Dynamic Markov Decision Policies for Delay Constrained Wireless Scheduling

Michael J. Neely , Sucha Supittayapornpong

***Abstract*— We consider a one-hop wireless system with a small number of delay constrained users and a larger number of users without delay constraints. We develop a scheduling algorithm that reacts to time varying channels and maximizes throughput utility (to within a desired proximity), stabilizes all queues, and satisfies the delay constraints. The problem is solved by reducing the constrained optimization to a set of weighted stochastic shortest path problems, which act as natural generalizations of max-weight policies to Markov decision networks. We also present approximation results for the corresponding shortest path problems, and discuss the additional complexity and delay incurred as compared to systems without delay constraints. The solution technique is general and applies to other constrained stochastic decision problems.**

***Index Terms*— Queueing systems, Network analysis and control, Markov processes**

I. Introduction

This paper considers delay-aware scheduling in a multi-user wireless uplink or downlink with K delay-constrained users and N delay-unconstrained users, each with different transmission channels. The system operates in slotted time with normalized slots $t \in \{0, 1, 2, \ldots\}$. Every slot, a random number of new packets arrive from each user. Packets are queued for eventual transmission, and every slot the scheduler looks at the queue backlog and the current channel states and chooses one channel to serve. The number of packets transmitted over that channel depends on its current channel state. The goal is to stabilize all queues, satisfy average delay constraints for the delay-constrained users, and drop as few packets as possible.

Without the delay constraints, this problem is a classical *opportunistic scheduling* problem, and can be solved with efficient max-weight algorithms based on Lyapunov drift and Lyapunov optimization (see [2] and references therein). The delay constraints make the problem a much more complex *Markov Decision Problem* (MDP). While general methods for solving MDPs exist (see, for example, [3][4][5]), they typically suffer from a curse of dimensionality. Specifically, the number of queue state vectors grows exponentially in the number of queues. Thus, a general problem with many queues has an intractably large state space. This creates non-polynomial implementation complexity for offline approaches such as linear programming [3][4], and non-polynomial complexity and/or learning time for online or quasi online/offline approaches such as Q-learning [6][7].

We do not solve this fundamental curse of dimensionality. Rather, we avoid this difficulty by focusing on the special structure that arises in a wireless network with a *relatively small number of delay-constrained users* (say, $K \leq 5$), but with an arbitrarily large number of users without delay constraints (so that N can be large). This is an important scenario, particularly in cases when the number of "best effort" users in a network is much larger than the number of delay-constrained users. We develop a solution that, on each slot, requires a computation that has a complexity that depends exponentially in K, but only polynomially in N. Further, the resulting convergence times and delays are polynomial in the total number of queues $K + N$. Our solution uses a concept of *forced renewals* that introduces a deviation from optimality that can be made arbitrarily small with a corresponding polynomial tradeoff in convergence time. Finally, we show that a simple Robbins-Monro iteration can be used to approximate the required computations when channel and traffic statistics are unknown. Our methods are general and can be applied to other MDPs for networks with similar structure.

Related prior work on delay optimality for multi-user opportunistic scheduling under special symmetric assumptions is developed in [8][9][10], and single-queue delay optimization problems are treated in [11][12][13][14] using dynamic programming and Markov Decision theory. Approximate dynamic programming algorithms are applied to multi-queue switches in [15] and shown to perform well in simulation. Optimal asymptotic energy-delay tradeoffs are developed for single queue systems in [16], and optimal energy-delay and utility-delay tradeoffs for multi-queue systems are treated in [17][18]. The algorithms of [17][18] have very low complexity and provably converge quickly even for large networks, although the tradeoff-optimal delay guarantees they achieve do not necessarily optimize the coefficient multiplier in the delay expression.

Our approach in the present paper treats the MDP problem associated with delay constraints using Lyapunov drift and Lyapunov optimization theory [2]. This theory has been used to stabilize queueing networks [8] and provide utility optimization [2][19][20][21][22][23][24] via simple *max-weight* principles. We extend the max-weight principles to treat networks with *Markov decisions*, where the network costs depend on both the control actions taken and the current state (such as the queue state) the system is in. For each cost constraint we define a *virtual queue*, and show that the constrained

This material was presented in part at the 48th IEEE Conf. on Decision and Control (CDC), Shanghai, China, Dec. 2009 [1].

The authors are with the Electrical Engineering department at the University of Southern California, Los Angles, CA.

This material is supported in part by one or more of the following: the DARPA IT-MANET program grant W911NF-07-0028, the NSF Career grant CCF-0747525.

MDP can be solved using Lyapunov drift theory implemented over a variable-length frame, where "max-weight" rules are replaced with weighted stochastic shortest path problems. This is similar to the Lagrange multiplier approaches used in the related works [13][14] that treat power minimization for single-queue wireless links with an average delay constraint. The work in [13] uses stochastic approximation with a 2-timescale argument and a limiting ordinary differential equation. The work in [14] treats a single-queue MIMO system using primal-dual updates [25]. Our virtual queues are similar to the Lagrange Multiplier updates in [13][14]. However, we treat multi-queue systems, and we use a different analytical approach that emphasizes stochastic shortest paths over variable length frames. Because of this, our approach can be used in conjunction with a variety of existing techniques for solving shortest path problems (see, for example, [6]). We use a Robbins-Monro technique that is adapted to this context, together with a *delayed queue analysis* to uncorrelate past samples from current queue states. Our resulting algorithm has an implementation complexity that grows exponentially in the number of delay-constrained queues K, but polynomially in the number of delay-unconstrained queues N. Further, we obtain polynomial bounds on convergence times and delays.

The next section describes the network model. Section III presents the weighted stochastic shortest-path algorithm. Section IV describes approximate implementations, and Section V presents a simple simulation example.

II. Network Model

Consider a one-hop wireless queueing network that operates in discrete time with timeslots $t \in \{0, 1, 2, \ldots\}$. The network has K *delay-constrained queues* and N *stability-constrained queues*, for a total of $K + N$ queues indexed by sets $\mathcal{K} \triangleq \{1, \ldots, K\}$ and $\mathcal{N} \triangleq \{K+1, \ldots, K+N\}$. The queues store fixed-length packets for transmission over their wireless channels. Every timeslot, new packets randomly arrive to each queue. Let $\boldsymbol{A}(t) = (A_1(t), \ldots, A_{K+N}(t))$ represent the random packet arrival vector, being a vector of non-negative integers. The stability-constrained queues have an infinite buffer space. The delay-constrained queues have a finite buffer space that can store b packets (for some positive integer b). The network channels can vary from slot to slot. Let $\boldsymbol{S}(t) = (S_1(t), \ldots, S_{K+N}(t))$ be the channel state vector on slot t, representing conditions that affect transmission rates. The stacked vector $[\boldsymbol{A}(t), \boldsymbol{S}(t)]$ is assumed to be independent and identically distributed (i.i.d.) over slots, with possibly correlated entries on the same slot.

Every slot t, the network controller observes the channel states $\boldsymbol{S}(t)$ and chooses a *transmission rate vector* $\boldsymbol{\mu}(t) = (\mu_1(t), \ldots, \mu_{K+N}(t))$, being a vector of non-negative integers. The choice of $\boldsymbol{\mu}(t)$ is constrained to a set $\Gamma_{\boldsymbol{S}(t)}$ that depends on the current $\boldsymbol{S}(t)$. A simple example is a system with ON/OFF channels where the controller can transmit a single packet over at most one ON channel per slot, as in [8]. In this example, $\boldsymbol{S}(t)$ is a binary vector of channel states, and $\Gamma_{\boldsymbol{S}(t)}$ restricts $\boldsymbol{\mu}(t)$ to be a binary vector with at most one non-zero entry and with $\mu_i(t) = 0$ whenever $S_i(t) = 0$. We assume that for each possible channel state vector $\boldsymbol{S}$, the set $\Gamma_{\boldsymbol{S}}$ has the natural property that for any $\boldsymbol{\mu} \in \Gamma_{\boldsymbol{S}}$, any non-negative integer vector $\boldsymbol{\mu}'$ that is entrywise less than or equal to $\boldsymbol{\mu}$ is also in $\Gamma_{\boldsymbol{S}}$. In addition to constraining $\boldsymbol{\mu}(t)$ to take values in $\Gamma_{\boldsymbol{S}(t)}$ every slot t, we shall soon also restrict the $\mu_k(t)$ values for the delay-constrained queues $k \in \mathcal{K}$ to be at most the current number of packets in queue k. This is a natural restriction, although we *do not* place such a restriction on the stability-constrained queues $n \in \mathcal{N}$. This is a technical detail that will be important later, when we show that the *effective dimension* of the resulting Markov decision problem is K, independent of the number of stability-constrained queues N.

Let $\boldsymbol{Q}(t) = (Q_1(t), \ldots, Q_{K+N}(t))$ represent the vector of current queue backlogs, and define $d_n(t) = A_n(t) - \mu_n(t)$. The queue dynamics for the stability-constrained queues are:

$$Q_n(t+1) = \max[Q_n(t) + d_n(t), 0] \ \forall n \in \mathcal{N} \qquad (1)$$

where the $\max[\cdot, 0]$ operation allows, in principle, a service variable $\mu_n(t)$ to be independent of whether or not $Q_n(t)$ is empty. If $Q_n(t) < \mu_n(t)$, the transmitter only transmits the $Q_n(t)$ packets available over channel n, and the residual capability of transmitting $\mu_n(t) - Q_n(t)$ additional packets is either wasted or used with idle fill.

The delay-constrained queues have a different queue dynamic. Because of the finite buffer, we must allow packet dropping. Let $D_k(t)$ be the number of dropped packets on slot t. The queue dynamics for the delay-constrained queues are given by:

$$Q_k(t+1) = Q_k(t) - \mu_k(t) - D_k(t) + A_k(t) \ \forall k \in \mathcal{K} \quad (2)$$

Note that this does not have any $\max[\cdot, 0]$ operation, because we will force the $\mu_k(t)$ and $D_k(t)$ decisions to be such that we never serve or drop packets that we do not have. The precise constraints on these decision variables are given after the introduction of a *forced renewal event*, defined in the next subsection.

A. Forced Renewals

We want to force the delay-constrained queues to repeatedly visit a *renewal state* of being simultaneously empty. Thus, at the end of every slot, with probability $\phi > 0$ we independently drop all unserved packets in all delay constrained queues $k \in \mathcal{K}$. The stability-constrained queues do not experience such forced drops. Specifically, let $\phi(t)$ be an i.i.d. Bernoulli process that is 1 with probability ϕ every slot t, and 0 otherwise. Assume $\phi(t)$ is independent of $[\boldsymbol{A}(t), \boldsymbol{S}(t)]$. If $\phi(t) = 1$, we say slot t experiences a *forced renewal event*. The decision options for $\mu_k(t)$ and $D_k(t)$ for $k \in \mathcal{K}$ are then additionally constrained as follows: If $\phi(t) = 0$, then:

$$\begin{aligned} \mu_k(t) &\in \{0, 1, \ldots, Q_k(t)\} \\ D_k(t) &\in \{\max[A_k(t) + Q_k(t) - b, 0], \ldots, A_k(t)\} \end{aligned}$$

so that during normal operation, we can serve at most $Q_k(t)$ packets from queue k (so new arrivals cannot be served), and we can drop only new arrivals, necessarily dropping any new

arrivals that would exceed the finite buffer capacity. However, if $\phi(t) = 1$ we have:

$$\begin{aligned} \mu_k(t) &\in \{0, 1, \ldots, Q_k(t)\} \\ D_k(t) &= Q_k(t) - \mu_k(t) + A_k(t) \end{aligned}$$

so that $\mu_k(t)$ is constrained as before, but $D_k(t)$ is then equal to the remaining packets (if any) at the end of the slot.

