Max Weight Learning Algorithms for Scheduling in Unknown Environments

Michael J. Neely , Scott T. Rager, Thomas F. La Porta

***Abstract*—We consider a discrete time queueing system where a controller makes a 2-stage decision every slot. The decision at the first stage reveals a hidden source of randomness with a control-dependent (but unknown) probability distribution. The decision at the second stage generates an attribute vector that depends on this revealed randomness. The goal is to stabilize all queues and optimize a utility function of time average attributes, subject to an additional set of time average constraints. This setting fits a wide class of stochastic optimization problems, including multi-user wireless scheduling with dynamic channel measurement decisions, and wireless multi-hop routing with multi-receiver diversity and opportunistic routing decisions. We develop a simple max-weight algorithm that learns efficient behavior by averaging functionals of previous outcomes.**

***Index Terms*—Wireless networks, opportunistic routing, queueing analysis, overhead and feedback**

I. Introduction

We consider a system with N queues that operate in discrete time with unit timeslots $t \in \{0, 1, 2, \ldots\}$. Let $\boldsymbol{Q}(t) = (Q_1(t), \ldots, Q_N(t))$ be the vector of queue backlogs. The dynamics for each $n \in \{1, \ldots, N\}$ are:

$$Q_n(t+1) = \max[Q_n(t) - b_n(t), 0] + a_n(t) \qquad (1)$$

where $a_n(t)$ and $b_n(t)$ are arrival and service processes. The arrivals $a_n(t)$ can be a sum of exogenous arrivals from traffic sources and endogenous arrivals from other network nodes, and so multi-hop networks can be treated using this framework. Multi-hop problems are treated in more detail in Section V. The units of $Q_n(t)$, $a_n(t)$, $b_n(t)$ depend on the context of the system, and these variables can be non-integer. Every slot t, a controller makes a 2-stage control decision that affects queue dynamics and generates a collection of random *network attributes*. Specifically, for each slot t there is a pair of random events $\eta(t) = [\beta(t), \omega(t)]$. The value of $\beta(t)$ is assumed to be known at the beginning of slot t and can be a multi-dimensional quantity, such as a vector of new arrivals or channel states for slot t. The process $\beta(t)$ is assumed to be i.i.d. over slots with distribution $\pi(\beta)$. However, the distribution $\pi(\beta)$ is unknown. Based on knowledge of $\beta(t)$, the controller first chooses an action $k(t)$ from a finite set of K "stage-1" control actions, given by an action set $\mathcal{K} = \{1, \ldots, K\}$.

After the action $k(t) \in \mathcal{K}$ is chosen, the additional random event $\omega(t)$ is revealed. The events $\omega(t)$ are conditionally i.i.d. with distribution $\pi_k(\omega)$ over all slots for which $k(t) = k$. The distribution functions $\pi_k(\omega)$ are unknown. Based on knowledge of the revealed $\omega(t)$ vector, the controller makes an additional decision $\alpha(t)$, where $\alpha(t)$ is chosen from some abstract set $\mathcal{A}_{\eta(t)}$ that possibly depends on $\eta(t)$. This decision affects the service rates and arrival processes of the queues on slot t, and additionally generates *attribute vectors* $\boldsymbol{x}(t) = (x_1(t), \ldots, x_M(t))$, $\boldsymbol{y}(t) = (y_0(t), y_1(t), \ldots, y_L(t))$ for some non-negative integers M, L. These are determined by arbitrary functions of $k(t)$, $\eta(t)$, and $\alpha(t)$:

$$a_n(t) = \hat{a}_n(k(t), \eta(t), \alpha(t)) \ , \ b_n(t) = \hat{b}_n(k(t), \eta(t), \alpha(t))$$
$$x_m(t) = \hat{x}_m(k(t), \eta(t), \alpha(t)) \ , \ y_l(t) = \hat{y}_l(k(t), \eta(t), \alpha(t))$$

The $x_m(t)$ and $y_l(t)$ quantities can represent additional rewards earned or penalties expended by the network on slot t, such as throughput admitted at different parts of the network, or powers expended by different network components.

For a given control policy, define time average expectation $\overline{\boldsymbol{x}} = (\overline{x}_1, \ldots, \overline{x}_M)$ as follows:[1]

$$\overline{\boldsymbol{x}} = \lim_{t\to\infty} \tfrac{1}{t} \textstyle\sum_{\tau=0}^{t-1} \mathbb{E}\left\{\boldsymbol{x}(\tau)\right\}$$

Define $\overline{\boldsymbol{y}} = (\overline{y}_0, \overline{y}_1, \ldots, \overline{y}_L)$ similarly. The goal is to develop an algorithm that makes decisions over time to solve:

$$\text{Minimize:} \quad \overline{y}_0 + f(\overline{\boldsymbol{x}}) \qquad (2)$$
$$\text{Subject to:} \quad \overline{y}_l \leq 0 \ \forall l \in \{1, \ldots, L\} \qquad (3)$$
$$k(t) \in \{1, \ldots, K\}, \alpha(t) \in \mathcal{A}_{\eta(t)} \ \forall t \qquad (4)$$
$$\text{Queues } Q_n(t) \text{ stable } \forall n \in \{1, \ldots, N\} \qquad (5)$$

where $f(\boldsymbol{x})$ is a continuous and convex function of $\boldsymbol{x} \in \mathbb{R}^M$, and where a discrete time queue $Q(t)$ is defined to be *stable* if:[2]

$$\limsup_{t\to\infty} \tfrac{1}{t} \textstyle\sum_{\tau=0}^{t-1} \mathbb{E}\left\{|Q(\tau)|\right\} < \infty$$

The above definition uses an absolute value of $Q(t)$, even though the processes $Q_n(t)$ in (1) are always non-negative, because we shall soon introduce *virtual queues* that can possibly be negative. We can use $M = 0$ and $f(\boldsymbol{x}) = 0$ in the above problem if there are no $\boldsymbol{x}(t)$ attributes and we seek only

This work was presented in part at the Information Theory and Applications Workshop (ITA), La Jolla, CA, February 2009.

M. J. Neely is with the Electrical Engineering department at the University of Southern California, Los Angles, CA. S. T. Rager and T. F. La Porta are with Pennsylvania State University, University Park, PA.

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, NSF grant 0964479, the Network Science Collaborative Technology Alliance sponsored by the U.S. Army Research Laboratory W911NF-09-2-0053.

[1]For simplicity of exposition, we temporarily assume the limiting time average expectation exists.

[2]This definition of queue stability is often called *strong stability*.

to minimize the time average of $y_0(t)$, rather than minimizing a convex function of a time average.

As an example, suppose that $f(\boldsymbol{x}) = 0$, there are no $x_m(t)$ attributes, and we define $y_0(t) = -admit(t)$, being -1 times the total amount of new packets admitted to the network on slot t. Further, for $l \in \{1, \ldots, L\}$ define $y_l(t) = p_l(t) - p_l^{av}$, where $p_l(t)$ represents power incurred by some component l in the network on slot t, and p_l^{av} is a desired average power constraint to be enforced. Then the problem of minimizing $\overline{y}_0$ subject to queue stability and to $\overline{y}_l \leq 0$ is one that seeks to maximize admitted throughput subject to stability and to average power constraints $\overline{p}_l \leq p_l^{av}$ at each network component $l \in \{1, \ldots, L\}$. As another example, suppose there are no $y_l(t)$ attributes. However, suppose that $x_m(t)$ is the admitted data of flow type m in the network on slot t, and $f(\boldsymbol{x}) = -\sum_{m=1}^{M} \log(1 + x_m)$ is -1 times a concave utility function of admitted data. Then the problem seeks to maximize a concave utility function of the throughput vector subject to queue stability. This particular example is considered in more detail in the simulation example of Section IV.

The problem (2)-(5) is similar to those treated in [2][3][4][5] using Lyapunov optimization and a drift-plus-penalty method, and in [6][7] using fluid limits. The problem can be partly addressed using these prior techniques in the following special cases:

- (Special Case 1) There is no "stage-1" control action $k(t)$, so that the revealed randomness $\omega(t)$ does not depend on any control decision.
- (Special Case 2) The distribution functions $\pi_k(\omega)$ are known.

An example of Special Case 1 is the problem of minimizing time average power expenditure in a multi-user wireless downlink (or uplink) with random time-varying channel states that are known at the beginning of every slot. Simple max-weight transmission policies are known to solve such problems, even without knowledge of the probability distributions for the channels or packet arrivals [4]. An example of Special Case 2 is the same system with the additional assumption that there is a cost to measuring channels at the beginning of each slot. In this example, we have the option of either measuring the channels (and thus having the hidden random channel states revealed to us) or transmitting blindly. Such problems are treated in [8][9], which use max-weight algorithms that include an expectation with respect to the known joint channel state distribution. Estimating the joint channel state distribution can be difficult when there are many (possibly correlated) channels. For example, if there are N channels and 1024 possible states per channel, there are 1024^N probabilities to be estimated in the joint channel state distribution.

Another important example is that of dynamic packet routing and transmission scheduling in a multi-commodity, multi-hop network with probabilistic channel errors and multi-receiver diversity. The Diversity Backpressure Routing (DIVBAR) algorithm of [10] reduces this problem to a 2-stage max-weight problem where each node decides which of the K commodities to transmit at the first stage. After transmission, the random vector of neighbor successes is revealed, and the "stage-2" packet forwarding decision is made. The solution given in [10] requires knowledge of the joint transmission success probabilities for all neighboring nodes.

In this paper, we provide a framework for solving such problems without having a-priori knowledge of the underlying probability distributions. Our approach uses the observation that, rather than requiring an estimate of the full probability distributions, all that is needed is an estimate of a set of expected *max-weight functionals* that depend on these distributions. These can be efficiently estimated from the history of previous events. Our analysis also provides a framework for treating time average equality constraints via a new virtual queue structure, and yields a generalized approximation theory for stochastic network optimization problems. Examples in Sections IV and V consider utility maximization and multi-hop networking.

