Energy Optimal Control for Time Varying Wireless Networks

Michael J. Neely

***Abstract*— We develop a dynamic control strategy for minimizing energy expenditure in a time varying wireless network with adaptive transmission rates. The algorithm operates without knowledge of traffic rates or channel statistics, and yields average power that is arbitrarily close to the minimum possible value achieved by an algorithm optimized with complete knowledge of future events. Proximity to this optimal solution is shown to be inversely proportional to network delay. We then present a similar algorithm that solves the related problem of maximizing network throughput subject to peak and average power constraints. The techniques used in this paper are novel and establish a foundation for stochastic network optimization.**

***Index Terms*— Stochastic Optimization, Queueing Analysis, Ad-Hoc Networks, Distributed Algorithms, Mobile Networks**

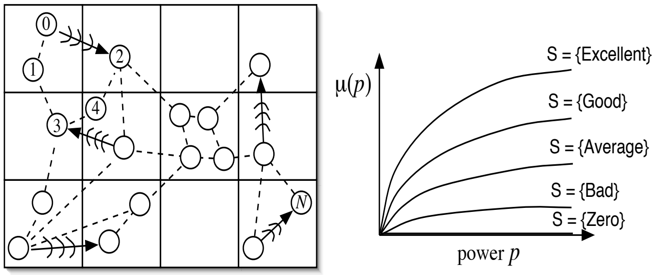

Fig. 1. A cell-partitioned wireless network, and an example set of rate-power curves for 5 different channel states.

I. Introduction

Wireless systems operate over time varying channels that are influenced by random environmental conditions, wireless fading, and power allocation decisions. To improve performance and meet the ever increasing demand for high throughput and low delay, modern wireless devices are designed with channel monitoring capabilities and rate adaptive technology. Such technology is currently being implemented for cellular communication with High Data Rate (HDR) services [3], and the ability to measure and react to channel information is expected to improve significantly.[1] It is of central importance to develop control strategies that take maximum advantage of this information to improve network performance and energy efficiency.

In this paper, we develop throughput optimal control strategies that conform to peak power constraints while minimizing average power expenditure. This design goal is crucial in all modern wireless scenarios, regardless of whether transmissions take place at a basestation, a hand-held unit, or at a node within an ad-hoc sensor network. Indeed, peak power constraints are important in systems with fixed hardware saturation levels or external environment regulations, while average power levels are important to extend network lifetime in systems with limited energy resources.

Here, we consider an ad-hoc network with N nodes and L wireless links, as shown in Fig. 1. We assume a slotted structure with slots equal to 1 time unit. Packets randomly arrive to the network every timeslot and must be delivered to their destinations, perhaps by routing over multi-hop paths. The transmission rates of each data link are determined every timeslot by link channel conditions and network power allocation decisions according to an L-dimensional rate function $\vec{\mu}(\vec{P}(t), \vec{S}(t))$, where $\vec{P}(t)$ is a vector of power allocations and $\vec{S}(t)$ is a vector of parameters describing the current channel conditions. For most of this paper, we assume that all nodes maintain the same locations relative to one another for the duration of the network operation. Although the network topology remains fixed in this scenario, link conditions may vary dramatically due to environmental effects, local mobility, or wireless fading. Extensions to networks with arbitrary mobility patterns are developed in Section VI.

Power vectors are restricted to a compact set Π of acceptable power allocations, so that $\vec{P}(t) \in \Pi$ for all t. The set Π includes the peak power constraints for each node together with any additional constraints the network might impose on instantaneous transmissions. All of our results are presented for general power sets Π and general rate functions $\vec{\mu}(\vec{P}, \vec{S})$. An example of concave rate-power curves for one data link with a discrete set of possible channel states is shown in Fig. 1. Such curves might also depend on the signal to interference ratio at the intended receiver, so that $\mu_l(\vec{P}, \vec{S})$ for a given link l is determined by the full vector of power allocations and channel states [22] [12] [23]. However, to simplify the multiple access control layer while capturing the geographic structure and interference properties of ad-hoc networks, we focus our implementation examples on a *cell partitioned network model*.

Under this model, the network region is divided into cells, each containing a distinct set of nodes. Specifically, we define $cell(n)$ as the cell of each node $n \in \{1, \ldots, N\}$, and define $tran(l)$ and $rec(l)$ as the transmitting and receiving nodes associated with a given wireless link $l \in \{1, \ldots, L\}$. We assume that each cell can support at most one active link transmission per timeslot, and that nodes can transmit only to other nodes in the same cell or in adjacent cells. That is, the feasible power set Π includes the constraint that if $P_l > 0$ for some link l, then $P_{\tilde{l}} = 0$ for all links $\tilde{l} \neq l$ such that

Michael J. Neely is with the Department of Electrical Engineering, University of Southern California, Los Angeles, CA 90089 USA (email: mjneely AT usc.edu, web: http://www-rcf.usc.edu/∼mjneely).

This work was presented in part at the IEEE INFOCOM conference, March 2005, and at the USC CSI Research Review, March 2004.

[1] Indeed, it is claimed in [4] that channel measurements can be obtained almost as often as the symbol rate of the link in certain local area wireless networks.

$cell(tran(\tilde{l})) = cell(tran(l))$. We further assume that the transmission rate of each link depends only on the channel state and the power allocated to that link, so that $\vec{\mu}(\vec{P}, \vec{S}) = (\mu_1(P_1, S_1), \ldots, \mu_L(P_L, S_L))$. This structure arises if nodes in neighboring cells transmit over orthogonal frequency bands. In this way, if a node is transmitting then it cannot concurrently receive from nodes within the same cell, and any data it receives from adjacent cells must be on a different frequency band. If the cell structure is rectinlinear, it is well known that only 9 orthogonal subbands are required to ensure all neighboring cells have distinct subbands, and this number can be reduced to 7 if cells are arranged according to a hexagonal pattern. While the cell partitioned structure is not critical to our analysis, it simplifies exposition and allows scheduling decisions to be decoupled cell by cell. Relaxations of the model or further restrictions on power assignment can easily be incorporated by modifying the set constraint Π or the rate function $\vec{\mu}(\vec{P}, \vec{S})$.

The goal of this paper is to develop a power allocation and routing algorithm that supports all incoming traffic while minimizing average power expenditure. We develop a robust policy that does not require knowledge of input rates or channel probabilities yet uses a total power expenditure that is arbitrarily close to the minimum average power expended by a system optimized with complete knowledge of future events. Distance to this minimum power level is controlled by a parameter V effecting an explicit tradeoff in average end-to-end network delay.

Previous work in the area of power allocation for wireless systems can be categorized into static optimization solutions [5]-[12] and dynamic control algorithms [14]-[13]. In [5], a utility optimization problem is presented for a static wireless downlink, and pricing schemes are developed to enable power allocations to converge to a fair allocation vector. Linear programming, geometric programming, and other convex optimization methods are considered in [6]-[10] for routing and power allocation problems in wireless systems and sensor networks. Such techniques rely on the mathematical theory of Lagrangian duality (see, for example, [28]). This theory was applied in the landmark paper [29] to develop mechanisms for optimal static resource allocation in a non-wireless network. We note that convex optimization approaches traditionally yield single-operating point solutions, which may not be well suited to cases when optimal networking involves *dynamic allocation* of resources. Indeed, in [12] it is shown that minimizing energy in a static ad-hoc network with interference involves the computation of a *periodic transmission schedule*, yielding dramatic improvements over any fixed resource allocation. A similar scheduling problem is shown to be NP-complete in [11].

Prior work in the area of *stochastic* optimization and dynamic control for wireless networks considers much smaller systems with more a-priori statistical information, including [23] [24] [25] for energy efficient scheduling in single queue systems, and [26] [27] for multi-user downlinks with infinite backlog. A downlink with randomly arriving traffic and peak and average power constraints is considered in [13] using a theory of Lyapunov drift, although the algorithm requires perfect knowledge of channel probabilities in order to meet the average power requirement. Lyapunov theory can be used to design stabilizing power allocation and routing algorithms that do not require knowledge of arrival rates or channel statistics in cases where there are only peak power constraints on the wireless devices [22]. Historically, Lyapunov theory has been extremely useful in the development of stable queue control policies for radio networks and switching systems [14]-[22]. However, there was previously no Lyapunov method for performing queueing network optimization (such as stabilizing a network with minimum average power).

In this paper, we develop a simple Lyapunov drift technique that enables system stability and performance optimization to be achieved simultaneously [2] [1] [30]. The technique extends the Lyapunov methods of [14]-[22] and bridges the gap between convex optimization theory and stochastic queueing control problems. We note that alternative approaches to stochastic network optimization have recently appeared in [31] [32] [33] using fluid model transformations and/or stochastic gradient theory. Our Lyapunov technique is similar to the notion of a stochastic gradient (see Chapters 4-5 of [2] for a comparison between static gradient search methods and Lyapunov scheduling), although it was developed from a queueing stability perspective and yields explicit bounds on average power and delay.

For simplicity of exposition and to highlight the issues of power allocation, in the first half of this paper we consider only single-hop networks with no routing. The paper is organized as follows: In the next section we consider a motivating example of a 2-user wireless downlink. In Section III we develop a control policy for minimizing average power for one-hop networks. In Section IV we treat a related problem of maximizing throughput subject to peak and average power constraints (for cases when traffic is either supportable or insupportable). Extensions to multi-hop networks and mobile networks are treated in Sections V and VI, and simulations are presented in Section VII.

II. A Simple Example

To illustrate the decisions involved in energy-optimal scheduling, we consider the following example of a two-queue wireless downlink, where a single node (labeled 'node 0') transmits data to two different stations over downlink channels 1 and 2 (as in Fig. 1 in the case when only node 0 is active). The system operates in slotted time, and every slot the channel states are measured, power allocation decisions are made, and new arrivals are queued according to their destinations.

Let $U_1(t)$ and $U_2(t)$ represent the current backlog queued for transmission to destinations 1 and 2, respectively, and consider the decision of whether or not to allocate power to channel 1. Clearly no power should be allocated if $U_1(t) = 0$. When $U_1(t) > 0$, we must decide whether to allocate power on the current slot or wait for a more energy-efficient future channel state. In this example, we consider only ON/OFF power constraints and assume that either no power is allocated to any channel, or full power of 1 Watt is allocated to either channel 1 or channel 2. Link conditions for each channel 1

	t	0	1	2	3	4	5	6	7	8
Arrivals	$A_1(t)$	3	0	3	0	0	1	0	1	0
	$A_2(t)$	2	0	1	0	1	1	0	0	0
Channels	$S_1(t)$	G	G	M	M	G	G	M	M	G
	$S_2(t)$	M	M	B	M	B	M	B	G	B
Max $U_i\mu_i$	$U_1(t)$	0	$\boxed{3}$	0	$\boxed{3}$	$\boxed{1}$	0	1	1	$\boxed{2}$
Policy	$U_2(t)$	0	2	$\boxed{2}$	2	2	$\boxed{3}$	$\boxed{2}$	$\boxed{1}$	0
Better	$U_1(t)$	0	3	3	6	$\boxed{6}$	$\boxed{3}$	1	1	$\boxed{2}$
Choices	$U_2(t)$	0	2	2	$\boxed{3}$	1	2	3	$\boxed{3}$	0

Fig. 2. An example set of arrivals, channel conditions, and queue backlogs for a two queue wireless downlink under two different scheduling algorithms, illustrating the power efficiency gains enabled by having full knowledge of future arrivals and channel states.

and 2 vary between '$Good$,' '$Medium$,' and 'Bad' states:

$$(P_1(t), P_2(t)) \in \Pi \triangleq \{(0,0), (1,0), (0,1)\}$$

$$S_1(t), S_2(t) \in \{G, M, B\}$$

Assume identical rate functions for $i = 1, 2$, given by:

$$\mu_i(0, S_i) = 0 \text{ units/slot} \quad \text{for all } S_i \in \{G, M, B\}$$
$$\mu_i(1, G) = 3, \mu_i(1, M) = 2, \mu_i(1, B) = 1 \text{ (units/slot)}$$

That is, a link can transmit 3 units of data in the '$Good$' state, 2 units in the '$Medium$' state, and 1 in the 'Bad' state.

Let $A_1(t)$ and $A_2(t)$ represent the number of new data units arriving during slot t and destined for nodes 1 and 2, respectively. Queueing dynamics proceed according to the equation:

$$U_i(t+1) = \max[U_i(t) - \mu_i(P_i(t), S_i(t)), 0] + A_i(t)$$

Suppose arrivals $A_i(t)$ and channel states $S_i(t)$ for the first 9 timeslots $t \in \{0, \ldots, 8\}$ are as given in Fig. 2, and consider the policy of allocating power to the channel with the largest rate-backlog product $U_i(t)\mu_i(1, S_i(t))$. This policy can be shown to stabilize the system whenever possible [15] [19] [21], although it is not necessarily energy-efficient. According to the figure, both queues are empty at time $t = 0$ when arrivals enter the system according to vector $(A_1(0), A_2(0)) = (3, 2)$, resulting in a backlog vector $(U_1(1), U_2(1)) = (3, 2)$ at the beginning of slot 1. Because the channel states at slot 1 are given by $(S_1(1), S_2(1)) = (G, M)$, the rate-backlog indices for channels 1 and 2 at slot 1 are given by $U_1(t)\mu_1(1, S_1(t)) = 9$, $U_2(t)\mu_2(1, S_2(t)) = 4$, so that the Max $U_i\mu_i$ policy places full power to channel 1 (as indicated by the boxed values in the figure).

Because there were no new arrivals during slot 1, the resulting backlog vector at time $t = 2$ is given by $(U_1(t), U_2(t)) = (0, 2)$, as shown in the figure. The policy proceeds by expending 1 Watt of power for time $t \in \{1, \ldots, 8\}$, and the scheduling decision at slot $t = 8$ will leave the system again empty at time $t = 9$. If the same arrival and channel patterns were extended periodically every 9 timeslots, the Max $U_i\mu_i$ policy would allocate 1 Watt of power 8 timeslots out of every 9, yielding a time average power consumption of $P_{av} = 8/9$ Watt. Similar power consumption levels are observed when the policy is simulated for random arrivals and channel states with the same steady state distributions as this example (see Section VII).