We shall optimize the system under the assumption that the forced renewal process $\phi(t)$ is uncontrollable. This provides an analyzable system that lends itself to simple approximations, as shown in later parts of the paper. While these forced renewals create inefficiency in the system, the rate of dropped packets due to forced renewals is at most $(Kb + \sum_{k=1}^{K} \mathbb{E}[A_k(t)])\phi$, which assumes the worst case of dropping a full buffer plus all new arrivals every renewal event. This value can be made arbitrarily small with a small choice of ϕ. For problems such as minimizing the average drop rate subject to delay constraints in the delay-constrained queues and stability in the stability-constrained queues, it can be shown that this $O(\phi)$ term bounds the gap between system optimality without forced renewals and system optimality with forced renewals (see Appendix A). In Theorem 1 we show the disadvantage of using a small value of ϕ is that our average queue bound for the stability-constrained queues is $O(1/\phi)$.

Define a *renewal frame* as the sequence of slots starting just after a renewal event and ending at the next renewal event. Assume that all delay-constrained queues are initially empty, so that time 0 starts the first renewal frame. Define $t_0 = 0$, and let t_r for $r \in \{1, 2, \ldots\}$ represent the sequence that marks the beginning of each renewal frame. For $r \in \{0, 1, 2, \ldots\}$, define $T_r \triangleq t_{r+1} - t_r$ as the duration of the rth renewal frame. Note that $\{T_r\}_{r=0}^{\infty}$ are i.i.d. geometric random variables with $\mathbb{E}[T_r] = 1/\phi$.

B. Markov Decision Notation

Define $\omega(t) \triangleq [\boldsymbol{A}(t), \boldsymbol{S}(t)]$ as the observed arrivals and channels of the network on slot t, and define the random network event $\Omega(t) \triangleq [\omega(t), \phi(t)]$. Then $\Omega(t)$ is i.i.d. over slots. The control decision constraints of the previous section can be summarized with the following simple notation: Let $\mathcal{Z} \triangleq \{0, 1, \ldots, b\}^K$ be the K-dimensional state space for the delay-constrained queues, and let $z(t) \triangleq (Q_k(t))_{k \in \mathcal{K}}$ represent the current state of these queues. Every slot t, the controller observes the random event $\Omega(t)$ and the queue state $z(t)$, and makes a *control action* $\alpha(t)$, which determines all decision variables $\mu_i(t)$ for $i \in \{1, \ldots, K + N\}$ and $D_k(t)$ for $k \in \mathcal{K}$. Control action $\alpha(t)$ is chosen in a set $\mathcal{A}_{\Omega(t), z(t)}$ that depends on $\Omega(t)$ and $z(t)$. All of the decision variables described in the previous subsection are constrained only in terms of $\Omega(t)$ and $z(t)$. In particular, the queue states $Q_n(t)$ for $n \in \mathcal{N}$ do not constrain the decisions.

Recall that $d_n(t) \triangleq A_n(t) - \mu_n(t)$. Then $\alpha(t)$, $\Omega(t)$, $z(t)$ together affect the vector $\boldsymbol{d}(t) = (d_n(t))_{n \in \mathcal{N}}$ through a deterministic function $\hat{d}_n(\alpha(t), \Omega(t), z(t))$:

$$d_n(t) = \hat{d}_n(\alpha(t), \Omega(t), z(t)) \ \forall n \in \mathcal{N} \tag{3}$$

Further, $\alpha(t)$, $\Omega(t)$, $z(t)$ together define the *transition probabilities* from $z(t)$ to $z(t+1)$, defined for all states i and j in $\mathcal{Z}$:

$$P_{ij}(\alpha, \Omega) = Pr[z(t+1) = j | z(t) = i, \alpha(t) = \alpha, \Omega(t) = \Omega] \tag{4}$$

From the equation (2) we find that $P_{ij}(\alpha, \Omega) \in \{0, 1\}$, so that next states $z(t+1)$ are deterministic given $\alpha(t)$, $\Omega(t)$, $z(t)$. Finally, we define a general *penalty vector* $\boldsymbol{y}(t) = (y_0(t), y_1(t), \ldots, y_L(t))$, for some integer $L \geq 0$, where penalties $y_l(t)$ are deterministic functions of $\alpha(t)$, $\Omega(t)$, $z(t)$:

$$y_l(t) \triangleq \hat{y}_l(\alpha(t), \Omega(t), z(t)) \tag{5}$$

For example, penalty $y_0(t)$ can be defined as the total number of dropped packets on slot t by defining $y_0(t) = \sum_{k \in \mathcal{K}} D_k(t)$, which is indeed a function of $\alpha(t)$, $\Omega(t)$, $z(t)$.

We assume throughout that all of the above deterministic functions are bounded, so that there is a finite constant β such that for all $l \in \{0, 1, \ldots, L\}$, all $n \in \mathcal{N}$, and all slots t we have:

$$|y_l(t)| \leq \beta \ , \ |d_n(t)| \leq \beta \tag{6}$$

C. The Optimization Problems

A control policy is a method for choosing actions $\alpha(t) \in \mathcal{A}_{\Omega(t), z(t)}$ over slots $t \in \{0, 1, 2, \ldots\}$. We restrict to causal policies that make decisions with knowledge of the past but without knowledge of the future. Suppose a particular control policy is given. Define time averages $\overline{Q}_n$ and $\overline{y}_l$ for $n \in \mathcal{N}$ and $l \in \{0, 1, \ldots, L\}$ by:

$$\begin{aligned} \overline{Q}_n &\triangleq \limsup_{t \to \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}[Q_n(\tau)] \\ \overline{y}_l &\triangleq \limsup_{t \to \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}[y_l(\tau)] \end{aligned}$$

Our goal is to design a control policy to solve the following stochastic optimization problem:

$$\text{Minimize:} \quad \overline{y}_0 \tag{7}$$

$$\text{Subject to:} \quad \overline{y}_l \leq 0 \ \forall l \in \{1, \ldots, L\} \tag{8}$$

$$\overline{Q}_n < \infty \ \forall n \in \mathcal{N} \tag{9}$$

$$\alpha(t) \in \mathcal{A}_{\Omega(t), z(t)} \ \forall t \in \{0, 1, 2, \ldots\} \tag{10}$$

That is, we desire to minimize the time average of the $y_0(t)$ penalty, subject to time average constraints on the other penalties, and subject to queue stability (called *strong stability*) for all stability-constrained queues. The general structure (7)-(10) fits a variety of network optimization problems. For example, if we define $y_0(t)$ as the sum packet drops $\sum_{k \in \mathcal{K}} D_k(t)$, define $L = K$, and define $y_k(t) = Q_k(t) - Q_{av}$ for all $k \in \mathcal{K}$ (for some positive constant Q_{av}), then the problem (7)-(10) seeks to minimize the total packet drop rate, subject to an average backlog of at most Q_{av} in all delay-constrained queues $k \in \mathcal{K}$, and subject to stability of all stability-constrained queues $n \in \mathcal{N}$.

Alternatively, to enforce an average *delay* constraint W_{av} at all queues $k \in \mathcal{K}$ (for some positive number W_{av}), we can define penalties:

$$y_k(t) = Q_k(t) - (A_k(t) - D_k(t))W_{av} \quad \forall k \in \mathcal{K}$$

Note that the time average of $(A_k(t) - D_k(t))$ is the number $\tilde{\lambda}_k$, the average arrival rate of (non-dropped) packets to queue k. Hence, the constraint $\overline{y}_k \leq 0$ is equivalent to:

$$\overline{Q}_k - \tilde{\lambda}_k W_{av} \leq 0$$

However, by Little's theorem [26] we have $\overline{Q}_k = \tilde{\lambda}_k \overline{W}_k$, where $\overline{W}_k$ is the average delay for queue k, and so the constraint $\overline{y}_k \leq 0$ ensures $\overline{W}_k \leq W_{av}$ (assuming $\tilde{\lambda}_k > 0$).

In the following, we develop a dynamic algorithm that can come arbitrarily close to solving the problem (7)-(10). Our solution is general and applies to any other discrete time Markov decision problem on a general finite state space $\mathcal{Z}$, random events $\Omega(t) = [\omega(t), \phi(t)]$ (for forced renewal process $\phi(t)$), control actions $\alpha(t)$ in a general set $\mathcal{A}_{\Omega(t),z(t)}$, queue equations (1) with $d_n(t)$ given in the form (3), transition probabilities in the form (4), and penalties in the form (5).

D. Slackness Assumptions

Suppose the problem (7)-(10) is *feasible*, so that there exists a policy that satisfies the constraints. It can be shown that the constraint $\overline{Q}_n < \infty$ implies that $\overline{d}_n \leq 0$ (the converse is not necessarily true) [27]. Thus, the following modified problem is feasible whenever the original one is:

$$\begin{aligned}
\text{Minimize:} \quad & \overline{y}_0 && (11)\\
\text{Subject to:} \quad & \overline{y}_l \leq 0 \ \forall l \in \{1, \ldots, L\} && (12)\\
& \overline{d}_n \leq 0 \ \forall n \in \mathcal{N} && (13)\\
& \alpha(t) \in \mathcal{A}_{\Omega(t),z(t)} \ \forall t \in \{0, 1, 2, \ldots\} && (14)
\end{aligned}$$

Define y_0^{opt} as the infimum of $\overline{y}_0$ for the problem (11)-(14), necessarily being less than or equal to the corresponding infimum of the original problem (7)-(10).[1] We show in Theorem 1 that, under a mild slackness condition, the value of y_0^{opt} can be approached arbitrarily closely while maintaining $\overline{Q}_n < \infty$ for all queues $n \in \mathcal{N}$. Thus, y_0^{opt} is also the infimum of $\overline{y}_0$ for the original problem (7)-(10).

The problem (11)-(14) is a constrained Markov decision problem (MDP) with state $(\Omega(t), z(t))$. Under mild assumptions (such as this state space being finite, and the action space $\mathcal{A}_{\Omega,z}$ being finite for each (Ω, z)) the MDP has an *optimal stationary policy* that chooses actions $\alpha(t) \in \mathcal{A}_{\Omega(t),z(t)}$ every slot t as a stationary and possibly randomized function of the state $(\Omega(t), z(t))$ only. We call such policies (Ω, z)*-only policies*. Because this system experiences regular renewals, the performance of any (Ω, z)-only policy can be characterized by ratios of expectations over one renewal frame. Thus, we make the following assumption.

[1]Recall that y_0^{opt} is defined assuming forced renewals of probability ϕ. Thus, y_0^{opt} is typically within a gap of $O(\phi)$ of the minimum cost without such forced renewals (see Appendix A).

Assumption 1: There is an (Ω, z)-only policy $\alpha_1^*(t)$ that satisfies the following over any renewal frame:

$$\begin{aligned}
\frac{\mathbb{E}\left[\sum_{\tau=t_r}^{t_r+T_r-1} y_0^*(\tau)\right]}{1/\phi} &= y_0^{opt} && (15)\\
\frac{\mathbb{E}\left[\sum_{\tau=t_r}^{t_r+T_r-1} d_n^*(\tau)\right]}{1/\phi} &\leq 0 \ \forall n \in \mathcal{N} && (16)\\
\frac{\mathbb{E}\left[\sum_{\tau=t_r}^{t_r+T_r-1} y_l^*(\tau)\right]}{1/\phi} &\leq 0 \ \forall l \in \{1, \ldots, L\} && (17)
\end{aligned}$$

where T_r is the size of the renewal frame, with $\mathbb{E}[T_r] = 1/\phi$, and $y_l^*(\tau)$, $d_n^*(\tau)$ are values under the policy $\alpha^*(t)$ on slot τ of the renewal frame.

We emphasize that Assumption 1 is mild and holds whenever the problem (11)-(14) is feasible and has an optimal stationary policy (i.e., an optimal (Ω, z)-only policy). We now make the following stronger assumption that there exists an (Ω, z)-only policy that can meet the constraints (16)-(17) with "ϵ-slackness," without caring what average value of $y_0(t)$ this policy generates. This assumption is related to standard "Slater-type" assumptions in optimization theory [25].

