Revenue Maximization in Time-Varying Multi-Hop Wireless Networks: A Dynamic Pricing Approach

Yang Song, Chi Zhang, *Member, IEEE,* Yuguang Fang, *Fellow, IEEE,* and Phone Lin, *Senior Member, IEEE*

***Abstract*—In this paper, we study a wireless multi-hop network where multiple flows co-exist and share the network resource collectively. Each flow is associated with a user which has specific requirements on its tradeoff between cost and quality of service. To support heterogeneous transmissions efficiently, we propose a quality-aware dynamic pricing algorithm, namely, QADP, which provably maximizes the overall network revenue while maintaining the stability of the network. Our proposed scheme enjoys the merit of self-adaptability due to its online nature.**

***Index Terms*—Revenue Maximization, Network Stability, Dynamic Pricing, Service Differentiation**

I. Introduction

In RECENT years, multi-hop wireless networks have made significant advance in both academic and industrial aspects. Besides traditional data services, multimedia transmissions become an indispensable component of network traffic nowadays. For example, people can watch live games while listening to online musical stations at the same time. Therefore, supporting multimedia services in multi-hop wireless networks effectively and efficiently has received intensive attention from the community.

Multimedia flows usually impose application-specific requirements on the minimum average attainable data rates and different flows may have distinct rate requirements. In addition, multimedia flows, especially wireless video transmissions, usually impose additional requirements on maximum end-to-end delays. For example, a multimedia stream for video surveillance may need a lower data transmission rate compared to high quality video-on-demand movie transmissions, whereas a much more stringent delay requirement is imposed. Therefore, the network inclines to allocate more network resource to those delay-imperative multimedia transmissions.

Manuscript received on 16 August 2011; revised on 16 January 2012 and 8 April 2012. This work was supported in part by the U.S. National Science Foundation under Grants CNS-0721744 and CNS-0916391. The work of Y. Fang was also partially supported by the 111 project under grant B08038 with Xidian University, Xi'an, China. The work of P. Lin was partially supported by NSC 100-2221-E-002-176 and NSC 100-2221-E-002-184.

Y. Song is with IBM Research, Hawthorne, NY (e-mail: yangsong@us.ibm.com).

C. Zhang is with the School of Information Science and Technology, University of Science and Technology of China (e-mail: chizhang@ustc.edu.cn).

Y. Fang is with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611 (e-mail: fang@ece.ufl.edu). He was also a Changjiang Scholar Chair Professor with the National Key Laboratory of Integrated Services Networks, Xidian University, Xi'an 710071, China.

P. Lin is with the Department of Computer Science and Information Engineering, National Taiwan University, Taipei 106, R.O.C. (e-mail: plin@pcs.csie.ntu.edu.tw).

Digital Object Identifier 10.1109/JSAC.2012.120809.

In this paper, we investigate the resource allocation problem in multi-hop wireless networks, from a network administrator's perspective. For each flow, the network charges a certain amount of admission fee in order to build up a system-wide revenue. The price imposed on each flow is subject to adaptation in order to obtain the optimum revenue. In addition, the network provides a set of *service levels*, denoted by $\ell = 1, \cdots, L$ where level one has the highest priority in the system with respect to delay guarantees. Note that L can be arbitrarily large. Each multimedia flow, according to its application requirement, proposes a service level request to the network. For example, a background traffic for movie downloading might have level ten whereas a VoD online movie transmission may demand a service level of two. Therefore, it is desired to design an efficient pricing algorithm which provides a service differentiation solution. To achieve this, we propose a quality-aware dynamic pricing algorithm, namely, QADP, which provably accumulates a network revenue that is arbitrarily close to the optimum solution while maintaining network stability under time varying channel conditions. Moreover, the guaranteed maximum average end-to-end delay for service level one traffic is j times less than that of level j transmissions, where $j = 1, \cdots, L$. From a cost-QoS tradeoff standpoint, if a multimedia flow demands level one service, the transmissions have the minimum end-to-end delay bound and thus represent the highest priority yet the imposed price might be significantly higher than others. Therefore, in the quality of experience (QoE) context, our framework allows each flow to determine the service level it demands, with an anticipation that the price charged by the network and the experienced QoS will reflect the selected priority level. For example, if one flow enjoys a better QoS, the cost charged by the network will also increase and hence may cause a degraded quality of experience from the user's viewpoint. The tradeoff between cost and network service experienced is captured by the objective function of each client user, which generally represents the quality of experience requirement of the user. Our framework provides the flexibility by allowing users to exploit the tradeoff between cost and experienced QoS in order to achieve a desired balance to fulfill their QoE requirements. In addition, the QADP algorithm is self-adaptive to the changes of statistical characteristics of time-varying channels.

II. Related Work

The interconnected queue network model has attracted significant attention since the seminal work of [1] where the well-known *MaxWeight* scheduling algorithm is proposed.

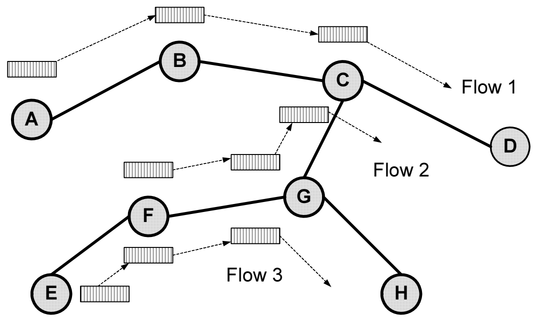

Fig. 1. Network topology with interconnected queues.

Neely extends the results into a general time varying setting in [2], [3], based on which a pioneering stochastic network optimization technique is developed [4], [5]. Unfortunately, in the literature, few work has been devoted to addressing the issue of prioritizing multimedia flows in practical wireless settings. In addition, although revenue maximization problem has been studied extensively in the literature, in general, either the quality provisioning issue is not particularly addressed [6]–[9], or the network model is restricted to wired networks where the channel conditions of the system are assumed to be time-invariant and remain unchanged [10], [11]. In [12], the issue of dynamic pricing is studied to maximize the average profit where only single hop wireless networks are considered. [13]–[15] and [16] addressed the pricing scheme design in spectrum sharing among multiple cognitive radios. However, the issues of network stability and scheduling were not discussed.