Now consider the alternate policy of waiting until slot 3 to allocate power, and then making decisions as shown in the figure. These decisions also leave the system empty at slot 9, but yield an average power expenditure of $P_{av} = 5/9$ Watt over the 9 slot interval. Average power can be further reduced if channel states and arrivals are extended periodically (or probabilistically) over the infinite time horizon, and it can be shown that the minimum average power required to stabilize such a system is given by $P_{av}^* = 0.518$.

The above example illustrates the energy gains available by more intelligent scheduling. In cases where power can be allocated as a continuous variable, more complex decisions are involved: Should we exploit better channel states by transmitting at higher data rates with the same power level, or by transmitting at the same data rate with reduced power? In the next section, we develop a simple decision making strategy that does not require knowledge of future events, traffic rates, or channel statistics, yet yields an average power expenditure that is arbitrarily close to optimal.

III. SINGLE HOP NETWORKS

Consider the wireless network of Fig. 1 with N nodes and L links, where each link corresponds to a directed transmission from one node to another. Packets randomly arrive to the system and are queued according to their destinations. This is a single-hop network, and hence incoming data is associated with a particular transmission link $l \in \{1, \ldots, L\}$ and is assumed to leave the network once it is transmitted. Let $A_l(t)$ represent the amount of bits arriving for transmission over link l during slot t, and let $U_l(t)$ represent the current queue backlog (or 'unfinished work') in queue l. Let $\vec{S}(t)$ and $\vec{P}(t)$ represent the L-dimensional vectors of channel states and power allocations. In vector notation, the queueing dynamics are:

$$\vec{U}(t+1) = \max[\vec{U}(t) - \vec{\mu}(\vec{P}(t), \vec{S}(t)), 0] + \vec{A}(t) \qquad (1)$$

where $\vec{\mu}(\vec{P}, \vec{S})$ is the rate function associated with the given physical layer modulation and coding strategies used for wireless communication.

We assume that there are a finite number of channel state vectors $\vec{S}$, and that $\vec{\mu}(\vec{P}, \vec{S})$ is a continuous function of the power vector $\vec{P}$ for each channel state $\vec{S}$.[2] Every timeslot a power vector $\vec{P}(t)$ is chosen in reaction to queue backlog and current channel conditions, subject to the constraint that $\vec{P}(t) \in \Pi$ for all t, where Π is a compact set of acceptable power vectors. Throughout this paper, we use these general rate functions and set constraints to present our main results. However, in all examples of distributed implementation, we assume the rate function has the structure: $\vec{\mu}(\vec{P}, \vec{S}) = (\mu_1(P_1, S_1), \ldots, \mu_L(P_L, S_L))$. Further, in our examples we assume that Π consists of all vectors $\vec{P}$ satisfying the cell-partition constraint (i.e., that if $P_l > 0$ for some link l, then $P_{\tilde{l}} = 0$ for all $\tilde{l} \neq l$ satisfying $cell(tran(l)) = cell(tran(\tilde{l}))$),

[2] Our results hold more generally for any (potentially discontinuous) rate-power curve that satisfies the *upper semi-continuity property* [2].

and such that each entry P_l is limited by a peak value P_{peak} according to either the continuous power constraint $0 \leq P_l \leq P_{peak}$ or the discrete ON/OFF power constraint $P_l \in \{0, P_{peak}\}$.

A. Minimum Power For Stability

Here we characterize the minimum average power required to stabilize the system. We begin with a precise definition of stability in terms of the *overflow function* $g(M)$ associated with a queue with unfinished work process $U(t)$:

$$g(M) \triangleq \limsup_{t\rightarrow\infty} \frac{1}{t}\sum_{\tau=0}^{t} Pr[U(\tau) > M]$$

The function $g(M)$ represents the largest limiting fraction of time the unfinished work is above the value M.

Definition 1: A queue with unfinished work process $U(t)$ is *stable* if $g(M) \rightarrow 0$ as $M \rightarrow \infty$. A *network of queues is stable* if all individual queues are stable.

In the special case when queue backlog evolves according to an ergodic Markov chain with a countably infinite state space, then this notion of stability is equivalent to the existence of a steady state probability distribution for the chain. However, the above stability definition is more general in that it does not require a countably infinite state space, nor does it require ergodicity. A more detailed discussion of stability issues is given in [34] [35] [2].

Assume that inputs and channel processes are ergodic with arrival rates $\vec{\lambda} = (\lambda_l)$ and channel probabilities $\pi_{\vec{S}}$. In [2], the network capacity region Λ is defined as the closure of the set of all rate vectors stabilizable under some power allocation algorithm that conforms to the power constraint $\vec{P}(t) \in \Pi$. The following theorem specifies the minimum average power required for network stability, among the class of all algorithms with complete knowledge of future events.

Theorem 1: (Minimum Power for Stability) If the network is stabilizable (so that $\vec{\lambda} \in \Lambda$), the minimum power required for stability is given by P_{av}^*, where P_{av}^* is the solution to the following nonlinear optimization problem (defined in terms of auxiliary probability variables $\alpha_k^{\vec{S}}$ and power vectors $\vec{P}_k^{\vec{S}}$ for all $\vec{S}$ and for $k \in \{1, 2, \ldots\}$):

$$\begin{array}{ll}\text{Minimize:} & P_{av} = \sum_{\vec{S}}\pi_{\vec{S}}\sum_{k=1}^{\infty}\alpha_k^{\vec{S}}\vec{1}'\vec{P}_k^{\vec{S}} \\ \text{Subject to:} & \vec{\mu}_{av} \triangleq \sum_{\vec{S}}\pi_{\vec{S}}\sum_{k=1}^{\infty}\alpha_k^{\vec{S}}\vec{\mu}(\vec{P}_k^{\vec{S}}, \vec{S}) \geq \vec{\lambda} \\ & \vec{P}_k^{\vec{S}} \in \Pi\,, \alpha_k^{\vec{S}} \geq 0 \quad \text{for all } k, \vec{S} \\ & \sum_{k=1}^{L+2}\alpha_k^{\vec{S}} = 1 \quad \text{for all } \vec{S}\end{array}$$

Further, the value of P_{av}^* is unchanged if the optimization problem above is restricted to use at most $L + 2$ auxiliary variables $\alpha_k^{\vec{S}}$ and $\vec{P}_k^{\vec{S}}$ for any channel state $\vec{S}$.

The theorem indicates that minimum power for stability is achieved among the class of stationary policies that measure the current channel state $\vec{S}(t)$ and then randomly allocate a power vector $\vec{P}_k^{\vec{S}}$ with probability $\alpha_k^{\vec{S}}$. This minimum is attained by a particular policy because the number of channel states is finite and the power set Π is compact. The value of P_{av}^* is the resulting average power of this stationary policy, and $\vec{\mu}_{av}$ is the resulting time average transmission rate vector. This is expressed in the following corollary to Theorem 1.

Corollary 1: If channel states $\vec{S}(t)$ are i.i.d. over slots, minimum power for stability is given by the value P_{av}^*, minimized over the class of all stationary randomized algorithms that make decisions based only on the current channel state $\vec{S}(t)$, and yielding for all timeslots t:

$$\sum_l \mathbb{E}\left\{P_l(t) \mid \boldsymbol{H}(t)\right\} = P_{av}^*$$

$$\vec{\mu}_{av} \triangleq \mathbb{E}\left\{\vec{\mu}\left(\vec{P}(t), \vec{S}(t)\right) \mid \boldsymbol{H}(t)\right\} \geq \vec{\lambda} \quad (2)$$

where $\boldsymbol{H}(t)$ represents the history of past channel states during slots $\tau \in \{0, 1, \ldots, t-1\}$.

As the corollary assumes channel states are i.i.d. over slots, the above expectation is the same for all slots t. Theorem 1 is proven via the following two claims: (Claim 1) No algorithm can achieve stability with a smaller average power P_{av}^*, and (Claim 2) any rate vector $\vec{\lambda}$ strictly interior to Λ can be stabilized with an average power that is arbitrarily close to P_{av}^*. Claim 1 is proven in Appendix A by extending the dimensionality of the system from L to $L+1$ and applying Caratheodory's Theorem [28] (where the $k \in \{1, \ldots, L+2\}$ result is also obtained). Below we prove Claim 2:

Proof: (Claim 2) The network capacity region Λ is proven in [2] to consist of all rate vectors $\vec{\lambda}$ such that a stationary power allocation rule exists satisfying (2). The value of P_{av}^* is by definition the average power consumption, minimized over all such stationary rules. If $\vec{\lambda}$ is strictly interior to Λ, there exists a positive value ϵ such that $\vec{\lambda} + \vec{\epsilon} \in \Lambda$ (where $\vec{\epsilon}$ is the L-dimensional vector with all entries equal to ϵ). It follows that there exists a stationary power allocation rule satisfying:

$$\mathbb{E}\left\{\vec{\mu}\left(\vec{P}(t), \vec{S}(t)\right)\right\} \geq \vec{\lambda} + \vec{\epsilon} > \vec{\lambda}$$

and we define $P_{av}^*(\epsilon)$ as the minimum average power consumed by any such stationary policy. The time average transmission rate of each queue is strictly larger than the arrival rate, and hence the network is stable [2]. This holds for arbitrarily small values of ϵ. Furthermore, it is not difficult to show:[3]

$$P_{av}^* \leq P_{av}^*(\epsilon) \leq \left(1 - \frac{\epsilon}{\epsilon_{max}}\right)P_{av}^* + \frac{\epsilon}{\epsilon_{max}}P_{av}^*(\epsilon_{max})$$

where ϵ_{max} is the largest scalar such that $(\lambda_l + \epsilon_{max}) \in \Lambda$. It follows that $P_{av}^*(\epsilon) \rightarrow P_{av}^*$ as $\epsilon \rightarrow 0$, so that stability can be attained with power that is arbitrarily close to P_{av}^*. □

We note that the concept of randomization used in Theorem 1 is vitally important to treat the general case where the set $\{\vec{\mu}(\vec{P}, \vec{S}) \mid \vec{P} \in \Pi\}$ is not necessarily convex. Otherwise, optimality can be achieved by a strategy that allocates a fixed power vector $\vec{P}^{\vec{S}}$ whenever the channel is in state $\vec{S}$. Note that even if there are only two possible channel states for every link, the total number of channel state vectors is 2^L. Thus, while the above static optimization defines the minimum

[3] The right hand side of the inequality follows by noting $P_{av}^*(\epsilon)$ is less than or equal to the average power associated with the mixed strategy that uses the P_{av}^* strategy with probability $1 - \epsilon/\epsilon_{max}$ and the $P_{av}(\epsilon_{max})$ strategy with probability ϵ/ϵ_{max}.

power level P^*_{av}, it is not practical to envision solving the optimization via standard techniques, even if the channel state probabilities $\pi_{\vec{S}}$ are fully known. In the next section we overcome this problem by developing a novel *stochastic optimization* technique.

B. An Energy-Optimal Control Algorithm

Here we develop a practical control algorithm that stabilizes the system and expends an average power that is arbitrarily close to the minimum power solution P^*_{av}. For simplicity of exposition, we assume the arrival vectors $\vec{A}(t)$ are i.i.d. over timeslots with arrival rate $\mathbb{E}\{\vec{A}(t)\} = \vec{\lambda}$, and that the channel state vectors $\vec{S}(t)$ are i.i.d. over timeslots with channel probabilities $\pi_{\vec{S}}$.[4] The algorithm below uses an arbitrary control parameter $V > 0$ that affects a tradeoff in average queueing delay.

Energy-Efficient Control Algorithm (EECA): Every timeslot, observe the current levels of queue backlog $\vec{U}(t)$ and channel states $\vec{S}(t)$ and allocate a power vector $\vec{P}(t) = (P_1, \ldots, P_L)$ according to the following optimization:

$$\begin{aligned} &\text{Maximize:} && \textstyle\sum_{l=1}^{L} \left[2U_l(t)\mu_l(\vec{P}, \vec{S}(t)) - VP_l\right] \qquad (3)\\ &\text{Subject to:} && \vec{P} = (P_1, \ldots, P_L) \in \Pi \end{aligned}$$

The EECA algorithm is similar to the power allocation algorithm of maximizing $\sum_l U_l \mu_l(\vec{P}, \vec{S})$ [21] [15] [19], with the exception that the optimization metric is modified by a weighted power term $-VP_l$ for each link l. It is interesting to note that the resulting metric is similar to the index policy of [27] developed for minimizing power in a system with infinite backlog and no queueing. However, the "index" that is used in [27] is a constant Lagrange multiplier that is pre-computed based on channel probability information, while our "index" includes a dynamic queue state $U_l(t)$ that is updated from slot to slot but requires no pre-computation or a-priori statistical knowledge.

Distributed Implementation: For cell-partitioned networks, we have $\vec{\mu}(\vec{P}, \vec{S}) = (\mu_1(P_1, S_1), \ldots, \mu_L(P_L, S_L))$. In this case, the above optimization is implemented according to the following simple algorithm: Each node measures the channel state $S_l(t)$ for each of its own outgoing links l and computes a *quality value* Q_l, where Q_l is the maximum value of $2U_l(t)\mu_l(P_l, S_l(t)) - VP_l$ over either the continuous interval $0 \le P_l \le P_{peak}$ or the 2-valued set $P_l \in \{0, P_{peak}\}$. Define $\tilde{P}_l$ as the *quality maximizing power level* for link l. Define Ω_n as the set of links $l \in \{1, \ldots, L\}$ such that $tran(l) = n$. Each node n then computes l^*_n and Q^*_n, defined as follows:

$$l^*_n \triangleq \arg\max_{l \in \Omega_n} Q_l \ , \ \ Q^*_n \triangleq Q_{l^*_n}$$

The value of Q^*_n is the contribution that node n brings to the summation in (3) if it is chosen for transmission. Each node then broadcasts its value of Q^*_n to all other nodes in its cell, and the node n with the largest Q^*_n is selected to transmit in that cell (ties are broken arbitrarily). Transmission takes place over link $l = l^*_n$, with power level $\tilde{P}_l$. In cases where each cell can support more than one transmission, the algorithm is simply implemented by selecting the *set* of nodes with the largest quality metrics.

Example 1: Under the ON/OFF constraint $P_l \in \{0, P_{peak}\}$, the power $\tilde{P}_l$ for each link l is given by:

$$\tilde{P}_l = \begin{cases} P_{peak} & \text{if } 2U_l(t)\mu_l(P_{peak}, S_l(t)) > VP_{peak} \\ 0 & \text{else} \end{cases}$$

In this case, we see that power is allocated only when the backlog exceeds a channel state dependent threshold.