II. Algorithm Design

A. Boundedness Assumptions

Assume that the functions $\hat{b}_n(\cdot)$, $\hat{y}_l(\cdot)$, $\hat{x}_m(\cdot)$ are arbitrary (possibly nonlinear, non-convex, discontinuous), but are bounded as follows:

$$y_{l,min} \leq \hat{y}_l(\cdot) \leq y_{l,max} \ , \ x_{m,min} \leq \hat{x}_m(\cdot) \leq x_{m,max}$$
$$0 \leq \hat{b}_n(\cdot) \leq b_{n,max}$$

where $y_{l,min}, y_{l,max}$, $x_{m,min}, x_{m,max}$, $b_{n,max}$ are finite constants. Assume the $\hat{a}_n(\cdot)$ functions are non-negative, and for any (possibly randomized) control choices for $k(t)$ and $\alpha(t)$ they satisfy

$$\mathbb{E}\left\{\hat{a}_n(k(t), \eta(t), \alpha(t))^2\right\} \leq A_n^2$$

for some finite constant A_n^2 that represents an upper bound on the second moment. For example, suppose new arrivals $a_n(t)$ are Poisson with rate λ_n, and do not depend on control actions $k(t)$, $\alpha(t)$. Then we can have $\eta(t) = [\beta(t), \omega(t)]$ with $\beta(t) = (a_1(t), \ldots, a_N(t))$, and for $n \in \{1, \ldots, N\}$ we have:

$$\begin{aligned} \hat{a}_n(k(t), \eta(t), \alpha(t)) &= a_n(t) \\ \mathbb{E}\left\{\hat{a}_n(k(t), \eta(t), \alpha(t))^2\right\} &= \lambda_n + \lambda_n^2 \end{aligned}$$

B. Transforming the Problem with Auxiliary Variables

Let $\boldsymbol{\gamma}(t) = (\gamma_1(t), \ldots, \gamma_M(t))$ be a vector of *auxiliary variables*, where for each $m \in \{1, \ldots, M\}$ and each slot t, the value $\gamma_m(t)$ is chosen subject to:

$$x_m^{min} - \sigma \leq \gamma_m(t) \leq x_m^{max} + \sigma \ \ \forall m \in \{1, \ldots, M\} \quad (6)$$

for some value $\sigma \geq 0$ (to be chosen later). Let $\boldsymbol{\gamma}(t) = (\gamma_1(t), \ldots, \gamma_M(t))$ be a vector of $\gamma_m(t)$ components for $m \in \{1, \ldots, M\}$. Define the time average expectation $\overline{\boldsymbol{\gamma}}(t)$ as follows:

$$\overline{\boldsymbol{\gamma}}(t) \triangleq \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\left\{\boldsymbol{\gamma}(\tau)\right\}$$

Define time average expectations $\overline{y}_l(t)$, $\overline{x}_m(t)$ similarly. Then it is not difficult to show that the problem (2)-(5) is equivalent

to the following:

$$\text{Min.:} \ \limsup_{t\to\infty}[\overline{y}_0(t)+f(\overline{\boldsymbol{\gamma}}(t))] \quad (7)$$

$$\text{Subj. to:} \ \limsup_{t\to\infty}\overline{y}_l(t)\leq 0 \ \forall l\in\{1,\ldots,L\} \quad (8)$$

$$\lim_{t\to\infty}[\overline{x}_m(t)-\overline{\gamma}_m(t)]=0 \ \forall m\in\{1,\ldots,M\} \quad (9)$$

$$k(t)\in\{1,\ldots,K\} \ , \alpha(t)\in\mathcal{A}_{\eta(t)} \ \forall t \quad (10)$$

$$\boldsymbol{\gamma}(t) \text{ satisfies (6) } \forall t \quad (11)$$

$$\text{Queues } Q_n(t) \text{ stable } \forall n\in\{1,\ldots,N\} \quad (12)$$

Indeed, the equality constraint (9) indicates that the auxiliary variable $\gamma_m(t)$ can be used as a proxy for $x_m(t)$, so that the above problem is equivalent to (2)-(5). These auxiliary variables are similar to those we introduced in [3][2][11] for optimizing a convex function of a time average penalty in a stochastic network. If $M=0$ and $f(\cdot)=0$, then no auxiliary variables are needed.

C. Virtual Queues and Lyapunov Drift

As in [4][2], for each each $l\in\{1,\ldots,L\}$, we enforce the inequality constraint (8) with a *virtual queue* $Z_l(t)$ with dynamics:

$$Z_l(t+1)=\max[Z_l(t)+y_l(t),0] \quad (13)$$

To enforce the equality constraint (9), for each $m\in\{1,\ldots,M\}$ we define a virtual queue $H_m(t)$ that can be possibly negative, with a new update structure:

$$H_m(t+1)=H_m(t)+\gamma_m(t)-x_m(t) \quad (14)$$

It is not difficult to show that the constraints (8),(9) are satisfied whenever all virtual queues $Z_l(t)$ and $H_m(t)$ are strongly stable [12].

Now define $\boldsymbol{\Theta}(t)\triangleq[\boldsymbol{Q}(t);\boldsymbol{Z}(t);\boldsymbol{H}(t)]$ as the vector of all actual and virtual queue values. To stabilize the queues, we define the following Lyapunov function:

$$L(\boldsymbol{\Theta}(t)) \triangleq \tfrac{1}{2}\sum_{n=1}^{N}Q_n(t)^2+\tfrac{1}{2}\sum_{l=1}^{L}Z_l(t)^2+\tfrac{1}{2}\sum_{m=1}^{M}H_m(t)^2$$

Intuitively, this Lyapunov function is large whenever one of the queues is large, and so keeping it small maintains stable queues. Define $\Delta(t)=L(\boldsymbol{\Theta}(t+1))-L(\boldsymbol{\Theta}(t))$. Let V be a non-negative parameter used to control the proximity of our algorithm to the optimal solution of (7)-(12). Using the framework of [2], we consider a control policy that observes the queue backlogs $\boldsymbol{\Theta}(t)$ and takes control actions on each slot t that minimize a bound on the following "drift-plus-penalty" expression:

$$\mathbb{E}\left\{\Delta(t)+V[y_0(t)+f(\boldsymbol{\gamma}(t))]\,|\,\boldsymbol{\Theta}(t)\right\}$$

Lemma 1: (The Drift-Plus-Penalty Bound) For any control policy, we have for any slot τ:

$$\begin{aligned}\mathbb{E}\left\{\Delta(\tau)+V[y_0(\tau)+f(\boldsymbol{\gamma}(\tau))]\,|\,\boldsymbol{\Theta}(\tau)\right\}\leq\\ B+V\mathbb{E}\left\{y_0(\tau)+f(\boldsymbol{\gamma}(\tau))\,|\,\boldsymbol{\Theta}(\tau)\right\}\\ +\sum_{n=1}^{N}Q_n(\tau)\mathbb{E}\left\{a_n(\tau)-b_n(\tau)\,|\,\boldsymbol{\Theta}(\tau)\right\}\\ +\sum_{l=1}^{L}Z_l(\tau)\mathbb{E}\left\{y_l(\tau)|\boldsymbol{\Theta}(\tau)\right\}\\ +\sum_{m=1}^{M}H_m(\tau)\mathbb{E}\left\{\gamma_m(\tau)-x_m(\tau)|\boldsymbol{\Theta}(\tau)\right\}\end{aligned} \quad (15)$$

where B is a finite constant that depends on the bounds for the $\hat{a}_n(\cdot)$, $\hat{b}_n(\cdot)$, $\hat{y}_l(\cdot)$, $\hat{x}_m(\cdot)$ functions.

Proof: See Appendix A. ∎

As in [2], our strategy every slot τ is to observe queue backlogs $\boldsymbol{\Theta}(\tau)$ and then make control actions $k(\tau)$, $\alpha(\tau)$, $\boldsymbol{\gamma}(\tau)$ that attempt to minimize:

$$\mathbb{E}\left\{RHS(\boldsymbol{\Theta}(\tau),k(\tau),\alpha(\tau),\boldsymbol{\gamma}(\tau))\,|\,\boldsymbol{\Theta}(\tau)\right\} \quad (16)$$

where the above conditional expectation represents the right-hand-side of the drift inequality (15). However, to enable system learning, it is important to allow for *approximate scheduling*.

Assumption A1 (Approximate Scheduling): Every slot τ the queue backlogs $\boldsymbol{\Theta}(\tau)$ are observed and control decisions $k(\tau)\in\mathcal{K}$, $\alpha(\tau)\in\mathcal{A}_{\eta(\tau)}$, and $\boldsymbol{\gamma}(\tau)$ satisfying (6) are made to ensure:

$$\begin{aligned}\mathbb{E}\left\{RHS(\boldsymbol{\Theta}(\tau),k(\tau),\alpha(\tau),\boldsymbol{\gamma}(\tau))|\boldsymbol{\Theta}(\tau)\right\}\leq\\ \mathbb{E}\left\{RHS(\boldsymbol{\Theta}(\tau),k^*(\tau),\alpha^*(\tau),\boldsymbol{\gamma}^*(\tau))|\boldsymbol{\Theta}(\tau)\right\}\\ +\mathbb{E}\left\{\tilde{C}(t)|\boldsymbol{\Theta}(t)\right\}+V\epsilon_V+\sum_{n=1}^{N}Q_n(\tau)\epsilon_Q\\ +\sum_{l=1}^{L}Z_l(\tau)\epsilon_Z+\sum_{m=1}^{M}|H_m(\tau)|\epsilon_H\end{aligned} \quad (17)$$

where $k^*(\tau)$, $\alpha^*(\tau)$, $\boldsymbol{\gamma}^*(\tau)$ are any other decisions that satisfy the required constraints (10)-(11), ϵ_V, ϵ_Q, ϵ_Z, ϵ_H, C are non-negative constants, and $\tilde{C}(t)$ is a random variable such that $\mathbb{E}\left\{\tilde{C}(t)\right\}\leq C$. We have $C=\epsilon_V=\epsilon_Q=\epsilon_Z=\epsilon_H=0$ if decisions exactly minimize (16) every slot, although this would typically require knowledge of the underlying probability distributions $\pi_k(\omega)$.

D. Feasibility and Slackness

Assume the problem (2)-(5) is *feasible*, so that it is possible to satisfy its constraints using some algorithm, and define $y_0^{opt}+f^{opt}$ as the infimum value of (2) over all feasible algorithms. In [12] it is shown that if the algorithm is feasible, then for all $\delta>0$ there exists a stationary and randomized algorithm that makes a control action $k^*(\tau)$ as a (possibly randomized) function of the observed $\beta(\tau)$, revealing a corresponding random vector $\omega^*(\tau)$, and chooses a stage-2 control action $\alpha^*(\tau)\in\mathcal{A}_{\eta^*(\tau)}$ as a potentially randomized function of

$\eta^*(\tau) = [\beta(\tau), \omega^*(\tau)]$ such that for any slot τ and independent of $\mathbf{\Theta}(\tau)$:

$$\mathbb{E}\left\{\hat{y}_0(k^*(\tau), \eta^*(\tau), \alpha^*(\tau))\right\} + f(\boldsymbol{\gamma}^{opt}) \leq y_0^{opt} + f^{opt} + \delta \quad (18)$$

$$\mathbb{E}\left\{\hat{y}_l(k^*(\tau), \eta^*(\tau), \alpha^*(\tau))\right\} \leq \delta \ \forall l \in \{1, \ldots, L\} \quad (19)$$

$$\mathbb{E}\left\{\hat{a}_n(k^*(\tau), \eta^*(\tau), \alpha^*(\tau))\right\} - \mathbb{E}\left\{\hat{b}_n(k^*(\tau), \eta^*(\tau), \alpha^*(\tau))\right\} \leq \delta \ \forall n \in \{1, \ldots, N\} \quad (20)$$

$$\left|\mathbb{E}\left\{\hat{x}_m(k^*(\tau), \eta^*(\tau), \alpha^*(\tau))\right\} - \gamma_m^{opt}\right| \leq \delta \ \forall m \in \{1, \ldots, M\} \quad (21)$$

where $\boldsymbol{\gamma}^{opt} = (\gamma_1^{opt}, \ldots, \gamma_M^{opt})$ is a vector that satisfies $f(\boldsymbol{\gamma}^{opt}) = f^{opt}$. Intuitively, this means that if the problem is feasible, then we can achieve the desired constraints and performance objective arbitrarily closely over the class of stationary and randomized algorithms. The use of $\delta > 0$ is necessary to treat general cases where infimums are not necessarily achievable by a single randomized policy, such as when the function $\hat{y}_l(k^*, \eta^*, \alpha^*)$ is not continuous in α^*, and/or the set $\mathcal{A}_{\eta^*}$ is not compact. In most cases where mild closure properties are satisfied, (18)-(21) hold with $\delta = 0$, although such closure properties are not needed in any of our analysis.