III. Revenue Maximization in Multi-hop Wireless Networks

A. System Model

We consider a static multi-hop wireless network represented by a directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, illustrated in Figure 1, where $\mathcal{N}$ is the set of nodes and $\mathcal{E}$ is the set of links. The numbers of nodes and links in the network are denoted by N and E, respectively. A link is denoted either by $e \in \mathcal{E}$ or $(a, b) \in \mathcal{E}$ where a and b are the transmitter and the receiver of the link. Time is slotted, i.e., $t = 0, 1, \cdots$. For link (a, b), the instantaneous channel condition at time slot t is denoted by $S_{a,b}(t)$. For example, $S_{a,b}(t)$ can represent the time varying fading factor on link (a, b) at time t, or the packet loss ratio of the particular link. Denote $\mathbf{S}(t)$ as the channel condition vector on all links. We assume that $\mathbf{S}(t)$ remains constant during a time slot. However, $\mathbf{S}(t)$ may change on slot boundaries. We assume that there are a finite but arbitrarily large number of possible channel condition vectors and $\mathbf{S}(t)$ follows an arbitrary yet unknown distribution. At each time slot t, given the channel state vector $\mathbf{S}(t)$, the network controller chooses a *link schedule*, denoted by $I(t)$, from a feasible set $\Upsilon_{\mathbf{S}(t)}$, which is restricted by factors such as underlying interference model, duplex constraints or peak power limitations. For a wireless link (a, b), the link data rate $\mu_{a,b}(t)$ is a function of $I(t)$ and $\mathbf{S}(t)$. We denote $\boldsymbol{\mu}(t)$ as the vector of link rates of all links at time slot t.

There are C commodities, a.k.a., flows, in the network, where each commodity, say $c, c = 1, 2, \cdots, C$, is associated with a routing path $P_c = \{c(0), c(1), \cdots, c(\kappa_c)\}$ where $c(0)$ and $c(\kappa_c)$ are the source and the destination node of flow c while $c(j)$ denotes the j-th hop node on its path. Without loss of generality, we assume that every node in the network initiates at most one flow[1]. However, multiple flows can intersect at any node in the network. Each node maintains a separate queue for every flow that passes through it. For each flow c, denote $A_c(t)$ as the exogenous arrival to the *transport layer* of node $c(0)$ during time slot t. We assume that the stochastic arrival process, i.e., $A_c(t)$, has an expected average rate of λ_c. For a single queue, define the *overflow function* [2] as

$$g(B) = \limsup_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} Pr(Q(\tau) > B) \tag{1}$$

where $Q(\tau)$ is the *queue backlog* at time τ. We say the queue is *stable* if $\lim_{B\to\infty} g(B) \to 0$ [2], [3] and a network is stable if all the queues in this network are stable.

Denote $\boldsymbol{\lambda} = \{\lambda_1, \cdots, \lambda_C\}$ as the *arrival rate vector* of the network. Note that all the arrival rate vectors are defined in an average sense. A flow control mechanism is implemented where during time slot t, an amount of $R_c(t)$ traffic is admitted to the network layer for flow c. The *network capacity region*, a.k.a., the *network stability region*, denoted by Ω, is defined as all the admission rate vectors that can be supported by the network, in the sense that there exists a policy that stabilizes the network under this admission rate. In this work, we consider a heavy traffic scenario where the arrival rate vector $\boldsymbol{\lambda}$ lies outside of the capacity region Ω for all time slots. Moreover, in this paper, we are particularly interested in heterogeneous transmissions, such as multimedia traffic, where each flow c has specific requirements on its tradeoff between cost and QoS metrics, e.g., rate and delay. To be specific, each flow c has a minimum data rate requirement α_c as well as a service level request, denoted by $\ell_c, 1 \leq \ell_c \leq L$. Denote $\boldsymbol{\alpha} = \{\alpha_1, \cdots, \alpha_C\}$ and $\boldsymbol{\ell} = \{\ell_1, \cdots, \ell_C\}$ as the *minimum rate vector* and the *service level request vector* of the network where ℓ_1 has the highest priority in terms of the delay guarantees provided by the network. It is worth noting that the smaller (α_c, l_c) is, the more expensive the cost (from the flow's point of view) is expected to occur. Throughout this paper, we assume that the minimum rate vector, i.e., $\boldsymbol{\alpha}$, is *inside* of the network capacity region Ω. Since that if $\boldsymbol{\alpha}$ is inherently not feasible, we cannot expect to find any policy to meet those demands and the only solution is to increase the network's information-theoretic capacity by traditional methods such as adding more channels, radios, enabling network coding, or utilizing MIMO techniques with multiple antennas.

B. Problem Formulation

Denote the *queue backlog* of node n for flow c as $Q_n^c(t)$. Note that $Q_{c(\kappa_c)}^c \equiv 0$ since whenever a packet reaches the destination, it is considered as leaving the network. The queue updating dynamic of $Q_n^c(t)$ is given as follows. For $j = 1, \cdots, \kappa_c - 1$, we have

$$Q_{c(j)}^c(t+1) \leq [Q_{c(j)}^c(t) - \mu_{c(j),c}^{out}(t)]^+ + \mu_{c(j),c}^{in}(t) \tag{2}$$

[1]If a node initiates more than one flow in the network, we can replace this node with multiple duplicate nodes and the following analysis still holds.

and for $j = 0$,

$$Q^c_{c(j)}(t+1) = [Q^c_{c(j)}(t) - \mu^{out}_{c(j),c}(t)]^+ + R_c(t) \tag{3}$$

where $[x]^+$ denotes $\max(x, 0)$ and $\mu^{in}_{n,c}(t)$, $\mu^{out}_{n,c}(t)$ represent the allocated data rate of the incoming link and the outgoing link of node n, by the scheduling algorithm, with respect to flow c. Note that (2) is an inequality since the previous hop node may have less packets to transmit than the allocated data rate $\mu^{in}_{c(j),c}(t)$.

During time slot t, $p_c(t)$ is charged for flow c as the *per unit flow* price. The functionality of the price is not only to control the admitted flows, but also, more importantly, to build up a system-wide revenue from the network's perspective. We further assume that each flow is associated with a particular user and thus we will use *flow* and *user* interchangeably. Every user c is assumed to have a concave, differentiable utility function $\mathcal{O}_c(R_c(t))$ which reflects the degree of satisfaction by transmitting with data rate $R_c(t)$. At time slot t, user c selects a data rate which optimizes the net income, a.k.a., *surplus*, i.e.,

$$R_c(t) = \text{argmax}_r \left(\mathcal{O}_c(r) - r \times p_c(t)\right) \quad \forall c = 1, \cdots, C, \tag{4}$$

where $r \in [0, R^{\max}_c]$ and $R^{\max}_c$ is the upper bound of admitted traffic of flow c during one time slot, i.e., $R_c(t) \leq R^{\max}_c, \forall c, t$. For example, $R^{\max}_c$ can represent the hardware limitation on the maximum volume of traffic that a node can admit during one time slot. Note that r is the admitted rate by user which is different from the link rate $\mu_{a,b}(t)$. In this work, we assume that the user's backlog queue will have sufficient packet to transmit. Therefore, with (4), the network accumulates revenue while performing admission control. In this paper, as an example, we consider that

$$\mathcal{O}_c(r) = \log(1 + r) \tag{5}$$

which is the natural logarithmic function. We emphasize that the following analysis can be extended to other heterogeneous forms of utility functions straightforwardly. From a QoE perspective, the utility function captures the actual *experience* of the client user which is more than the network service received. According to budget and sensitivity to price changes, each client user is able to choose an appropriate utility function which represents the QoE preference of the user. In the following sections, we will show that our proposed scheme can also ensure the network resources allocated to users are determined by the QoE requirements of users which are reflected by their utility functions and service level requests jointly. The assumption of (4) and (5) is only for simplicity, since it represents the knowledge of the service provider on the user's sensitivity to price settings. In other words, we assume that the service provider has certain knowledge or expectation on how users would respond provided an imposed price, a.k.a., the price of elasticity of demand (PED) [17], which can be determined by various of statistical methods such as historical data analysis or conjoint analysis approach [18]. Furthermore, note that the fairness issue of multiple flows can be solved by choosing utility functions properly. For example, a utility function of $\log(r)$ represents the proportional fairness among competitive flows. For more discussions, refer to [19] and [20]. Our framework can be extended to other objective functions such as sigmoid function which represents a wide range of real-time multimedia traffic scenarios.