Example 2: Suppose we have a continuous constraint $0 \le P_l \le P_{peak}$ and that rate functions have a logarithmic profile: $\mu_l(P, S) = \log(1 + \gamma_S P)$, where γ_S is an attenuation/noise coefficient associated with channel state S. In this case, the optimal power level is a continuous function of the queue backlog. Indeed, for any link l with channel state $S_l(t) = S$ and queue backlog $U_l(t) = U$, the quality maximizer $\tilde{P}_l$ is a critical point of $2U\mu_l(P, S) - VP$ over the interval $0 \le P \le P_{peak}$. Differentiating with respect to power, we have:

$$\frac{d}{dP}\left[2U\mu_l(P, S) - VP\right] = \frac{2U\gamma_S}{1 + \gamma_S P} - V$$

and it easily follows that:

$$\tilde{P}_l = \min\left[\max\left[\frac{2U_l(t)}{V} - \frac{1}{\gamma_{S_l(t)}}, 0\right], P_{peak}\right] \qquad \Box$$

To evaluate the above algorithm, define A^2_{max}, μ^{out}_{max}, and B as follows:

$$\begin{aligned} A^2_{max} &\triangleq \max_{n} \sum_{l \in \Omega_n} \mathbb{E}\left\{A^2_l\right\} \\ \mu^{out}_{max} &\triangleq \max_{\{n, \vec{S}, \vec{P} \in \Pi\}} \sum_{l \in \Omega_n} \mu_l(\vec{P}, \vec{S}) \\ B &\triangleq A^2_{max} + (\mu^{out}_{max})^2 \qquad (4) \end{aligned}$$

Now assume that $\vec{\lambda}$ is *strictly interior* to the network capacity region Λ, and define the scalar value ϵ_{max} as the largest value that can be added to each component of $\vec{\lambda}$ so that the resulting vector is still within the capacity region, i.e., $(\lambda_l + \epsilon_{max}) \in \Lambda$.

Theorem 2: If $\vec{\lambda}$ is *strictly interior* to Λ, then the EECA algorithm with any $V > 0$ stabilizes the system, with a resulting average congestion bound given by:

$$\overline{\sum_l U_l} \triangleq \limsup_{t \to \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \left[\sum_l \mathbb{E}\left\{U_l(\tau)\right\}\right] \le \frac{BN + VNP_{peak}}{2\epsilon_{max}}$$

Furthermore, average power $\overline{P}_{av}$ is given by:

$$\overline{P}_{av} \triangleq \limsup_{t \to \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \left[\sum_l \mathbb{E}\left\{P_l(\tau)\right\}\right] \le P^*_{av} + BN/V$$

where P^*_{av} is the minimum power solution of the optimization in Theorem 1.

Thus, the V parameter can be chosen so that BN/V is arbitrarily small, yielding average power that is arbitrarily close to the optimum. However, the congestion bound grows linearly

[4] We note that the i.i.d. assumptions are not necessary, and the same algorithms can be used for general ergodic arrivals and channels, resulting in modified but more involved delay expressions [2].

with V. By Little's Theorem, average backlog is proportional to average bit delay. Hence, average power can be pushed arbitrarily close to the minimum value, with a corresponding linear increase in average delay. This holds because the V parameter effectively determines the amount by which the time average transmission rate vector $\vec{\mu}_{av}$ is larger than the input rate $\vec{\lambda}$. Pushing $\vec{\mu}_{av}$ downward towards $\vec{\lambda}$ decreases average power consumption while increasing queueing delay. Theorem 2 is proved in the next subsection using a novel drift argument.

C. Performance Analysis

To prove the performance results of the previous subsection, we first establish a novel Lyapunov drift technique enabling stability and performance optimization to be achieved simultaneously. Let $\vec{U}(t)$ be a vector process of queue backlogs that evolves according to some probability law. To measure aggregate network congestion, define a *Lyapunov function* $L(\vec{U})$ as the sum of squares of the individual queue backlogs: $L(\vec{U})\triangleq\sum_l U_l^2$. Let $\vec{P}(t) = (P_1(t), \ldots, P_L(t))$ represent a process of non-negative auxiliary control variables. Let $g(\vec{P})$ be any non-negative cost function of the vector $\vec{P}$, and let g^* represent a target cost value. The goal is to stabilize the $\vec{U}(t)$ process while keeping the time average cost of $g(\vec{P}(t))$ near or below the value of g^*. (Note that if $\vec{P}$ represents a power vector and $g(\vec{P}) = \sum_l P_l$, then minimizing cost corresponds to minimizing time average power).

Define the one-step *conditional Lyapunov drift* $\Delta(\vec{U}(t))$ as follows:[5]

$$\Delta(\vec{U}(t))\triangleq\mathbb{E}\left\{L(\vec{U}(t+1)) - L(\vec{U}(t)) \mid \vec{U}(t)\right\} \tag{5}$$

where the expectation is taken over the potential randomness of the channel state and control decision during slot t, given the current backlog vector $\vec{U}(t)$.

Lemma 1: (Lyapunov Drift with Performance Optimization) If there are positive constants V, B, ϵ such that for all timeslots t and all vectors $\vec{U}(t)$ the one-step conditional Lyapunov drift satisfies:

$$\Delta(\vec{U}(t)) + V\mathbb{E}\left\{g(\vec{P}(t)) \mid \vec{U}(t)\right\} \leq \\ B - \epsilon\sum_l U_l(t) + Vg^* \tag{6}$$

then the system is stable and time average backlog satisfies:

$$\overline{\sum_l U_l}\triangleq\limsup_{M\to\infty}\frac{1}{M}\sum_{\tau=0}^{M-1}\sum_l\mathbb{E}\left\{U_l(\tau)\right\} \leq \frac{B+Vg^*}{\epsilon}$$

while time average cost satisfies:

$$\overline{g}\triangleq\limsup_{M\to\infty}\frac{1}{M}\sum_{\tau=0}^{M-1}\mathbb{E}\left\{g(\vec{P}(\tau))\right\} \leq g^* + B/V$$

From the above statement, it is clear that if the V parameter can be increased while holding all other constants fixed, then the time average cost can be pushed arbitrarily near or below the target cost level g^*, with a corresponding tradeoff in average queue backlog.

Proof: The drift condition is satisfied for all timeslots t. Taking an expectation of (6) with respect to the distribution of $\vec{U}(t)$ and using the law of iterated expectations yields:

$$\mathbb{E}\left\{L(\vec{U}(t+1)) - L(\vec{U}(t))\right\} + V\mathbb{E}\left\{g(P\vec{(t)})\right\} \leq \\ B - \epsilon\sum_l\mathbb{E}\left\{U_l(t)\right\} + Vg^*$$

Summing over timeslots $t \in \{0, \ldots, M-1\}$ and dividing by M yields:

$$\frac{\mathbb{E}\left\{L(\vec{U}(M))-L(\vec{U}(0))\right\}}{M} + \frac{V}{M}\sum_{\tau=0}^{M-1}\mathbb{E}\left\{g(\vec{P}(\tau))\right\} \leq \\ B - \frac{\epsilon}{M}\sum_{\tau=0}^{M-1}\sum_l\mathbb{E}\left\{U_l(\tau)\right\} + Vg^* \tag{7}$$

By non-negativity of the Lyapunov function and of the $g(\vec{P})$ function, a simple manipulation of (7) yields:

$$\frac{1}{M}\sum_{\tau=0}^{M-1}\sum_l\mathbb{E}\left\{U_l(\tau)\right\} \leq \frac{B+Vg^*+\mathbb{E}\left\{L(\vec{U}(0))\right\}/M}{\epsilon}$$

Taking limits of the above inequality as $M \to \infty$ yields the time average backlog bound. In [2] [22], it is shown that this time average backlog bound implies system stability.

Similarly, by again manipulating (7) we obtain:

$$\frac{1}{M}\sum_{\tau=0}^{M-1}\mathbb{E}\left\{g(\vec{P}(\tau))\right\} \leq g^* + \frac{B}{V} + \frac{\mathbb{E}\left\{L(\vec{U}(0))\right\}}{VM} \tag{8}$$

Taking limits as $M \to \infty$ yields the result. □

The art of stochastic optimal networking is designing a strategy to ensure the drift condition of Lemma 1 is satisfied. In the remainder of this section, we illustrate the technique with a constructive proof of Theorem 2. The first step is to establish a general expression for Lyapunov drift under any power allocation policy.

Lemma 2: If arrivals $\vec{A}(t)$ are i.i.d. every slot with rates $\mathbb{E}\{\vec{A}(t)\} = \vec{\lambda} = (\lambda_1, \ldots, \lambda_L)$, then the conditional Lyapunov drift under any power allocation policy satisfies:

$$\Delta(\vec{U}(t))\triangleq\mathbb{E}\left\{L(\vec{U}(t+1) - L(\vec{U}(t)) \mid \vec{U}(t)\right\} \leq \\ BN - 2\sum_l U_l(t)\left[\mathbb{E}\left\{\mu_l(\vec{P}(t), \vec{S}(t)) \mid \vec{U}(t)\right\} - \lambda_l\right] \tag{9}$$

where B is defined in (4). □

The lemma follows simply by squaring the dynamical queueing equation (1) and taking expectations, and is proved in Appendix B. We now massage the right hand side of (9) into a form suitable for application of Lemma 1 by adding the same value to both sides of the inequality. We have:

$$\Delta(\vec{U}(t)) + V\sum_l\mathbb{E}\left\{P_l(t) \mid \vec{U}(t)\right\} \leq BN + 2\sum_l U_l(t)\lambda_l \\ -\mathbb{E}\left\{\sum_l\left[2U_l(t)\mu_l(\vec{P}(t), \vec{S}(t)) - VP_l(t)\right] \mid \vec{U}(t)\right\} \tag{10}$$

The design principle behind the EECA algorithm of section III-B is now apparent: *Given $\vec{U}(t)$ at time t, the EECA algorithm (3) is designed to minimize the right hand side of inequality (10) over all possible power allocation strategies.*

Suppose now that $\vec{\lambda}$ is strictly interior to the capacity region Λ, and let ϵ be a positive value such that $\vec{\lambda} + \vec{\epsilon} \in \Lambda$. Because channel states are i.i.d. over slots, from Corollary 1 it follows

[5] Strictly speaking, proper notation for the conditional Lyapunov drift should be $\Delta(\vec{U}(t), t)$, as the expectation may also depend on the timeslot. However, we use the more concise notation $\Delta(\vec{U}(t))$, which should be understood as a formal representation of the right hand side of (5).

that there exists a stationary randomized power allocation strategy that chooses power independent of queue backlog and yields for all timeslots t:

$$\begin{aligned} \sum_l \mathbb{E}\left\{P_l(t) \,\middle|\, \vec{U}(t)\right\} &= P_{av}(\epsilon) && (11) \\ \mathbb{E}\left\{\mu_l\left(\vec{P}(t), \vec{S}(t)\right) \mid \vec{U}(t)\right\} &\geq \lambda_l + \epsilon \quad \text{(for all } l\text{)} && (12) \end{aligned}$$

where $P_{av}(\epsilon)$ is the minimum power required to stabilize the data rates $\vec{\lambda} + \vec{\epsilon}$. Note that $P_{av}(\epsilon) \to P_{av}^*$ as $\epsilon \to 0$. Because this stationary rule is simply a particular power allocation strategy, the final term in (10) under the EECA algorithm is less than or equal to the resulting value under the stationary rule. However, this value in (10) under the stationary rule can be explicitly calculated using (11) and (12), and we have:

$$\begin{aligned} \Delta(\vec{U}(t)) + V\sum_l \mathbb{E}\left\{P_l(t) \mid \vec{U}(t)\right\} \leq BN + 2\sum_l U_l(t)\lambda_l \\ - \left[2\sum_l U_l(t)(\lambda_l + \epsilon) - VP_{av}(\epsilon)\right] \end{aligned}$$

Canceling the $U_l(t)\lambda_l$ terms in the above expression yields:

$$\begin{aligned} \Delta(\vec{U}(t)) + V\sum_l \mathbb{E}\left\{P_l(t) \mid \vec{U}(t)\right\} \leq BN - 2\epsilon\sum_l U_l(t) \\ + VP_{av}(\epsilon) \end{aligned}$$

The above expression is in the exact form specified in Lemma 1 in the case $g(\vec{P}) = \sum_l P_l$. It follows that time average unfinished work satisfies:

$$\overline{\sum_l U_l} \leq \frac{BN + VP_{av}(\epsilon)}{2\epsilon} \leq \frac{BN + VNP_{peak}}{2\epsilon} \qquad (13)$$

and time average power satisfies:

$$\overline{P}_{av} \triangleq \overline{\sum_l P_l} \leq P_{av}(\epsilon) + BN/V \qquad (14)$$

The performance bounds in (13) and (14) hold for any value $\epsilon > 0$ such that $\vec{\lambda} + \vec{\epsilon} \in \Lambda$. However, the particular choice of ϵ only affects the bound calculation and does not affect the EECA allocation policy or change any sample path of system dynamics. We can thus optimize the bounds in (13) and (14) separately over all possible ϵ values. The bound in (14) is clearly minimized by taking a limit as $\epsilon \to 0$, yielding:

$$\overline{\sum_l P_l} \leq P_{av}^* + BN/V$$

Conversely, the bound in (13) is minimized by considering the largest feasible ϵ such that $\vec{\lambda} + \vec{\epsilon} \in \Lambda$ (defined as ϵ_{max}), yielding:

$$\overline{\sum_l U_l} \leq \frac{BN + VNP_{peak}}{2\epsilon_{max}}$$

This proves Theorem 2.

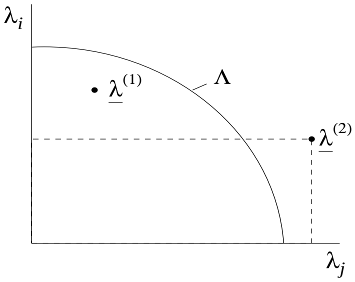

Fig. 3. A capacity region Λ (illustrated in 2 dimensions) with a rate vector $\vec{\lambda}^{(1)}$ strictly in the interior. The rate vector $\vec{\lambda}^{(2)}$ is outside of the capacity region.