The next assumption states that the constraints are not only feasible, but have a useful slackness property, analogous to a Slater condition for static optimization problems [13].

Assumption A2 (Slackness of Constraints): There is a value $\epsilon_{max} > 0$ together with a stationary and randomized policy that makes stage-1 and stage-2 control decisions $\tilde{k}(\tau) \in \mathcal{K}$ and $\tilde{\alpha}(\tau) \in \mathcal{A}_{\tilde{\eta}(\tau)}$, possibly different than the decisions in (18)-(21), such that:

$$\mathbb{E}\left\{\hat{y}_l(\tilde{k}(\tau), \tilde{\eta}(\tau), \tilde{\alpha}(\tau))\right\} \leq -\epsilon_{max} \ \forall l \in \{1, \ldots, L\} \quad (22)$$

$$\mathbb{E}\left\{\hat{a}_n(\tilde{k}(\tau), \tilde{\eta}(\tau), \tilde{\alpha}(\tau)) - \hat{b}_n(\tilde{k}(\tau), \tilde{\eta}(\tau), \tilde{\alpha}(\tau))\right\} \leq -\epsilon_{max} \ \forall n \in \{1, \ldots, N\} \quad (23)$$

Theorem 1: (Performance Theorem) Suppose we use parameters $V \geq 0$ and $\sigma > 0$, and Assumptions A1 and A2 hold. Define f_{min}, f_{max} as the minimum and maximum values of $f(\boldsymbol{\gamma})$ over the constraints (6), and assume these are finite. Suppose ϵ_Q, ϵ_Z, ϵ_H are small enough, and that σ is large enough, so that:

$$\epsilon_Q < \epsilon_{max} \ , \ \epsilon_Z < \epsilon_{max} \ , \ \epsilon_H < \sigma \quad (24)$$

Then all queues are strongly stable and so all constraints (8)-(12) hold (and hence all constraints of the original problem (2)-(5) hold). Further:

$$\limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1}\sum_{n=1}^{N}\mathbb{E}\left\{Q_n(\tau)\right\} \leq \frac{B + C + V(\epsilon_V + d)}{\epsilon_1} \quad (25)$$

$$\limsup_{t\to\infty}[\overline{y}_0(t) + f(\overline{\boldsymbol{x}}(t))] \leq y_0^{opt} + f^{opt} + \epsilon_V + \epsilon_2 + \frac{B + C}{V} \quad (26)$$

where constants B, C are from Lemma 1 and Assumption A1, respectively, and:

$$\begin{aligned} \epsilon_1 &\triangleq \min[\epsilon_{max} - \epsilon_Q, \epsilon_{max} - \epsilon_Z, \sigma - \epsilon_H] \\ d &\triangleq y_{0,max} - y_{0,min} + f_{max} - f_{min} \\ \epsilon_2 &\triangleq d\max[\epsilon_Q/\epsilon_{max}, \epsilon_Z/\epsilon_{max}, \epsilon_H/\sigma] \end{aligned}$$

Additional queue and utility bounds are derived for a finite horizon in (29) and (30) of the proof. Note that $\epsilon_1 = \min[\epsilon_{max}, \sigma]$ and $\epsilon_2 = \epsilon_V = 0$ if the exact minimization of the $RHS(\cdot)$ function in (16) is performed every slot t. This shows that achieved cost can be pushed to within $O(1/V)$ of optimality, with a corresponding $O(V)$ tradeoff in queue backlog, as in [2]. The difference from [2] is that we consider the 2-stage structure, generalized approximate scheduling, and arbitrary convex cost functions $f(\boldsymbol{x})$ (without the entrywise non-increasing or non-decreasing assumptions in [2]). This latter part is achieved by the new virtual queue structure (14) that enforces an equality constraint. This structure can also be used to treat additional time average equality constraints.

Proof: (Theorem 1) Using Assumption A1 with (15) yields for any slot τ:

$$\begin{aligned} \mathbb{E}\left\{\Delta(\tau) + V[y_0(\tau) + f(\boldsymbol{\gamma}(\tau))] \mid \mathbf{\Theta}(\tau)\right\} \leq \\ B + \mathbb{E}\left\{\tilde{C}(t)|\mathbf{\Theta}(t)\right\} + V\epsilon_V + \epsilon_H\sum_{m=1}^{M}|H_m(\tau)| \\ + V\mathbb{E}\left\{y_0'(\tau) + f(\boldsymbol{\gamma}'(\tau)) \mid \mathbf{\Theta}(\tau)\right\} \\ + \sum_{n=1}^{N} Q_n(\tau)\mathbb{E}\left\{a_n'(\tau) - b_n'(\tau) + \epsilon_Q \mid \mathbf{\Theta}(\tau)\right\} \\ + \sum_{l=1}^{L} Z_l(\tau)\mathbb{E}\left\{y_l'(\tau) + \epsilon_Z|\mathbf{\Theta}(\tau)\right\} \\ + \sum_{m=1}^{M} H_m(\tau)\mathbb{E}\left\{\gamma_m'(\tau) - x_m'(\tau)|\mathbf{\Theta}(\tau)\right\} \end{aligned} \quad (27)$$

for any alternative decisions $k'(\tau)$, $\alpha'(\tau)$, $\boldsymbol{\gamma}'(t)$ that satisfy (11), with corresponding random event $\eta'(\tau) = [\beta(\tau), \omega'(\tau)]$, variables $a_n'(\tau) = \hat{a}_n(k'(\tau), \eta'(\tau), \alpha'(\tau))$, and variables $b_n'(\tau)$, $x_m'(\tau)$, $y_l'(\tau)$ defined similarly in terms of $k'(\tau)$, $\eta'(\tau)$, $\alpha'(\tau)$. Now fix $\delta > 0$, and consider the following policy:

$$(k'(\tau), \alpha'(\tau), \gamma'(\tau)) = \begin{cases} (k^*(\tau), \alpha^*(\tau), \boldsymbol{\gamma}^{opt}) & \text{w.p. } 1-p \\ (\tilde{k}(\tau), \tilde{\alpha}(\tau), \tilde{\boldsymbol{\gamma}}) & \text{w.p. } p \end{cases}$$

where p is some probability (to be specified later), "w.p." stands for "with probability," $(k^*(\tau), \alpha^*(\tau), \boldsymbol{\gamma}^{opt})$ is the stationary, randomized policy of (18)-(21), $(\tilde{k}(\tau), \tilde{\alpha}(\tau), \tilde{\boldsymbol{\gamma}})$ is the policy of (22)-(23) from Assumption A2, and where $\tilde{\gamma}_m$ is defined:

$$\tilde{\gamma}_m = \mathbb{E}\left\{\hat{x}_m(\tilde{k}(\tau), \tilde{\eta}(\tau), \tilde{\alpha}(\tau))\right\} - sign(H_m(\tau))\sigma \ \forall m \in \{1, \ldots, M\}$$

where $sign(H_m(\tau))$ is 1 if $H_m(\tau) \geq 0$, and -1 else. Using this policy with (18)-(23) and taking a limit as $\delta \to 0$ yields

for all n, l, m:

$$\begin{aligned}
\mathbb{E}\left\{y_0'(\tau) + f(\boldsymbol{\gamma}'(\tau))|\boldsymbol{\Theta}(\tau)\right\} &\leq (1-p)(y_0^{opt} + f^{opt}) \\
&\quad + p(y_{0,max} + f_{max}) \\
\mathbb{E}\left\{a_n'(\tau) - b_n'(\tau) + \epsilon_Q|\boldsymbol{\Theta}(\tau)\right\} &\leq -p\epsilon_{max} + \epsilon_Q \\
\mathbb{E}\left\{y_l'(\tau) + \epsilon_Z|\boldsymbol{\Theta}(\tau)\right\} &\leq -p\epsilon_{max} + \epsilon_Z \\
H_m(\tau)\mathbb{E}\left\{\gamma_m'(\tau) - x_m'(\tau)|\boldsymbol{\Theta}(\tau)\right\} &\leq -p\sigma|H_m(\tau)|
\end{aligned}$$

Plugging these into (27) yields:

$$\begin{aligned}
&\mathbb{E}\left\{\Delta(\tau) + V[y_0(\tau) + f(\boldsymbol{\gamma}(\tau))] \,|\, \boldsymbol{\Theta}(\tau)\right\} \leq \\
&\quad B + \mathbb{E}\left\{\tilde{C}(t)|\boldsymbol{\Theta}(t)\right\} + V\epsilon_V \\
&\quad + V(1-p)(y_0^{opt} + f^{opt}) + Vp(y_{0,max} + f_{max}) \\
&\quad - \sum_{n=1}^{N} Q_n(\tau)(p\epsilon_{max} - \epsilon_Q) - \sum_{l=1}^{L} Z_l(\tau)(p\epsilon_{max} - \epsilon_Z) \\
&\quad - \sum_{m=1}^{M} |H_m(\tau)|(p\sigma - \epsilon_H) \qquad (28)
\end{aligned}$$

Inequality (28) holds for any probability p. Using $p = 1$ and rearranging terms yields:

$$\begin{aligned}
\mathbb{E}\left\{\Delta(\tau)|\boldsymbol{\Theta}(\tau)\right\} &\leq B + \mathbb{E}\left\{\tilde{C}(t)|\boldsymbol{\Theta}(t)\right\} + V(\epsilon_V + d) \\
&\quad - \epsilon_1\left[\sum_{n=1}^{N} Q_n(\tau) + \sum_{l=1}^{L} Z_l(\tau) + \sum_{m=1}^{M} |H_m(\tau)|\right]
\end{aligned}$$

Taking expectations of both sides of the above, summing over $\tau \in \{0, 1, \ldots, t-1\}$, and rearranging terms (as in [2]), yields:

$$\begin{aligned}
&\frac{\mathbb{E}\left\{L(\boldsymbol{\Theta}(t))\right\} - \mathbb{E}\left\{L(\boldsymbol{\Theta}(0))\right\}}{\epsilon_1 t} \\
&+ \frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\left\{\sum_{n=1}^{N} Q_n(\tau) + \sum_{l=1}^{L} Z_l(\tau) + \sum_{m=1}^{M} |H_m(\tau)|\right\} \\
&\quad \leq \frac{B + C + V(\epsilon_V + d)}{\epsilon_1} \qquad (29)
\end{aligned}$$

Using non-negativity of $L(\boldsymbol{\Theta}(t))$ and taking limits proves strong stability of all queues, ensures all constraints (8)-(12) hold, and proves (25).