From the network administrator's perspective, the overall network-wide revenue is the target to be maximized. Meanwhile, the stability of the network as well as the service requirements from multimedia flows need to be addressed. Formally speaking, the objective of the network is to find an optimal policy to

<u>Revenue Maximization Problem:</u>

$$\text{maximize} \quad D = \liminf_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} O(\tau) \tag{6}$$

s.t.

(a) the network is stable,
(b) the minimum data rate requirements, i.e., $\boldsymbol{\alpha}$, are satisfied,
(c) the guaranteed maximum end-to-end delays for multiple multimedia flows are prioritized according to the service levels of $\boldsymbol{\ell}$,

where

$$O(t) = E\left(\sum_c R_c(t) p_c(t)\right) \tag{7}$$

is the *expected* overall network revenue during time slot t, with respected to the randomness of arrival processes and channel variations.

IV. Quality-Aware Dynamic Pricing (QADP) Algorithm

In this section, we propose an online policy, i.e., QADP algorithm, which solves the revenue maximization problem in (6).

Let μ^{max} be the maximum data rate on *any* link of the network, which may be determined by factors such as the number of antennas, modulation schemes and coding policies. In addition, for each flow c, we introduce a *virtual queue* $Y_c(t)$ which is initially empty, and the queue updating dynamic is defined as

$$Y_c(t+1) = [Y_c(t) - R_c(t)]^+ + \alpha_c \quad \forall c. \tag{8}$$

Note that virtual queues are easy to implement. For example, the source node of flow c, i.e., $c(0)$, can maintain a software based counter to measure the backlog updates of virtual queue $Y_c(t)$. In addition, for each flow c, we define

$$\delta_c = N(\mu^{\max})^2 + (R^{\max}_c)^2 + \frac{1}{2}(\alpha_c)^2 \quad \forall c. \tag{9}$$

Denote $\theta^1, \cdots, \theta^C$ as the weights which will be calculated and assigned to all flows, where C denotes the number of flows in the network. Let J be a tunable[2] positive large number determined by the network. In addition, we assume a maximum value of the allocated weight, denoted by $\theta^{\max}$, i.e., $\theta^c \leq \theta^{\max}, \forall c$. The proposed QADP algorithm is given as follows.

[2]The impact of J on the performance of QADP algorithm will be clarified shortly.

QADP ALGORITHM:

- ***Part I: Weight Assignment***
  For all multimedia transmissions, find the flow with the minimum value of $\alpha_c \times \ell_c, c = 1, \cdots, C$, say, flow j. For each flow c, assign an associated *weight*, denoted by θ^c, which is calculated by

$$\theta^c = \frac{\theta^{\max} \times \alpha_j \times \ell_j}{\alpha_c \times \ell_c}, \quad \forall c = 1, \cdots, C. \tag{10}$$

- ***Part II: Dynamic Pricing***
  For every time slot t, the source node of flow c, i.e., $c(0)$, measures the value of $Q^c_{c(0)}(t)$ and $Y_c(t)$. If $Q^c_{c(0)}(t) > Y_c(t)$, the instantaneous admission price is set as

$$p_c(t) = \sqrt{\frac{\theta^c \left(Q^c_{c(0)}(t) - Y_c(t)\right)}{J}} \tag{11}$$

  and $p_c(t) = 0$ otherwise.
- ***Part III: Scheduling***
  For every time slot t, find a link schedule $I^*(t)$, from the feasible set $\Upsilon_{\mathbf{S}(t)}$, which solves

$$\max_{I(t) \in \Upsilon_{\mathbf{S}(t)}} \sum_{(a,b) \in \mathcal{E}} \mu_{a,b}(t) \xi_{a,b} \tag{12}$$

  where

$$\xi_{a,b} = \max_{c:(a,b) \in P_c} \left(\theta^c (Q^c_a(t) - Q^c_b(t))\right) \tag{13}$$

  if $\exists c$, such that $(a, b) \in P_c$, and $\xi_{a,b} = 0$ otherwise.

It is worth noting that Part I of QADP can be precalculated before actual transmissions. The value of θ^c represents the "importance" of flow c and remains unchanged unless the values of $(\boldsymbol{\alpha}, \boldsymbol{\ell})$ are updated, by which a new weight calculation is triggered.

The dynamic pricing part is the key component of QADP. By following (11), not only the incoming admitted rates can be regulated effectively, but also the overall average network revenue can be maximized, as will be shown shortly. Note that after the weight assignment, in order to compute $p_c(t)$, the source node of flow c, i.e., $c(0)$, which is considered as the edge node of the network, requires only local information, i.e., current backlogs of the source data queue and the virtual queue. Note that in practice, a fixed price, or flat price, can be utilized for the sake of simplicity, where the fixed price is calculated based on the underlying dynamic pricing scheme. In this work, we focus on dynamic pricing schemes in order to understand the performance gain introduced by our proposed service differentiation solution. The design of fixed pricing scheme is out of the scope of this paper and remains as future research.

The third part of QADP is a *weighted* extension to the well-known *MaxWeight* scheduling algorithm [1], [2], [21]. Instead of the exact difference of queue backlogs, we deliberately select the weighted difference of queue backlogs as the weight of a particular link in the scheduling algorithm. Intuitively, if a flow is assigned with a larger value of θ^c, the links associated with it will have a higher possibility of being selected for transmissions by QADP. Therefore, by assigning proper values of θ^c to flows with different requirements, a service differentiation can be achieved. In addition, as indicated by (11), a higher priority needs to pay at a higher price. Therefore, a service differentiation solution which strikes the balance between cost and multimedia delivery quality is achieved. Note that to calculate (12), QADP needs to solve a complex optimization problem which requires a global information on channel states, i.e., $\mathbf{S}(t)$. However, availed of the prosperous development of distributed scheduling schemes, such as [22]–[25], the difficulty of centralized computation can be lessened.

V. PERFORMANCE ANALYSIS

In this section, we provide the main result on the performance of QADP algorithm.

Theorem 1: Define D^* as the optimum solution of (6). For QADP algorithm, we have

(a) Revenue Maximization

$$\liminf_{t \to \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} O(\tau) \geq D^* - \frac{K}{J} \tag{14}$$

where K is a constant and is given by

$$K = \sum_c \theta^c \delta_c \tag{15}$$

and δ_c is defined in (9).

(b) Network Stability
The network is stable under QADP algorithm.