Assumption 2: There is a value $\epsilon > 0$ and an (Ω, z)-only policy $\alpha_2^*(t)$ (typically different from policy $\alpha_1^*(t)$ in Assumption 1) that satisfies the following over any renewal frame:

$$\begin{aligned}
\frac{\mathbb{E}\left[\sum_{\tau=t_r}^{t_r+T_r-1} d_n^*(\tau)\right]}{1/\phi} &\leq -\epsilon \ \forall n \in \mathcal{N} && (18)\\
\frac{\mathbb{E}\left[\sum_{\tau=t_r}^{t_r+T_r-1} y_l^*(\tau)\right]}{1/\phi} &\leq -\epsilon \ \forall l \in \{1, \ldots, L\} && (19)
\end{aligned}$$

We show in Theorem 1 that systems that satisfy Assumption 2 with larger values of ϵ can operate with smaller average queue sizes in the stability-constrained queues.

E. An Example where Assumptions 1 and 2 are Satisfied

Consider the problem of minimizing the average drop rate subject to an average queue backlog constraint of Q_k^{av} in each queue $k \in \mathcal{K}$, and to stability in each queue $n \in \mathcal{N}$ (such a problem is considered in the simulation example of Section V). The buffer size b in each delay-constrained queue is finite, so that the set $\mathcal{Z}$ is finite. Suppose the set Ω is also finite, and the action space $\mathcal{A}_{\Omega,z}$ is finite for each (Ω, z). Time averages achievable in constrained Markov decision problems with forced renewals and with finite state and action spaces can be shown to also be achievable by stationary policies. Thus, if the problem is feasible, then Assumption 1 holds.

Now assume the arrival rate vector $(\lambda_n)_{n\in\mathcal{N}}$ for the stability-constrained queues is interior to the N-dimensional *capacity region* for those queues (see [27] for a discussion of the capacity region). Specifically, suppose there is a value $\delta > 0$ such that $(\lambda_n + \delta)_{n\in\mathcal{N}}$ is in this capacity region. Assume that $Q_k^{av} > 0$ for all $k \in \mathcal{K}$, and define $\epsilon = \min[\delta, \min_{k\in\mathcal{K}} Q_k^{av}]$. Consider the (Ω, z)-only policy that drops all arrivals of all delay-constrained queues (so $Q_k(t) = 0$ for all $k \in \mathcal{K}$), and that chooses transmission rates for the stability-constrained

queues to support the rates $(\lambda_n + \delta)_{n\in\mathcal{N}}$. This policy yields average backlog at least ϵ less than the required constraint Q_k^{av} in each queue $k \in \mathcal{K}$, and yields $\overline{d}_n = \lambda_n - \overline{\mu}_n \leq -\epsilon$ for all $n \in \mathcal{N}$, and so Assumption 2 is satisfied.

III. The Dynamic Control Algorithm

To solve the problem (7)-(10), we extend the framework of [2] to a case of variable length frames (see related analysis, without renewal frame structures, in [19][20][21][22][28][23][24]). Specifically, for each of the L penalty constraints $\overline{y}_l \leq 0$, we define a *virtual queue* $X_l(t)$ that is initialized to zero and that has dynamic update equation:

$$X_l(t+1) = \max[X_l(t) + y_l(t), 0] \ \forall l \in \{1, \ldots, L\} \quad (20)$$

where $y_l(t) = \hat{y}_l(\alpha(t), \Omega(t), z(t))$ is the lth penalty incurred on slot t by a particular action $\alpha(t) \in \mathcal{A}_{\Omega(t),z(t)}$. The intuition is that if the virtual queue $X_l(t)$ is stable, then the time average of $y_l(t)$ must be non-positive. This turns the time average constraint into a simple queue stability problem.

A. Lyapunov Drift

Define $\boldsymbol{X}(t)$ as a vector of all virtual queues $X_l(t)$ for $l \in \{1, \ldots, L\}$. Define $\boldsymbol{\Theta}(t)$ as the combined vector of all virtual queues and all stability-constrained queues:

$$\boldsymbol{\Theta}(t) \triangleq [\boldsymbol{X}(t), (Q_n(t))_{n\in\mathcal{N}}]$$

Assume all queues are initially empty, so that $\boldsymbol{\Theta}(0) = \mathbf{0}$. Define the following quadratic function:

$$L(t) \triangleq \frac{1}{2}\sum_{n\in\mathcal{N}} Q_n(t)^2 + \frac{1}{2}\sum_{l=1}^{L} X_l(t)^2$$

Let t_r be the start of a renewal frame, with duration T_r. Define $\Delta(t_r)$ as follows:

$$\Delta(t_r) \triangleq L(t_r + T_r) - L(t_r) \quad (21)$$

The conditional expectation of $\Delta(t_r)$, given the queue backlogs $\boldsymbol{\Theta}(t_r)$, is called the *frame-based conditional Lyapunov drift*. It is important to note that the implemented policy $\alpha(t)$ may not be stationary and/or may depend on the queue values $\boldsymbol{\Theta}(t)$ (which can be different on each renewal interval). Thus, the actual system events are not necessarily i.i.d. over different renewal frames. However, these frames are useful because we will analytically compare the frame based Lyapunov drift of the actual policy to the corresponding drifts of the (Ω, z)-only policies of Assumptions 1 and 2.

Lemma 1: (Lyapunov Drift) Under any network control policy that chooses $\alpha(\tau) \in \mathcal{A}_{\Omega(\tau),z(\tau)}$ for all slots τ during a renewal frame $\tau \in \{t_r, \ldots, t_r + T_r - 1\}$, and for any initial queue values $\boldsymbol{\Theta}(t_r)$, we have:

$$\mathbb{E}\left[\Delta(t_r)|\boldsymbol{\Theta}(t_r)\right] \leq B/\phi^2 + \mathbb{E}\left[G(t_r)|\boldsymbol{\Theta}(t_r)\right] \quad (22)$$

where $G(t_r)$ is defined:

$$G(t_r) \triangleq \sum_{n\in\mathcal{N}} Q_n(t_r) \sum_{\tau=t_r}^{t_r+T_r-1} d_n(\tau) + \sum_{l=1}^{L} X_l(t_r) \sum_{\tau=t_r}^{t_r+T_r-1} y_l(\tau) \quad (23)$$

and where B is a finite constant defined:

$$B \triangleq \frac{(2-\phi)\beta^2(N+L)}{2}$$

where we recall β is the bound in (6).

Proof: For any $l \in \{1, \ldots, L\}$ and any $\tau \in \{t_r, \ldots, t_r + T_r - 1\}$ we have by squaring (20) and using $\max[x, 0]^2 \leq x^2$:

$$\begin{aligned} X_l(\tau+1)^2 &\leq (X_l(\tau) + y_l(\tau))^2 \\ &= X_l(\tau)^2 + y_l(\tau)^2 + 2X_l(\tau)y_l(\tau) \\ &= X_l(\tau)^2 + y_l(\tau)^2 + 2X_l(t_r)y_l(\tau) \\ &\quad +2[X_l(\tau) - X_l(t_r)]y_l(\tau) \\ &\leq X_l(\tau)^2 + \beta^2 + 2X_l(t_r)y_l(\tau) + 2\beta^2(\tau - t_r) \end{aligned}$$

where the final inequality holds because the change in $X_l(\tau)$ on any slot is at most β, as is the magnitude of $y_l(\tau)$. Summing the above over $\tau \in \{t_r, \ldots, t_r + T_r - 1\}$ and dividing by 2 yields:

$$\begin{aligned} \frac{X_l(t_r+T_r)^2 - X_l(t_r)^2}{2} &\leq \frac{T_r\beta^2 + \beta^2 T_r(T_r-1)}{2} \\ &\quad + X_l(t_r)\sum_{\tau=t_r}^{t_r+T_r-1} y_l(\tau) \quad (24) \\ &= \frac{\beta^2 T_r^2}{2} + X_l(t_r)\sum_{\tau=t_r}^{t_r+T_r-1} y_l(\tau) \quad (25) \end{aligned}$$

where (24) uses the identity:

$$\sum_{\tau=t_r}^{t_r+T_r-1} (\tau - t_r) = T_r(T_r-1)/2$$

Similarly, it can be shown for any $n \in \mathcal{N}$:

$$\begin{aligned} \frac{Q_n(t_r+T_r)^2 - Q_n(t_r)^2}{2} &\leq \frac{\beta^2 T_r^2}{2} \\ &\quad + Q_n(t_r)\sum_{\tau=t_r}^{t_r+T_r-1} d_n(\tau) \quad (26) \end{aligned}$$

Summing (25) and (26) over $l \in \{1, \ldots, L\}$, $n \in \mathcal{N}$, taking conditional expectations, and noting that the second moment of a geometric random variable T_r with success probability ϕ is given by $(2-\phi)/\phi^2$ proves the result. □

B. The Frame-Based Drift-Plus-Penalty Algorithm

Let $V \geq 0$ be a non-negative parameter that we use to affect proximity to the optimal solution. Our dynamic algorithm initializes all virtual and actual queue states to 0, and designates $t_0 = 0$ as the start of the first renewal frame. Then:

- For each frame $r \in \{0, 1, 2, \ldots\}$, observe the vector of virtual and actual queues $\boldsymbol{\Theta}(t_r)$ and implement a policy over the course of the frame to minimize the following "drift-plus-penalty" expression:

$$\mathbb{E}\left[G(t_r) + V\sum_{\tau=t_r}^{t_r+T_r-1} y_0(\tau) \,|\, \boldsymbol{\Theta}(t_r)\right] \quad (27)$$

- During the course of the frame, update virtual and actual queues every slot by (1) and (20), and update state $z(t)$ by (4). At the end of the frame, go back to the preceding step.

The decision rule (27) generalizes the drift-plus-penalty rule in [2][28] to a variable frame system. The problem of designing a policy to minimize (27) over one frame is a *weighted stochastic shortest path problem*, where weights are virtual and actual queue backlogs at the start of the frame. Finding such a policy is non-trivial, and often can only be done approximately. The next sub-section analyzes the algorithm under the assumption that we have a procedure to approximate (27) every frame. Section IV considers various approximation methods.

C. Performance Theorem

For constants $C \geq 0$, $\delta \geq 0$, define a (C, δ)-*approximation* of (27) to be a policy for choosing $\alpha(t)$ over a frame (consisting of slots $\tau \in \{t_r, \ldots, t_r + T_r - 1\}$) that yields a drift-plus-penalty that is less than or equal to that of any other policy, plus an error term parameterized by C and δ:

$$\mathbb{E}\left[G(t_r) + V\sum_{\tau=t_r}^{t_r+T_r-1} y_0(\tau) \mid \mathbf{\Theta}(t_r)\right] \leq \mathbb{E}\left[G^*(t_r) + V\sum_{\tau=t_r}^{t_r+T_r-1} y_0^*(\tau) \mid \mathbf{\Theta}(t_r)\right] + C + \delta\sum_{n\in\mathcal{N}} Q_n(t_r) + \delta\sum_{l=1}^{L} X_l(t_r) + V\delta \quad (28)$$

where $G^*(t_r)$ and $y_0^*(\tau)$ represent (23) and (5), respectively, under any alternative algorithm $\alpha^*(t)$ that can be implemented during the slots $\tau \in \{t_r, \ldots, t_r + T_r - 1\}$ of the frame. Note that an exact minimization of the stochastic shortest path problem (27) is a (C, δ)-approximation for $C = \delta = 0$.

Theorem 1: Suppose Assumptions 1 and 2 hold for a given $\epsilon > 0$. Fix $V \geq 0$, $C \geq 0$, $\delta \geq 0$, and suppose we use a (C, δ)-approximation every frame. If $\epsilon > \phi\delta$, then all desired constraints (8)-(10) are satisfied. Further, for all positive integers R, the average queue sizes satisfy:

$$\frac{1}{R}\sum_{r=0}^{R-1}\left[\sum_{n\in\mathcal{N}}\mathbb{E}\left[Q_n(t_r)\right] + \sum_{l=1}^{L}\mathbb{E}\left[X_l(t_r)\right]\right] \leq \frac{B/\phi + C\phi + V(\phi\delta + 2\beta)}{\epsilon - \phi\delta} \quad (29)$$

Further, the time average penalty satisfies:

$$\limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\left[y_0(\tau)\right] \leq y_0^{opt} + \frac{B/\phi + C\phi}{V} + \phi\delta[1 + (\beta - y_0^{opt})/\epsilon] \quad (30)$$

Suppose our implementation of the stochastic shortest path problem every frame is accurate enough to ensure $\delta = 0$. Then from (30) and (29), the time average of $y_0(t)$ can be made arbitrarily close to (or below) y_0^{opt} as V is increased, with a tradeoff in average queue size that is linear in V. The dependence on the ϕ parameter is also apparent: While we desire ϕ to be small to minimize the disruptions due to forced renewals, a small value of ϕ implies a larger value of B/ϕ in (30) and (29). Note also that the average size of each stability-constrained queue affects its average delay, and the average size of each virtual queue affects the convergence time required for its constraint to be closely met.