Alternatively, using $p = \max[\epsilon_Q/\epsilon_{max}, \epsilon_Z/\epsilon_{max}, \epsilon_H/\sigma]$ in (28) yields:

$$\begin{aligned}
&\mathbb{E}\left\{\Delta(\tau) + V[y_0(\tau) + f(\boldsymbol{\gamma}(\tau))]|\boldsymbol{\Theta}(\tau)\right\} \leq \\
&\quad B + \mathbb{E}\left\{\tilde{C}(t)|\boldsymbol{\Theta}(t)\right\} + V\epsilon_V \\
&\quad + V(1-p)(y_0^{opt} + f^{opt}) + Vp(y_{0,max} + f_{max})
\end{aligned}$$

Taking iterated expectations of both sides and summing over $\tau \in \{0, \ldots, t-1\}$ as before yields:

$$\begin{aligned}
&\frac{\mathbb{E}\left\{L(\boldsymbol{\Theta}(t))\right\} - \mathbb{E}\left\{L(\boldsymbol{\Theta}(0))\right\}}{Vt} + \frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\left\{y_0(\tau) + f(\boldsymbol{\gamma}(\tau))\right\} \\
&\qquad \leq y_0^{opt} + f^{opt} + \frac{B+C}{V} + \epsilon_V + pd
\end{aligned}$$

Using non-negativity of $L(\boldsymbol{\Theta}(t))$ and Jensen's inequality yields:

$$\begin{aligned}
\overline{y}_0(t) + f(\overline{\boldsymbol{\gamma}}(t)) &\leq y_0^{opt} + f^{opt} + (B+C)/V + \epsilon_V + pd \\
&\quad + \mathbb{E}\left\{L(\boldsymbol{\Theta}(0))\right\}/(Vt) \qquad (30)
\end{aligned}$$

The result (26) follows by taking a limit as $t \to \infty$ and using continuity of $f(\boldsymbol{\gamma})$ and (9). ■

E. The Drift-Plus-Penalty Algorithm

Define:

$$\begin{aligned}
P_k(\eta(t), \alpha(t), \boldsymbol{\Theta}(t)) &\triangleq V\hat{y}_0(k, \eta(t), \alpha(t)) \\
&\quad + \sum_{l=1}^{L} Z_l(t)\hat{y}_l(k, \eta(t), \alpha(t)) \\
&\quad + \sum_{n=1}^{N} Q_n(t)[\hat{a}_n(k, \eta(t), \alpha(t)) - \hat{b}_n(k, \eta(t), \alpha(t))] \\
&\quad - \sum_{m=1}^{M} H_m(t)\hat{x}_m(k, \eta(t), \alpha(t)) \qquad (31)
\end{aligned}$$

This consists of those terms of the right hand side of (15) that do not involve the auxiliary variables $\boldsymbol{\gamma}(t)$. Suppose that the action $\alpha(t)$ and the set $\mathcal{A}_{\eta(t)}$ has the structure $\alpha(t) = [J(t), I(t)]$ and $\mathcal{A}_{\eta(t)} = \mathcal{J}_{\beta(t)} \times \mathcal{I}_{\eta(t)}$, where $J(t)$ is chosen in a set $\mathcal{J}_{\beta(t)}$ (a set depending only on $\beta(t)$), and $I(t)$ is chosen in the set $\mathcal{I}_{\eta(t)}$. Suppose that:

$$P_k(\eta(t), \alpha(t), \boldsymbol{\Theta}(t)) = R(\beta(t), J(t)) + Y_k(\eta(t), I(t), \boldsymbol{\Theta}(t))$$

so that it can be decomposed into a term that depends only on $J(t)$ and $\beta(t)$, and a term that does not depend on $J(t)$. Such a structure always exists in the trivial form when $J(t)$ is null, and $R(\cdot)$ is zero. However, this structure is useful, for example, when $\beta(t)$ represents random arrivals and $J(t)$ a flow control decision, which can be optimally chosen independently of the $k(t)$ decision and the $\omega(t)$ outcome. The algorithm that minimizes the $RHS(\cdot)$ expression (16) observes $\beta(t)$, $\boldsymbol{\Theta}(t)$ every slot t, and makes decisions as follows:

- (Auxiliary Variables) Observe $\boldsymbol{H}(t)$. Choose $\boldsymbol{\gamma}(t)$ subject to (6) to minimize:
$$Vf(\boldsymbol{\gamma}(t)) + \textstyle\sum_{m=1}^{M} H_m(t)\gamma_m(t)$$
- (Stage-1 Decision $k(t)$) Observe $\beta(t)$, $\boldsymbol{\Theta}(t)$. Choose $k(t)$ as the integer $k \in \{1, \ldots, K\}$ with the smallest value of $e_k(t)$, defined (recall that $\eta(t) = [\beta(t), \omega(t)]$):
$$\begin{aligned}
&e_k(t) \triangleq \\
&\mathbb{E}\left\{\min_{I \in \mathcal{I}_{\eta(t)}} Y_k(\eta(t), I, \boldsymbol{\Theta}(t))|\boldsymbol{\Theta}(t), k(t) = k, \beta(t)\right\} \qquad (32)
\end{aligned}$$
where the expectation in (32) is with respect to the distribution $\pi_k(\omega)$ for the random $\omega(t)$ event given $k(t) = k$.
- (Stage-2 Decision $\alpha(t) = [J(t), I(t)]$) Let $k(t)$ be the stage-1 decision, observe the resulting $\eta(t) = [\beta(t), \omega(t)]$, and observe $\boldsymbol{\Theta}(t)$. Choose action $I(t) \in \mathcal{I}_{\eta(t)}$ to minimize $Y_{k(t)}(\eta(t), I(t), \boldsymbol{\Theta}(t))$. Choose action $J(t) \in \mathcal{J}_{\beta(t)}$ to minimize $R(\beta(t), J(t))$.[3]
- (Queue Updates) Update the virtual and actual queues via (1), (13), (14).

[3]If the infimum of the $Y_k(\cdot)$ and/or $R(\cdot)$ functions over $I(t) \in \mathcal{I}_{\eta(t)}$ and $J(t) \in \mathcal{J}_{\beta(t)}$ cannot be achieved, we can simply get within C of the infimum and use Assumption A1 with $C > 0$.

This algorithm yields $\epsilon_V = \epsilon_Q = \epsilon_Z = \epsilon_H = C = 0$. However, we cannot compute the $e_k(t)$ values needed for the stage-1 decision without knowledge of the $\pi_k(\omega)$ distributions, and hence we need to use efficient estimates.

III. Estimating the Max-Weight Functional

There may be some stage-1 control options $k \in \mathcal{K}$ that generate outcomes $\omega(t)$ for which the outcome under another option $\tilde{k} \in \mathcal{K}$ can be *inferred*. For example, suppose we have a wireless system where the options are to measure none of the channels, measure some of the channels, or measure them all, where the $\omega(t)$ outcome is a vector of channel states for all measured channels, having "null" values for channels not measured. Then the outcome associated with measuring any subset of the channels can equally be inferred by measuring *all* channels. Define $\tilde{\mathcal{K}}$ as the subset of $\mathcal{K}$ such that the outcome for any $k \in \mathcal{K}$ can be inferred by the outcome for some $\tilde{k} \in \tilde{\mathcal{K}}$. To persistently explore all possible stage-1 actions, on every slot t we independently declare an *exploration event* with probability θ, where $0 < \theta < 1$. If an exploration event occurs, we independently choose option $\tilde{k}(t) \in \tilde{\mathcal{K}}$ uniformly over all options in $\tilde{\mathcal{K}}$ (regardless of the state of the network on this slot). We say that the action is a *k-inferred exploration event* if the outcome for option k can be inferred from the outcome for the chosen $\tilde{k}(t)$ (note that there can be many k for which the same event is a k-inferred exploration event).

Define opt_θ as the optimal cost for the problem (2)-(5) subject to the additional constraints that we use exploration events of probability θ. For simplicity, we measure our cost performance with respect to opt_θ. While $opt_\theta \geq y_0^{opt} + f^{opt}$, the understanding is that the gap is small if θ is small. Indeed, it can be shown that $|opt_\theta - (y_0^{opt} + f^{opt})| \leq O(\theta)$. Such probabilistic exploration events can be built into the framework of the system by re-defining the observed random event $\beta(t)$ to have the following structure: $\beta'(t) = [\phi(t), \beta(t)]$, where $\beta(t)$ is the original event, and $\phi(t)$ is chosen independently and i.i.d. every slot in the set $\{0\} \cup \tilde{\mathcal{K}}$, with $Pr[\phi(t) = 0] = 1 - \theta$ and $Pr[\phi(t) = \tilde{k}] = \theta/|\tilde{\mathcal{K}}|$ for all $\tilde{k} \in \tilde{\mathcal{K}}$, where $|\tilde{\mathcal{K}}|$ represents the number of elements in $\tilde{\mathcal{K}}$. If $\phi(t) = 0$, then no exploration event occurs. If $\phi(t) = \tilde{k}$, then the structure of the $\hat{a}_n(\cdot)$, $\hat{b}_n(\cdot)$, $\hat{y}_l(\cdot)$, $\hat{x}_m(\cdot)$ functions, which depend on $\beta'(t) = [\phi(t), \beta(t)]$, force the stage-1 decision to be $\tilde{k}$ by having outcomes identical to choosing $\tilde{k}$ whenever any other choice $k \neq \tilde{k}$ is chosen.

A. Estimating the $e_k(t)$ value — Approach 1

Define an integer W that represents a *moving average window size*. For each stage-1 option $k \in \mathcal{K}$ and each time t, define $\omega_1^{(k)}(t), \ldots, \omega_W^{(k)}(t)$ as the outcome that would have occurred at the time $\tau \leq t$ at which the wth latest k-inferred exploration event took place. Define $\eta_w^{(k)}(t) \triangleq [\beta(t), \omega_w^{(k)}(t)]$ as the outcome with the current $\beta(t)$ value but the previously observed $\omega_w^{(k)}(t)$ value. Define the estimate $\hat{e}_k(t)$ as follows:

$$\hat{e}_k(t) \triangleq \frac{1}{W} \sum_{w=1}^{W} \min_{I \in \mathcal{I}_{\eta_w^{(k)}(t)}} \left[Y_k(\eta_w^{(k)}(t), I, \boldsymbol{\Theta}(t)) \right]$$

In the case when there have not yet been W previous k-inferred exploration events by time t, the estimate $\hat{e}_k(t)$ is taken with respect to the (fewer than W) events, and is set to zero if no such events have occurred. The estimates $\hat{e}_k(t)$ can be viewed as empirical averages using the current queue backlogs $\boldsymbol{\Theta}(t) = [\boldsymbol{Q}(t); \boldsymbol{Z}(t); \boldsymbol{H}(t)]$ and the current $\beta(t)$, but using the outcomes $\omega_w^{(k)}(t)$ observed on previous exploration events.

Note that one might define $\hat{e}_k(t)$ according to an average over the past W slots on which an outcome of stage-1 decision k can be inferred, rather than restricting to sampling at exploration events. The reason we have used exploration events is to overcome the subtle "inspection paradox" issues involved in sampling the previous $\omega(\tau)$ outcomes. Indeed, even though outcomes $\omega(\tau)$ are generated in an i.i.d. way every slot in which $k(\tau) = k$ is chosen, the distribution of the last-seen outcome ω that corresponds to a particular decision k may be *skewed* in favor of creating larger penalties. This is because our algorithm may choose to avoid decision k for a longer period of time if this last outcome was non-favorable. Sampling at random exploration events ensures that our samples indeed form an i.i.d. sequence. An additional difficulty remains: Even though these samples $\{\omega_w^{(k)}(t)\}$ form an i.i.d. sequence, they are *not* independent of the queue values $\boldsymbol{\Theta}(t)$, as these prior outcomes have influenced the current queue states. We overcome this difficulty in Section III-D via a delayed-queue analysis.