IV. Average Power Constraints

In this section we consider a related problem of maximizing network throughput subject to both peak and average power constraints. Specifically, we consider the same one-hop network of the previous section, but assume that each node $n \in \{1, \ldots, N\}$ must satisfy the average power constraint:

$$\lim_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1}\left[\sum_{l\in\Omega_n} \mathbb{E}\left\{P_l(\tau)\right\}\right] \leq P_{av}^n \qquad (15)$$

where Ω_n is the set of all outgoing links of node n, and P_{av}^n is the average power constraint of node n.

Using a proof similar to that given in Theorem 1, it can be shown that the new capacity region Λ reduces to the set of all rates $\vec{\lambda}$ for which there exists a stationary randomized power allocation scheme that makes decisions based only on the current channel state $\vec{S}(t)$, and such that (2) is satisfied for all t, and the additional constraints $\mathbb{E}\left\{\sum_{l\in\Omega_n} P_l(t) \mid \boldsymbol{H}(t)\right\} \leq P_{av}^n$ are also satisfied for all t and all $n \in \{1, \ldots, N\}$. Here we consider cases where the arrival rate vector $\vec{\lambda}$ is *either inside the capacity region or outside of the capacity region*. This requires an additional set of admission control decisions to be made on top of the power allocation decisions, as only a fraction of the arriving traffic can be successfully delivered if inputs exceed capacity (see Fig. 3).

Let $R_l(t)$ represent the packets accepted into the network at queue l on timeslot t (where $R_l(t) \leq A_l(t)$, that is, $R_l(t)$ is the portion of new arrivals that are accepted on slot t, where the remaining data is dropped). Define $\overline{R}_l \triangleq \lim_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1} \mathbb{E}\left\{R_l(\tau)\right\}$ as the long term expected admission rate into queue l, and let $\vec{R}_{av} = (\overline{R}_1, \ldots, \overline{R}_L)$. The goal is to design a joint strategy for power allocation and admission control that satisfies all power constraints while maximizing the weighted throughput metric $\sum_l \theta_l \overline{R}_l$ (where θ_l values are arbitrary positive weights) subject to the demand requirement $\vec{R}_{av} \leq \vec{\lambda}$ and the stability requirement $\vec{R}_{av} \in \Lambda$. Define $\vec{R}^*$ as the optimal admission rate vector for this problem. This optimum could in principle be computed if the arrival rates $\vec{\lambda}$ and the capacity region Λ were known in advance. Below we design a practical algorithm that performs arbitrarily close to the utility of $\vec{R}^*$.

A. The Virtual Power Queue

We first establish a novel mechanism for ensuring the average power constraints are met at every node. To this end, each node n maintains a *virtual power queue* with occupancy $X_n(t)$ equal to the maximum excess power expended beyond the average power constraint over any interval ending at slot t. Indeed, defining $X_n(0) = 0$, we propagate the $X_n(t)$ values as follows:

$$X_n(t+1) = \max[X_n(t) - P_{av}^n, 0] + \sum_{l\in\Omega_n} P_l(t) \tag{16}$$

Thus, the $X_n(t)$ process acts as a single server queue with constant server rate given by the average power constraint P_{av}^n, with 'arrivals' given by the total power allocated for outgoing transmissions of node n on the current timeslot. The intuition behind this construction is given by the following observation: *If a power allocation algorithm conforms to the power constraint $\vec{P}(t) \in \Pi$ for all t while stabilizing all actual queues $U_l(t)$ and all virtual queues $X_n(t)$ (for $l \in \{1, \ldots, L\}$, $n \in \{1, \ldots, N\}$), then the strategy also satisfies the average power constraints for each node.* This observation holds because if the excess backlog $X_n(t)$ in virtual power queue n is stabilized, it must be the case that the time average 'power arrivals' $\sum_{l\in\Omega_n} \overline{P}_l$ (corresponding to time average power expenditure in node n) is less than or equal to the 'service rate' P_{av}^n. Formally, this observation is stated according to the following lemma (proven in Appendix D):

Lemma 3: If the virtual power queue $X_n(t)$ is stable and satisfies $\limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1} \mathbb{E}\{X_n(\tau)\} < \infty$, then $\limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1}\sum_{l\in\Omega_n} \mathbb{E}\{P_l(\tau)\} \leq P_{av}^n$.

B. An Energy Constrained Control Algorithm (ECCA)

We use the virtual power queues in the following energy constrained control algorithm. Assume the weights θ_l are known to the controllers, and let $V > 0$ represent an arbitrary control parameter.

Admission Control: Every timeslot and for each queue l, we allow the full set of new arrivals $A_l(t)$ into the queue whenever $U_l(t) \leq V\theta_l/2$. Else, we drop all new arrivals for queue l entering on that timeslot.

Power Allocation: Allocate power $\vec{P}(t) = \vec{P}$ according to the following optimization:

$$\text{Max: } \sum_{n=1}^N \sum_{l\in\Omega_n} \left[U_l(t)\mu_l(\vec{P}, \vec{S}(t)) - X_n(t)P_l\right] \tag{17}$$

$$\text{Subject to: } \vec{P} \in \Pi$$

The virtual power queues $X_n(t)$ are then updated via (16).

Note that distributed implementation of this algorithm for the case $\vec{\mu}(\vec{P}, \vec{S}) = (\mu_1(P_1, S_1), \ldots, \mu_L(P_L, S_L))$ is similar to the implementation of EECA given in Section III-B. The only difference here is that the quality maximizing values $\tilde{P}_l$ are computed by using the value $2X_n(t)$ instead of the scalar V [compare (17) and (3)]. To simplify the analysis of the above algorithm, we additionally assume that the total arrivals to any node are bounded by a constant value A_{max} every timeslot, that is, $\sum_{l\in\Omega_n} A_l(t) \leq A_{max}$ for all nodes n. Further, we make the following additional system assumptions:

Property 1: If $(P_1, \ldots, P_L) \in \Pi$, then setting one or more of the entries P_i to zero yields another vector that is contained in Π.

Property 2: There exists a finite value $\beta > 0$ such that:

$$\mu_i(\vec{P}, \vec{S}) \leq \mu_i(\vec{P} - P_i\vec{e}_i, \vec{S}) + \beta P_i$$

for all power vectors $\vec{P} \in \Pi$, all channel states $\vec{S}$, and all links $i \in \{1, \ldots, L\}$ (where $\vec{e}_i$ is an L dimensional unit vector with a "1" in the ith entry and zeros in all other entries).

In the case when rate functions are differentiable with respect to power, the value β represents the maximum directional derivative with respect to power, maximized over all links and channel states.

Theorem 3: For any input rate vector $\vec{\lambda}$, the above ECCA algorithm conforms to both peak and average power constraints and yields queue backlog and excess energy that is deterministically upper bounded for all t as follows:

$$\begin{aligned} U_l(t) &\leq U_l^{max} \triangleq \frac{V\theta_l}{2} + A_{max} \\ X_n(t) &\leq X_n^{max} \triangleq \frac{\beta V}{2}\left(\max_{l\in\Omega_n}\{\theta_l\}\right) + \beta A_{max} + P_{peak}^n \end{aligned}$$

for all nodes n and all links l, where P_{peak}^n is the maximum value of $\sum_{l\in\Omega_n} P_l$ over $\vec{P} \in \Pi$. Further, if arrivals and channel states are i.i.d. over timeslots, then the algorithm achieves a throughput performance bound of:

$$\overline{\sum_l \theta_l R_l} \geq \sum_l \theta_l R_l^* - \frac{(B+C)N}{V} \tag{18}$$

where B is defined as in (4), and where:

$$C \triangleq \frac{1}{N}\sum_{n=1}^N \left[(P_{peak}^n)^2 + (P_{av}^n)^2\right] \tag{19}$$

Note that the queue backlog is bounded for every instant of time. Hence, the algorithm yields the same performance if all buffers are finite with buffer size $\mathit{Buffer} = V\theta_{max}/2 + A_{max}$. In systems with finite buffers, the parameter V could be defined according to this equation, resulting in performance:

$$\overline{\sum_l \theta_l R_l} \geq \sum_l \theta_l R_l^* - \frac{(B+C)N\theta_{max}}{2(\mathit{Buffer} - A_{max})}$$

and hence performance can be pushed arbitrarily close to optimality by increasing the buffer size. The excess energy bound is also very strong, and implies that the total energy expended by node n over any interval of size T slots is less than or equal to $TP_{av}^n + X_n^{max}$ (and in particular the average power constraints are satisfied). It is remarkable that these performance guarantees do not depend on the channel statistics or arrival rates.

C. Performance Analysis

We analyze the above strategy in a manner similar to the EECA algorithm of the previous section. In particular, the $R_l(t)$ variables play the role of packet arrivals $A_l(t)$:

$$U_l(t+1) = \max[U_l(t) - \mu_l(\vec{P}(t), \vec{S}(t)), 0] + R_l(t) \tag{20}$$

The virtual queue backlogs $\vec{X}(t)$ evolve according to (16). Define the Lyapunov function $L(\vec{U},\vec{X}) = \sum_l U_l^2 + \sum_n X_n^2$, and define the one-step drift:

$$\Delta(\vec{U}(t),\vec{X}(t)) \triangleq \mathbb{E}\left\{L(\vec{U}(t+1),\vec{X}(t+1)) - L(\vec{U}(t),\vec{X}(t)) \mid \vec{U}(t),\vec{X}(t)\right\}$$

To simplify formulas, below we use the shortened notation $\Delta, \mu_l, \vec{U}$, and $\vec{X}$ to represent $\Delta(\vec{U}(t))$, $\mu_l(\vec{P}(t),\vec{S}(t))$, $\vec{U}(t)$, and $\vec{X}(t)$.

Lemma 4: The one-step drift satisfies:

$$\Delta \leq N(B+C) - 2\sum_l U_l \mathbb{E}\left\{\mu_l - R_l \mid \vec{U},\vec{X}\right\} -2\sum_n X_n\left[P_{av}^n - \mathbb{E}\left\{\sum_{l\in\Omega_n} P_l \mid \vec{U},\vec{X}\right\}\right]$$

The lemma follows by summing the corresponding drift of the actual queues and virtual queues (using update equations (20) and (16), compare with Lemma 2), and the derivation is omitted for brevity. Adding and subtracting the term $V\sum_l \theta_l \mathbb{E}\left\{R_l \mid \vec{U},\vec{X}\right\}$ to the right hand side of the drift expression and rearranging terms yields:

$$\begin{aligned}\Delta \leq N(B+C) + V\sum_l \theta_l \mathbb{E}\left\{R_l \mid \vec{U},\vec{X}\right\} - 2\sum_n X_n P_{av}^n \\ + \sum_l (2U_l - V\theta_l)\mathbb{E}\left\{R_l \mid \vec{U},\vec{X}\right\} \\ -2\sum_{n=1}^{N}\sum_{l\in\Omega_n}\mathbb{E}\left\{U_l\mu_l - X_nP_l \mid \vec{U},\vec{X}\right\}\end{aligned}$$

The design methodology of the ECCA algorithm is now apparent: *The admission control algorithm minimizes the second to last term of the above expression over all possible admission decisions, and the power allocation algorithm minimizes the last term of the above expression over all possible power decisions.*

In particular, the optimal input rate vector $\vec{R}^* = (R_1^*,\ldots,R_L^*)$ could in principle be achieved by the simple backlog-independent admission control algorithm of including all new arrivals $A_l(t)$ for a given link l and slot t independently with probability $\alpha_l = R_l^*/\lambda_l$, yielding:

$$\mathbb{E}\left\{R_l \mid \vec{U},\vec{X}\right\} = \mathbb{E}\left\{R_l\right\} = \alpha_l\mathbb{E}\left\{A_l\right\} = R_l^* \quad (21)$$

Likewise, because $\vec{R}^* \in \Lambda$, there must exist a stationary power allocation policy that chooses power independent of backlog and yields:

$$\mathbb{E}\left\{\mu_l \mid \vec{U},\vec{X}\right\} = \mathbb{E}\left\{\mu_l\right\} \geq R_l^* \quad (22)$$

$$\mathbb{E}\left\{\sum_{l\in\Omega_n} P_l \mid \vec{U},\vec{X}\right\} = \mathbb{E}\left\{\sum_{l\in\Omega_n} P_l\right\} \leq P_{av}^n \quad (23)$$

Plugging in the expectations (21)-(23) of the particular backlog-independent policies into the last two terms of the above drift expression for the ECCA algorithm thus preserves the bound, and yields:

$$\begin{aligned}\Delta(\vec{U}(t),\vec{X}(t)) \leq N(B+C) \\ +V\textstyle\sum_l \theta_l \mathbb{E}\left\{R_l(t) \mid \vec{U},\vec{X}\right\} - V\sum_l \theta_l R_l^* \quad (24)\end{aligned}$$

where we have canceled the common terms $\sum_l 2U_lR_l^*$ and $\sum_n X_nP_{av}^n$. Taking expectations of (24) with respect to $\vec{U},\vec{X}$ and summing from $t=0$ to $t=M-1$ yields:

$$\begin{aligned}\frac{1}{M}\sum_{\tau=0}^{M-1}\sum_l \theta_l\mathbb{E}\left\{R_l(\tau)\right\} \geq \sum_l \theta_l R_l^* - \frac{N(B+C)}{V} \\ -\mathbb{E}\left\{L(\vec{U}(0),\vec{X}(0))\right\}/(MV)\end{aligned}$$

which yields (18) as $M\to\infty$.

Furthermore, the backlog bound $U_l(t) \leq U_l^{max}$ follows immediately from the definition of the ECCA admission control policy: No new arrivals are admitted if $U_l(t) > V\theta_l/2$, so that $U_l(t) \leq V\theta_l/2 + A_{max}$ for all t (where in the worst case we add an amount A_{max} when backlog is exactly at the $V\theta_l/2$ threshold).