D. Proof of Theorem 1

We first prove (29), and then (30).

Proof: (Theorem 1 part 1—Queue Bounds) Let t_r be the start of a renewal time. From (22) we have:

$$\begin{aligned}
&\mathbb{E}\left[\Delta(t_r) + V\sum_{\tau=t_r}^{t_r+T_r-1} y_0(\tau) \mid \mathbf{\Theta}(t_r)\right] \\
&\leq \frac{B}{\phi^2} + \mathbb{E}\left[G(t_r) + V\sum_{\tau=t_r}^{t_r+T_r-1} y_0(\tau) \mid \mathbf{\Theta}(t_r)\right] \\
&\leq \frac{B}{\phi^2} + C + \mathbb{E}\left[G^*(t_r) + V\sum_{\tau=t_r}^{t_r+T_r-1} y_0^*(\tau) \mid \mathbf{\Theta}(t_r)\right] \\
&\quad + \delta\sum_{n\in\mathcal{N}} Q_n(t_r) + \delta\sum_{l=1}^{L} X_l(t_r) + V\delta \quad (31)
\end{aligned}$$

where $G^*(t_r)$ and $y_l^*(\tau)$ are for any alternative policy $\alpha^*(t)$. Using the fact that $|y_0^*(\tau) - y_0(\tau)| \leq 2\beta$ for all τ, and $\mathbb{E}\left[T_r\right] = 1/\phi$, we have:

$$\begin{aligned}
\mathbb{E}\left[\Delta(t_r)|\mathbf{\Theta}(t_r)\right] &\leq \frac{B}{\phi^2} + C + \frac{2\beta V}{\phi} + \mathbb{E}\left[G^*(t_r) \mid \mathbf{\Theta}(t_r)\right] \\
&\quad + \delta\sum_{n\in\mathcal{N}} Q_n(t_r) + \delta\sum_{l=1}^{L} X_l(t_r) + V\delta \quad (32)
\end{aligned}$$

Now consider the (Ω, z)-only policy $\alpha_2^*(t)$ from Assumption 2 (equations (18)-(19)), which makes decisions independent of $\mathbf{\Theta}(t_r)$ to yield (using the definition of $G(t_r)$ in (23)):

$$\mathbb{E}\left[G^*(t_r)|\mathbf{\Theta}(t_r)\right] \leq \frac{-\epsilon}{\phi}\left[\sum_{n\in\mathcal{N}} Q_n(t_r) + \sum_{l=1}^{L} X_l(t_r)\right]$$

Substituting the above into the right-hand-side of (32) gives:

$$\begin{aligned}
\mathbb{E}\left[\Delta(t_r)|\mathbf{\Theta}(t_r)\right] &\leq B/\phi^2 + C + V(2\beta/\phi + \delta) \\
&\quad + (\delta - \epsilon/\phi)\left[\sum_{n\in\mathcal{N}} Q_n(t_r) + \sum_{l=1}^{L} X_l(t_r)\right] \quad (33)
\end{aligned}$$

Taking expectations of the above and using the definition of $\Delta(t_r)$ gives:

$$\begin{aligned}
\mathbb{E}\left[L(t_{r+1})\right] - \mathbb{E}\left[L(t_r)\right] &\leq B/\phi^2 + C + V(2\beta/\phi + \delta) \\
&\quad + (\delta - \epsilon/\phi)\left[\sum_{n\in\mathcal{N}}\mathbb{E}\left[Q_n(t_r)\right] + \sum_{l=1}^{L}\mathbb{E}\left[X_l(t_r)\right]\right]
\end{aligned}$$

Summing the above over $r \in \{0, \ldots, R-1\}$ (for some positive integer R), dividing by R, and using the fact that $\mathbb{E}\left[L(t_0)\right] = 0$ gives:

$$\begin{aligned}
\frac{\mathbb{E}\left[L(t_R)\right]}{R} &\leq B/\phi^2 + C + V(2\beta/\phi + \delta) \\
&\quad + \frac{(\delta - \epsilon/\phi)}{R}\sum_{r=0}^{R-1}\left[\sum_{n\in\mathcal{N}}\mathbb{E}\left[Q_n(t_r)\right] + \sum_{l=1}^{L}\mathbb{E}\left[X_l(t_r)\right]\right]
\end{aligned}$$

Rearranging terms and using $\mathbb{E}[L(t_R)] \geq 0$ and $\epsilon > \phi\delta$ proves (29). While (29) samples only at the start of renewal frames, it can easily be used to show all virtual and actual queues are *strongly stable* (see Appendix B), and hence all desired inequality constraints are met [27]. □

Proof: (Theorem 1 part 2 — Performance Bound) Define probability $\gamma \triangleq \delta\phi/\epsilon$. This is a valid probability because $\epsilon > \phi\delta$ by assumption. We consider a new policy $\alpha^*(t)$ implemented over the frame $\tau \in \{t_r, \ldots, t_r + T_r - 1\}$. The policy $\alpha^*(t)$ is a randomized mixture of the (Ω, z)-only policies from Assumptions 1 and 2: At the start of the frame, independently flip a biased coin with probabilities γ and $1-\gamma$, and carry out one of the two following policies for the full duration of the renewal interval:

- With probability γ: Use policy $\alpha_2^*(t)$ from Assumption 2 for the duration of the renewal frame, which yields (18)-(19).
- With probability $1-\gamma$: Use policy $\alpha_1^*(t)$ from Assumption 1 for the duration of the renewal frame, which yields (15)-(17).

Note that this policy $\alpha^*(t)$ is independent of $\boldsymbol{\Theta}(t_r)$. With $\alpha^*(t)$, from (15) we have:

$$\mathbb{E}\left[\sum_{\tau=t_r}^{t_r+T_r-1} y_0^*(\tau)|\boldsymbol{\Theta}(t_r)\right] \leq \frac{\gamma\beta + (1-\gamma)y_0^{opt}}{\phi} \quad (34)$$

We also have from (16)-(17) and (18)-(19):

$$\begin{aligned}\mathbb{E}\left[\sum_{\tau=t_r}^{t_r+T_r-1} y_l^*(\tau)|\boldsymbol{\Theta}(t_r)\right] &\leq \frac{-\gamma\epsilon}{\phi} = -\delta \ \forall l \in \{1,\ldots,L\} \\ \mathbb{E}\left[\sum_{\tau=t_r}^{t_r+T_r-1} d_n^*(\tau)|\boldsymbol{\Theta}(t_r)\right] &\leq \frac{-\gamma\epsilon}{\phi} = -\delta \ \forall n \in \mathcal{N} \quad (35)\end{aligned}$$

Plugging (34)-(35) into (31) yields:

$$\begin{aligned}\mathbb{E}\left[\Delta(t_r) + V\sum_{\tau=t_r}^{t_r+T_r-1} y_0(\tau) \mid \boldsymbol{\Theta}(t_r)\right] &\leq B/\phi^2 + C + V\delta \\ &\quad + \frac{V}{\phi}[\gamma\beta + (1-\gamma)y_0^{opt}]\end{aligned}$$

Taking expectations gives:

$$\begin{aligned}\mathbb{E}[L(t_{r+1})] - \mathbb{E}[L(t_r)] + V\mathbb{E}\left[\sum_{\tau=t_r}^{t_r+T_r-1} y_0(\tau)\right] \leq \\ B/\phi^2 + C + V\delta + \frac{V}{\phi}[\gamma\beta + (1-\gamma)y_0^{opt}]\end{aligned}$$

Summing over $r \in \{0, \ldots, R-1\}$ and dividing by VR/ϕ gives the following for all $R > 0$:

$$\frac{1}{R/\phi}\mathbb{E}\left[\sum_{\tau=0}^{t_R-1} y_0(\tau)\right] \leq [\gamma\beta + (1-\gamma)y_0^{opt} + \delta\phi] + \frac{B/\phi + C\phi}{V}$$

Using $\gamma = \delta\phi/\epsilon$ shows the right-hand-side of the above inequality is the same as the right-hand-side of the desired inequality (30). Finally, in Appendix B it is shown that:

$$\limsup_{R\to\infty} \frac{\mathbb{E}\left[\sum_{\tau=0}^{t_R-1} y_0(\tau)\right]}{R/\phi} \geq \limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1} \mathbb{E}[y_0(\tau)] \quad (36)$$

□

IV. Approximating the Stochastic Shortest Path Problem

Consider now the stochastic shortest path problem (27). Here we describe several approximation options and their performance. We note that the techniques and results in this section draw from standard contraction results used in dynamic programming with discounted rewards [29][6][30].

For simplicity, assume the state space $(\Omega(t), z(t))$ is finite, and the action space $\mathcal{A}_{\Omega(t),z(t)}$ is finite for all $(\Omega(t), z(t))$. Without loss of generality, assume we start at time 0 and have (possibly non-zero) backlogs $\boldsymbol{\Theta} = \boldsymbol{\Theta}(0)$. Let T be the renewal interval size. For every step $\tau \in \{0, \ldots, T-1\}$, define $c_{\boldsymbol{\Theta}}(\alpha(\tau), \Omega(\tau), z(\tau))$ as the incurred cost assuming that the queue state at the beginning of the renewal is $\boldsymbol{\Theta}(0)$:

$$\begin{aligned}c_{\boldsymbol{\Theta}}(\alpha(\tau), \Omega(\tau), z(\tau)) \triangleq &\sum_{n\in\mathcal{N}} Q_n(0)\hat{d}_n(\alpha(\tau), \Omega(\tau), z(\tau)) \\ &+ \sum_{l=1}^{L} X_l(0)\hat{y}_l(\alpha(\tau), \Omega(\tau), z(\tau)) \\ &+ V\hat{y}_0(\alpha(\tau), \Omega(\tau), z(\tau)) \quad (37)\end{aligned}$$

Let $\alpha^{ssp}(\tau)$ denote the optimal control action on slot τ for solving the stochastic shortest path problem, given that the controller first observes $\Omega(\tau)$ and $z(\tau)$. Define $\tilde{\mathcal{Z}} \triangleq \mathcal{Z} \cup \{renewal\}$, where we have added a new state "$renewal$" to represent the renewal state, which is the termination state of the stochastic shortest path problem. Appropriately adjust the transition probabilities $P_{ij}(\alpha, \Omega)$ to account for this new state [29][6][30]. Define $\boldsymbol{J} = (J_z)|_{z\in\tilde{\mathcal{Z}}}$ as a vector of optimal costs, where J_z is the minimum expected sum cost to the renewal state given that we start in state z, and $J_{renewal} = 0$. By basic dynamic programming theory [29][6], the optimal control action on each slot τ (given $\Omega(\tau)$ and $z(\tau)$) is:

$$\begin{aligned}\alpha(\tau) = \arg\min_{\alpha\in\mathcal{A}_{\Omega(\tau),z(\tau)}} [c_{\boldsymbol{\Theta}}(\alpha, \Omega(\tau), z(\tau)) + \\ \textstyle\sum_{y\in\tilde{\mathcal{Z}}} P_{z(\tau),y}(\alpha, \Omega(\tau))J_y] \quad (38)\end{aligned}$$