(c) Service Differentiation
By following QADP algorithm, any *feasible* minimum data rate requirements $\boldsymbol{\alpha}$ can be satisfied. In addition, the guaranteed maximum average end-to-end delays for multimedia flows with service level j are j times larger than that of level one transmissions.

It can be observed that in (14), the achieved performance of QADP algorithm can be pushed arbitrarily close to the optimum solution D^* by selecting a sufficiently large value of J. However, as will be shown in Section VI, a large J also increases the average network delay. The proof of Theorem 1 is provided in the following.

A. Proof of Revenue Maximization

Since the minimum rate vector $\boldsymbol{\alpha}$ is assumed to lie *inside* the capacity region Ω, there exists a small positive number $\tilde{\epsilon} > 0$ such that $\boldsymbol{\alpha} + \tilde{\epsilon}\mathbf{1} \in \Omega$ where $\mathbf{1}$ is a unity vector with dimension C.

Lemma 1: For any *feasible* input rate vector $\boldsymbol{\vartheta}$, there exists a stationary[3] randomized policy, denoted by *RAND*, which generates

$$E\left(\mu^{out}_{n,c} - \mu^{in}_{n,c} - \vartheta^c_n(t)\right) = 0 \quad \forall n, c, t \tag{16}$$

and

$$E(\vartheta^c_{c(0)}(t)) \geq \alpha_c + \tilde{\epsilon} \quad \forall t, c \tag{17}$$

[3]Stationary means that the probabilistic structure of the randomized policy does not change with different values of queue backlogs.

where $\vartheta_n^c(t)$ is the exogenous arrival on node n for flow c during time slot t, and hence it is nonzero at the source node of session c only.

The proof of Lemma 1 follows similar lines as in [2], [4], [5] and is omitted.

Recall that a virtual queue $Y_c(t)$ is introduced for every flow c and the queue updating dynamic is given by (8). As a result, the minimum data rate requirement is converted to a queue stability problem since if the virtual queue $Y_c(t)$ is stable, the average service rate, i.e., the time average of $R_c(t)$, needs to be greater than the average arrival rate, i.e., α_c. Define $\boldsymbol{Z}(t) = [\boldsymbol{Q}(t); \boldsymbol{Y}(t)]$ as all the real data queues and virtual queues at time slot t. If we can ensure that the network is stable with respect to $\boldsymbol{Z}(t)$, the backlogs of real queues are bounded and the minimum rate requirements are achieved at the same time.

Define a system-wide *potential function (PF)* as

$$PF(\boldsymbol{Z}(t)) = \sum_c PF^c(\boldsymbol{Z}(t)) \tag{18}$$

where

$$PF^c(\boldsymbol{Z}(t)) = \frac{1}{2}\left(\sum_n \theta^c (Q_n^c(t))^2 + \theta^c (Y_c(t))^2\right). \tag{19}$$

Note that $PF(\boldsymbol{Z}(t))$ is a scalar-valued nonnegative function. Define

$$\Delta(\boldsymbol{Z}(t)) = E\left(PF(\boldsymbol{Z}(t+1)) - PF(\boldsymbol{Z}(t)) | \boldsymbol{Z}(t)\right) \tag{20}$$

as the *drift* of the potential function $PF(\boldsymbol{Z}(t))$.

For flow c, we take the square of both sides of (2), (3), and (8) to obtain

$$\begin{aligned} &PF^c(\boldsymbol{Z}(t+1)) - PF^c(\boldsymbol{Z}(t)) \\ &\leq \Xi_c + \theta^c Q_{c(0)}^c(t) R_c(t) - \theta^c Y_c(t)(R_c(t) - \alpha_c) \\ &- \sum_n \theta^c Q_n^c(t)\left(\mu_{n,c}^{out}(t) - \mu_{n,c}^{in}(t)\right) \end{aligned} \tag{21}$$

where

$$\Xi_c = \theta^c \left(N(\mu^{\max})^2 + (R_c^{\max})^2 + \frac{1}{2}(\alpha_c)^2\right). \tag{22}$$

Note that (21) is summed over the whole network. If node n is not on the path of flow c, $\mu_{n,c}^{in}(t) = \mu_{n,c}^{out}(t) = 0$. Moreover, $\mu_{n,c}^{in}(t) = 0$ for the source node of flow c and $\mu_{n,c}^{out}(t) = 0$ for the destination node of flow c. Next, we sum over all flows to derive the network-wide *potential function difference* as

$$\begin{aligned} &PF(\boldsymbol{Z}(t+1)) - PF(\boldsymbol{Z}(t)) \\ &\leq K - \sum_{n,c} \theta^c Q_n^c(t)(\mu_{n,c}^{out}(t) - \mu_{n,c}^{in}(t)) \\ &+ \sum_c \theta^c Q_{c(0)}^c(t) R_c(t) - \sum_c \theta^c Y_c(t)(R_c(t) - \alpha_c) \end{aligned}$$

where $K = \sum_c \Xi_c$. Therefore, for a positive constant J, we have

$$\begin{aligned} &\Delta(\boldsymbol{Z}(t)) - JE\left(\sum_c R_c(t) p_c(t) | \boldsymbol{Z}(t)\right) \\ &\leq K - \sum_{n,c} \theta^c Q_n^c(t) E\left(\mu_{n,c}^{out}(t) - \mu_{n,c}^{in}(t) | \boldsymbol{Z}(t)\right) \\ &+ \sum_c \theta^c Q_{c(0)}^c(t) E(R_c(t) | \boldsymbol{Z}(t)) \\ &- \sum_c \theta^c Y_c(t) E(R_c(t) - \alpha_c | \boldsymbol{Z}(t)) \\ &- JE\left(\sum_c R_c(t) p_c(t) | \boldsymbol{Z}(t)\right). \end{aligned} \tag{23}$$

Note that (23) is general and holds for any possible policy.

Lemma 2: QADP algorithm minimizes the RHS of (23) over all possible policies.

Proof: The proof is deferred to the Appendix. ■

For an arbitrarily small positive constant $0 < \epsilon \leq \epsilon^{\max}$, define the *$\epsilon$-reduced network capacity region*, Ω_ϵ, as all possible input rate vectors such that

$$\Omega_\epsilon = \{\boldsymbol{\lambda} | \bar{\boldsymbol{\lambda}} + \boldsymbol{1} \cdot \epsilon \in \Omega\} \tag{24}$$

where $\bar{\boldsymbol{\lambda}}$ is feasible rate vector and $\boldsymbol{1}$ is a unit vector, and Ω is the original network capacity region. We will discuss about how to obtain $\epsilon^{\max}$ shortly.

Define D_ϵ^* as the optimum value of the *reduced problem* to (6) where Ω is replaced by Ω_ϵ. It can be verified that [3]

$$\lim_{\epsilon \to 0} D_\epsilon^* \to D^* \tag{25}$$

where D^* is the optimum value of the original revenue maximization problem in (6), i.e., the target of QADP algorithm.