Likewise, by definition of the ECCA power allocation algorithm together with Properties 1 and 2 of Section IV-B, we have that for any node n, any link $l\in\Omega_n$, and time t such that $X_n(t) > \beta U_l(t)$:

$$\begin{aligned}U_l(t)\mu_l(\vec{P},\vec{S}(t)) - X_n(t)P_l \leq U_l(t)\mu_l(\vec{P}-P_l\vec{e}_l,\vec{S}(t)) \\ +\beta U_l(t)P_l - X_n(t)P_l \\ \leq U_l(t)\mu_l(\vec{P}-P_l\vec{e}_l,\vec{S}(t))\end{aligned}$$

where the inequality is achieved if and only of $P_l = 0$. Therefore, if $X_n(t) > \beta U_l(t)$, then the ECCA algorithm necessarily chooses $P_l(t) = 0$. Thus, if $X_n(t) > \beta\max_{l\in\Omega_n} U_l^{max}$, then $P_l(t) = 0$ for all $l\in\Omega_n$ and so by (16) the $X_n(t)$ value cannot further increase. It follows that $X_n(t) \leq \beta\max_{l\in\Omega_n} U_l^{max} + P_{peak}^n$ for all t, proving Theorem 3.

V. Multi-Hop Networks

Here we consider the same network as before but assume that data can be routed over multi-hop paths to reach its destination. We optimize over all possible power allocation and routing algorithms. Thus, incoming data is not necessarily associated with any particular link, and so we redefine the arrival processes in terms of the origin and destination of the data: $A_n^c(t) \triangleq$ amount of data exogenously arriving to node n at slot t that is destined for node c. All data (from any source node) that is destined for a particular node $c\in\{1,\ldots,N\}$ is defined as *commodity* c data. Data is stored in each node according to its destination, and we let $U_n^c(t)$ represent the current backlog of commodity c data in node n.

Suppose power vector $\vec{P}(t)$ is allocated in slot t, so that the transmission rate over a link l is $\mu_l(\vec{P}(t),\vec{S}(t))$. A *routing decision* must be made to establish which commodity to transfer over link l. In general, multiple commodities could be transfered over the same link simultaneously,[6] and we define routing variables $\mu_l^c(t)$ as the rate allocated to commodity c data over link l during slot t. The problem is to allocate power every timeslot according to the power constraint $\vec{P}(t)\in\Pi$ and then to route data according to the *link rate constraint*:

$$\sum_{c=1}^{N}\mu_l^c(t) \leq \mu_l(\vec{P}(t),\vec{S}(t)) \quad (25)$$

[6] We find that the capacity achieving solution needs only route a single commodity over any link during a timeslot.

Recall that Ω_n is the set of all links l such that $tran(l) = n$. Further define Θ_n as the set of all links l such that $rec(l) = n$. The resulting 1-step queueing equation for backlog $U_n^c(t)$ thus satisfies (for $c \neq n$):

$$\begin{aligned} U_n^c(t+1) \leq \max[U_n^c(t) - \sum_{l\in\Omega_n} \mu_l^c(t), 0] & \\ +A_n^c(t) + \sum_{l\in\Theta_n} \mu_l^c(t) & \end{aligned} \tag{26}$$

The above expression is an inequality rather than an equality because the incoming commodity c data to node n may be less than $\sum_{l\in\Theta_n}\mu_l^c(t)$ if the corresponding transmitting nodes have little or no data of this commodity waiting to be transfered.

In [22] [2], the network layer capacity region Λ is defined as the closure of the set of rate matrices $(\lambda_n^{(c)})$ that can be stably supported, considering all possible power allocation and routing strategies. There, it was shown that any rate matrix $(\lambda_n^{(c)}) \in \Lambda$ is supportable via a randomized algorithm for choosing power allocations $\vec{P}(t)$ and routing variables $\mu_l^c(t)$. Here, we assume the rate matrix is inside the capacity region, and develop an energy efficient stabilizing algorithm. However, note that the objective of minimizing average power expenditure in a multihop network may place an unfair power burden on centrally located nodes that are used by many others. Thus, to balance power more evenly, we consider the more general objective of minimizing the time average of $\sum_n g_n(\sum_{l\in\Omega_n} P_l(t))$, where $g_n(p)$ is any convex increasing cost function of the power expended by node n.

Define $U_n^n(t) = 0$ for all t, and define:

$$\begin{aligned} A_{max}^2 &\triangleq \max_n \sum_c \mathbb{E}\left\{(A_n^c)^2\right\} \\ \mu_{max}^{in} &\triangleq \max_{\{n,\vec{S},\vec{P}\in\Pi\}} \sum_{l\in\Theta_n} \mu_l(\vec{P},\vec{S}) \\ D &\triangleq (A_{max} + \mu_{max}^{in})^2 + (\mu_{max}^{out})^2 \end{aligned} \tag{27}$$

Let $L(\underline{U}) = \sum_{n,c}(U_n^c)^2$. The one-step drift $\Delta(\underline{U})$ for any policy is found by squaring the dynamical equation (26) as in Lemma 2, and is given in [22] as follows:

Lemma 5: If arrivals and channel states are i.i.d. over timeslots, then:

$$\begin{aligned} \Delta(\underline{U}(t)) + V\textstyle\sum_n \mathbb{E}\left\{g_n(\sum_{l\in\Omega_n}P_l(t)) \mid \underline{U}(t)\right\} \leq DN & \\ +2\textstyle\sum_{n,c} U_n^c(t)\lambda_n^{(c)} & \\ -\textstyle\sum_n \mathbb{E}\Big\{\sum_{l\in\Omega_n}\sum_c 2\mu_l^c(t)(U_{tran(l)}^c(t) - U_{rec(l)}^c(t)) & \\ -Vg_n(\textstyle\sum_{l\in\Omega_n}P_l(t))\,\big|\,\underline{U}(t)\Big\} & \end{aligned}$$

The above drift expression for multi-hop networks is the same as that given in [22], with the exception that we have added the optimization metric $V\sum_n \mathbb{E}\left\{g_n(\sum_{l\in\Omega_n}P_l(t)) \mid \underline{U}(t)\right\}$. Minimizing the right hand side in the above drift expression over all power allocations satisfying $\vec{P} \in \Pi$ and all routing strategies satisfying (25) leads to the following *multi-hop EECA algorithm*:

1) For all links l, find the commodity $c_l^*(t)$ such that:

$$c_l^*(t) = \arg\max_c \left\{U_{tran(l)}^c(t) - U_{rec(l)}^c(t)\right\}$$

and define:

$$W_l^*(t) = \max[U_{tran(l)}^{c_l^*}(t) - U_{rec(l)}^{c_l^*}(t), 0]$$

2) *Power Allocation:* Choose a power vector $\vec{P}(t) \in \Pi$ that maximizes:

$$\sum_n \left[\sum_{l\in\Omega_n} 2\mu_l(\vec{P},\vec{S}(t))W_l^* - Vg_n(\sum_{l\in\Omega_n}P_l)\right] \tag{28}$$

3) *Routing:* Over link l, if $W_l^*(t) > 0$, transmit commodity $c_l^*(t)$ in the amount of $\mu_l(\vec{P}(t),\vec{S}(t))$, using idle fill if necessary (in cases when there is not enough data to transmit).

Distributed Implementation: Given a cell partitioned network with backlog values $U_a^c(t)$ for all neighbor nodes a, the distributed method for allocating power and choosing which node transmits in every cell is similar to the implementation of EECA in Section III-B, with the exception that the quality maximizer values $\tilde{P}_l$ now maximize $\left[2\mu_l(P_l, S_l(t))W_l^* - Vg_{tran(l)}(P_l)\right]$, representing the contribution to (28) if link l is chosen for transmission (recall that, under the cell partition model, a given node n may activate only one outgoing link $l \in \Omega_n$ during a timeslot).

To find the backlog values of neighbors, note that for rectilinear networks there are at most 10 queues that change their backlog values during a timeslot in any given cell. This is because the transmitting node may transmit to another node in the same cell (increasing the queue level of the transmitted commodity in the receiving node, and decreasing it in the transmitting node), and there are at most 8 other data receptions in the same cell (due to potential tranmissions from the 8 adjacent cells). Knowledge of backlog levels in neighboring nodes can thus be maintained by broadcasting the backlog changes to all nodes in the same cell and in adjacent cells. Each update requires a triplet of information: (n, c, δ), where n is the node, c is the commodity that was changed, and δ is the amount of the change. Thus, the bandwidth of the broadcast control channel must be sufficient to support the transmission of up to 10 update triplets per cell per timeslot.

Let $\mathcal{D}$ represent the set of all (n, c) pairs for which there are valid network queues $U_n^{(c)}(t)$ (so that $(n, n) \notin \mathcal{D}$), and let $U_n^{(c)}(t) \triangleq 0$ for all $(n, c) \notin \mathcal{D}$.

Theorem 4: If the rate matrix $(\lambda_n^{(c)})$ is interior to the capacity region Λ, then the above multihop EECA algorithm for routing and power allocation stabilizes the network and yields a time average congestion bound of:

$$\overline{\sum_{nc} U_n^c} \leq \frac{DN + V\sum_n g_n(P_{peak})}{2\epsilon_{max}}$$

(where ϵ_{max} is the largest ϵ such that $(\lambda_n^{(c)} + \epsilon 1_n^{(c)}) \in \Lambda$, with $1_n^{(c)}$ being an indicator function equal to 1 if $(n, c) \in \mathcal{D}$, and 0 else). Further, the time average cost satisfies:

$$\overline{\sum_n g_n} \triangleq \lim_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\left[\sum_n \mathbb{E}\left\{g_n\left(\sum_{l\in\Omega_n}P_l(\tau)\right)\right\}\right] \leq g^* + \frac{DN}{V}$$

where g^* represents the minimum time average cost of any stabilizing policy.

The proof is similar to the proof of Theorem 2, and so we present only an outline: The dynamic algorithm minimizes the final term of Lemma 5 over all policies. In [22] it is shown that if there is an ϵ such that $(\lambda_n^{(c)} + \epsilon 1_n^{(c)}) \in \Lambda$, then a single stationary power allocation and routing strategy can be developed to satisfy $\mathbb{E}\{\mu_l^c(t)\} = f_l^c$, where the (f_l^c) values are multi-commodity flows such that for all $(n, c) \in \mathcal{D}$:

$$\sum_{l\in\Omega_n} f_l^c - \sum_{l\in\Theta_n} f_l^c = \lambda_n^{(c)} + \epsilon$$

(recall that Ω_n and Θ_n respectively represent the set of outgoing and incoming links for node n). Thus, under this stationary policy we have:

$$\sum_n \sum_{l\in\Omega_n} \sum_c \mathbb{E}\{\mu_l^c(t)\}\,(U_{tran(l)}^c(t) - U_{rec(l)}^c(t))$$

$$= \sum_{nc} U_n^c(t) \left(\sum_{l\in\Omega_n} \mathbb{E}\{\mu_l^c(t)\} - \sum_{l\in\Theta_n} \mathbb{E}\{\mu_l^c(t)\} \right) \quad (29)$$

$$= \sum_{nc} U_n^c(t)(\lambda_n^{(c)} + \epsilon) \quad (30)$$

where (29) follows by switching the sums and (30) follows by noting $U_n^c(t) \triangleq 0$ whenever $(n, c) \notin \mathcal{D}$. Further, the stationary policy also satisfies $\sum_n V\mathbb{E}\{g(\sum_{l\in\Omega_n} P_l(t))\} = g^*(\epsilon)$, where $g^*(\epsilon)$ is the minimum cost for stabilizing rates $(\lambda_n^{(c)} + \epsilon 1_n^{(c)})$ and satisfies $g^*(\epsilon) \to g^*$ as $\epsilon \to 0$. Plugging these particular policies into the last term of the drift expression in Lemma 5 thus preserves the bound and yields:

$$\Delta(\underline{U}(t)) + V\sum_n \mathbb{E}\{g_n(\sum_{l\in\Omega_n} P_l(t)) \mid \underline{U}(t)\} \le DN - 2\sum_{n,c} U_n^c(t)\epsilon + Vg^*$$

which yields the result upon application of Lemma 1. □

We note that the multi-hop EECA algorithm delivers all data to its destination *without knowing the network topology*. The algorithm effectively accomplishes this by expending initial energy transmitting data to neighbors in order to learn efficient routes, which emerge from backlog information.

A. Multi-hop ECCA

Here we show that the ECCA algorithm can also be extended to a multi-hop setting, providing maximum throughput subject to average power constraints. In particular, each node n maintains its own virtual power queue $X_n(t)$, but makes decisions based on a differential backlog metric rather than absolute backlog. We assume that Properties 1 and 2 of Section IV-B hold.

However, there is one important modification that we make in order to ensure queue levels in both the actual queues and the virtual power queues remain bounded for all time: We enforce the additional constraint that no node can transfer data of a particular commodity to a relay node that is not the destination of that commodity *unless* the differential backlog of that commodity between the two nodes is greater than or equal to a fixed value γ. The value γ is chosen large enough to ensure the resulting backlog of that commodity in the relay at time $t + 1$ is not larger than the corresponding backlog in the transmitting node at time t. Note that the most data that can enter a node in a single slot is $\mu_{max}^{in} + A_{max}$ (considering the sum of the maximum endogenous and exogenous arrivals). Hence, setting $\gamma = \mu_{max}^{in} + A_{max}$ ensures the condition is satisfied. This transmission restriction can then easily be implemented by defining a modified optimal commodity $\tilde{c}_l^*(t)$ and a modified differential backlog metric $\tilde{W}_l^*(t)$ for every link l as follows. First, we define constants γ_l^c for each link l and each commodity c:

$$\gamma_l^c = \begin{cases} \gamma & \text{if } rec(l) \neq c \\ 0 & \text{else} \end{cases}$$

That is, the γ_l^c value is equal to 0 whenever the receiver of link l is the destination of commodity c data, and is equal to γ when the receiver of link l is *not* the destination and hence would act as a relay for commodity c data. The $\tilde{c}_l^*(t)$ and $\tilde{W}_l^*(t)$ values are then defined as follows:

$$\tilde{c}_l^*(t) = \arg\max_c \left\{ U_{tran(l)}^c(t) - U_{rec(l)}^c(t) - \gamma_l^c \right\}$$

$$\tilde{W}_l^*(t) = \max[U_{tran(l)}^{c_l^*}(t) - U_{rec(l)}^{c_l^*(t)} - \gamma_l^{c_l^*(t)}, 0] \quad (31)$$

Note from (31) that $\tilde{W}_l^*(t) > 0$ only if the differential backlog is larger than γ. Consider now the objective of maximizing the weighted throughput $\sum_{nc} \theta_n^c \overline{R}_n^c$ (subject to network peak and average power constraints), for arbitrary positive weights θ_n^c. With this goal, we define the following *Multi-hop ECCA Algorithm* as follows:

1) *Admission Control:* Every timeslot and for each input (n, c), we accept the full set of exogenous arrivals $A_n^c(t)$ whenever $U_n^c(t) \le V\theta_n^c/2$. Else, all new arrivals $A_n^c(t)$ are rejected on that timeslot.
2) *Power Allocation:* Allocate power $\vec{P}(t) = \vec{P}$ according to the following optimization:

   Max: $\sum_{n=1}^N \sum_{l\in\Omega_n} \left[\tilde{W}_l^*(t)\mu_l(\vec{P}, \vec{S}(t)) - X_n(t)P_l \right]$

   Subject to: $\vec{P} \in \Pi$
3) *Virtual Queue Update:* The virtual queues $X_n(t)$ are updated according to (16).
4) *Routing:* Over link l, if $\tilde{W}_l^*(t) > 0$, transmit commodity $\tilde{c}_l^*(t)$ in the amount of $\mu_l(\vec{P}(t), \vec{S}(t))$, using idle fill if necessary.