This policy is easily implemented provided that the J_z values are known. It is well known that the $\boldsymbol{J}$ vector satisfies the following vector dynamic programming equation:[2]

$$\boldsymbol{J} = \mathbb{E}\left[\min_{\alpha_z\in\mathcal{A}_{\Omega,z}} [\boldsymbol{c}_{\boldsymbol{\Theta}}(\alpha_z, \Omega) + P(\alpha_z, \Omega)\boldsymbol{J}]\right] \quad (39)$$

where we have used an *entry-wise min* (possibly with different α_z actions being used for minimizing each entry $z \in \tilde{\mathcal{Z}}$). Further, $\boldsymbol{c}_{\boldsymbol{\Theta}}(\alpha_z, \Omega)$ is defined as a vector with entries $c_{\boldsymbol{\Theta}}(\alpha_z, \Omega, z)$, and $P(\alpha_z, \Omega) = (P_{zy}(\alpha_z, \Omega))$ is the matrix of transition probabilities for Ω and control action α_z. The expectation in (39) is over the distribution of the i.i.d. process Ω. Because $\Omega(t)$ has the structure $\Omega(t) = [\omega(t), \phi(t)]$, where $\omega(t)$ is the random outcome for slot t and $\phi(t)$ is an independent Bernoulli process that has forced renewals with probability ϕ,

[2]One can also derive (39) by defining a value function $H(z, \Omega)$, writing the Bellman equation in terms of $H(z(t+1), \Omega(t+1))$, taking an expectation with respect to the i.i.d. $\Omega(t)$, $\Omega(t+1)$, and defining $J(z) \triangleq \mathbb{E}_{\Omega(t)}[H(z, \Omega(t))]$.

we can re-write the above vector equation as:

$$\boldsymbol{J} = \phi\mathbb{E}\left[\min_{\alpha_z\in\mathcal{A}_{[\omega,1],z}} \boldsymbol{c}^{(1)}_{\boldsymbol{\Theta}}(\alpha_z,\omega)\right] + (1-\phi)\mathbb{E}\left[\min_{\alpha_z\in\mathcal{A}_{[\omega,0],z}}\left[\boldsymbol{c}^{(0)}_{\boldsymbol{\Theta}}(\alpha_z,\omega) + P^{(0)}(\alpha_z,\omega)\boldsymbol{J}\right]\right] \quad (40)$$

where:

$$\begin{aligned}\boldsymbol{c}^{(1)}_{\boldsymbol{\Theta}}(\alpha_z,\omega) &\triangleq \boldsymbol{c}_{\boldsymbol{\Theta}}(\alpha_z,[\omega,1])\\ \boldsymbol{c}^{(0)}_{\boldsymbol{\Theta}}(\alpha_z,\omega) &\triangleq \boldsymbol{c}_{\boldsymbol{\Theta}}(\alpha_z,[\omega,0])\\ P^{(0)}(\alpha_z,\omega) &\triangleq P(\alpha_z,[\omega,0])\end{aligned}$$

We assume the transition probabilities $P^{(0)}(\alpha_z,\omega)$ are known (recall that these are indeed known binary values as described in the model of Section II-B). We next show how to compute an approximation of $\boldsymbol{J}$ based on random samples of $\omega(t)$ and using a classic Robbins-Monro iteration.

A. Estimation Through Random i.i.d. Samples

Suppose we have an infinite sequence of random variables arranged in batches with batch size W, with ω_{kw} denoting the wth sample of batch k. All random variables are i.i.d. with probability distribution the same as $\omega(t)$, and all are independent of the queue state $\boldsymbol{\Theta}$ that is used for this stochastic shortest path problem. Consider the following two mappings Ψ and $\tilde{\Psi}$ from a $\boldsymbol{J}$ vector to another $\boldsymbol{J}$ vector, where the second is implemented with respect to a particular batch k:

$$\Psi\boldsymbol{J} \triangleq \phi\mathbb{E}\left[\min_{\alpha_z\in\mathcal{A}_{[\omega,1],z}} \boldsymbol{c}^{(1)}_{\boldsymbol{\Theta}}(\alpha_z,\omega)\right] + (1-\phi)\mathbb{E}\left[\min_{\alpha_z\in\mathcal{A}_{[\omega,0],z}}\left[\boldsymbol{c}^{(0)}_{\boldsymbol{\Theta}}(\alpha_z,\omega) + P^{(0)}(\alpha_z,\omega)\boldsymbol{J}\right]\right] \quad (41)$$

$$\tilde{\Psi}\boldsymbol{J} \triangleq \phi\frac{1}{W}\sum_{w=1}^{W}\min_{\alpha_z\in\mathcal{A}_{[\omega_{kw},1],z}} \boldsymbol{c}^{(1)}_{\boldsymbol{\Theta}}(\alpha_z,\omega_{kw}) + (1-\phi)\frac{1}{W}\sum_{w=1}^{W}\min_{\alpha_z\in\mathcal{A}_{[\omega_{kw},0],z}}\Big[\boldsymbol{c}^{(0)}_{\boldsymbol{\Theta}}(\alpha_z,\omega_{kw}) + P^{(0)}(\alpha_z,\omega_{kw})\boldsymbol{J}\Big] \quad (42)$$

where the min is entrywise over each vector entry. The expectation in (41) is implicitly conditioned on a given $\boldsymbol{\Theta}$ vector, and is with respect to the random ω, which is independent of $\boldsymbol{\Theta}$. We note that both $\Psi\boldsymbol{J}$ and $\tilde{\Psi}\boldsymbol{J}$ are vectors with size determined by the size of the state space $\mathcal{Z}$. For a system with K delay-constrained queues, the size of $\mathcal{Z}$ is exponential in K. Thus, any computation of the map $\Psi\boldsymbol{J}$ or $\tilde{\Psi}\boldsymbol{J}$ must update a number of entries that is exponential in K. This is why we desire K to be small, even though the number of stability-constrained queues N can be large.

The mapping Ψ cannot be implemented without knowledge of the distribution of ω (so that the expectation can be computed), whereas the mapping $\tilde{\Psi}$ can be implemented as a "simulation" over the W random samples ω_{kw} (assuming such samples can be generated or obtained). However, the expected value of $\tilde{\Psi}\boldsymbol{J}$ is exactly equal to $\Psi\boldsymbol{J}$. Thus, given an initial vector $\boldsymbol{J}_k$ for use in step k, we can write $\tilde{\Psi}\boldsymbol{J}_k = \Psi\boldsymbol{J}_k + \boldsymbol{\eta}_k$, where $\boldsymbol{\eta}_k$ is a zero-mean vector random variable. Specifically, the vector $\boldsymbol{\eta}_k$ satisfies:

$$\mathbb{E}\left[\boldsymbol{\eta}_k \mid \boldsymbol{J}_k\right] = \boldsymbol{0}$$

Thus, while the vector $\boldsymbol{\eta}_k$ is *not* independent of $\boldsymbol{J}_k$, each entry is *uncorrelated* with any deterministic function of $\boldsymbol{J}_k$. That is, for each entry i and any deterministic (and measurable) function $f(\cdot)$ we have via iterated expectations:

$$\mathbb{E}\left[\eta_k[i]f(\boldsymbol{J}_k)\right] = \mathbb{E}\left[f(\boldsymbol{J}_k)\mathbb{E}\left[\eta_k[i] \mid \boldsymbol{J}_k\right]\right] = 0 \quad (43)$$

For $k \in \{0,1,2,\ldots\}$ we have the iteration:

$$\boldsymbol{J}_{k+1} = \frac{1}{k+1}\tilde{\Psi}\boldsymbol{J}_k + \frac{k}{k+1}\boldsymbol{J}_k \quad (44)$$

This iteration is a classic *Robbins-Monro* stochastic approximation algorithm. It can be shown that the $\boldsymbol{J}$ vector remains deterministically bounded for all k [31], and that Ψ and $\tilde{\Psi}$ satisfy the requirements of Proposition 4.6 in Section 4.3.4 of [6]. Thus the above iteration is in the standard form for stochastic approximation theory, and ensures that:

$$\lim_{k\to\infty}\boldsymbol{J}_k = \boldsymbol{J}^* \quad \text{with prob. 1}$$

where $\boldsymbol{J}^*$ is the cost vector associated with the optimal stochastic shortest path problem, that is, it is the solution to (40) and thus satisfies $\boldsymbol{J}^* = \Psi\boldsymbol{J}^*$. This holds for any batch size W (including the simplest case $W=1$), although taking larger batches reduces the variance of the per-batch estimation and may improve overall convergence speed.

B. Recursive Methods for Ψ

Contraction results for general stochastic shortest path problems are given in [6]. The following is a related result with a simpler form that holds because of our forced renewal structure. For a given vector $\boldsymbol{X}$, define $||\boldsymbol{X}||$ as the maximum absolute value of $\boldsymbol{X}$:

$$||\boldsymbol{X}|| \triangleq \max_i |X_i|$$

It is not difficult to show that for any vector $\boldsymbol{X}$ and any probability matrix P with rows that sum to 1, and with a number of columns equal to the size of $\boldsymbol{X}$, we have $||P\boldsymbol{X}|| \leq ||\boldsymbol{X}||$.

Lemma 2: For any vectors $\boldsymbol{X}$, $\boldsymbol{Y}$ of the same size as $\boldsymbol{J}^*$, we have:

$$||\Psi\boldsymbol{X} - \Psi\boldsymbol{Y}|| \leq (1-\phi)||\boldsymbol{X} - \boldsymbol{Y}||$$

Proof: Omitted (see [31] and related results in [6]). □

This simple result yields the following approximation bounds for k iterations of the map Ψ: Define $\boldsymbol{J}_0$ as any initial guess of $\boldsymbol{J}^*$, and for $k \in \{1,2,3,\ldots\}$ define $\boldsymbol{J}_k = \Psi\boldsymbol{J}_{k-1}$. Because $\Psi\boldsymbol{J}^* = \boldsymbol{J}^*$, we have:

$$\begin{aligned}||\boldsymbol{J}_k - \boldsymbol{J}^*|| &= ||\Psi\boldsymbol{J}_{k-1} - \Psi\boldsymbol{J}^*||\\ &\leq (1-\phi)||\boldsymbol{J}_{k-1} - \boldsymbol{J}^*||\end{aligned}$$

By recursion, it easily follows that for all $k \in \{0,1,2,\ldots\}$ we have:

$$||\boldsymbol{J}_k - \boldsymbol{J}^*|| \leq (1-\phi)^k||\boldsymbol{J}_0 - \boldsymbol{J}^*|| \quad (45)$$

Because the renewal frame size is independent of the policy, and has average $1/\phi$, it is not difficult to show that $\boldsymbol{J}^* \leq c_{max}/\phi$, where c_{max} is the largest possible magnitude of of $c_{\boldsymbol{\Theta}}(\alpha(\tau), \Omega(\tau), z(\tau))$ for slot τ in the frame (such a constant exists and is finite because of the boundedness assumptions). Therefore, defining $\boldsymbol{J}_0 = \boldsymbol{0}$ and using (45) yields:

$$||\boldsymbol{J}_k - \boldsymbol{J}^*|| \leq (1-\phi)^k c_{max}/\phi$$

By the definition of $c_{\boldsymbol{\Theta}}(\cdot)$ in (37), it can be shown that c_{max} is a sum of terms that are proportional to V, $Q_n(t_r)$, and $Z_l(t_r)$. Further, in [31] it is shown that the deviation in the optimal cost when (38) is used with an approximate value $\boldsymbol{J}_k$, rather than $\boldsymbol{J}^*$, deviates from $\boldsymbol{J}^*$ by at most:

$$\frac{2(1-\phi)||\boldsymbol{J}_k - \boldsymbol{J}^*||}{\phi}$$

Hence, the above two bounds can be used to compute a value k that provides explicit approximation values for C and δ for use in Theorem 1.