Specifically, we denote $r_{\epsilon,c}^*(0), r_{\epsilon,c}^*(1), \cdots, r_{\epsilon,c}^*(t), \cdots$ and $p_{\epsilon,c}^*(0), p_{\epsilon,c}^*(1), \cdots, p_{\epsilon,c}^*(t), \cdots$ as the optimum sequences of admitted rates and prices, for flow c, which achieve D_ϵ^*. Define $\widetilde{r_c^\epsilon}$ as the time average of the optimum sequence of $r_{\epsilon,c}^*(0), r_{\epsilon,c}^*(1), \cdots, r_{\epsilon,c}^*(t), \cdots$. Therefore, following the definition of (24), we have $\widetilde{r_c^\epsilon} + \epsilon \in \Omega$. By Lemma 1, we claim that there exists a randomized policy, denoted by *RAND*, which yields

$$E(\mu_{n,c}^{out}(t) - \mu_{n,c}^{in}(t) - r_{\epsilon,c}^*(t)) = \epsilon \quad \forall c, n = c(0) \tag{26}$$

and

$$E(\mu_{n,c}^{out}(t) - \mu_{n,c}^{in}(t)) = \epsilon \quad \forall c, n \neq c(0) \tag{27}$$

and

$$E(r_{\epsilon,c}^*(t) + \epsilon) \geq \alpha_c + \tilde{\epsilon} \quad \forall c. \tag{28}$$

Denote the RHS of (23) as Ψ. Without loss of generality, we assume $\epsilon \leq \tilde{\epsilon}$. Therefore, for randomized policy *RAND*, we have

$$\begin{aligned} \Psi^{RAND} \leq K &- \epsilon\left(\sum_{n,c} \theta^c Q_n^c(t) + \sum_c \theta^c Y_c(t)\right) \\ &- JE\left(\sum_c r_{\epsilon,c}^*(t) p_{\epsilon,c}^*(t) | \boldsymbol{Z}(t)\right). \end{aligned} \tag{29}$$

Following Lemma 2, we conclude that for QADP algorithm,

$$\Delta(\boldsymbol{Z}(t)) - JE\left(\sum_c R_c(t)p_c(t)|\boldsymbol{Z}(t)\right) \le \Psi^{QADP}$$
$$\le \Psi^{RAND} \le K - \epsilon\left(\sum_{n,c}\theta^c Q_n^c(t) + \sum_c \theta^c Y_c(t)\right)$$
$$-JE\left(\sum_c r^*_{\epsilon,c}(t)p^*_{\epsilon,c}(t)|\boldsymbol{Z}(t)\right). \quad (30)$$

We take expectation with the distribution of $\boldsymbol{Z}(t)$, on both sides of (30), and take a sum on time slots $\tau = 0, \cdots, T-1$ and attain

$$\frac{1}{T}\sum_{\tau=0}^{T-1}\epsilon E\left(\sum_{n,c}\theta^c Q_n^c(\tau) + \sum_c \theta^c Y_c(\tau)\right)$$
$$+\frac{1}{T}\sum_{\tau=0}^{T-1} JE\left(\sum_c r^*_{\epsilon,c}(\tau)p^*_{\epsilon,c}(\tau)\right)$$
$$\le K + \frac{1}{T}\sum_{\tau=0}^{T-1} JO(\tau) + \frac{E(PF(\boldsymbol{Z}(0)))}{T} \quad (31)$$

where the nonnegativity of the potential function is utilized. We take $\liminf_{T\to\infty}$ on both sides of (31) and have[4]

$$\liminf_{T\to\infty}\frac{1}{T}\sum_{\tau=0}^{T-1} O(\tau)$$
$$\ge \liminf_{T\to\infty}\frac{1}{T}\sum_{\tau=0}^{T-1} E\left(\sum_c r^*_{\epsilon,c}(\tau)p^*_{\epsilon,c}(\tau)\right) - \frac{K}{J}.$$

Therefore, by (25), taking the limit of $\epsilon \to 0$ yields the performance bound of (14) in Theorem 1.

B. Proof of Network Stability

To prove the stability of the network, we take $\limsup$ on (31) and have

$$\limsup_{T\to\infty}\frac{1}{T}\sum_{\tau=0}^{T-1} E\left(\sum_{n,c}\theta^c Q_n^c(\tau) + \sum_c \theta^c Y_c(\tau)\right)$$
$$\le \frac{K + JO^{\max}}{\epsilon} \quad (32)$$

if it satisfies that $O(t) \le O^{\max}$ for all time slot t. Recall that the above analysis applies to any $0 < \epsilon \le \epsilon^{\max}$. In addition, by the definition of (24), we have

$$\epsilon^{\max} = \mu^{\max} - \max_i \alpha_i \text{ where } i = 1, \cdots, C \quad (33)$$

where $\mu^{\max}$ is the maximum possible data rate on a link and is assumed bounded. Moreover, by (4) and (5), we have $O^{\max} = C$ where C is the number of flows in the network. Therefore, we have

$$\limsup_{T\to\infty}\frac{1}{T}\sum_{\tau=0}^{T-1} E\left(\sum_{n,c}\theta^c Q_n^c(\tau) + \sum_c \theta^c Y_c(\tau)\right) \le \frac{K + JC}{\epsilon^{\max}}.$$

[4]We assume that the initial queue sizes of real data queues and virtual queues are bounded.

Using the fact that

$$X(t) \le Y(t)\ \forall t \Rightarrow \limsup X(t) \le \limsup Y(t), \quad (34)$$

we have, for every flow c, the average queue length on its routing path is bounded by

$$\limsup_{T\to\infty}\frac{1}{T}\sum_{\tau=0}^{T-1} E\left(\sum_{j=0}^{\kappa_c} Q^c_{c(j)}(\tau)\right) \le \frac{K + JC}{\theta^c \epsilon^{\max}}. \quad (35)$$

Finally, by applying Markov Inequality, we conclude that all data queues in the network are stable, based on the fact that the RHS of (35) is bounded.

C. Proof of Service Differentiation

Similar to (35), for every virtual queue $Y_c(t)$, we can obtain

$$\limsup_{T\to\infty}\frac{1}{T}\sum_{\tau=0}^{T-1} E(Y_c(\tau)) \le \frac{K + JC}{\theta^c \epsilon^{\max}}. \quad (36)$$

Therefore, by similar analysis and the definition of virtual queues, we conclude that the virtual queues are stable and thus the minimum data rate requirements imposed by multimedia flows are achieved.