B. Estimating the $e_k(t)$ value — Approach 2

The estimation of Approach 1 does not require knowledge of the $\pi_k(\omega)$ distributions. However, evaluation of $\hat{e}_k(t)$ requires W minimizations of $Y_k(\cdot)$ on each slot t, according to the value of each particular $\omega_w^{(k)}(t)$ outcome. Here we describe an approach that has a per-timeslot computation that is independent of W, in the special case when the $Y_k(\cdot)$ function does not depend on $\beta(t)$. Again let W be an integer moving average window size, and define $\omega_1^{(k)}(t), \ldots, \omega_W^{(k)}(t)$ the same as in Approach 1. Further define $\boldsymbol{\Theta}_1^{(k)}(t), \ldots, \boldsymbol{\Theta}_W^{(k)}(t)$ as the *queue backlogs* seen at the corresponding times of these exploration events. Define an estimate $e'_k(t)$ as follows:

$$e'_k(t) \triangleq \frac{1}{W} \sum_{w=1}^{W} \min_{I \in \mathcal{I}_{\eta_w^{(k)}(t)}} \left[Y_k(\eta_w^{(k)}(t), I, \boldsymbol{\Theta}_w^{(k)}(t)) \right]$$

The $e'_k(t)$ estimate is adjusted appropriately if fewer than W k-inferred exploration events have occurred. This approach is different from Approach 1 in that the current queue backlogs are not used. These minimizations can be done on every slot τ on which an exploration event occurs and $\tilde{k} \in \tilde{\mathcal{K}}$ is chosen, and the results can be stored and used for later use. This requires one minimization of $Y_k(\cdot)$ for each $k \in \mathcal{K}$ such that outcome $\omega^{(k)}(\tau)$ can be inferred from the outcome $\omega(\tau)$. This requires a number of minimizations that is at most K per exploration event, regardless of the window size W.

C. The Max-Weight Learning Algorithm

Let θ be a given exploration probability, $\sigma > 0$, $V \geq 0$ be given parameters, and W a positive integer window size.[4]

[4]A variable length window size algorithm with $W(t)$ that grows with t is also analyzed in [1] and shown to achieve exact optimality subject to a weaker "mean rate stability" constraint.

Define $\tilde{K} \triangleq |\tilde{\mathcal{K}}|$. The *Max-Weight Learning Algorithm* is as follows.

- (Initialization) Let $\mathbf{\Theta}(-\tilde{K}W) = \mathbf{0}$, and run the system over slots $t = \{-W\tilde{K}, -W\tilde{K}+1, \ldots, -1\}$, choosing each stage-1 decision option $\tilde{k} \in \tilde{\mathcal{K}}$ in a fixed round-robin order, choosing $\alpha(t)$ to minimize $P_{k(t)}(\eta(t), \alpha(t), \mathbf{\Theta}(t))$, and choosing $\boldsymbol{\gamma}(t)$ according to the auxiliary variable selection algorithm. This ensures that we have W independent samples by time 0, and creates a possibly non-zero initial queue state $\mathbf{\Theta}(0)$. Next perform the following sequence of actions for each slot $t \geq 0$.
- (Stage-1 Decisions) Independently with probability θ, decide to have an exploration event. If there is an exploration event, choose $k(t)$ uniformly over all options in $\tilde{\mathcal{K}}$. If there is no exploration event, then under Approach 1 we observe current queue backlogs $\mathbf{\Theta}(t)$ and compute $\hat{e}_k(t)$ for each $k \in \{1, \ldots, K\}$ (using window size W). We then choose $k(t)$ as the index $k \in \{1, \ldots, K\}$ that minimizes $\hat{e}_k(t)$ (breaking ties arbitrarily). Under Approach 2, if there is no exploration event we choose $k(t)$ to minimize $e'_k(t)$.
- (Stage-2 Decisions) Choose auxiliary variables $\boldsymbol{\gamma}(t)$ and actions $\alpha(t) \in \mathcal{A}_{\eta(t)}$ as before, using the $k(t)$ value from stage 1, and its resulting $\omega(t)$.
- (Past Value Storage) For Approach 1, store the $\omega(t)$ vector in memory as appropriate. For Approach 2, store the costs from minimizing $Y_k(\cdot)$ in memory as appropriate.
- (Queue Updates) Update virtual and actual queues via (1), (13), (14).

D. Analysis of the Max-Weight Learning Algorithm

For brevity, we analyze only Approach 1 (Approach 2 analysis is given in [1]). Our goal is to compute parameters C, ϵ_V, ϵ_Q, ϵ_Z, ϵ_H for (17) that can be plugged into Theorem 1.

Theorem 2: (Performance Under Approach 1) Suppose Approach 1 is implemented using an exploration probability $\theta > 0$, a positive integer window size W, and a fixed parameter $V \geq 0$. Then condition (17) of Assumption A1 holds with:

$$C = D_2 W K \tilde{K}/\theta \quad , \quad \epsilon_V = \epsilon_Q = \epsilon_Z = \epsilon_H = K D_1/\sqrt{W} \tag{33}$$

where D_1, D_2 are constants independent of queue backlog and of V, W, θ (and depend on the worst case moments of arrival, service, and attribute functions). Alternatively, if the algorithm is implemented using W samples of $\omega(t)$ that are somehow independently generated with the correct distribution (but not taken from the past), then the same holds with $C = 0$.

Proof: See Appendix B. ∎

It follows that if the fixed window size W is chosen to be suitably large, then the ϵ_V, ϵ_Q, ϵ_Z, ϵ_H constants will be small enough to satisfy the conditions (24) required for Theorem 1.

IV. Adaptive Channel Measurement in a Wireless Downlink

Here we consider an example of maximizing throughput-utility in a 3-queue wireless system with joint flow control and opportunistic transmission. The stage-1 decision chooses whether to measure channels or to transmit blindly, where measuring channels incurs an overhead that reduces throughput by $1/2$, with the advantage of knowing what rates can currently be achieved on each channel. The detailed model is in Section IV-B. We first present the simulation results.

A. Simulation Results

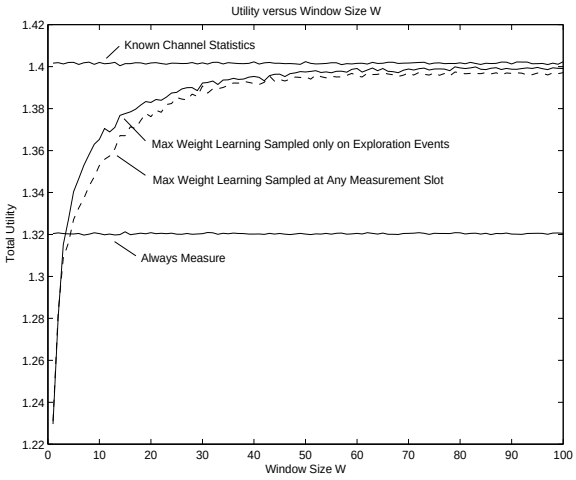

Fig. 1. Total utility versus W with fixed $V = 100$.

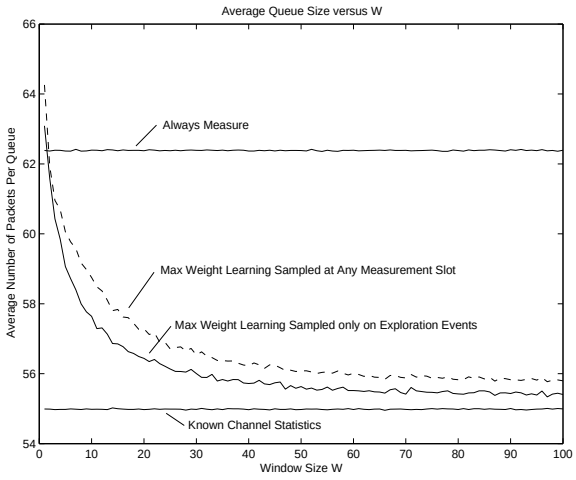

Fig. 2. Average queue backlog versus W with fixed $V = 100$.

The simulation results are plotted versus the moving average window size W in Figs. 1 and 2, with a fixed $V = 100$. Fig. 1 shows the simulated utility achieved by the optimal statistics-aware policy with perfect knowledge of $e_k(t)$. This policy is independent of W, and the small variations are due to the independent simulation runs. The figure also shows how the utility of the (statistics unaware) max-weight learning algorithm (Approach 1) increases to this optimal utility as the window size W is increased. Also plotted is the utility under a max-weight learning policy that samples at *any* measurement slot, not necessarily slots that are exploration events. The utility is slightly worse for small W, but seems to converge to the same value for large W, suggesting that the "inspection

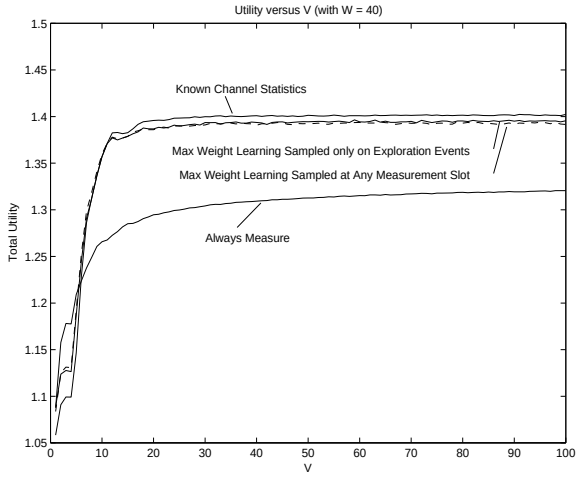

Fig. 3. Total utility versus V with fixed $W = 40$.

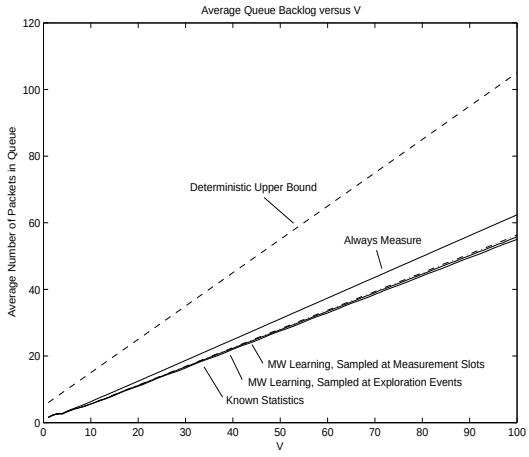

Fig. 4. Average backlog versus V with fixed $W = 40$.

paradox" issues are minimal when the window size is large. For comparison purposes, the figure also plots the utility achieved by a policy that always decides to measure the channel, showing that performance is degraded. Fig. 2 plots the average backlog in each of the queues under these same experiments. The same qualitative performance is observed: The statistics-aware policy performs the best (having the lowest backlog), the "always measure" policy performs the worst, and the max-weight learning algorithms improve as W is increased.

Fig. 3 shows how total utility increases as V is increased, for a fixed window size $W = 40$. It is seen that performance improves quickly with V, coming very close to optimal after $V = 20$. There is a small gap between the optimal utility (achieved by the statistics-aware algorithm) and the max-weight learning algorithms. This gap would be reduced if we used a larger window size W. Finally, it can be shown that worst case backlog is $Q_n(t) \leq V + 4 + \sigma$ for all queues n and all slots t (see analysis in next subsection). This deterministic bound is plotted in Fig. 4 along with the average queue values.