C. Recursive Methods for $\tilde{\Psi}$

The difficulty in iterating the map $\Psi\boldsymbol{J}$ is that it requires full knowledge of the underlying probability distributions to compute the associated expectations. An approximation of this is to use $\tilde{\Psi}$ from (42). Specifically, assume we have W i.i.d. samples $\omega_1, \ldots, \omega_W$. Then the $\tilde{\Psi}$ function is:

$$\tilde{\Psi}\boldsymbol{J} \triangleq \phi \frac{1}{W}\sum_{w=1}^{W} \min_{\alpha_z \in \mathcal{A}_{[\omega_w,1],z}} \boldsymbol{c}^{(1)}_{\boldsymbol{\Theta}}(\alpha_z, \omega_w) +$$
$$(1-\phi)\frac{1}{W}\sum_{w=1}^{W} \min_{\alpha_z \in \mathcal{A}_{[\omega_w,0],z}} \left[\boldsymbol{c}^{(0)}_{\boldsymbol{\Theta}}(\alpha_z, \omega_w) + P^{(0)}(\alpha_z, \omega_w)\boldsymbol{J}\right]$$

Define $\tilde{\boldsymbol{J}}_0$ as any initial vector, and for $k \in \{1, 2, 3, \ldots\}$ define $\tilde{\boldsymbol{J}}_k = \tilde{\Psi}\boldsymbol{J}_{k-1}$. Using the same proof technique as Lemma 2 and equation (45) it is easy to show that for any $W > 0$, $\tilde{\Psi}$ is also a contraction that satisfies for any $\boldsymbol{X}$ and $\boldsymbol{Y}$:

$$||\tilde{\Psi}\boldsymbol{X} - \tilde{\Psi}\boldsymbol{Y}|| \leq (1-\phi)||\boldsymbol{X} - \boldsymbol{Y}||$$

Thus, it has a unique fixed point $\tilde{\boldsymbol{J}}^*$ satisfying $\tilde{\Psi}\tilde{\boldsymbol{J}}^* = \tilde{\boldsymbol{J}}^*$, and for all $k \in \{1, 2, 3, \ldots\}$ we have:

$$||\tilde{\boldsymbol{J}}_k - \tilde{\boldsymbol{J}}^*|| \leq (1-\phi)^k ||\tilde{\boldsymbol{J}}_0 - \tilde{\boldsymbol{J}}^*||$$

The value $\tilde{\boldsymbol{J}}^*$ is typically not the same as $\boldsymbol{J}^*$. It represents the optimal cost vector in a modified system where the ω vector is i.i.d. with the same distribution as the empirical average given over the W samples. Intuitively, $\tilde{\boldsymbol{J}}^*$ becomes a better approximation for $\boldsymbol{J}^*$ when the number of samples W is large. This is because the iteration for $\tilde{\Psi}\boldsymbol{J}$ uses a summation of bounded i.i.d. random variables to approximate an expectation, and the error of such an approximation goes to zero as the number of samples W is increased. Formal convergence as $W \to \infty$ can be derived using continuity and contraction properties of the Bellman iteration (see related results in [6]).

D. Sampling From the Past and Delayed Queue Analysis

It remains to be seen how one can obtain the required i.i.d. samples without knowing the probability distribution for ω. In this subsection, we describe a technique that uses previous samples of the $\omega(\tau)$ values.

We first obtain a collection of W i.i.d. samples of $\omega(t)$. Consider a given renewal time t_r, and suppose that the time t_r is large enough so that we can obtain W samples according to the following procedure: Let $\omega_1 \triangleq \omega(t_r)$, $\omega_2 \triangleq \omega(t_r - 1)$, $\omega_3 \triangleq \omega(t_r - 2), \ldots, \omega_W \triangleq \omega(t_r - W + 1)$. Because $\omega(t)$ is i.i.d. over slots (and because our renewal times are chosen randomly and independently), it is easy to see that $\{\omega_1, \ldots, \omega_W\}$ form an i.i.d. sequence.

A subtlety now arises: Even though the $\{\omega_1, \ldots, \omega_W\}$ sequence is i.i.d., these samples are *not* independent of the queue backlog $\boldsymbol{\Theta}(t_r)$ at the beginning of the renewal. This is because these values have influenced the queue states. This makes it challenging to directly analyze a Robbins-Monro iteration. Indeed, the expectation in (41) can be viewed as a conditional expectation given a certain queue backlog at the beginning of the renewal interval, which is $\boldsymbol{\Theta}(t_r)$ for the rth renewal. This conditioning does not affect (41) when $\omega(t)$ is chosen independently of initial queue backlog, and so the random samples in (42) are also assumed to be chosen independent of the initial queue backlog, which is not the case if we sample from the past.

To avoid this difficulty and ensure the samples are both i.i.d. and independent of the queue states that form the weights in our stochastic shortest path problem, we use a *delayed queue analysis* as in the related queueing problem [32] (see also related work on using delayed samples for Robbins-Monro iterations in [33][34]). Let t_{start} denote the slot on which sample ω_W is taken, and let $\boldsymbol{\Theta}(t_{start})$ represent the queue backlogs at that time. It follows that the i.i.d. samples are also independent of $\boldsymbol{\Theta}(t_{start})$. Hence, the bounds derived for the iteration technique in the previous section can be applied when the iterates use $\boldsymbol{\Theta}(t_{start})$ as the backlog vector. Let $\boldsymbol{J}_{\boldsymbol{\Theta}(t_r)}$ denote the optimal solution to the problem (39) for a queue backlog $\boldsymbol{\Theta}(t_r)$ at the beginning of our renewal time t_r, and let $\boldsymbol{J}_{\boldsymbol{\Theta}(t_{start})}$ denote the corresponding optimal solution for a problem that starts with initial queue backlog $\boldsymbol{\Theta}(t_{start})$. Then there are $W-1$ slots in between t_{start} and t_r. Because the maximum change in any queue on one slot is bounded by β, we want to claim that an algorithm which computes the stochastic shortest path using the $\boldsymbol{\Theta}(t_{start})$ queue values gives a result that is within an additive constant of the algorithm which uses $\boldsymbol{\Theta}(t_r)$. Such an additive constant can be viewed as the C constant in Theorem 1. This can be justified using the next lemma, which bounds the deviation of the optimal costs associated with two general queue backlog vectors.

Let $\boldsymbol{\Theta}_1$ and $\boldsymbol{\Theta}_2$ be two different queue backlog vectors, and let $\boldsymbol{J}_{\boldsymbol{\Theta}_1}$ and $\boldsymbol{J}_{\boldsymbol{\Theta}_2}$ represent the optimal frame costs corresponding to $\boldsymbol{\Theta}_1$ and $\boldsymbol{\Theta}_2$, respectively. Define the constant γ as follows:

$$\gamma \triangleq \sup_{\alpha_z, \Omega} ||\boldsymbol{c}_{\boldsymbol{\Theta}_1}(\alpha_z, \Omega) - \boldsymbol{c}_{\boldsymbol{\Theta}_2}(\alpha_z, \Omega)|| \quad (46)$$

where $\boldsymbol{c}_{\boldsymbol{\Theta}}(\alpha_z, \Omega)$ is the vector, indexed by z, with the zth

entry given by (37) using backlog vector $\boldsymbol{\Theta}$. Note from (37) that γ is independent of V (as the V term in (37) cancels out in the subtraction), and is proportional to the maximum penalty value times the maximum *difference* in any queue backlog entry in $\boldsymbol{\Theta}_1$ and its corresponding entry in $\boldsymbol{\Theta}_2$. Thus γ is also independent of the actual size of the backlog vectors, and depends only on their *difference*, being bounded by a constant proportional to $W\beta$.

Lemma 3: For the vectors $\boldsymbol{\Theta}_1$ and $\boldsymbol{\Theta}_2$, and for the γ value defined in (46), we have:

(a) The difference between $\boldsymbol{J}_{\boldsymbol{\Theta}_1}$ and $\boldsymbol{J}_{\boldsymbol{\Theta}_2}$ satisfies:

$$||\boldsymbol{J}_{\boldsymbol{\Theta}_1} - \boldsymbol{J}_{\boldsymbol{\Theta}_2}|| \leq \frac{\gamma}{\phi}$$

(b) Let $\alpha_1(t)$ denote the policy decisions at time t under the policy that makes optimal decisions subject to queue backlogs $\boldsymbol{\Theta}_1$, and define $\boldsymbol{J}_{21}^{mis}$ as the expected sum cost over a frame of a *mismatched policy* that incurs costs according to backlog vector $\boldsymbol{\Theta}_2$ but makes decisions according to $\alpha_1(t)$ (and hence has the same decisions as the optimal policy for $\boldsymbol{\Theta}_1$). Then:

$$\boldsymbol{J}_{\boldsymbol{\Theta}_2} \leq \boldsymbol{J}_{21}^{mis} \leq \boldsymbol{J}_{\boldsymbol{\Theta}_1} + \mathbf{1}\frac{\gamma}{\phi}$$

where $\mathbf{1}$ is a vector of all 1 values with the same dimension as $\boldsymbol{J}_{\boldsymbol{\Theta}_1}$.

Proof: Omitted for brevity (see [31]). □

V. Simulation

In this section, we simulate the frame-based drift-plus-penalty algorithm in Section III-B for the simple network in Fig. 1. The algorithm utilizes the classic Robbins-Monro iteration, based on samples from the past, to approximate the weighted stochastic shortest path problem (40). This is because solving (40) exactly is computationally expensive, would require full probability knowledge, and may not be practical for implementation.

The network in Fig. 1 consists of one delay-constrained queue and ten stability-constrained queues, so that $\mathcal{K} = \{1\}$ and $\mathcal{N} = \{2, 3, \ldots, 11\}$. The size of the delay-constrained queue is limited to $b = 10$ packets. Random packet arrivals are i.i.d. Bernoulli processes with $Pr[A_n(t) = 1] = 0.06$ for $n \in \mathcal{N}$ and $Pr[A_1(t) = 1] = 0.4$. Each network channel has a binary state and is active (ON-state) with probability $Pr[S_i(t) = 1] = 0.18775$ for $i \in \mathcal{N}$ and $Pr[S_1(t) = 1] = 0.5$. The forced renewal probability is $Pr[\phi(t) = 1] = 0.01$.

In this simulation, we consider a problem of minimizing the average number of dropped packets. For the delay-constrained

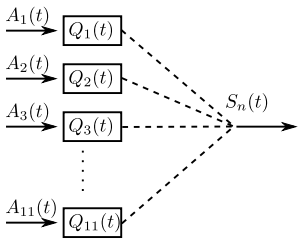

Fig. 1. A network with 1 delay-constrained queue (queue 1), and 10 stability-constrained queues.

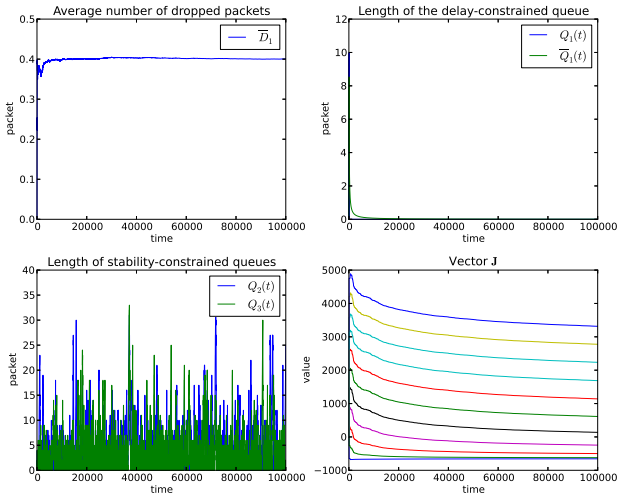

Fig. 2. Behavior in the system with $V = 0$.

queue $Q_1(t)$, the average backlog is limited to 1.2. Define $y_0(t) = D_1(t)$ and $y_1(t) = Q_1(t) - 1.2$. Then an optimization for this simulation is

$$\begin{aligned} \text{Minimize} \quad & \overline{y}_0 \\ \text{Subject to} \quad & \overline{y}_1 \leq 0 \\ & \bar{Q}_n < \infty \text{ for all } n \in \{2, 3, \ldots, 11\}. \end{aligned}$$

The simulation follows the frame-based drift-plus-penalty algorithm in Section III-B with the Robbins-Monro iteration (44). A batch size is set to be $W = 50$, so that we store the most recent 50 samples (using less than 50 in the initial slots if $\tau < 50$). Note that the number of samples is half of the average frame size, $1/\phi = 100$. Every forced renewal slot t_r, the algorithm uses the batch to approximate the mapping $\tilde{\Psi}\boldsymbol{J}$ in (42), and then updates $\boldsymbol{J}$ according to (44). After updating $\boldsymbol{J}$, every decision in frame r is decided from the simple rule (38). Then all delay-constrained, stability-constrained, and virtual queues are updated as in (1), (2), and (20).