Next, we show that QADP indeed provides a service differentiation solution on the guaranteed maximum end-to-end delays for all multimedia flows. Denote the actual experienced average delay of flow c as ω_c. By Little's Law, ω_c is approximated[5] by

$$\omega_c = \frac{\text{average overall queue length on the path of flow } c}{\text{average incoming rate of flow } c}$$
$$= \frac{\limsup_{T\to\infty}\frac{1}{T}\sum_{\tau=0}^{T-1} E\left(\sum_{j=0}^{\kappa_c} Q^c_{c(j)}(\tau)\right)}{\limsup_{T\to\infty}\frac{1}{T}\sum_{\tau=0}^{T-1} E(R_c(\tau))}. \quad (37)$$

In light of the stability of virtual queue $Y_c(t)$, we have $\limsup_{T\to\infty}\frac{1}{T}\sum_{\tau=0}^{T-1} E(R_c(\tau)) \ge \alpha_c$. Hence, we can obtain

$$\omega_c \le \frac{K + JC}{\theta^c \epsilon^{\max}\alpha_c}. \quad (38)$$

Equivalently speaking, to differentiate the guaranteed maximum delay bound, we need to find a set of weights such that

$$\theta^c \alpha_c \ell_c = \theta^d \alpha_d \ell_d \quad (39)$$

is satisfied for any pair of multimedia flows c and d. Therefore, it is straightforward to verify that the weight assignment algorithm in QADP indeed provides a service differentiation solution where the guaranteed maximum end-to-end delays of multimedia flows are distinguished according to application-dependent service level requests, which completes the proof of Theorem 1. The performance of QADP algorithm will be evaluated numerically in Section VI.

VI. Simulations

A. Single-hop Wireless Cellular Networks

We first consider a single-hop wireless cellular network with downlink multimedia transmissions, as shown in Figure

[5]Note that we consider a heavy loaded network. Propagation delays are assumed to be negligible compared to queueing delays and thus are omitted.

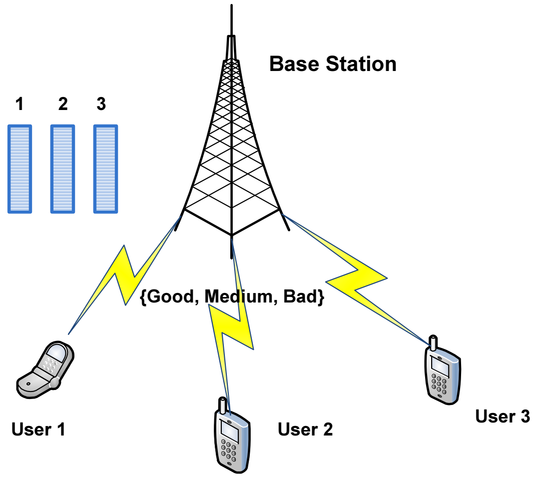

Fig. 2. A single-hop wireless cellular network with three users.

2. The base station (BS) is associated with three users with infinite backlogged traffic. A separate queue is maintained by the base station for every user. In addition, at each time slot, BS can only transmit to one particular user. A wireless link is assumed to have three equally possible channel states, i.e., *Good, Medium, Bad*. The corresponding transmission rates for three channel states are 20, 15 and 10 bits per slot, respectively.

Without loss of generality, we assume that the minimum average rate requirements for user $1, 2, 3$ are $\boldsymbol{\alpha} = [1, 2, 3]$ bits per slot. In addition, to provide service differentiation, the network offers three prioritized service levels, e.g., *Platinum, Gold* and *Silver*, where level *Platinum* possesses the highest priority in terms of end-to-end delay upper bound. We assume that user 1 has a service level request for *Platinum* while user 2 and 3 demand for level *Gold* and *Silver*, respectively, according to the upper layer applications. Other system parameters are assumed to be $R_c^{\max} = 20$ bits per slot for all flows and $\theta^{\max} = 100$. We next implement QADP algorithm for different values of J where $J = [50, 100, 500, 1000, 5000, 10000, 20000, 50000, 100000]$. Every experiment is simulated for 500000 time slots.

Figure 3 depicts the system revenue, i.e., the solution of (6) by QADP algorithm, with respect to different values of J. As shown in (14) and demonstrated pictorially in Figure 3, the achieved system revenue by QADP converges gradually to the optimum solution as J grows. Note that the values of system revenues are almost indistinguishable when $J \geq 50000$. Figure 4 illustrates the actual experienced average delays of all three flows with different values of J, where the delays of user 1 to user 3 are compared from left to right. It is worth noting that, not only the maximum guaranteed end-to-end delays are distinguished analytically by QADP, but also the actual experienced average delays are prioritized for all three flows with distinct service level requests. More specifically, user 1, i.e., the *Platinum* user, enjoys a delay which is less than half of that of user 2 and one third of that of user 3, for all values of J, as demonstrated in Figure 4. However, as shown by Figure 3 and Figure 4 jointly, while a larger value of J yields an improvement on the performance of QADP, the end-to-end delays of all three flows are augmented concurrently. Therefore, by tuning J, a tradeoff between optimality and average delays can be achieved. The time average admitted rates of multimedia flows are shown in Table I. We observe that in all cases, the average rates of flows exceed the minimum data rate requirements specified by $\boldsymbol{\alpha}$.

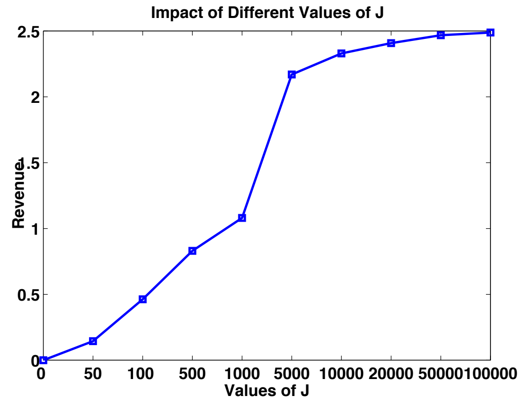

Fig. 3. Impact of different values of J on the performance of QADP.

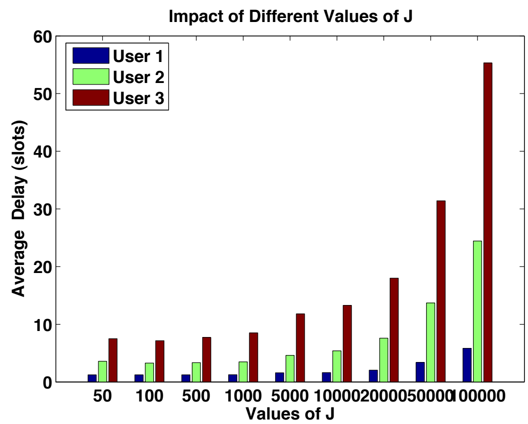

Fig. 4. Impact of different values of J on the average experienced delays.

TABLE I
AVERAGE ADMITTED RATES FOR MULTIMEDIA FLOWS

	Flow 1	Flow 2	Flow 3
J = 50	4.01	2.71	3.07
J = 100	4.04	2.72	3.09
J = 500	4.05	2.91	3.11
J = 1000	4.31	3.07	3.16
J = 5000	4.51	3.57	3.42
J = 10000	4.63	3.94	3.62
J = 20000	4.75	4.28	4.09
J = 50000	5.09	4.71	4.61
J = 100000	5.13	4.88	4.85

The sample paths of price adaptations and queue backlog evolutions are illustrated in Figure 5 and Figure 6, respectively, for the first 100 time slots with $J = 50000$. Note that, in Figure 5, the prices imposed for three users are dynamically adjusted at every time slot. It is worth noting that whenever a data queue in Figure 6 has a tendency to build up, the price imposed by QADP algorithm, as shown in Figure 5, rises correspondingly, which in turn discourages the excessive admitted rate and thus all queues in the network remain bounded. As a result, the stability of the network is achieved.