B. System Details

The system has 3 queues with time-varying channel states $\boldsymbol{S}(t) = (S_1(t), S_2(t), S_3(t))$, where $S_n(t) \in \{0, 2, 4\}$ and represents the number of packets that can be transmitted on channel n on slot t. However, $\boldsymbol{S}(t)$ is unknown and there are two stage-1 options every slot: Measure all channels ($k(t) = 1$) or measure no channels ($k(t) = 0$). Measuring channels yields $\omega(t) = \boldsymbol{S}(t)$, while measuring no channels yields $\omega(t) = Null$. There is only one type of exploration event: Measuring all channels (so that $\tilde{K} = 1$). This is because the result of transmitting blindly can be inferred by knowledge of the channel measurements. There are randomly arriving packets every slot, denoted by a process $\beta(t) = (\beta_1(t), \beta_2(t), \beta_3(t))$, where $\beta_n(t) \in \{0, 1, 2\}$ for all t and all n. *Flow control decisions* $J(t) = \boldsymbol{a}(t) = (a_1(t), a_2(t), a_3(t))$ are made every slot, where $a_n(t)$ is the amount of newly arriving packets admitted to queue n on slot t, subject to the constraints: $0 \leq a_n(t) \leq \beta_n(t)$ for all t and all $n \in \{1, 2, 3\}$. Any data that is not admitted is dropped (similar to [3][2]). Transmission decisions are made after $\omega(t)$ is observed, and are given by $I(t) = \boldsymbol{r}(t)$, where $\boldsymbol{r}(t) = (r_1(t), r_2(t), r_3(t))$ and is subject to $r_n(t) \in \{0, 2, 4\}$ for all n and and $r_n(t)$ is non-zero for at most one $n \in \{1, 2, 3\}$. The queue dynamics are given by (1) with:

$$\hat{a}_n(k(t), \beta(t), \boldsymbol{a}(t)) = a_n(t)$$
$$\hat{b}_n(k(t), \omega(t), \boldsymbol{r}(t)) = \begin{cases} c r_n(t) 1\{S_n(t) \geq r_n(t)\} & \text{if measure} \\ r_n(t) 1\{S_n(t) \geq r_n(t)\} & \text{else} \end{cases}$$

where $1\{S_n(t) \geq r_n(t)\}$ is an indicator function that is 1 if $S_n(t) \geq r_n(t)$, and is zero else, and $c = 1/2$ is a fraction representing channel measurement overhead. This model is similar to [8][9], with the exception that we have no a-priori knowledge of the channel distributions.

Define the attribute $x_n(t) = a_n(t)$. The goal is to maximize a concave utility function of the time average admitted traffic:

$$\text{Maximize: } \textstyle\sum_{n=1}^{3} g_n(\overline{a}_n) \quad , \quad \text{Subject to: Queue stability}$$

where $\overline{a}_n = \lim_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\{a_n(\tau)\}$, and where $g_n(a) = \log(1 + a)$. This fits the framework with $M = N = 3$, $L = 0$, $y_0(\cdot) = 0$, and $f(\boldsymbol{x}) = -\sum_{n=1}^{3} g_n(x_n)$. We have $x_{min} = 0$ and $x_{max} = 2$. We define $\sigma = 1$, and extend the utility functions concavely over the range $-\sigma \leq a \leq 2 + \sigma$ for the sake of the auxiliary variable optimization:

$$g(a) = \begin{cases} \log(1 + a) & \text{if } 0 \leq a \leq 2 + \sigma \\ -a & \text{if } -\sigma \leq a < 0 \end{cases}$$

The auxiliary variable optimization decouples for all $n \in \{1, 2, 3\}$ into choosing $\gamma_n(t)$ subject to $-\sigma \leq \gamma_n(t) \leq 2 + \sigma$ to minimize $-V g(\gamma_n(t)) + H_n(t)\gamma_n(t)$. The solution is given by:

$$\gamma_n(t) = \begin{cases} -\sigma & \text{if } H_n(t) > V \\ \min\left[\frac{V}{H_n(t)} - 1, 2 + \sigma\right] & \text{if } 0 < H_n(t) \leq V \\ 2 + \sigma & \text{if } H_n(t) \leq 0 \end{cases} \tag{34}$$

The optimal flow control decisions $J(t) = (a_1(t), a_2(t), a_3(t))$ are (for $n \in \{1,2,3\}$):

$$a_n(t) = \begin{cases} \beta_n(t) & \text{if } Q_n(t) \leq H_n(t) \\ 0 & \text{if } Q_n(t) > H_n(t) \end{cases} \quad (35)$$

The $Y_k(\eta(t), \boldsymbol{r}(t), \boldsymbol{\Theta}(t))$ function is given by:

$$Y_k(\eta(t), \boldsymbol{r}(t), \boldsymbol{\Theta}(t)) = -\sum_{i=1}^{3} Q_n(t)\hat{b}_n(k(t), \eta(t), \boldsymbol{r}(t))$$

It can be seen from (34) that $\gamma_n(t) = -\sigma$ whenever $H_n(t) > V$, and hence, by the $H_n(t)$ updates in (14), $H_n(t)$ cannot increase on such a slot. Hence $H_n(t) \leq V + 2 + \sigma$ for all t. It follows from (35) that queue backlog is deterministically bounded as follows: $Q_n(t) \leq V + 4 + \sigma$ for all $n \in \{1,2,3\}$ and all t. This deterministic bound was indeed upheld in the simulations.

We assume channel state vectors $\boldsymbol{S}(t)$ are i.i.d. over slots but correlated in each entry, as follows: Every slot t, $S_1(t)$ is chosen independently according to the distribution:

$$Pr[S_1(t) = 0] = Pr[S_1(t) = 2] = 2/5,\ Pr[S_1(t) = 4] = 1/5$$

Then with probability 0.2 we have: $S_3(t) = S_2(t) = S_1(t)$. With probability 0.8, we have $Pr[S_2(t) = 2] = 1$, and we have $S_3(t)$ distributed independently according to the same distribution as $S_1(t)$. Packets are assumed to arrive independently over slots and channels, with $Pr[\beta_n(t) = 2] = Pr[\beta_n(t) = 0] = 1/2$. We use an exploration probability $\theta = 0.01$ and a measurement loss factor $c = 1/2$, and we run all simulations over 2 million slots.

V. Multi-Hop Diversity Backpressure Routing

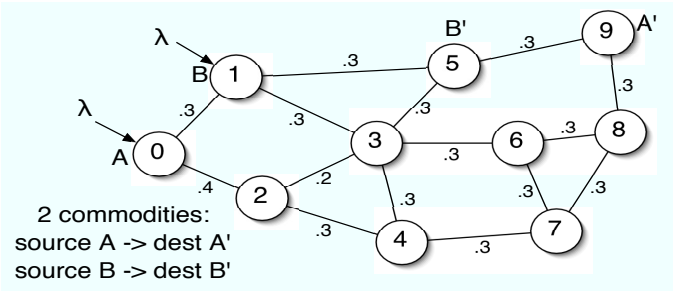

Fig. 5. An illustration of the graph showing link success probabilities for the multi-hop network.

Here we consider the multi-hop network model of [10]. The network of interest is shown in Fig. 5. There are two traffic types: Traffic type A has source node 0 and destination 9. New packets of type A arrive to node 0 according to an i.i.d. Bernoulli process with rate λ_A packets/slot. Traffic type B has source node 1 and destination 5, and this traffic arrives to node 1 according to an i.i.d. Bernoulli process with rate λ_B packets/slot. Every slot, each node chooses a single packet to transmit. This packet is successfully received at each of the neighboring links of the graph with their corresponding link success probabilities, as shown in Fig. 5. For simplicity, we assume the system uses orthogonal channels, so that there is no interference and multiple packets can be received (from different transmitter nodes) at one node on the same slot. Further, we assume all link successes are independent over links and over slots. For example, if node 0 transmits a packet on slot t, then it is successfully received with probability 0.3 at node 1, and independently with probability 0.4 at node 2.

The 2-stage decision structure at each node is as follows: There is no initial $\beta(t)$ information. In the stage-1 decision on slot t, every node chooses whether to transmit a packet of type A or B (remaining idle, or equivalently, transmitting a "null" packet if no actual packets are available). The feedback information $\omega(t)$ represents ACK/NACK feedback that each transmitting node receives about reception on each of the possible outgoing links of the graph. Based on this feedback, the stage-2 decision $\alpha(t)$ chooses which single next-node takes responsibility for each transmitted packet (possibly keeping the packet in the same transmitting node for future transmission if none of the successful receivers are as desirable as the current node). The packet is then deleted from the previous queue, and retained only in the single queue chosen to take responsibility for the packet.

Let $Q_n^{(A)}(t)$ and $Q_n^{(B)}(t)$ be the current number of commodity A and commodity B packets, respectively, in node $n \in \{0, \ldots, 9\}$ on slot t. Note that $Q_9^{(A)}(t) = 0$ and $Q_5^{(B)}(t) = 0$ for all t, since packets that reach their destination are removed from the network. The queueing dynamics for commodity A are:

$$Q_n^{(A)}(t+1) = \max\left[Q_n^{(A)}(t) - \sum_{j=0}^{9} b_{nj}^{(A)}(t), 0\right] + \sum_{i=0}^{9} b_{in}^{(A)}(t) + a_n^{(A)}(t)$$

where $a_n^{(A)}(t)$ is the number of new type A packets that arrive exogenously to node n on slot t, being zero for all $n \neq 0$, and being Bernoulli with rate λ_A if $n = 0$. Further, $b_{ij}^{(A)}(t)$ is the number of type A packets that node i transfers to node j on slot t, being at most 1, and being 1 if and only if node i successfully transmitted a packet to node j, and then decided to choose node j to take responsibility of the packet in the stage-2 decision. Similar dynamics hold for commodity B. Further details on the structure and constraints of $b_{ij}^{(A)}(t)$ and $b_{ij}^{(B)}(t)$ are given in [10].

For simplicity, we consider a problem that only seeks to stabilize the queues, so that there are no attributes $y_l(t)$, $x_m(t)$. In [10] it is shown that the optimal max-weight stage-2 decision is as follows: For each node n that transmits a packet of commodity c on slot t (where $c = A$ or $c = B$), define $\mathcal{S}_n(t)$ as the set of nodes consisting of node n together with all other nodes that successfully received the packet. Then choose the next-hop as the node $j \in \mathcal{S}_n(t)$ that maximizes the *differential backlog* $Q_n^{(c)}(t) - Q_j^{(c)}(t)$, breaking ties arbitrarily. This is a simple decision that requires no knowledge of the success probability information. Thus, the main complexity of the algorithm is in choosing the stage-1 decision of which commodity, A or B, is transmitted. It is shown in [10] that the optimal max-weight stage-1 decision is as follows: For each node n, let $\mathcal{D}_n$ be the set of possible next-hop nodes

(including node n). For each commodity $c \in \{A, B\}$, *rank order* the nodes in $\mathcal{D}_n$ from highest differential backlog to lowest differential backlog, breaking ties arbitrarily so that a specific rank order is obtained. This represents the preference order for next-hop routing. Then for each $c \in \{A, B\}$ and each node n, compute:

$$e^{(c)}(t) \triangleq \sum_{j \in \mathcal{D}_n} \max[Q_n^{(c)}(t) - Q_j^{(c)}(t), 0]\phi_{nj}^{(c)}(t)$$

where $\phi_{nj}^{(c)}(t)$ is the probability that, given node n transmits a commodity c packet, node j successfully receives the packet, but no other nodes with better rank successfully receive the packet. This of course requires knowledge of the joint success probability distribution at all nodes (which may not have product form if successes are correlated, even though in our simulation example we assume independent successes). The values of $e^{(A)}(t)$ and $e^{(B)}(t)$ are then compared, and we transmit commodity A whenever $e^{(A)}(t) \geq e^{(B)}(t)$, and B else. This algorithm is called the Diversity Backpressure Routing Algorithm (DIVBAR) [10]. The $e^{(A)}(t)$ and $e^{(B)}(t)$ values in this case correspond to those in (32) of the general max-weight learning algorithm.

