Cross-Layer Rate Optimization for Proportional Fairness in Multi-hop Wireless Networks with Random Access

Xin Wang and Koushik Kar, *Member, IEEE*

***Abstract*— In this paper, we address the rate control problem in a multi-hop random access wireless network, with the objective of achieving proportional fairness amongst the end-to-end sessions. The problem is considered in the framework of nonlinear optimization. Compared to its counterpart in a wired network where link capacities are fixed, rate control in a multi-hop random access network is much more complex and requires joint optimization at both the transport and link layers. This is due to the fact that the attainable throughput on each link in the network is 'elastic' and is typically a non-convex and non-separable function of the transmission attempt rates. Two cross-layer algorithms, a dual based algorithm and a penalty based algorithm, are proposed in this paper to solve the rate control problem in a multi-hop random access network. Both algorithms can be implemented in a distributed manner, and work at the link layer to adjust link attempt probabilities and at the transport layer to adjust session rates. We prove rigorously that the two proposed algorithms converge to the globally optimal solutions. Simulation results are provided in support of our conclusions.**

***Index Terms*— Cross-layer optimization, Proportional fairness, Random access networks.**

I. Introduction

The objective of rate control is generally to use the available bandwidth to the full while maintaining a certain level of "fairness" amongst the sessions in the network. In wired networks, the problem of rate control has been extensively researched, e.g., [7], [9]. It has been shown that in wired networks, since the feasible rate region can be represented by a set of simple, separable, convex constraints, globally fair rates are attainable via distributed approaches based on convex programming.

In wireless networks, the capacity of a link is not a fixed quantity, and depends on the specific MAC (Medium Access Control) protocol used. MAC protocols are designed to reduce collisions, to ensure high system throughput, and to distribute the available bandwidth fairly among the competing nodes. A prominent feature of the wireless network is that its feasible rate region is typically a complex non-convex and non-separable function of the MAC control parameters like the transmission probabilities or back-off window sizes. Therefore, approaches and results on rate control in wired networks are not readily applicable to a wireless scenario.

This work was supported by the National Science Foundation under grants ECS-0330203 and CNS-0448316.

The authors are with the Department of Electrical, Computer and Systems Engineering, Rensselaer Polytechnic Institute, NY 12180, USA. E-mail: wangx5@rpi.edu, koushik@ecse.rpi.edu.

Since the feasible rate region in a wireless network depends on the MAC protocol and parameters, the end-to-end rate optimization question must be considered in a cross-layer framework, i.e., the rate control strategy must be implemented at both the link layer and the transport layer. In this paper, we study the end-to-end *proportionally fair* [6] rate allocation problem in a multi-hop random access network with general topology. Specifically, we address the problem of how to introduce a cooperation between the link layer and the transport layer so that aggregate utilities of all end-to-end sessions are maximized. The problem is formulated as an optimization problem and two algorithms are proposed to solve the problem in a distributed manner.

The first algorithm is a dual-based algorithm. At the higher (transport) layer, end-to-end sessions adjust their rates in a distributed manner so as to attain proportionally fair session rates given specific link rates. At the lower (link) layer, the link attempt probabilities are adjusted with local information, so that the bandwidth bottlenecks are alleviated and the aggregate utilities can be further increased. In this manner, the link layer and the transport layer cooperate with each other and achieve end-to-end proportionally fair rates in the distributed manner.

It is worth noting that, every time the attempt probabilities (and hence the link rates) are adjusted at the link layer, the algorithm at the transport layer will compute the optimal end-to-end session rates, along with the optimal "link prices", under the given link rates. After that, the algorithm at the link layer adjusts the link attempt probabilities accordingly, using information on the link prices and the link attempt probabilities in the local neighborhood. Therefore the proposed rate control algorithm works at a longer time scale at the link layer and at a shorter time scale at the transport layer. We show that this algorithm essentially adjusts the link attempt probabilities in an dual descent direction, and the algorithm converges to the globally optimal solutions.

Although the formulated rate control problem appears to be non-convex, we show that it is equivalent to a convex programming problem through certain transformations. More importantly, the transformed convex problem can be solved in a distributed manner. A penalty-based algorithm using subgradient method is proposed for this transformed convex program. In each iteration of this algorithm, a link updates its attempt probability using the attempt probabilities of the links in its neighborhood. At the same time, the end user updates its session rate using aggregate traffic load and capacity information for the links on its path. In this approach, the transport and link layer algorithms work at the same time scale; this algorithm also converges and achieves the globally optimal rates.