As before, we assume that no node receives more than A_{max} bits of data exogenously during a single slot. Arrivals and channel states are assumed to be i.i.d. every slot, and the network is assumed to be initially empty at time 0. The following theorem establishes performance of the Multi-hop ECCA algorithm.

Theorem 5: For arbitrary rate matrices $(\lambda_n^{(c)})$ (possibly outside of the network capacity region), the ECCA algorithm ensures:

(a) $U_n^c(t) \le \theta_{max}V/2 + A_{max}$ for all time t and for all queues (n, c), where $\theta_{max} \triangleq \max_{nc}\{\theta_n^c\}$.

(b) All nodes satisfy their average power constraints. Specifically, the total power expenditure of any node n over any set of T slots is no more than $TP_{av}^n + X^{max}$, where X^{max}

represents a bound on the worst case virtual power queue backlog $X_n(t)$, and is given by:

$$X_n(t) \leq X^{max} \triangleq \frac{\beta V \theta_{max}}{2} + \beta A_{max} + P_{peak}$$

where β is defined according to Property 2 of Section IV-B.

(c) The resulting weighted throughput values satisfy:

$$\overline{\sum_{nc} \theta_n^c R_n^c} \geq \sum_{nc} \theta_n^c R_{nc}^* - \tilde{C}N/V \tag{32}$$

where $\tilde{C} \triangleq C + D + 2\gamma\mu_{max}^{out}$, C, D are defined in (19), (27), and where (R_{nc}^*) is the optimal throughput matrix (maximizing the weighted sum of throughput over all feasible rates).

The proof of part (c) uses a Lyapunov drift argument similar to the proofs in the previous sections, and is given in Appendix C. Below we prove parts (a) and (b).

Proof: (Parts (a) and (b)) We prove (a) by induction over timeslots. Let $U_{max}(t)$ represent the maximum backlog of any commodity in any node of the network at timeslot t. Assume that $U_{max}(t) \leq \theta_{max}V/2 + A_{max}$ (this clearly holds for $t = 0$). We prove the same is true at time $t+1$.

Consider any node n with backlog $U_n^c(t+1)$ at time $t+1$. We assume that $n \neq c$, as we have $U_n^n(t) = 0$ for all t. If node n received endogenous data of commodity c that was transmitted by some other node a at time t, then $U_a^c(t) - U_n^c(t) \geq \gamma$ (otherwise, the modified differential backlog for the link (a, n) would be zero and hence no data would be transmitted under the Multi-hop ECCA algorithm). As $U_n^c(t)$ can increase by at most γ every timeslot, we have $U_n^c(t+1) \leq U_n^c(t) + \gamma$, and:

$$U_n^c(t+1) \leq U_a^c(t) \leq U_{max}(t) \leq \theta_{max}V/2 + A_{max}$$

In the alternate case when node n did *not* receive any endogenous data of commodity c at time t, then the commodity c backlog can only increase due to exogenous arrivals. If there were no exogenous arrivals, then clearly we have $U_n^c(t+1) \leq U_n^c(t) \leq \theta_{max}V/2 + A_{max}$. Otherwise, if there were exogenous arrivals, it must be the case that $U_n^c(t) \leq \theta_{max}V/2$ (otherwise the ECCA admission control algorithm would reject all commodity c data exogenously arriving to node n during slot t), and hence $U_n^c(t+1) \leq \theta_{max}V/2 + A_{max}$. Thus, in all cases we have $U_n^c(t+1) \leq \theta_{max}V/2 + A_{max}$, proving part (a). Because all queues $U_n^c(t)$ are uniformly bounded, all differential backlogs $\tilde{W}_l^*(t)$ are also bounded by the same value, and hence (by the same argument as given in Theorem 3 for the single-hop ECCA algorithm), the virtual power queues $X_n(t)$ must be bounded by X^{max} for all time. □

VI. Mobile Networks

Here we treat extensions to fully mobile networks. Note that up to this point, we have assumed that nodes remain in their respective cells for all time and hence link dynamics depend only on random fading, environmental effects, or local "in-cell" mobility. In this section, we consider link dynamics that also depend on topological changes arising from full user mobility. In particular, for the cell partitioned network model, we assume that every timeslot nodes randomly choose to either remain in their same cell or visit another cell. It turns out that the *same algorithms* developed in the previous sections can be applied to this mobile network model, although the node mobility impacts the delay analysis as well as the algorithm implementation.

A. Delay Analysis

Note that the link condition between any two nodes of a mobile network depends on the node mobility process, which may not be i.i.d. over timeslots. For example, if nodes move according to a Markovian random walk, then channel state variations are correlated over timeslots because future channel states depend on current node locations. Thus, the performance bounds developed in the previous sections cannot be directly applied, as these results assume channel states that are independent from slot to slot. However, we note that the *same algorithms* developed for the i.i.d channel model can be applied directly to networks with *arbitrary ergodic* channel state variations, including models where channel states depend on Markovian random walks (or any other mobility pattern that exhibits a steady state).

Indeed, it can be shown under these more general channel models that stability is maintained, and performance also converges to optimal performance as the control parameter V is increased. However, the non-i.i.d. channel model alters the average delay bound. This can be shown by repeating the same Lyapunov drift arguments over the course of K timeslots, rather than just a single timeslot. The value of K is chosen for the analysis to be large enough so that the mobility process sampled every K slots is "sufficiently close" to an i.i.d. process (and hence, the value of K may depend on the network size). Rather than repeating all of our analysis using K-slot drift, we simply note that the end result yields a delay expression that is (roughly) scaled by a factor of K. The interested reader is referred to [2] [22] for analytical details of such K-slot drift arguments.

B. Implementation Complexity

Mobile networks present an additional challenge of *implementation complexity*. Recall that the multi-hop network control algorithms of Section V require knowledge of the backlog levels of all commodities in neighboring nodes. In cell partitioned networks, it was shown in Section V that there are at most 10 backlog changes per cell per timeslot. Hence, in networks *without* mobility, nodes can infer the backlog levels of their neighbors by keeping track of the backlog updates. However, in mobile networks, two nodes might suddenly become neighbors after being apart for a long period of time, and hence neither knows the backlog levels of the other. In this case, all backlog values must be exchanged. This is simple in the case when there is only a single commodity or a small number of commodities being delivered over the network, but requires $O(N)$ queue updates in the general case where each node maintains $N-1$ internal queues that store data destined for each of the other network nodes. It is possible to reduce this control overhead by updating only $O(1)$ of the queue components every timeslot, and having each node maintain a running *estimate* of the queue backlogs of all other queues. However, this complexity reduction can increase network delay by a factor proportional to the estimation error (see [2]). In the mobile network simulations of the next section, we assume that nodes are aware of the queue backlogs of their neighbors on every slot.

VII. Simulations

Here we present simulation results of our network control algorithms. Consider first the two-queue downlink example of Section II. Packets arrive to the system according to Poisson processes with rates $\lambda_1 = 8/9, \lambda_2 = 5/9$, which are the same as the empirical rates obtained by averaging over the first 9 timeslots of the example in Fig. 2. Channel states arise as i.i.d. vectors $(S_1(t), S_2(t))$ every slot. The probability of each vector state is matched to the empirical occurrence frequency in the example, so that $Pr[(G, M)] = 3/9, Pr[(M, B)] = 2/9, Pr[(M, M)] = 1/9$, etc. We first simulate the policy of serving the queue with the largest rate-backlog index $U_i(t)\mu_i(t)$, a strategy that stabilizes the system whenever possible but does not necessarily make energy efficient decisions [15] [19] [21]. The simulation was run for 10 million timeslots. The resulting average power is $P_{av} = 0.898$ Watts, and the resulting time average backlog is 2.50 packets.

Next, we consider the EECA algorithm, where power allocation decisions are determined by the solution of the optimization problem (3). First note that $A^2_{max} = \sum_{i=1}^{2} \lambda_i^2 + \lambda_i = 2.54$, $\mu^{out}_{max} = 3$, and hence from (4) we have $B = 11.54$. It follows from Theorem 2 that the resulting average power differs from optimality by no more than $11.54/V$, where V is the control parameter of the algorithm (note that $N = 1$ in this example). Furthermore, it can be shown that $\epsilon_{max} = 0.489$ for this example, and hence by Theorem 2 we know the average backlog in the system satisfies the following inequality:

$$\overline{U}_1 + \overline{U}_2 \leq \frac{11.54 + V}{0.978}$$

By Little's Theorem, dividing both sides of the above inequality by $(\lambda_1 + \lambda_2)$ yields an upper bound on average delay.

We simulated the EECA algorithm for 20 different values of the control parameter V, ranging from 1 to 10^4. Each simulation was run for 10 million timeslots. In Fig. 4 the resulting average power is plotted against the time average backlog. The corresponding upper bound is also shown in the figure. We find that average power decreases to its minimum value of 0.518 Watts as the control parameter V is increased, with a corresponding tradeoff in average delay. In Fig. 5 we plot average backlog versus the V parameter together with the backlog bound, illustrating that average delay grows linearly in V, as suggested by the performance bound. As a point of reference, we note that at $V = 50$, the average power is 0.53 Watts and the average sum backlog is 21.0 packets.

A. Minimum Energy Scheduling for Mobile Networks

Here we consider an ad-hoc mobile network with 28 users and a cell structure arranged as a 4×4 grid, as shown in Fig. 6. For simplicity, we assume there can be at most one transmission per cell per timeslot, and that all transmissions use full power of 1 Watt. We assume transmission rates are adaptive, and that 3 packets can be transfered if the receiver is in the same cell as the transmitter, while only 1 packet can be transfered if the receiver is in one of the adjacent cells to the North, South, East, or West. Data arrives to each node according to a Bernoulli arrival process with rate $\lambda = 0.5$ packets/slot (so that a single packet arrives with probability 0.5, else no packet arrives). We assume source-destination pairs are given by the grouping $1 \leftrightarrow 2$, $3 \leftrightarrow 4$, ..., $27 \leftrightarrow 28$, so that node 1 packets are destined for node 2 and node 2 packets are destined for node 1, node 3 packets are destined for node 4 and node 4 packets are destined for node 3, etc.

We simulate the multi-hop EECA algorithm for both a Markovian random walk model and an i.i.d. mobility model, with the objective of minimizing total power expenditure. In the Markovian mobility model, every timeslot nodes independently move to a neighboring cell either to the North, South, East, or West, with equal probability. In the case when a node on the edge of the network attempts to move in an infeasible

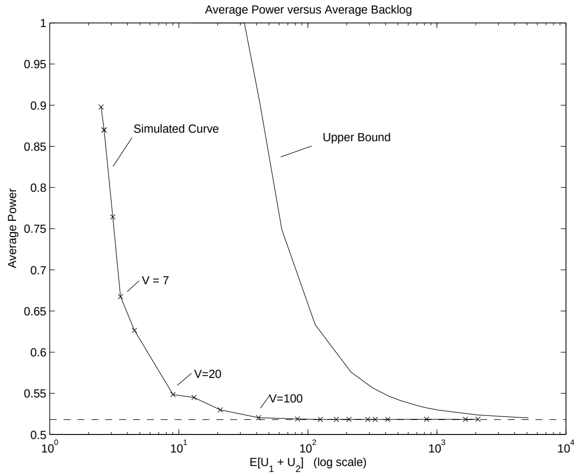

Fig. 4. Average power versus average backlog for a two queue downlink under the EECA algorithm.

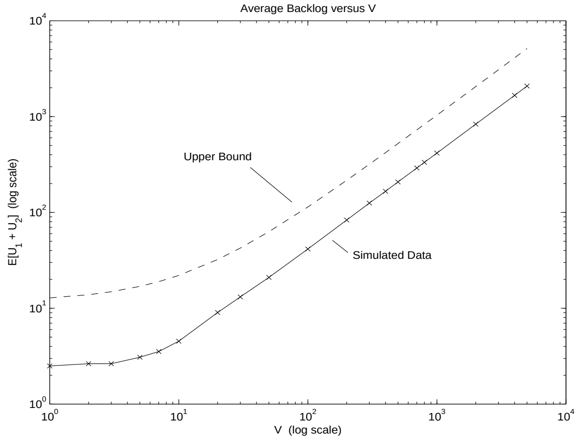

Fig. 5. Average backlog versus the V parameter from 10 million iterations of the EECA algorithm for a two queue downlink. The analytical upper bound is also plotted.

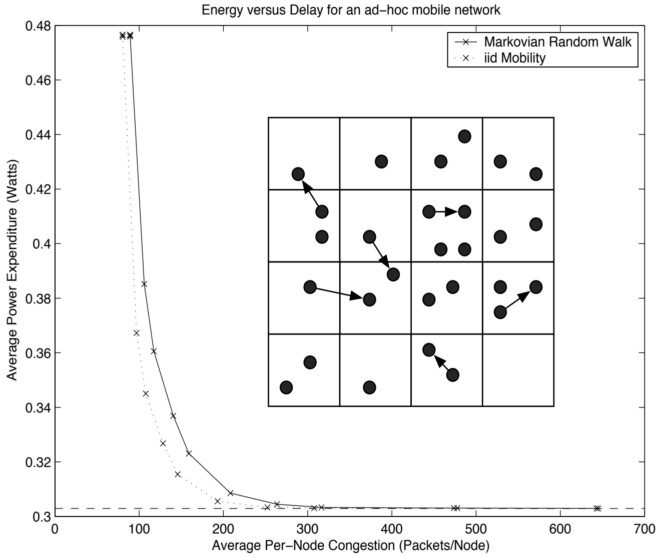

Fig. 6. An ad-hoc mobile network with adaptive transmission rates, and the resulting per node average power expenditure versus average node congestion for V between 0 and 200.

direction, it simply stays in its current cell. In the i.i.d. mobility model, nodes randomly choose new cell locations every timeslot independently and uniformly over the set of all 16 cells. It is not difficult to show that both mobility models have the same steady state node location distribution. Hence, the network capacity region and the minimum average power requried for stability are exactly the same for both mobility models (recall that Theorem 1 implies that the minimum power for stability depends only on the steady state channel distribution). In this case, the minimum power for stability under the given traffic load can be exactly computed, and is equal to 0.303 Watts.