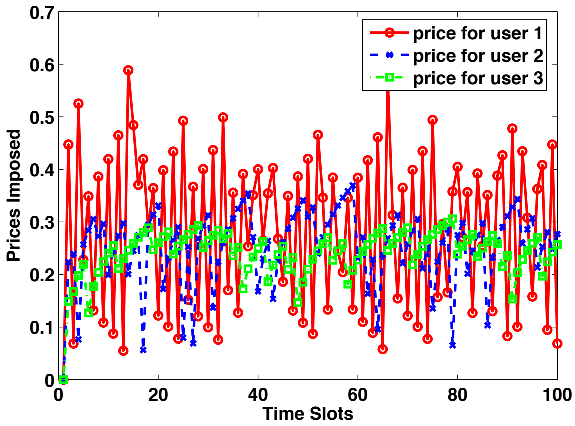

Fig. 5. Price dynamics in QADP for all users.

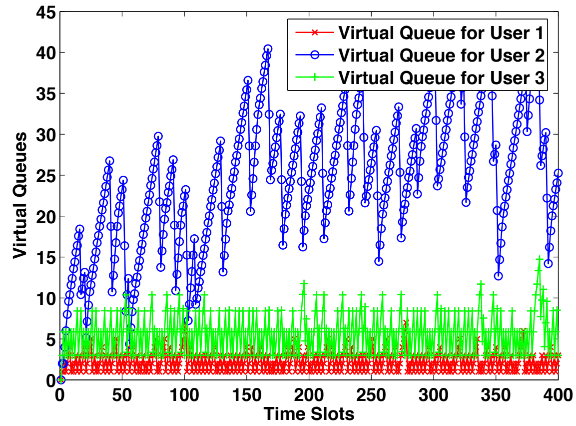

Fig. 7. Virtual queue backlog updates in QADP for $J = 50000$.

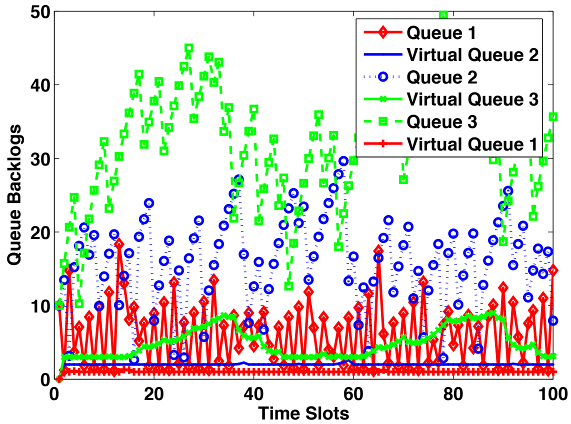

Fig. 6. Queue backlog dynamics for all users.

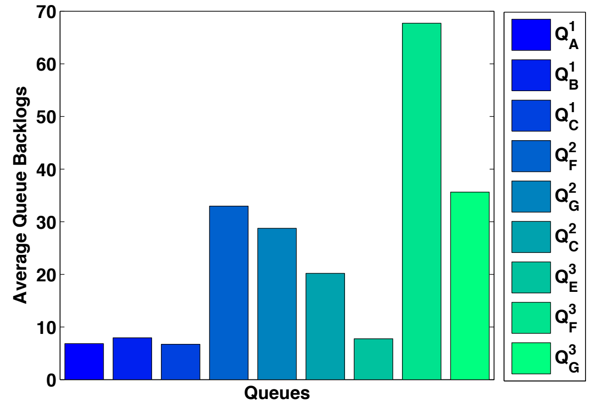

Fig. 8. Average queue backlogs in the network for J = 50000.

B. Multi-hop Wireless Networks

We next consider a multi-hop wireless network with a topology shown in Figure 1. There are three multimedia flows exist in the network, denoted by *Flow 1, 2, 3*. The routing paths of flows are specified by $P_1 = \{A, B, C, D\}$, $P_2 = \{F, G, C, D\}$ and $P_3 = \{E, F, G, H\}$. Without loss of generality, we assume a two-hop interference model which represents the general IEEE 802.11 MAC protocols [23], [24]. Other configurations are the same as the single-hop scenario described above except that the possible link rates are assumed to be $40, 30, 20$ bits per slot for three channel conditions. We observe that in this network topology, link $C \rightarrow D$ and link $F \rightarrow G$ are shared by two different flows. Therefore, in the scheduling part of QADP, the particular flow with a larger weighted queue backlog difference should be selected.

In Figure 7, we specifically depict the dynamics of three virtual queues for the first 400 time slots with $J = 50000$. Unlike the single-hop case in Figure 6, the virtual queues behave remarkably different in this multi-hop scenario. It is worth noting that while the virtual queues of user 1 and 3 have relatively low occupancies, the virtual queue associated with user 2 suffers a larger average backlog. Intuitively, due to the underlying two-hop interference model, link $G \rightarrow C$ needs to be scheduled exclusively in the network for successful transmissions. In other words, link $G \rightarrow C$ is the bottleneck of the network. Therefore, to ensure network-wide stability, a much more stringent regulation is enforced on the admitted rate of flow 2. As a consequence, although remains bounded, the virtual queue of flow 2 accumulates more backlogs compared to other competitive flows. In addition, we compare the time average queue backlogs of all data queues in the network, from left to right, in Figure 8. We can observe that the data queues on the path of flow 1 have fewer average backlogs due to the highest priority with respect to the service level, i.e., *Platinum*. On the contrary, the queues on the path of flow 3 have larger backlogs compared to other two flows. It is noticeable that Q_F^3 and Q_G^3 have considerably larger average queue sizes. This is because that Q_F^3 has to share the link rate of $F \rightarrow G$ with Q_F^2. Nevertheless, Q_F^3 possesses a smaller share of bandwidth than Q_F^2 due to the lower prioritized service level associated with flow 3. Even worse yet, both link $F \rightarrow G$ and $G \rightarrow H$ have less opportunity to be scheduled due to their locations and the underlying interference model. Therefore, the average backlogs on the path of user 3 has higher occupancies compared to other two competitive flows. The average data rates are provided in Table II. Note that the minimum average rate requirements of all multimedia flows are satisfied simultaneously, as expected. The tradeoff between optimality and average delay, which is controlled by different values of J, as well as the network service differentiation

TABLE II
AVERAGE ADMITTED RATES FOR MULTIMEDIA FLOWS

	Flow 1	Flow 2	Flow 3
J = 50	3.04	2.08	3.88
J = 100	3.12	2.03	3.98
J = 500	3.17	2.04	4.60
J = 1000	3.30	2.03	5.04
J = 5000	4.13	2.01	6.45
J = 10000	4.36	2.02	7.32
J = 20000	5.08	2.03	8.12
J = 50000	6.43	2.01	9.18
J = 100000	7.63	2.01	9.89

in terms of delays are analogous to the single-hop scenario discussed above. Duplicated simulation figures are omitted.