The new max-weight learning implementation of DIVBAR, which does not need probability information, is as follows: First note that "success" events are independent of the packet commodity, and so we do not need to define any exploration events. In principle, we can assume each node transmits a packet every slot (being a null packet if it does not have any actual packets).[5] Each node n keeps a sample-path history $\omega_n(t,1), \ldots, \omega_n(t,W)$ of the sets of successful receivers of node n transmissions over the past W slots. It then computes the following estimates $\hat{e}^{(A)}(t)$ and $\hat{e}^{(B)}(t)$, written below as $\hat{e}^{(c)}(t)$ with $c \in \{A, B\}$:

$$\hat{e}^{(c)}(t) = \frac{1}{W} \sum_{w=1}^{W} \left[\sum_{j \in \mathcal{D}_n} \max[Q_n^{(c)}(t) - Q_j^{(c)}(t), 0] 1_{nj}^{(c)}(w,t) \right]$$

where $1_{nj}^{(c)}(w,t)$ is an indicator function that is 1 if node j successfully received a transmission from node n, but no other nodes with rank order better than j successfully received, in the wth sample of the sample past history.

We now present results for this max-weight learning algorithm along with results for an implementation of the algorithm that assumes *a priori* knowledge of all link success probabilities for comparison. These results were obtained from simulations performed using the Qualnet[6] network simulation environment. The network and channel success probabilities from Fig. 5 were used, and traffic rates of the two flows were assumed to be equal for all simulations, such that $\lambda_A = \lambda_B = \lambda$ packets/second. The time slot duration was set to 1 second, and each simulation run was performed for 1 million slots with throughput and occupancy results averaged over the entire run.

[5]In practice, we do not have to transmit null packets, and we could use the sample-path history of the past W transmissions, which may not be the past W slots if some of these slots did not have transmissions.

[6]http://www.scalable-networks.com

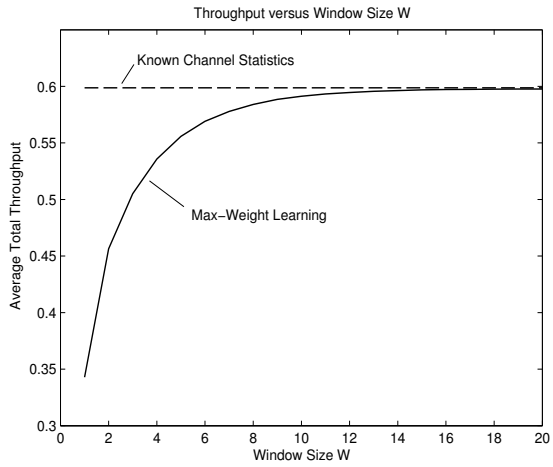

Fig. 6. Average total throughput versus W.

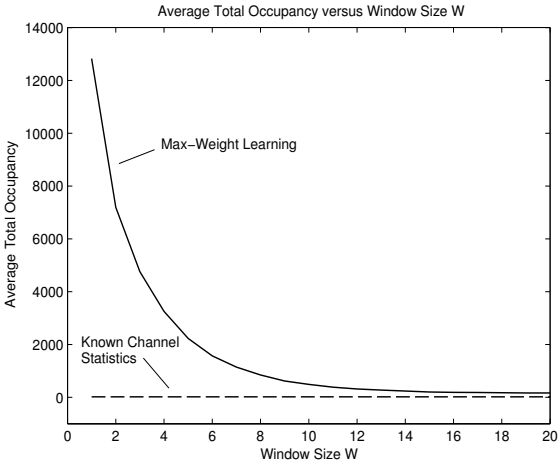

Fig. 7. Average total occupancy versus W. The horizontal asymptote for known channel statistics represents an average total queue backlog, summed over all queues, of 18.8 packets.

First, we consider the scenario with a fixed value of $\lambda = 0.3$ and a varying window size, W. Fig. 6 shows how the max-weight learning implementation approaches the optimal total average throughput of $\lambda * 2 = 0.6$ packets/second as the window size W is increased. In this case, a value of as low as $W = 15$ achieves full throughput. Fig. 7 shows the average total occupancy of all the queues in the network for the max-weight learning algorithm using the same values of W and for the implementation assuming full channel state knowledge. Again, the values approach the known channel statistics case, being $\overline{Q}_{tot} = 18.8$, as W is increased, also achieving virtually the same performance as the algorithm with *a priori* knowledge at $W = 15$.

We also ran simulations of the same scenario using a fixed value of $W = 15$ and simultaneously increasing input rates of both flows from 0 to 0.5 packets/slot. These results are displayed in Figs. 8 and 9. Since using a window size of this value achieves performance virtually equal to the algorithm assuming full knowledge of channel state distributions as shown above, only data from the max-weight learning version

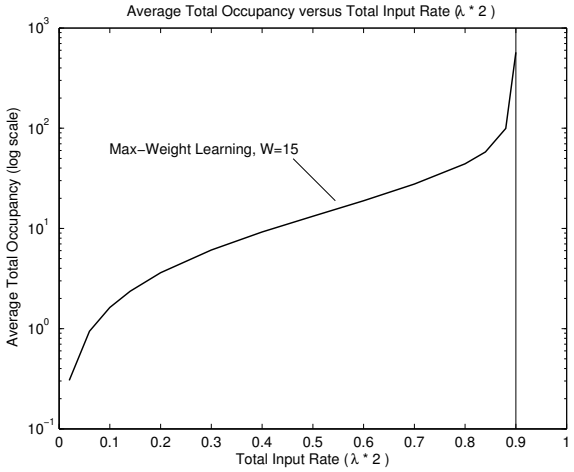

Fig. 8. Average total occupancy versus input rate, $\lambda * 2$.

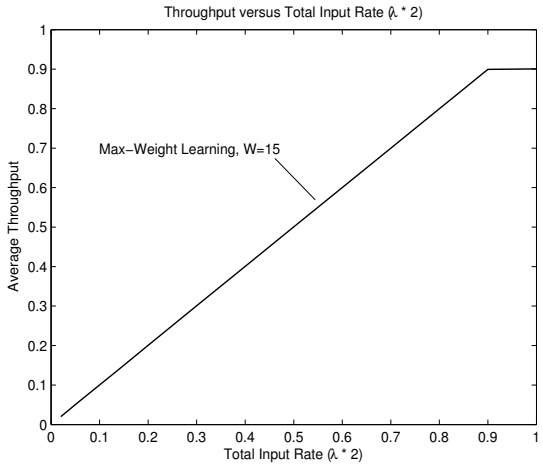

Fig. 9. Average throughput versus total input rate, $\lambda * 2$.

of the algorithm is shown in these two figures.

The average total occupancy of all queues in the network for increasing traffic loads is shown in Fig. 8 (with the y-axis in log scale). Here, queue stability is achieved while the input traffic remains within the network's capacity region of 0.9 packets/second. We can see the edge of this capacity region, however, as congestion approaches the asymptote at $\lambda * 2 = 0.9$. This same limit is also evident in Fig. 9, which plots the average delivered throughput versus the total input rate from the same set of simulations. While input traffic is less than 0.9 packets/second, the network's ability to support the offered traffic load is observed by the average throughput being equal to the average input rate. At $\lambda * 2 = 0.9$, however, we see that the network's capacity region is saturated and achieved throughput levels off. Note that this capacity limit of the network is also intuitive from an examination of the topology in Fig. 5. If we consider the minimum cut across the links $\{(1,5),(3,5),(8,9)\}$, the total of the link success probabilities is equal to 0.9, the same capacity achieved through simulation.

This results in the following interesting observation: The DIVBAR algorithm is designed to maximize throughput over all possible routing and scheduling strategies that send packets with reception success as described above, but does not necessarily achieve the (unsolved) *information theoretic* network capacity, which can also optimize over all network coding strategies on the same problem structure. However, in this case there is a min-cut that limits the symmetric throughput to $\lambda_A = \lambda_B \leq 0.45$ packets/slot (for a total throughput of 0.9) for any information theoretic strategy.[7] However, by analyzing the topology and probabilities for this example, it can be shown that there exists a 2-commodity flow that uses only pure routing and scheduling (without network coding) that achieves $\lambda_A = \lambda_B = 0.45$. Thus, the DIVBAR algorithm also achieves this and so for this example it achieves the symmetric information-theoretic capacity.

[7]We note that a remark in [10] incorrectly stated the capacity as $\lambda_A = \lambda_B = 0.455$ packets/slot, rather than 0.45 packets/slot.

VI. Conclusion

This work extends the drift-plus-penalty framework for stochastic network optimization to a context with 2-stage decisions and unknown distributions that govern the stochastics. This is useful in a variety of contexts, including transmission scheduling for wireless networks in unknown environments and with unknown channels. The learning algorithms developed here are based on estimates of expected max-weight functionals, and are much more efficient than algorithms that would attempt to learn the complete probability distributions associated with the system. Our analysis provides explicit bounds on the deviation from optimality, and the corresponding backlog tradeoffs, in terms of the sample size W and the control parameter V.

Simulations were provided for an example 3-user wireless system. Our simulations include a variation of the max-weight learning algorithm that samples at any measurement time. This policy is more difficult to analyze and provides slightly worse performance in the simulations considered here. However, it uses more recent channel realizations and thus may provide *better* performance for mobile networks where the channel probability distributions can change over time. Indeed, our sampling method here assumes each of the W samples is identically distributed. In networks with high mobility, one might place a limit on the amount of samples W that are assumed to be relevant to the existing topology. This creates an $O(1/\sqrt{W})$ performance gap for mobile networks due to the inability to adaptively estimate max-weight functionals for a changing channel state distribution.

Appendix A — Proof of Lemma 1

Here we prove Lemma 1. Squaring the $Z_l(t)$ update equation (13) gives:

$$\begin{aligned}(1/2)Z_l(t+1)^2 &\leq (1/2)(Z_l(t)+y_l(t))^2 \\ &= (1/2)Z_l(t)^2 + (1/2)y_l(t)^2 + Z_l(t)y_l(t)\end{aligned}$$

Similarly, squaring the $H_m(t)$ update equation (14) gives:

$$\frac{H_m(t+1)^2}{2} = \frac{H_m(t)^2}{2} + \frac{1}{2}(\gamma_m(t) - x_m(t))^2 + H_m(t)(\gamma_m(t) - x_m(t))$$

Finally, using the fact that $([Q-b,0]+a)^2 \leq Q^2+b^2+a^2+2Q(a-b)$ for non-negative numbers Q, b, a, and squaring (1) gives:

$$\frac{1}{2}Q_n(t+1)^2 \leq \frac{1}{2}Q_n(t)^2 + \frac{1}{2}(b_n(t)^2 + a_n(t)^2) + Q_n(t)(a_n(t) - b_n(t))$$

Summing the above over $l \in \{1,\ldots,L\}$, $m \in \{1,\ldots,M\}$, $n \in \{1,\ldots,N\}$ gives:

$$L(\mathbf{\Theta}(t+1)) - L(\mathbf{\Theta}(t)) \leq B(t) + \sum_{l=1}^{L} Z_l(t)y_l(t) + \sum_{m=1}^{M} H_m(t)(\gamma_m(t) - x_m(t)) + \sum_{n=1}^{N} Q_n(t)(a_n(t) - b_n(t))$$

where $B(t)$ is defined:

$$B(t) \triangleq \frac{1}{2}\sum_{l=1}^{L} y_l(t)^2 + \frac{1}{2}\sum_{m=1}^{M}(\gamma_m(t) - x_m(t))^2 + \frac{1}{2}\sum_{n=1}^{N}(b_n(t)^2 + a_n(t)^2)$$

The boundedness assumptions in Section II-A ensure that $\mathbb{E}\{B(t)|\mathbf{\Theta}(t)\} \leq B$ for some finite constant B. Taking conditional expectations of the above, given $\mathbf{\Theta}(t)$, proves the result.