Simulations were conducted using control parameters V in the range from 0 to 200, and the results are given in Fig. 6. In the figure, each data point represents an independent simulation for a particular value of V over the course of 4 million timeslots. The resulting per-node average power is plotted against the resulting per-node average queue congestion. From the figure, it is clear that under both mobility models, average power expenditure quickly converges to the minimum power level as the control parameter V is increased (and hence, delay is increased). The average delay under Markovian mobility is slightly larger than the delay under i.i.d. mobility. As an example set of data points, we note that for the Markovian mobility model at $V = 0$, the per-node average backlog is 89.2 packets (about 3.3 packets on average in each of the 27 internal queues), and per-node average power expenditure is 0.477 Watts. At $V = 40$, the per-node average backlog is 263.6 packets, and per-node average power expenditure is 0.305 Watts. For values of V beyond 50, average power expenditure differs from the optimal value of 0.303 only in the fourth or fifth significant digits, while average congestion continues to increase.

B. Heterogeneous Mobility and Maximum Throughput

Here we simulate the multi-hop ECCA algorithm for the same 4×4 cell partitioned network with 28 users, with the objective of maximizing total throughput subject to an average power constraint of 0.2 Watts for every node. For this maximum throughput metric, we set all θ_n^c values to 1. Source-destination pairs are the same as before, as are the packet transfer rates for in-cell and inter-cell transmission. However, here we consider the following heterogeneous mobility model: Users $0, 2, 4, 6, 8, 10, 12, 14$ are restricted to moving (uniformly) in the upper left 2×2 squares. Users $1, 3, 5, 7, 9, 11, 13, 15$ are immobile, and remain in the bottom right 2×2 squares (two in each of the four squares). All other users move uniformly throughout the network according to the Markovian mobility model.

We assume that one packet exogenously arrives to each node every timeslot, so that $A_{max} = 1$ (admission decisions for accepting/rejecting this packet are made according to the ECCA threshold rule). Further, we note that $\mu_{max}^{in} = 7$ in this case, as a given node can at most receive one in-cell transmission containing 3 packets and 4 adjacent-cell transmissions containing 1 packet. Adding this to the value A_{max} yields $\gamma = 8$, so that data is not passed to a relay node unless there is a differential backlog of 8 or more packets. According to Theorem 5, this implies that no queue will ever contain more than $V/2+1$ packets, and that the excess energy $X_n(t)$ at every node n is bounded by $\beta V/2 + \beta + 1$ (where $\beta = 3$ packets/Watt in this case). Simulation results are shown in Fig. 7. Each data point again represents a simulation over 4 million timeslots for a particular value of V, where V takes values in the range 10 to 250. In Fig. 7, it is shown that average per-node throughput quickly increases from 0.1 packets/slot (when $V = 10$) to 0.258 packets/slot (when $V = 250$). As an example, when $V = 40$ the per-node throughput is 0.257 packets/slot, and the per-node average backlog is 186.3 packets. Simulations also verify that average backlog grows linearly with V, although this data is omitted for brevity. We further note that the resulting average power expenditure at each node was almost exactly equal to the average power constraint of 0.2 Watts. This suggests that the power resources of all nodes are being fully utilized, and that total throughput would increase if the average power constraint is increased at any or all of the nodes.

These simulations verify that the given algorithms yield performance that can be pushed arbitrarily close to optimum, with a corresponding tradeoff in average backlog (and hence, average delay). While these particular simulations consider only the simple case of allocating either zero power or full power, we note that it is not difficult to implement the scheme for rate-power curves with a continuum of power choices. Furthermore, the simulations treat the simple case of cell partitioned networks, where there is no inter-cell interference. However, we emphasize that the theory we have developed applies to arbitrary rate-power curves $\vec{\mu}(\vec{P}, \vec{S})$ with any interference properties, although the resulting resource allocation problems (3) and (17) may be difficult to compute in a distributed manner. Distributed algorithms and random

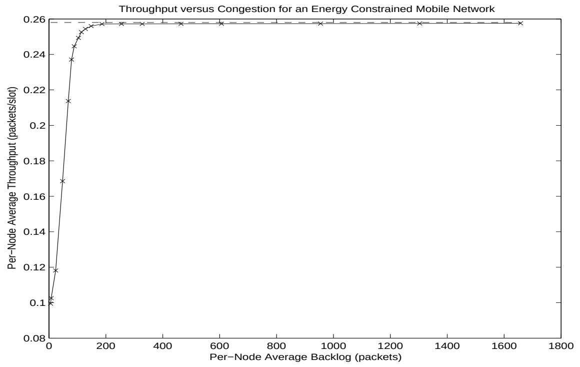

Fig. 7. Per-node throughput versus average per-node congestion for the ECCA algorithm with heterogeneous Markovian mobility.

access methods for approximately solving resource allocation problems in networks with interference are developed in [36] [22] [2] [37], and such algorithms can likely be applied in this context to yield simple distributed approximations of energy optimal control.

VIII. Conclusions

We have developed energy-efficient control strategies with performance that can be pushed arbitrarily close to optimal, with a corresponding tradeoff in average network delay. Our algorithms adapt to local link conditions without requiring knowledge of traffic rates, channel statistics, or global network topology. For simplicity of exposition, channels were modeled as being independent from slot to slot. However, the algorithms yield similar results for more general channel processes and node mobility processes, and are robust to situations when channel statistics or traffic loadings change over time [2]. The analysis presented here uses a new Lyapunov drift technique enabling stability and performance optimization to be achieved simultaneously. This research creates a general framework for designing practical control algorithms that are provably optimal.

Appendix A — Minimum Power for Stability

Here we prove Claim 1 of Theorem 1: *Consider any allocation rule for choosing $\vec{P}(t)$ subject to $\vec{P}(t) \in \Pi$, perhaps one that uses full knowledge of future arrivals and channel states. If the rule stabilizes the system, then:*

$$\underline{P}_{av} \triangleq \liminf_{t\rightarrow\infty} \frac{1}{t}\sum_{\tau=0}^{t-1}\left[\sum_{l} P_l(\tau)\right] \geq P_{av}^* \tag{33}$$

where P_{av}^* is the minimum power obtained from the optimization in Theorem 1, and $\underline{P}_{av}$ is the lim inf of the empirical average power expenditure.

To prove (33), we first establish some convenient notation. For each $\vec{S}$, define $T_{\vec{S}}(M)$ as the set of timeslots $t \in \{0, \ldots, M\}$ during which the channel state vector is equal to $\vec{S}$, and let $||T_{\vec{S}}(M)||$ represent the total number of such slots. Define the conditional empirical average of transmission rate and power consumption as follows:

$$\left(\vec{\mu}_{av}^{\vec{S}}(M); P_{av}^{\vec{S}}(M)\right) \triangleq \sum_{\tau\in T_{\vec{S}}(M)} \frac{\left(\vec{\mu}(\vec{P}(\tau),\vec{S}); \vec{1}'\vec{P}(\tau)\right)}{||T_{\vec{S}}(M)||}$$

Lemma 6: For every M, there exist probabilities $\alpha_k^{\vec{S}}(M)$ and power vectors $\vec{P}_k^{\vec{S}}(M) \in \Pi$ such that:

$$\vec{\mu}_{av}^{\vec{S}}(M) = \sum_{k=1}^{L+2} \alpha_k^{\vec{S}}(M)\vec{\mu}\left(\vec{P}_k^{\vec{S}}(M), \vec{S}\right) \tag{34}$$

$$P_{av}^{\vec{S}}(M) = \sum_{k=1}^{L+2} \alpha_k^{\vec{S}}(M)\vec{1}'\vec{P}_k^{\vec{S}}(M) \tag{35}$$

Proof: Define $\vec{\Phi}^{S}(\vec{P}) \triangleq \left(\vec{\mu}(\vec{P},\vec{S}); \vec{1}'\vec{P}\right)$ as a function mapping the L dimensional power vector into $L+1$ dimensional space. Then $\frac{1}{||T_{\vec{S}}(M)||}\sum_{\tau\in T_{\vec{S}}(M)} \vec{\Phi}(\vec{P}(\tau))$ is a convex combination of points in the image of the $L+1$ dimensional function $\vec{\Phi}^{S}(\vec{P})$ (for $\vec{P} \in \Pi$), and is therefore (by Caratheodory's theorem [28]) expressible by a convex combination of at most $L+2$ elements of the image. □

Now define:

$$\vec{\mu}_{av}(M) \triangleq \sum_{\vec{S}} \frac{||T_{\vec{S}}(M)||}{M}\vec{\mu}_{av}^{\vec{S}}(M) \tag{36}$$

$$P_{av}(M) \triangleq \sum_{\vec{S}} \frac{||T_{\vec{S}}(M)||}{M}P_{av}^{\vec{S}}(M) \tag{37}$$

For each M, the number of $\alpha_k^{\vec{S}}(M)$ and $\vec{P}_k^{\vec{S}}(M)$ values is at most $(L+2)Card(\{\vec{S}\})$ (where $Card(\{\vec{S}\})$ represents the number of possible channel state vectors). By compactness, we can thus find an appropriate subsequence of integers $\{M_i\}$ such that $M_i \rightarrow \infty$ and such that there exist limiting probabilities $\alpha_k^{\vec{S}}$ and power levels $\vec{P}_k^{\vec{S}} \in \Pi$ satisfying:

$$\vec{P}_k^{\vec{S}}(M_i) \rightarrow \vec{P}_k^{\vec{S}} \ , \ \alpha_k^{\vec{S}}(M_i) \rightarrow \alpha_k^{\vec{S}} \ , \ P_{av}(M_i) \rightarrow \underline{P}_{av} \tag{38}$$

Formally, such a subsequence $\{M_i\}$ with the above convergence properties is formed by first choosing a preliminary subsequence of integers $\{\tilde{M}_i\}$ such that $P_{av}(\tilde{M}_i) \rightarrow \underline{P}_{av}$ as $\tilde{M}_i \rightarrow \infty$. The values $\vec{P}_k^{\vec{S}}(\tilde{M}_i), \alpha_k^{\vec{S}}(\tilde{M}_i)$ can thus be viewed collectively (for all channel states $\vec{S}$ and all $k \in \{1, \ldots, L+2\}$) as an infinite sequence of vector values contained in a compact set, and hence there must exist a subsequence $\{M_i\}$ for which the values converge (as a vector) to a point in the set. Hence, the properties of (38) are satisfied.

Using (35) in (37), we have for each M_i:

$$P_{av}(M_i) = \sum_{\vec{S}} \frac{||T_{\vec{S}}(M_i)||}{M_i}\left[\sum_{k=1}^{L+2}\alpha_k^{\vec{S}}(M_i)\vec{1}'\vec{P}_k^{\vec{S}}(M_i)\right] \tag{39}$$

Because channel states are ergodic, we have $\frac{||T_{\vec{S}}(M_i)||}{M_i} \rightarrow \pi_{\vec{S}}$ for all $\vec{S}$. Thus, using (38) in (39) we have:

$$\begin{aligned}\underline{P}_{av} &= \lim_{i\rightarrow\infty} P_{av}(M_i) \\ &= \sum_{\vec{S}}\pi_{\vec{S}}\sum_{k=1}^{L+2}\alpha_k^{\vec{S}}\vec{1}'\vec{P}_k^{\vec{S}} \end{aligned} \tag{40}$$

and likewise, from (36), (34) and continuity of the rate function with respect to power, we have:[7]

$$\vec{\mu}_{av} \triangleq \lim_{i\to\infty} \mu_{av}(M_i) = \sum_{\vec{S}} \pi_{\vec{S}} \sum_{k=1}^{L+2} \alpha_k^{\vec{S}} \vec{\mu}\left(\vec{P}_k^{\vec{S}}, \vec{S}\right)$$

Now note that stability implies that the input rate to any queue is less than or equal to the $\liminf$ of the service rate (see [2], [21], Appendix D), so that $\vec{\lambda} \leq \vec{\mu}_{av}$. It follows that:

$$\vec{\lambda} \leq \sum_{\vec{S}} \pi_{\vec{S}} \sum_{k=1}^{L+2} \alpha_k^{\vec{S}} \vec{\mu}\left(\vec{P}_k^{\vec{S}}, \vec{S}\right) \tag{41}$$

From (40) and (41) it follows that $\underline{P}_{av}$ is the average power associated with a stationary power allocation scheme of the type specified by the optimization problem of Theorem 1. Because P_{av}^* is defined as the *minimum* average power over all such schemes, it follows that $\underline{P}_{av} \geq P_{av}^*$, completing the proof. □

Appendix B — The Drift Expression

Here we prove the drift expression of Lemma 2: Suppose arrivals $A_l(t)$ are i.i.d. every slot with rate $\mathbb{E}\{A_l(t)\} = \lambda_l$. For each queue l, consider the evolution equation $U_l(t+1) = \max[U_l(t) - \mu_l(\vec{P}(t), \vec{S}(t)), 0] + A_l(t)$ from (1). By squaring this equation and noting that $(\max[x, 0])^2 \leq x^2$, we obtain:

$$(U_l(t+1))^2 \leq (U_l(t))^2 + \mu_l^2 - 2U_l(t)(\mu_l - A_l) + A_l^2$$

where we have simplified the notation by writing μ_l and A_l in place of $\mu_l(\vec{P}(t), \vec{S}(t))$ and $A_l(t)$. Taking conditional expectations and summing over all l yields:

$$\begin{aligned}\Delta(\vec{U}(t)) \leq & \textstyle\sum_l \mathbb{E}\left\{\mu_l^2 + A_l^2 \mid \vec{U}(t)\right\} \\ & -2\textstyle\sum_l U_l(t)\left(\mathbb{E}\left\{\mu_l \mid \vec{U}(t)\right\} - \lambda_l\right)\end{aligned}$$