For a simple initial comparison, we use $V = 0$, so the algorithm puts no weight on minimizing $\overline{y}_0$ and only attempts to satisfy the desired constraints. Results from the algorithm until 1×10^5 slots are shown in Fig. 2. The system drops almost all packets in the delay-constrained queue (as expected), making its average queue size approach zero, as shown in the top graphs of Fig. 2. All stability-constrained queues are stable and have similar behavior, which is shown in the bottom-left of Fig. 2. The bottom-right of Fig. 2 shows the convergence of $\boldsymbol{J}$. This illustrates that the algorithm yields a feasible solution.

We next use $V = 100$, so the algorithm attempts to minimize dropping in queue 1. Behavior in the system for the first 2×10^5 slots are shown in Fig. 3. The figure shows the convergence of the algorithm. After 2×10^6 slots, the average rate of dropped packets is 0.080 packets/slot and the average backlog of the delay-constrained queue is 0.975. These values correspond to the data points plotted for $V = 100$ in Figs. 4 and 5. Compared to the result from $V = 0$, the average number of dropped packets decreases, while the backlog increases as a result of more aggressive admission. In addition, the algorithm

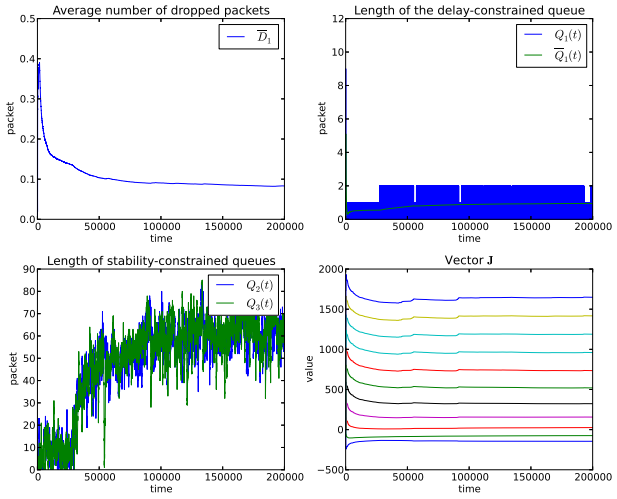

Fig. 3. Behaviors of the system with $V = 100$

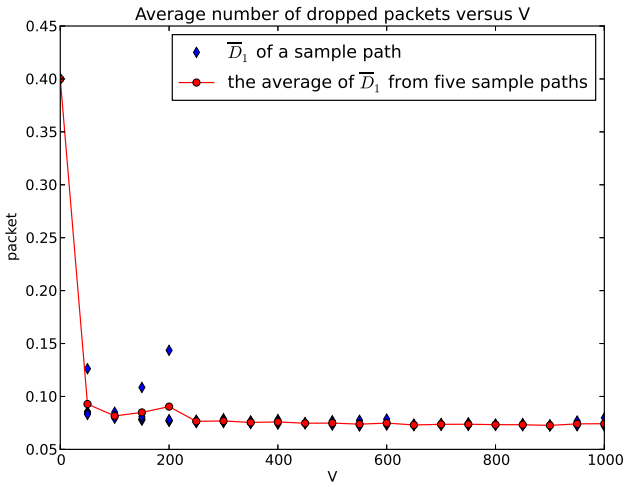

Fig. 4. Average number of dropped packets versus V

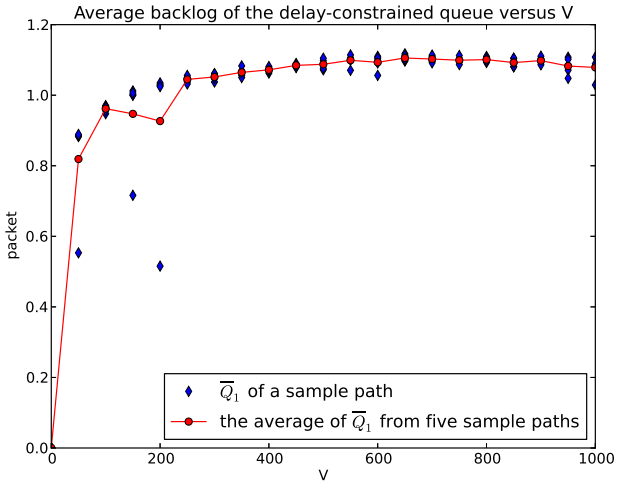

Fig. 5. Average backlog of the delay-constrained queue versus V

with $V = 100$ takes more time slots to converge.

Finally, the system is simulated for V in the range from 0 to 1000, as shown in Figs. 4 and 5. Each value of V is simulated over 5 independent runs. As V is increased, we expect the average drop rate to approach to optimality, with a corresponding increase in average queue sizes for the both delay-constrained and stability-constrained queues. This is exactly what happens. After 2×10^6 slots, the average number of dropped packets and the average number of backlogs are recorded. Then the average of the five values for each V is calculated. Additional simulation with $V = 10^4$ shows that the average rate of dropped packets is 0.073 packets/slot, and the average backlog is $\overline{Q}_1 = 1.15$ which is closer to 1.2 than the case with $V = 1000$.

For intuition about how these simulation results compare to the analytical optimum for this problem, note that the sum input rate (minus dropped packets) is $(\lambda_1 - \lambda_{drop}) + 10\lambda_2 = 1 - \lambda_{drop}$ packets/slot. This must be less than or equal to the maximum possible system output rate, being the probability that at least one of the 11 channels is ON: $1 - (1/2)(1 - .18775)^{10}$ packets/slot. It follows that any stabilizing strategy must satisfy $\lambda_{drop} \geq 0.0625$ packets/slot. However, one cannot achieve this value exactly because that would make the average backlog in queue 1 greater than the constraint 1.2. Further note that the forced renewal structure creates an optimality gap by an amount no more than the drops due to forced renewals (recall Appendix A). In our simulations, the rate of these drops is roughly $(1/100)\overline{Q}_1 \approx 0.0115$ packets/slot. This error bound is consistent with our simulated total drop rate of 0.073 (note that $0.0625 + 0.0115 = 0.0740$). Finally, note that since our algorithm optimizes decisions subject to assumed forced renewal events of probability $1/100$, it has an incentive to keep average queue size slightly below the 1.2 constraint to reduce the drops due to random forced renewals. We expect that an actual system optimality (without the forced renewal structure) would match the constraint $\overline{Q}_1 = 1.2$ exactly.

VI. Conclusions

We have developed an approach to the Markov Decision problems associated with a small number K of delay-constrained wireless users and a (possibly large) number N of stability-constrained queues. Our formulation allows optimization of general penalty functions subject to general penalty constraints, such as minimizing average packet drops subject to average backlog and/or average delay constraints at the delay-constrained queues, and subject to stability at the stability-constrained queues. Our approach uses a reduction to an online (unconstrained) weighted stochastic shortest path problem implemented over variable length frames. This generalizes the class of max-weight network control policies to networks with Markov decisions. The solution to the underlying stochastic shortest path problem has complexity that is exponential in the number of delay-constrained queues K, but polynomial in the number of delay-unconstrained queues N. A Robbins-Monro approximation technique was used to develop

several approximation algorithms for the stochastic shortest path problem. The solution technique is general and extends to other network problems with stochastic decisions.

APPENDIX A — BOUNDING THE INEFFICIENCY OF FORCED RENEWALS

Consider the problem of minimizing the average drop rate subject to delay constraints in the delay-constrained queues and stability in the stability-constrained queues. First consider the case without forced renewals. Assume the problem is feasible, and define $drop^{opt}$ as the infimum drop rate subject to the desired constraints. For simplicity, assume this infimum is achieved by a particular policy (else, we can consider a sequence of policies that approach $drop^{opt}$ arbitrarily closely). Let $[\boldsymbol{A}(t), \boldsymbol{S}(t)]$ be a particular sample path of arrivals and channels over $t \in \{0, 1, 2, \ldots\}$, and let $[\boldsymbol{\mu}^{opt}(t), \boldsymbol{D}^{opt}(t)]$ be the decisions made by the optimal policy in response to this sample path, where $\boldsymbol{D}^{opt}(t) = (D_k^{opt}(t))_{k\in\mathcal{K}}$. Thus, these decisions satisfy all required constraints, and:

$$\limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1}\sum_{k\in\mathcal{K}} \mathbb{E}\left[D_k^{opt}(\tau)\right] = drop^{opt} \quad (47)$$

Let $\boldsymbol{Q}^{opt}(t) = (Q_1^{opt}(t), \ldots, Q_{K+N}^{opt}(t))$ be the queue backlogs on slot t under this policy.

Now consider the same system with the same arrivals and channels $[\boldsymbol{A}(t), \boldsymbol{S}(t)]$, but introduce an independent forced renewal process $\phi(t)$, where forced renewals occur i.i.d. with probability ϕ. Consider a new policy $[\tilde{\boldsymbol{\mu}}(t), \tilde{\boldsymbol{D}}(t)]$ that acts on this sample path. Let $\tilde{\boldsymbol{Q}}(t)$ be the resulting queue backlog, with initial condition $\tilde{\boldsymbol{Q}}(0) \triangleq \boldsymbol{Q}^{opt}(0)$. Define $\tilde{L}_k(t)$ as the residual packets (plus arrivals) in queue $k \in \mathcal{K}$ after transmission on slot t:

$$\tilde{L}_k(t) \triangleq A_k(t) + \tilde{Q}_k(t) - \tilde{\mu}_k(t)$$

The policy $[\tilde{\boldsymbol{\mu}}(t), \tilde{\boldsymbol{D}}(t)]$ is defined as follows. For each slot $t \in \{0, 1, 2, \ldots\}$ we have for all $i \in \{1, \ldots, K+N\}$:

$$\tilde{\mu}_i(t) = \min[\mu_i^{opt}(t), \tilde{Q}_i(t)]$$

Further, for $k \in \mathcal{K}$ we have:

$$\tilde{D}_k(t) = \begin{cases} \min[D_k^{opt}(t), \tilde{L}_k(t)] & \text{if } \phi(t) = 0 \\ \tilde{L}_k(t) & \text{if } \phi(t) = 1 \end{cases}$$

Thus, the new policy mimics the decisions of the original policy, with the exception that it only transmits and drops packets it actually has. It is not difficult to see that this new policy satisfies $\tilde{Q}_i(t) \leq Q_i^{opt}(t)$ for all t and all $i \in \{1, \ldots, K+N\}$. Thus, the finite buffer size b is never violated in any queue. Further, because service is FIFO within a queue, all non-dropped packets have delay less than or equal to their delay in the original policy. Thus, the new policy satisfies all desired stability and delay constraints. Further, on each slot t we can write:

$$\tilde{D}_k(t) = \tilde{D}_k^A(t) + \tilde{D}_k^B(t)$$

where $\tilde{D}_k^A(t) \triangleq \min[D_k^{opt}(t), \tilde{L}_k(t)]$, and $\tilde{D}_k^B(t)$ are the additional packets (if any) that are dropped. It is clear that all packet drops in $\tilde{D}_k^B(t)$ are due to forced renewals. Further, by definition of $\tilde{D}_k^A(t)$ we have $\tilde{D}_k^A(t) \leq D_k^{opt}(t)$, and so:

$$\tilde{D}_k(t) \leq D_k^{opt}(t) + \tilde{D}_k^B(t)$$

Thus:

$$\begin{aligned} &\limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1}\sum_{k\in\mathcal{K}} \mathbb{E}\left[\tilde{D}_k(\tau)\right] \\ &\leq \limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1}\sum_{k\in\mathcal{K}} \mathbb{E}\left[D_k^{opt}(\tau)\right] \\ &\quad + \limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1}\sum_{k\in\mathcal{K}} \mathbb{E}\left[\tilde{D}_k^B(\tau)\right] \\ &\leq drop^{opt} + (Kb + \textstyle\sum_{k\in\mathcal{K}} \lambda_k)\phi \end{aligned}$$

where $\lambda_k \triangleq \mathbb{E}[A_k(t)]$, and we have used (47) together with the fact that the rate of drops due to forced renewals is at most $(Kb + \sum_{k=1}^{K} \lambda_k)\phi$. Thus, there exists a policy on the system with forced renewals that has a drop rate within $O(\phi)$ of $drop^{opt}$. It follows that the *optimal* policy on the system with forced renewals is also within $O(\phi)$ of $drop^{opt}$.