APPENDIX B — PROOF OF THEOREM 2

To prove Theorem 2, fix time t and define $\Omega(\mathbf{\Theta}(t))$ as follows:

$$\Omega(\mathbf{\Theta}(t)) \triangleq \mathbb{E}\left\{RHS(\mathbf{\Theta}(t), \hat{k}(t), \hat{\alpha}(t), \boldsymbol{\gamma}^{mw}(t)) \mid \mathbf{\Theta}(t)\right\} - \mathbb{E}\left\{RHS(\mathbf{\Theta}(t), k^{mw}(t), \alpha^{mw}(t), \boldsymbol{\gamma}^{mw}(t)) \mid \mathbf{\Theta}(t)\right\}$$

where $\hat{k}(t)$, $\hat{\alpha}(t)$ are the actions taken using the $\hat{e}_k(t)$ estimates of Approach 1, and $k^{mw}(t)$, $\alpha^{mw}(t)$ are the ideal max-weight actions based on perfectly known $e_k(t)$ values (for simplicity, assume "max-weight" actions that achieve the appropriate minimum exist, else we can take actions arbitrarily close to the infimum). Because these right-hand sides differ only in terms comprising the $e_k(t)$ expression defined in (32), we have:

$$\Omega(\mathbf{\Theta}(t)) = \mathbb{E}\left\{e_{\hat{k}(t)}(t) \mid \mathbf{\Theta}(t)\right\} - \min_{k\in\mathcal{K}}[e_k(t)]$$

where the expectation on the right hand side is over the decision $\hat{k}(t) = \arg\min_{k\in\mathcal{K}}[\hat{e}_k(t)]$. Now for each $k \in \mathcal{K}$, define $\delta_k(t) \triangleq \hat{e}_k(t) - e_k(t)$. We thus have:

$$\begin{aligned}
&\mathbb{E}\left\{e_{\hat{k}(t)}(t) \mid \mathbf{\Theta}(t)\right\} \\
&= \mathbb{E}\left\{\hat{e}_{\hat{k}(t)}(t) - \delta_{\hat{k}(t)}(t) \mid \mathbf{\Theta}(t)\right\} \\
&\leq \mathbb{E}\left\{\hat{e}_{\hat{k}(t)}(t) \mid \mathbf{\Theta}(t)\right\} + \mathbb{E}\left\{\max_{k\in\mathcal{K}}[-\delta_k(t)] \mid \mathbf{\Theta}(t)\right\} \\
&= \mathbb{E}\left\{\min_{k\in\mathcal{K}}[\hat{e}_k(t)] \mid \mathbf{\Theta}(t)\right\} + \mathbb{E}\left\{\max_{k\in\mathcal{K}}[-\delta_k(t)] \mid \mathbf{\Theta}(t)\right\} \\
&= \mathbb{E}\left\{\min_{k\in\mathcal{K}}[e_k(t) + \delta_k(t)] \mid \mathbf{\Theta}(t)\right\} + \\
&\quad \mathbb{E}\left\{\max_{k\in\mathcal{K}}[-\delta_k(t)] \mid \mathbf{\Theta}(t)\right\} \\
&\leq \min_{k\in\mathcal{K}}[e_k(t)] + \mathbb{E}\left\{\max_{k\in\mathcal{K}}[\delta_k(t)] + \max_{k\in\mathcal{K}}[-\delta_k(t)] \mid \mathbf{\Theta}(t)\right\} \\
&\leq \min_{k\in\mathcal{K}}[e_k(t)] + \sum_{k=1}^{K}\mathbb{E}\left\{|\delta_k(t)| \mid \mathbf{\Theta}(t)\right\}
\end{aligned}$$

It follows that:

$$\begin{aligned}
\Omega(\mathbf{\Theta}(t)) &\leq \sum_{k=1}^{K}\mathbb{E}\left\{|\delta_k(t)| \mid \mathbf{\Theta}(t)\right\} \\
&= \sum_{k=1}^{K}\mathbb{E}\left\{|\hat{e}_k(t) - e_k(t)||\mathbf{\Theta}(t)\right\}
\end{aligned}$$

Note that $\Omega(\mathbf{\Theta}(t))$ corresponds to the desired inequality (17), and hence it suffices to bound $\mathbb{E}\{|\hat{e}_k(t) - e_k(t)||\mathbf{\Theta}(t)\}$. Unfortunately, the W samples that generate $\hat{e}_k(t)$ are not necessarily conditionally i.i.d. given $\mathbf{\Theta}(t)$, but they *are* conditionally i.i.d. given $\mathbf{\Theta}(t - T_k(t))$, where $t - T_k(t)$ is the time of the earliest sampled k-inferred exploration event. Note that $T_k(t)$ is a sum of W geometric random variables, and has mean at most $W\tilde{K}/\theta$. We have:

$$\begin{aligned}
|\hat{e}_k(t) - e_k(t)| \leq\ & |\hat{e}_k(t) - \hat{e}_k(t - T_k(t))| \\
& + |\hat{e}_k(t - T_k(t)) - e_k(t - T_k(t))| \\
& + |e_k(t - T_k(t)) - e_k(t)|
\end{aligned}$$

The expectations of the first and last terms are proportional to the expectation of $T_k(t)$ multiplied by a constant $D_2/2$ that depends on the worst case moments of queue backlog change over one slot. The expectation of the middle term is just the expected difference between an expectation and an estimate based on W i.i.d. samples, and hence is $O(1/\sqrt{W})$. Specifically, due to the uniform bounds on first and second moments for all k, we can show for all $k \in \{1,\ldots,K\}$ [1]:

$$\begin{aligned}
&\mathbb{E}\{|\hat{e}_k(t - T_k(t)) - e_k(t - T_k(t))||\mathbf{\Theta}(t)\} \\
&\leq \frac{D_1}{\sqrt{W}}\left[V + \sum_{n=1}^{N} Q_n(t) + \sum_{l=1}^{L} Z_l(t) + \sum_{m=1}^{M}|H_m(t)|\right]
\end{aligned}$$

where D_1 is a constant independent of V, W, and queue sizes and depends on the second moments of the processes. We thus have:

$$\begin{aligned}
\Omega(\mathbf{\Theta}(t)) \leq\ & \tilde{C}(t) \\
& + \frac{D_1 K}{\sqrt{W}}\left[V + \sum_{n=1}^{N} Q_n(t) + \sum_{l=1}^{L} Z_l(t) + \sum_{m=1}^{M}|H_m(t)|\right]
\end{aligned}$$

where $\tilde{C}(t)$ is defined:

$$\tilde{C}(t) \triangleq \sum_{k=1}^{K} [|\hat{e}_k(t) - \hat{e}_k(t - T_k(t))| + |e_k(t - T_k(t)) - e_k(t)|]$$

Further, the unconditional expectation of $\tilde{C}(t)$ can be shown to be bounded by: $\mathbb{E}\{\tilde{C}(t)\} \leq C \triangleq D_2 K \tilde{K} W/\theta$. We thus have that (17) of Assumption A1 holds with C defined as above and $\epsilon_V = \epsilon_Q = \epsilon_Z = \epsilon_H = D_1 K/\sqrt{W}$.

REFERENCES

[1] M. J. Neely. Max weight learning algorithms with application to scheduling in unknown environments. *arXiv:0902.0630v1*, Feb. 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] M. J. Neely, E. Modiano, and C. Li. Fairness and optimal stochastic control for heterogeneous networks. *Proc. IEEE INFOCOM*, pp. 1723-1734, March 2005.

[4] 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.

[5] 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.

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

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

[8] C. Li and M. J. Neely. Energy-optimal scheduling with dynamic channel acquisition in wireless downlinks. *IEEE Transactions on Mobile Computing*, vol. 9, no. 4, pp. 527-539, April 2010.

[9] A. Gopalan, C. Caramanis, and S. Shakkottai. On wireless scheduling with partial channel-state information. *IEEE Transactions on Information Theory*, vol. 58, no. 1, pp. 403-420, Jan. 2012.

[10] M. J. Neely and R. Urgaonkar. Optimal backpressure routing in wireless networks with multi-receiver diversity. *Ad Hoc Networks (Elsevier)*, vol. 7, no. 5, pp. 862-881, July 2009.

[11] 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.

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

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

Michael J. Neely received B.S. degrees in both Electrical Engineering and Mathematics from the University of Maryland, College Park, in 1997. He was then awarded a Department of Defense NDSEG Fellowship for graduate study at the Massachusetts Institute of Technology, where he received an M.S. degree in 1999 and a Ph.D. in 2003, both in Electrical Engineering. In 2004 he joined the faculty of the Electrical Engineering Department at the University of Southern California, where he is currently an Assistant Professor. His research is in the area of stochastic network optimization for wireless and ad-hoc mobile networks. He received the NSF Career Award in January 2008 and the Viterbi School of Engineering Junior Research Award in 2009. Michael is a member of Tau Beta Pi and Phi Beta Kappa.

Scott T. Rager is a PhD candidate in the Computer Science and Engineering Department at Penn State University. He received a B.A. in Physics from the Slippery Rock University of Pennsylvania and a B.S. in Computer Engineering from Penn State both in 2009 after completing a cooperative dual-degree undergraduate program. He is presently a member of the Network and Security Research Center at Penn State, and his current research interests include mobile ad hoc networks and distributed protocol designs.

Thomas F. La Porta is a Distinguished Professor in the Computer Science and Engineering Department at Penn State. He received his B.S.E.E. and M.S.E.E. degrees from The Cooper Union, New York, NY, and his Ph.D. degree in Electrical Engineering from Columbia University, New York, NY. He joined Penn State in 2002. He is the Director of the Networking and Security Research Center at Penn State. Prior to joining Penn State, Dr. La Porta was with Bell Laboratories since 1986. He was the Director of the Mobile Networking Research Department in Bell Laboratories, Lucent Technologies where he led various projects in wireless and mobile networking. He is an IEEE Fellow, Bell Labs Fellow, received the Bell Labs Distinguished Technical Staff Award in 1996, and an Eta Kappa Nu Outstanding Young Electrical Engineer Award in 1996. He also won a Thomas Alva Edison Patent Awards in 2005 and 2009. His research interests include mobility management, signaling and control for wireless networks, security for wireless systems, mobile data systems, and protocol design.

Dr. La Porta was the founding Editor-in-Chief of the IEEE Transactions on Mobile Computing and served as Editor-in-Chief of IEEE Personal Communications Magazine. He was the Director of Magazines for the IEEE Communications Society and was on its Board of Governors for three years. He has published numerous papers and holds 35 patents.