Noting that the first term on the right hand side of the above expression is bounded by $N(\mu_{max}^{out})^2 + NA_{max}^2$ proves the result. □

Appendix C — Multi-Hop ECCA

Here we prove part (c) of Theorem 5. Define the Lyapunov function $L((U_n^c), \vec{X}) = \sum_{nc}(U_n^c)^2 + \sum_n X_n^2$. The dynamics of $U_n^c(t)$ and $X_n(t)$ proceed as in (26) and (16), with the exception that exogenous arrivals $A_n^c(t)$ in (26) are replaced with *admitted arrivals* $R_n^c(t)$. The conditional Lyapunov drift (conditioned on knowledge of $U_n^c(t), \vec{X}(t)$) thus satisfies:

$$\begin{aligned}\Delta\left((U_n^c(t)), \vec{X}(t)\right) \leq & (D+C)N + 2\sum_{nc} U_n^c(t)\mathbb{E}\{R_n^c(t)\} \\ & -2\sum_{nc}\sum_{l\in\Omega_n} \mathbb{E}\{\mu_l^c(t)\}(U_{tran(l)}^c(t) - U_{rec(l)}^c(t) - \gamma) \\ & -2\sum_n X_n(t)\left[P_{av}^n - \mathbb{E}\left\{\sum_{l\in\Omega_n} P_l\right\}\right]\end{aligned}$$

where the non-negative γ term has been conveniently added to the right hand side, and where all expectations above are implicitly conditioned on knowledge of $(U_n^c(t))$ and $\vec{X}(t)$. Adding and subtracting the optimization metric $V\sum_{nc}\theta_n^c\mathbb{E}\{R_n^c(t)\}$ and rearranging terms yields:

$$\begin{aligned}\Delta\left((U_n^c(t)), \vec{X}(t)\right) \leq & (D+C)N + V\sum_{nc}\theta_n^c\mathbb{E}\{R_n^c(t)\} \\ & +2\sum_{nc}[U_n^c(t) - \theta_n^c V/2]\,\mathbb{E}\{R_n^c(t)\} \\ & -2\sum_{nc}\sum_{l\in\Omega_n} \mathbb{E}\{\mu_l^c(t)\}(U_{tran(l)}^c(t) - U_{rec(l)}^c(t) - \gamma) \\ & -2\sum_n X_n(t)\left[P_{av}^n - \mathbb{E}\left\{\sum_{l\in\Omega_n} P_l\right\}\right]\end{aligned}$$

The Multi-Hop ECCA admission control policy was designed to minimize the third term on the right hand side of the above inequality, while the power allocation and routing policies were designed to minimize the fourth and fifth terms. Thus, the right hand side is less than or equal to the resulting expression when these terms use expectations corresponding to the optimal *stationary* control policy, where $\mathbb{E}\{R_n^c(t)\} = R_{nc}^*$, $\mathbb{E}\{\sum_{l\in\Omega_n} P_l\} \leq P_{av}^n$, and $\mathbb{E}\{\mu_l^c\} = f_l^c$, where (f_l^c) are flows satisfying $\sum_{nc}\sum_{l\in\Omega_n} f_l^c(U_{tran(l)}^c(t) - U_{rec(l)}^c(t)) = \sum_{nc} U_n^c(t)R_{nc}^*$ (see [2] [22]). Plugging these expressions into the right hand side of the above inequality creates many terms that can be cancelled, yielding:

$$\Delta \leq \tilde{C}N + V\sum_{nc}\theta_n^c\mathbb{E}\{R_n^c(t)\} - V\sum_{nc}\theta_n^c R_{nc}^*$$

which proves the result upon application of Lemma 1. □

Appendix D — Virtual Queues

Here we prove a general queue stability fact that directly implies the result of Lemma 3. Consider any discrete time queue with unfinished work function $U(t)$ and with any general arrival and transmission rate processes $A(t)$ and $\mu(t)$, where $U(t+1) = \max[U(t) - \mu(t), 0] + A(t)$. Suppose the queue is *strongly stable*, so that:

$$\limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\{U(\tau)\} = M < \infty$$

Lemma 7: If a queue is strongly stable and has a transmission rate $\mu(t)$ that is upper bounded, so that $\mu(t) \leq \mu_{max}$ for all t, then:

$$\liminf_{t\to\infty}\left[\overline{\mu}(t) - \overline{A}(t)\right] \geq 0$$

where $\overline{\mu}(t) \triangleq \frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\{\mu(\tau)\}$, and $\overline{A}(t) \triangleq \frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\{A(\tau)\}$.

Proof: First suppose that the following inequality holds:

$$\limsup_{t\to\infty}\mathbb{E}\{U(t)\}/t = 0 \tag{42}$$

Now note that for any time t we have:

$$U(t) \geq U(0) + \sum_{\tau=0}^{t-1}A(\tau) - \sum_{\tau=0}^{t-1}\mu(\tau)$$

Dividing by t and taking expectations yields:

$$\frac{\mathbb{E}\{U(t)\}}{t} \geq \frac{\mathbb{E}\{U(0)\}}{t} + \left[\overline{A}(t) - \overline{\mu}(t)\right]$$

[7] Upper semi-continuity can be used here to obtain the same inequality (41) for the more general case of discontinuous rate-power functions [2].

Taking the $\limsup$ of both sides and using (42) yields $0 \geq \limsup_{t\to\infty}\left[\overline{A}(t) - \overline{\mu}(t)\right]$, and hence $0 \leq \liminf_{t\to\infty}\left[\overline{\mu}(t) - \overline{A}(t)\right]$.

It thus suffices to prove that (42) is satisfied whenever the conditions of the lemma hold. To show this, suppose that (42) does *not* hold, so that there exists a value $\epsilon > 0$ and a subsequence of times $\{t_n\}$ such that $t_n \rightarrow \infty$ and $\mathbb{E}\left\{U(t_n)\right\}/t_n \geq \epsilon$. We reach a contradiction.

Choose any arbitrarily large constant V such that $V > M$, and let T_n denote the number of timeslots after time t_n until $\mathbb{E}\left\{U(t)\right\}$ crosses below the V threshold. If $\mathbb{E}\left\{U(t_n)\right\} < V$, then we define T_n to be 0. Note that T_n is finite for all n, as otherwise $\mathbb{E}\left\{U(t)\right\} \geq V$ for all $t \geq t_n$, which would contradict the fact that the $\limsup$ time average expected value of $U(t)$ is equal to M. Because transmission rates are upper bounded by μ_{max}, we have for any time $t > t_n$:

$$\begin{aligned}\mathbb{E}\left\{U(t)\right\} &\geq \mathbb{E}\left\{U(t_n)\right\} - \mu_{max}(t - t_n) \\ &\geq \epsilon t_n - \mu_{max}(t - t_n)\end{aligned}$$

and hence $\mathbb{E}\left\{U(t)\right\} \geq V$ whenever $(t - t_n) \leq (\epsilon t_n - V)/\mu_{max}$. It follows that $T_n \geq (\epsilon t_n - V)/\mu_{max}$. Hence:

$$\frac{T_n}{t_n} \geq \frac{\epsilon}{\mu_{max}} - \frac{V}{t_n \mu_{max}}$$

and so $\liminf_{t_n\to\infty} T_n/t_n \geq \epsilon/\mu_{max}$, implying that $\limsup_{t_n\to\infty} t_n/T_n \leq \mu_{max}/\epsilon$. However, note by definition that:

$$\begin{aligned}\frac{1}{t_n + T_n}\sum_{\tau=0}^{t_n+T_n-1}\mathbb{E}\left\{U(\tau)\right\} &\geq V\frac{T_n}{t_n + T_n} \\ &= V\frac{1}{t_n/T_n + 1}\end{aligned}$$

Taking limits of the above inequality and recalling that $M = \limsup_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\left\{U(\tau)\right\}$, we have:

$$M \geq V\frac{1}{\mu_{max}/\epsilon + 1}$$

However, this inequality holds for arbitrarily large values of V, contradicting the fact that M is finite. □

REFERENCES

[1] M. J. Neely. Energy optimal control for time varying wireless networks. *Proceedings of IEEE INFOCOM*, March 2005.

[2] 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.

[3] A. Jalali, R. Padovani, and R. Pankaj. Data throughput of cdma-hdr a high efficiency data rate personal communication wireless system. *IEEE Vehicular Technology Conference*, May 2000.

[4] L. Tsaur and D. C. Lee. Closed-loop architecture and protocols for rapid dynamic spreading gain adaptation in cdma networks. *IEEE Proc. of INFOCOM*, 2004.

[5] J. W. Lee, R. R. Mazumdar, and N. B. Shroff. Downlink power allocation for multi-class cdma wireless networks. *IEEE Proceedings of INFOCOM*, 2002.

[6] P. Marbach and R. Berry. Downlink resource allocation and pricing for wireless networks. *IEEE Proc. of INFOCOM*, 2002.

[7] L. Xiao, M. Johansson, and S. Boyd. Simultaneous routing and resource allocation for wireless networks. *Proc. of the 39th Annual Allerton Conf. on Comm., Control, Comput.*, Oct. 2001.

[8] D. Julian, M. Chiang, D. O'Neill, and S. Boyd. Qos and fairness constrained convex optimization of resource allocation for wireless cellular and ad hoc networks. *Proc. INFOCOM*, 2002.

[9] B. Krishnamachari and F. Ordonez. Analysis of energy-efficient, fair routing in wireless sensor networks through non-linear optimization. *IEEE Vehicular Technology Conference*, Oct. 2003.

[10] P. Giaccone, B. Prabhakar, and D. Shah. Energy constrained wireless switching. *Proc. of Allerton Conference on Communications, Control, and Computing*, Oct. 2003.

[11] U. C. Kozat, I. Koutsopoulos, and L. Tassiulas. A framework for cross-layer design of energy-efficient communication with qos provisioning in multi-hop wireless networks. *INFOCOM*, 2004.

[12] R. Cruz and A. Santhanam. Optimal routing, link scheduling, and power control in multi-hop wireless networks. *IEEE Proceedings of INFOCOM*, April 2003.

[13] E. M. Yeh and A. S. Cohen. Throughput optimal power and rate control for multiaccess and broadcast communications. *Proc. of the International Symposium on Information Theory*, June 2004.

[14] L. Tassiulas and A. Ephremides. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *IEEE Transacations on Automatic Control*, Vol. 37, no. 12, Dec. 1992.

[15] L. Tassiulas and A. Ephremides. Dynamic server allocation to parallel queues with randomly varying connectivity. *IEEE Trans. on Inform. Theory*, vol. 39, pp. 466-478, March 1993.

[16] P.R. Kumar and S.P. Meyn. Stability of queueing networks and scheduling policies. *IEEE Trans. on Automatic Control*, Feb. 1995.

[17] N. McKeown, V. Anantharam, and J. Walrand. Achieving 100% throughput in an input-queued switch. *Proc. INFOCOM*, 1996.

[18] N. Kahale and P. E. Wright. Dynamic global packet routing in wireless networks. *IEEE Proceedings of INFOCOM*, 1997.

[19] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, and P. Whiting. Providing quality of service over a shared wireless link. *IEEE Communications Magazine*, 2001.

[20] E. Leonardi, M. Mellia, F. Neri, and M. Ajmone Marsan. Bounds on average delays and queue size averages and variances in input-queued cell-based switches. *Proc. INFOCOM*, 2001.

[21] M. J. Neely, E. Modiano, and C. E. Rohrs. Power allocation and routing in multi-beam satellites with time varying channels. *IEEE Transactions on Networking*, Feb. 2003.

[22] M. J. Neely, E. Modiano, and C. E Rohrs. Dynamic power allocation and routing for time varying wireless networks. *IEEE Journal on Selected Areas in Communications*, January 2005.

[23] E. Uysal-Biyikoglu, B. Prabhakar, and A. El Gamal. Energy-efficient packet transmission over a wireless link. *IEEE/ACM Trans. Networking*, vol. 10, pp. 487-499, Aug. 2002.

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

[25] 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.

[26] L. Li and A. Goldsmith. Capacity and optimal resource allocation for fading broadcast channels: Part i: Ergodic capacity. *IEEE Trans. Inform. Theory*, ppl. 1083-1102, March 2001.

[27] X. Liu, E. K. P. Chong, and N. B. Shroff. A framework for opportunistic scheduling in wireless networks. *Computer Networks*, vol. 41, no. 4, pp. 451-474, March 2003.

[28] D. P. Bertsekas, A. Nedic, and A. E. Ozdaglar. *Convex Analysis and Optimization*. Boston: Athena Scientific, 2003.

[29] F.P. Kelly, A.Maulloo, and D. Tan. Rate control for communication networks: Shadow prices, proportional fairness, and stability. *Journ. of the Operational Res. Society*, 49, p.237-252, 1998.

[30] M. J. Neely, E. Modiano, and C. Li. Fairness and optimal stochastic control for heterogeneous networks. *Proceedings of IEEE INFOCOM*, March 2005.

[31] J. W. Lee, R. R. Mazumdar, and N. B. Shroff. Opportunistic power scheduling for dynamic multiserver wireless systems. *IEEE Transactions on Wireless Communications*, vol. 5, no.5, May 2006.

[32] A. Stolyar. Maximizing queueing network utility subject to stability: Greedy primal-dual algorithm. *Queueing Systems*, vol. 50, pp. 401-457, 2005.

[33] Y. H. Lin and R. Cruz. Opportunistic link scheduling, power control, and routing for multi-hop wireless networks over time-varying channels. *Proceedings of 43rd Annual Allerton Conference on Communication, Control, and Computing*, September 2005.

[34] M. El-Taha and S. Stidham Jr. *Sample-Path Analysis of Queueing Systems*. Kluwer Academic Publishers, Boston:, 1999.

[35] F. M. Guillemin and R. R. Mazumdar. On pathwise analysis and existence of empirical distributions for g/g/1 queues. *Stoc. Proc. Appl.*, Vol. 67 (1), 1997.

[36] M. Grossglauser and D. Tse. Mobility increases the capacity of ad-hoc wireless networks. *Proceedings of IEEE INFOCOM*, 2001.

[37] X. Lin and N. B. Shroff. The impact of imperfect scheduling on cross-layer rate control in wireless networks. *IEEE Proceedings of INFOCOM*, 2005.