VII. CONCLUSIONS

We consider a multi-hop wireless network where multiple flows share the network resource jointly. To maximize the overall network revenue while capturing the tradeoff between cost and quality of multimedia transmissions, we propose a dynamic pricing based algorithm, namely, QADP, which achieves a solution that is arbitrarily close to the optimum, subject to network stability. A weight assignment mechanism is introduced to address the service differentiation issue for heterogenous flows with different priorities. In this work, we assume that the information of channel states, i.e., $\mathbf{S}(t)$, is available for QADP algorithm, which is either acquired by the centrally or approximated by the distributed local scheduling algorithms such as [22]–[25]. As a future work, the incorporation of online channel probing mechanism seems interesting and needs further investigation.

APPENDIX

Proof of Lemma 2:
The scheduling part of QADP algorithm in Section IV is a weighted version of *MaxWeight* algorithm [1], [2], [21] which maximizes $\sum_{n,c} \theta^c Q_n^c(t)(\mu_{n,c}^{out}(t) - \mu_{n,c}^{in}(t)) = \sum_{(a,b)\in\mathcal{E}} \mu_{a,b}(t)\xi_{a,b}$ if $\exists c$, such that $(a,b) \in P_c$, for every time slot t. The dynamic pricing part of QADP is essentially maximizing

$$\sum_c \Big(\theta^c (Y_c(t) - Q_{c(0)}^c(t)) R_c(t) + J R_c(t) p_c(t)\Big) \qquad (40)$$

for every time slot. By (4) and (5), we see that QADP finds an optimum price $p_c^*(t)$ which maximizes

$$M = \theta^c (Y_c(t) - Q_{c(0)}^c(t))(\frac{1}{p_c(t)} - 1) + J(1 - p_c(t)). \quad (41)$$

Define $W = \theta^c (Q_{c(0)}^c(t) - Y_c(t))$.
Case 1: If $W > 0$, we have

$$M = J + W - (J p_c(t) + W \frac{1}{p_c(t)}). \qquad (42)$$

Obviously, M is a concave function and can be optimized at $p_c^*(t) = \sqrt{W/J}$.

Case 2: If $W \leq 0$, M is a decreasing function with respect to $p_c(t)$. Therefore, $p_c^*(t) = 0$.

REFERENCES

[1] L. Tassiulas and A. Ephremides, “Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks,” *IEEE Trans. Autom. Contr.*, vol. 37, pp. 1936–1949, 1992.
[2] M. J. Neely, E. Modiano, and C. E. Rohrs, “Dynamic power allocation and routing for time-varying wireless networks,” *IEEE J. Sel. Areas Commun.*, vol. 23, pp. 89–103, Jan. 2005.
[3] M. J. Neely, “Dynamic power allocation and routing for satellite and wireless networks with time varying channels,” Ph.D. dissertation, Masssachusetts Institute of Technology, LIDS, 2003.
[4] M. J. Neely, E. Modiano, and C.-P. Li, “Fairness and optimal stochastic control for heterogeneous networks,” *IEEE/ACM Trans. Netw.*, vol. 16, pp. 396–409, April 2008.
[5] M. J. Neely, “Energy optimal control for time varying wireless networks,” *IEEE Trans. Inf. Theory*, vol. 52, pp. 2915–2934, July 2006.
[6] L. He and J. Walrand, “Pricing differentiated internet services,” *IEEE INFOCOM*, 2005.
[7] T. Basar and R.Srikant, “Revenue-maximizaing pricing and capacity expansion in a many-users regime,” *IEEE INFOCOM*, 2002.
[8] R. K. Lam, D.-M. Chiu, and J. C. Lui, “On the access pricing and network scaling issues of wireless mesh networks,” *IEEE Trans. Comput.*, vol. 56, pp. 1456–1469, 2007.
[9] Shakkottai and R. Srikant, “Economics of network pricing with multiple isps,” *IEEE/ACM Trans. Netw.*, vol. 14, 2006.
[10] N. J. Keon and G. Anandalingam, “Optimal pricing for multiple services in telecommunications networks offering quality-of-service guarantees,” *IEEE/ACM Trans. Netw.*, vol. 14, pp. 66–80, 2003.
[11] Z. Zhang, D. Dey, and Y. Tan, “Pricing communication services with delay guarantee,” *INFORMS Journal on Computing*, vol. 19, pp. 248–260, 2007.
[12] L. Huang and M. J. Neely, “The optimality of two prices: Maximizing revenue in a stochastic communication system,” *IEEE/ACM Trans. Netw.*, vol. 18, pp. 406–419, 2010.
[13] D. Niyato and E. Hossain, “Competitive spectrum sharing in cognitive radio networks: A dynamic game approach,” *IEEE Trans. Wireless Commun.*, vol. 7, 2008.
[14] D. Niyato, E. Hossain, and L. Le, “Competitive spectrum sharing and pricing in cognitive wireless mesh networks,” *IEEE WCNC*, 2008.
[15] D. Niyato, E. Hossain, and Z. Han, “Dynamic spectrum access in ieee 802.22-based cognitive wireless networks: A game theoretic model for competitive spectrum bidding and pricing,” *IEEE Wireless Commun.*, 2009.
[16] ——, “Dynamics of multiple-seller and multiple-buyer spectrum trading in cognitive radio networks: A game-theoretic modeling approach,” *IEEE Trans. Mobile Computing*, 2009.
[17] P. Krugman and R. Wells, *Microeconomics*. Worth Publishers, 2009.
[18] J. Hair, B. Black, B. Babin, R. Anderson, and R. Tatham, *Multivariate Data Analysis*. Prentice Hall, 2006.
[19] M. Chiang, S. H. Low, A. R. Calderbank, and J. C. Doyle, “Layering as optimization decomposition: A mathematical theory of network architectures,” *Proc. IEEE*, vol. 95, pp. 255–312, Mar. 2007.
[20] R. Srikant, *The Mathematics of Internet Congestion Control*. Birkhauser Boston, 2003.
[21] A. Stolyar, “Maximizing queueing network utility subject to stability: Greedy primal-dual algorithm,” *Queueing Systems*, vol. 50, pp. 401–457, 2005.
[22] A. Brzezinski, G. Zussman, and E. Modiano, “Distributed throughput maximization in wireless mesh networks via pre-partitioning,” *IEEE/ACM Trans. Netw.*, 2008.
[23] C. Joo, “A local greedy scheduling scheme with provable performance guarantee,” *Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing (Mobihoc)*, 2008.
[24] A. Gupta, X. Lin, and R. Srikant, “Low-complexity distributed scheduling algorithms for wireless networks,” *IEEE INFOCOM*, 2007.
[25] A. Stolyar, “Dynamic distributed scheduling in random access networks,” *Journal of Applied Probability*, vol. 45, pp. 297–313, 2008.