\right)^{N(K-k)}$.
3) $E_3 = \{n \in \{1,2,\ldots,N\}$ users receive packet k successfully for the first time in time-slot y (i.e., the $(r+1)$st transmission of packet k) and the remaining $N-n$ users have successfully received it in the previous r transmissions opportunities of packet $k\}$. $P(E_3) = \sum_{n=1}^{N} \binom{N}{n} \left(1 - \bar{c}^r\right)^{N-n} \left(\bar{c}^r c\right)^n$.

The production of the probability of the aforementioned three events gives $P(Z^{\mathrm{RR}}(K) = y)$. Then, we get $F_{Z^{\mathrm{RR}}(K)}(x) = P(Z^{\mathrm{RR}}(K) \leq x)$ by summing up $P(Z^{\mathrm{RR}}(K) = y)$ for all $0 \leq y \leq x$.

References

[1] D. P. Bertsekas, *Dynamic Programming and Optimal Control*. Belmont, MA: Athena Scientific, 1995.

[2] P. P. Bhattacharya and A. Ephremides, "Optimal scheduling with strict deadlines," *IEEE Trans. Autom. Control*, vol. 34, no. 7, pp. 721–728, Jul. 1989.

[3] L. Bui, A. Eryilmaz, R. Srikant, and X. Wu, "Joint asynchronous congestion control and distributed scheduling for wireless networks," in *Proc. Infocom*, pp. 2072–2080.

[4] T. M. Cover and J. Thomas, *Elements of Information Theory*. Hoboken, NJ: Wiley, 1991.

[5] A. Eryilmaz, A. Ozdaglar, and M. Medard, "On delay performance gains from network coding," in *Proc. 40th Annu. Conf. Inf. Sci. Syst.*, Mar. 2006, pp. 864–870.

[6] A. Eryilmaz and R. Srikant, "Fair resource allocation in wireless networks using queue-length based scheduling and congestion control," in *Proc. 24th Annu. Joint Conf. IEEE Computer and Commun. Soc.*, Miami, FL, Mar. 2005, vol. 3, pp. 1794–1803.

[7] A. Eryilmaz and R. Srikant, "Joint congestion control, routing and MAC for stability and fairness in wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 14, no. 6, pp. 1514–1524, Aug. 2006, Special Issue on Nonlinear Optimization of Communication System.

[8] H. Gangammanavar and A. Eryilmaz, "Dynamic coding and rate-control for serving deadline-constrained traffic over fading channels," in *Proc. IEEE Int. Symp. Inf. Theory Proc.,*, 2010, pp. 1788–1792.

[9] I. H. Hou, V. Borkar, and P. R. Kumar, "A theory of QoS for wireless," in *Proc. IEEE INFOCOM*, Jun. 2009, pp. 486–494.

[10] I.-H. Hou and P. R. Kumar, "Admission control and scheduling for qos guarantees for variable-bit-rate applications on wireless channels," in *Proc. 10th ACM Int. Symp. Mobile ad hoc Netw. Comput.*, 2009, pp. 175–184.

[11] J. J. Jaramillo and R. Srikant, "Optimal scheduling for fair resource allocation in ad hoc networks with elastic and inelastic traffic," *IEEE/ACM Trans. Netw.*, vol. 19, no. 4, pp. 1125–1136, Aug. 2009.

[12] J. J. Jaramillo, R. Srikant, and L. Ying, "Scheduling for optimal rate allocation in ad hoc networks with heterogeneous delay constraints," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 5, pp. 979–987, May 2011.

[13] M. J. Neely, *Stochastic Network Optimization With Application to Communication and Queueing Systems*. San Rafael, CA: Morgan & Claypool, 2010.

[14] M. J. Neely, E. Modiano, and C.-P. Li, "Fairness and optimal stochastic control for heterogeneous networks," *IEEE/ACM Trans. Netw*, vol. 16, no. 2, pp. 396–409, Apr. 2008.

[15] M. J. Neely, E. Modiano, and C. E. Rohrs, "Dynamic power allocation and routing for time varying wireless networks," in *Proc. IEEE Infocom*, Apr. 2003, pp. 745–755.

[16] D. G. Pandelis and D. Teneketzis, "Stochastic scheduling in priority queues with strict deadlines," *Adv. Appl. Probab.*, vol. 26, pp. 258–279, 1994.

[17] D. Rajan, A. Sabharwal, and B. Aazhang, "Delay-bounded packet scheduling of bursty traffic over wireless channels," *IEEE Trans. Info. Theory*, vol. 50, no. 1, pp. 125–144, Jan. 2004.

[18] R. Srikant, *The Mathematics of Internet Congestion Control*. Boston, MA: Birkhäuser, 2004.

[19] R. N. Swamy and T. Javidi, "Optimal code length for bursty sources with deadlines," in *Proc. IEEE Int. Conf. Symp. Inf. Theory*, Piscataway, NJ, 2009, pp. 2694–2698.

[20] L. Tassiulas, "Scheduling and performance limits of networks with constantly varying topology," *IEEE Trans. Inf. Theory*, vol. 43, no. 3, pp. 1067–1073, May 1997.

[21] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Trans. Autom. Control*, vol. 37, no. 12, pp. 1936–1948, Dec. 1992.
[22] L. Tassiulas and A. Ephremides, "Dynamic server allocation to parallel queues with randomly varying connectivity," *IEEE Trans. Inf. Theory*, vol. 39, no. 2, pp. 466–478, Mar. 1993.
[23] Z.-X. Zhao, S. S. Panwar, and D. Towsley, "Queueing performance with impatient customers," in *Proc. INFOCOM*, Apr. 1991, vol. 1, pp. 400–409.

Ruogu Li (S'10) received his B.S. degree in Electronic Engineering from Tsinghua University, Beijing, in 2007. He is currently a Ph.D. student in Electrical and Computer Engineering at The Ohio State University. His research interests include optimal network control, wireless communication networks, low-delay scheduling scheme design, and cross-layer algorithm design.

Harsha Gangammanavar received his B.E. in Electrical Engineering from Visvesvaraya Technological University in 2007 and M.S. in Electrical Engneering from Ohio State University in 2010. He is currently a Ph.D. student in Operations Research at the Ohio State University. His research interests include Stochastic Programming, Optimal Control, and Large Scale Optimization algorithm design.

Atilla Eryilmaz (S'00–M'06) received his M.S. and Ph.D. degrees in Electrical and Computer Engineering from the University of Illinois at Urbana-Champaign in 2001 and 2005, respectively. Between 2005 and 2007, he worked as a Postdoctoral Associate at the Laboratory for Information and Decision Systems at the Massachusetts Institute of Technology. He is currently an Assistant Professor of Electrical and Computer Engineering at The Ohio State University. His research interests include communication networks, optimal control of stochastic networks, optimization theory, distributed algorithms, stochastic processes, and network coding. He received the NSF CAREER and Lumley Research Awards in 2010.