APPENDIX B — TIME AVERAGES

This appendix provides details for the proof of Theorem 1. Recall that t_r is the start time of the rth renewal frame, and T_r is the size of the frame, for $r \in \{0, 1, 2, \ldots\}$. The random variables T_r are i.i.d. and geometrically distributed with mean $1/\phi$ and second moment $(2-\phi)/\phi^2$. The queue sizes on slot t_r are independent of T_r. Recall that (29) implies there is a finite constant $D > 0$ such that for all $R > 0$:

$$\frac{1}{R}\sum_{r=0}^{R-1}\left[\sum_{n\in\mathcal{N}} \mathbb{E}[Q_n(t_r)] + \sum_{l=1}^{L} \mathbb{E}[X_l(t_r)]\right] \leq D \quad (48)$$

Claim 1: If (48) holds, then:

$$\limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1}\left[\sum_{n\in\mathcal{N}} \mathbb{E}[Q_n(\tau)] + \sum_{l=1}^{L} \mathbb{E}[X_l(\tau)]\right] < \infty$$

and so all queues $Q_n(t)$ and $X_l(t)$ are strongly stable.

Proof: (Claim 1) Define $H(t)$ as follows:

$$H(t) \triangleq \sum_{n\in\mathcal{N}} Q_n(t) + \sum_{l=1}^{L} X_l(t)$$

The sum of the (non-negative) $H(\tau)$ values over $\tau \in \{0, \ldots, R-1\}$ is less than or equal to the sum over the first R frames (because each frame is at least one slot), and so:

$$\begin{aligned} \sum_{\tau=0}^{R-1} H(\tau) &\leq \sum_{r=0}^{R-1}\sum_{\tau=t_r}^{t_r+T_r-1} H(\tau) \\ &\leq \sum_{r=0}^{R-1} T_r[H(t_r) + T_r\gamma] \end{aligned}$$

where $\gamma = (N+L)\beta$ is the maximum increase in $H(t)$ during one slot (recall (6)). Taking expectations and using $\mathbb{E}[H(t_r)T_r] = (1/\phi)\mathbb{E}[H(t_r)]$ yields:

$$\sum_{\tau=0}^{R-1} \mathbb{E}[H(\tau)] \leq \gamma R\mathbb{E}\left[T_0^2\right] + (1/\phi)\sum_{r=0}^{R-1} \mathbb{E}[H(t_r)]$$

Dividing by R and using (48) yields for all $R > 0$:

$$\frac{1}{R}\sum_{\tau=0}^{R-1} \mathbb{E}[H(\tau)] \leq \gamma\mathbb{E}\left[T_0^2\right] + D/\phi$$

Taking a limit as $R \to \infty$ proves the result. □

Claim 2: If $|y_0(\tau)| \leq \beta$ for all τ, then:

$$\limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1} \mathbb{E}\left[y_0(\tau)\right] \leq \limsup_{R\to\infty} \frac{\mathbb{E}\left[\sum_{\tau=0}^{t_R-1} y_0(\tau)\right]}{R/\phi}$$

Proof: (Claim 2) It suffices to assume $y_0(t) \geq 0$ for all t (else, just define $\tilde{y}_0(t) = y_0(t) + \beta$). Fix $\epsilon > 0$. Define $R(t) \triangleq \lceil (\phi+\epsilon)t \rceil$. For each integer $t > 0$, define $\psi(t) = 1$ if:

$$\sum_{r=0}^{R(t)-1} T_r < t$$

and define $\psi(t) = 0$ otherwise. Note that the average of the T_r values converges to $1/\phi$ with probability 1, and so $\lim_{t\to\infty}\mathbb{E}\left[\psi(t)\right] = 0$. Thus, because $y_0(t) \geq 0$ for all t:

$$\frac{1}{t}\sum_{\tau=0}^{t-1} y_0(\tau) \leq \frac{1}{t}\sum_{\tau=0}^{t_{R(t)}-1} y_0(\tau) + \beta\psi(t)$$

where the first term in the right-hand-side is an upper bound if $\psi(t) = 0$ (because $R(t)$ frames contains at least t slots if $\psi(t) = 0$), while the second is an upper bound otherwise. Taking expectations yields:

$$\begin{aligned}\frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\left[y_0(\tau)\right] &\leq \frac{R(t)}{t}\frac{1}{R(t)}\mathbb{E}\left[\sum_{\tau=0}^{t_{R(t)}-1} y_0(\tau)\right] \\ &\quad +\beta\mathbb{E}\left[\psi(t)\right] \\ &\leq (\frac{1}{t}+\phi+\epsilon)\frac{1}{R(t)}\mathbb{E}\left[\sum_{\tau=0}^{t_{R(t)}-1} y_0(\tau)\right] \\ &\quad +\beta\mathbb{E}\left[\psi(t)\right]\end{aligned}$$

Taking limits gives:

$$\limsup_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\left[y_0(\tau)\right] \leq (\phi+\epsilon)\limsup_{R\to\infty}\frac{1}{R}\mathbb{E}\left[\sum_{\tau=0}^{t_R-1} y_0(\tau)\right]$$

The above holds for all $\epsilon > 0$. Taking a limit as $\epsilon \to 0$ yields the result. □

REFERENCES

[1] M. J. Neely. Stochastic optimization for Markov modulated networks with application to delay constrained wireless scheduling. *Proc. IEEE Conf. on Decision and Control (CDC)*, Shanghai, China, pp. 4826-4833, Dec. 2009.

[2] L. Georgiadis, M. J. Neely, and L. Tassiulas. Resource allocation and cross-layer control in wireless networks. *Foundations and Trends in Networking*, vol. 1, no. 1, pp. 1-149, 2006.

[3] S. Ross. *Introduction to Probability Models*. Academic Press, 8th edition, Dec. 2002.

[4] E. Altman. *Constrained Markov Decision Processes*. Boca Raton, FL, Chapman and Hall/CRC Press, 1999.

[5] S. Meyn. *Control Techniques for Complex Networks*. Cambridge University Press, 2008.

[6] D. P. Bertsekas and J. N. Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, Belmont, Mass, 1996.

[7] J. Abounadi, D. Bertsekas, and V. S. Borkar. Learning algorithms for Markov decision processes with average cost. *SIAM Journal on Control and Optimization*, vol. 20, pp. 681-698, 2001.

[8] L. Tassiulas and A. Ephremides. Dynamic server allocation to parallel queues with randomly varying connectivity. *IEEE Transactions on Information Theory*, vol. 39, no. 2, pp. 466-478, March 1993.

[9] E. M. Yeh. *Multiaccess and Fading in Communication Networks*. PhD thesis, Massachusetts Institute of Technology, Laboratory for Information and Decision Systems (LIDS), 2001.

[10] A. Ganti, E. Modiano, and J. N. Tsitsiklis. Optimal transmission scheduling in symmetric communication models with intermittent connectivity. *IEEE Transactions on Information Theory*, vol. 53, no. 3, pp. 998-1008, March 2007.

[11] A. Fu, E. Modiano, and J. Tsitsiklis. Optimal energy allocation for delay-constrained data transmission over a time-varying channel. *Proc. IEEE INFOCOM*, 2003.

[12] M. Goyal, A. Kumar, and V. Sharma. Power constrained and delay optimal policies for scheduling transmission over a fading channel. *Proc. IEEE INFOCOM*, April 2003.

[13] N. Salodkar, A. Bhorkar, A. Karandikar, and V. S. Borkar. An on-line learning algorithm for energy efficient delay constrained scheduling over a fading channel. *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 4, pp. 732-742, May 2008.

[14] D. V. Djonin and V. Krishnamurthy. Q-learning algorithms for constrained Markov decision processes with randomized monotone policies: Application to MIMO transmission control. *IEEE Transactions on Signal Processing*, vol. 55, no. 5, pp. 2170-2181, May 2007.

[15] C. C. Moallemi, S. Kumar, and B. Van Roy. Approximate and data-driven dynamic programming for queuing networks. Submitted for publication, 2008.

[16] R. Berry and R. Gallager. Communication over fading channels with delay constraints. *IEEE Transactions on Information Theory*, vol. 48, no. 5, pp. 1135-1149, May 2002.

[17] M. J. Neely. Optimal energy and delay tradeoffs for multi-user wireless downlinks. *IEEE Transactions on Information Theory*, vol. 53, no. 9, pp. 3095-3113, Sept. 2007.

[18] M. J. Neely. Super-fast delay tradeoffs for utility optimal fair scheduling in wireless networks. *IEEE Journal on Selected Areas in Communications, Special Issue on Nonlinear Optimization of Communication Systems*, vol. 24, no. 8, pp. 1489-1501, Aug. 2006.

[19] A. Stolyar. Maximizing queueing network utility subject to stability: Greedy primal-dual algorithm. *Queueing Systems*, vol. 50, no. 4, pp. 401-457, 2005.

[20] A. Stolyar. Greedy primal-dual algorithm for dynamic resource allocation in complex networks. *Queueing Systems*, vol. 54, no. 3, pp. 203-220, 2006.

[21] A. Eryilmaz and R. Srikant. Fair resource allocation in wireless networks using queue-length-based scheduling and congestion control. *IEEE/ACM Transactions on Networking*, vol. 15, no. 6, pp. 1333-1344, Dec. 2007.

[22] A. Eryilmaz and R. Srikant. Joint congestion control, routing, and MAC for stability and fairness in wireless networks. *IEEE Journal on Selected Areas in Communications, Special Issue on Nonlinear Optimization of Communication Systems*, vol. 14, pp. 1514-1524, Aug. 2006.

[23] M. J. Neely, E. Modiano, and C. Li. Fairness and optimal stochastic control for heterogeneous networks. *Proc. IEEE INFOCOM*, pp. 1723-1734, March 2005.

[24] M. J. Neely. *Dynamic Power Allocation and Routing for Satellite and Wireless Networks with Time Varying Channels*. PhD thesis, Massachusetts Institute of Technology, LIDS, 2003.

[25] D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, MA, 1995.

[26] D. P. Bertsekas and R. Gallager. *Data Networks*. New Jersey: Prentice-Hall, Inc., 1992.

[27] M. J. Neely. *Stochastic Network Optimization with Application to Communication and Queueing Systems*. Morgan & Claypool, 2010.

[28] M. J. Neely. Energy optimal control for time varying wireless networks. *IEEE Transactions on Information Theory*, vol. 52, no. 7, pp. 2915-2934, July 2006.

[29] D. P. Bertsekas. *Dynamic Programming and Optimal Control, vols. 1 and 2*. Athena Scientific, Belmont, Mass, 1995.

[30] J. R. Norris. *Markov Chains*. Cambridge Series in Statistical and Probabilistic Mathematics, 1998.

[31] M. J. Neely. Stochastic optimization for Markov modulated networks with application to delay constrained wireless scheduling. *ArXiv Technical Report*, July 2011.

[32] M. J. Neely. Max weight learning algorithms with application to scheduling in unknown environments. *arXiv:0902.0630v1*, Feb. 2009.

[33] V. Dupač and U. Herkenrath. Stochastic approximation with delayed observations. *Biometrika*, vol. 72, no. 3, pp. 683-685, 1985.

[34] M. A. Mahmoud and A. A. Rasha. Stochastic approximation with compound delayed observations. *Mathematical and Computational Applications*, Vol. 10, no. 2, pp. 283-289, 2005.