The paper is organized as follows. Related work is discussed in Section II. In Section III, we describe the models for the network and the link layer, and formulate the rate control problem as an optimization question. The solution approach and the implementation of the dual-based algorithm are presented in Section IV. We then consider the transformed convex program and discuss the penalty-based algorithm in detail in Section V. The two algorithms are compared in Section VI. Simulation results are presented in Section VII, and the paper is concluded in Section VIII. All necessary proofs are provided in the appendix.

II. Related Work

There are several existing works that address the problem of fair bandwidth sharing at the link layer. In [12], Tassiulas *et al.* have proposed a centralized algorithm to attain max-min fair rate for a certain class of ad-hoc networks. On the other hand, Nadagopal *et al.* [10] and Ozugur *et al.* [11] have proposed decentralized algorithms that try to attain fair rate allocations. Kar *et al.* [16] have considered the proportional fairness problem in Aloha networks, both slotted and unslotted, and derived local topology based strategies that attain global fairness for single-hop flows. In [17], Wang *et al.* provide distributed algorithms to achieve max-min fair rates in Aloha networks. All these schemes can be viewed as rate control for single-hop flows. However, these results cannot be readily extended to the general rate control problem for end-to-end sessions in a multi-hop wireless network.

Several recent works have addressed related questions on the cross-layer design of wireless networks. In [13], Johansson *et al.* consider the problem of finding the jointly optimal end-to-end communication rates, routing, power allocation and transmission scheduling in a general wireless network. However the approach is based on nonlinear column generation which is difficult to implement in a distributed manner. In [14], Xiao *et al.* formulate the problem of simultaneous routing and resource allocation in a wireless network and propose distributed algorithms via dual decomposition. However, a basic assumption of their work is that the capacity of a wireless link is a concave function of link variables, which may not be true in general. In [8], the authors consider the problem of joint rate control and scheduling in a multi-hop wireless network. Through the dual approach, the authors decompose the original joint optimization problem into the rate control problem and the scheduling problem. The solution of the scheduling problem, however, requires maximizing the total weighted link capacity in the network, which is computationally expensive and difficult to implement in a distributed manner. In [15], Chiang proposes a distributed power control algorithm, that along with a TCP rate update mechanism, optimizes the end-to-end throughput in a wireless CDMA network. Although our work is closely related to [15], the problem considered and the approaches proposed in our work differ significantly from those in [15]. Unlike [15], we are interested in optimizing the transmission attempt probabilities at the lower layer, and not the transmission powers. Moreover, we propose both penalty- and dual-based approaches, which work at one and two timescales respectively; in contrast, the approach in [15] is a dual based approach which works at a single timescale.

III. Formulation

A. System Model

We consider a general wireless network, where all nodes need not be in the transmission range of each other. For simplicity, we assume a symmetric hearing matrix, i.e., node i can receive signal from node j if and only if node j can receive signal from node i. However, our analysis can be generalized to the case when this assumption does not hold.

A wireless network can be modeled as an undirected graph $G = (N, E)$, where N and E respectively denote the set of nodes and the set of undirected edges. An edge exists between two nodes if and only if they can receive each other's signals. A directed link (i, j) represents an active direct-communication pair, and L is the set of directed links. Note that there are $2|E|$ possible direct-communication pairs, but only a few pairs may be actively communicating.

Suppose the set of sessions (end-to-end flows) sharing the network be denoted by S. Let $L(s) \subseteq L$ denote the set of links that a session $s \in S$ uses, i.e., $L(s)$ is the set of links in session s's end-to-end path. For each link $(i, j) \in L$, let $S(i, j) = \{s \in S | (i, j) \in L(s)\}$ be the set of sessions that use link (i, j). Note that $(i, j) \in L(s)$ if and only if $s \in S(i, j)$. In the sequel we assume that both the set of sessions and the routing matrix are fixed. We also assume that all sessions are backlogged.

Further details on the link layer model are provided next. For any node i, the set of i's *neighbors*, $K_i = \{j : (i, j) \in E\}$, represents the set of nodes that can hear i. For any node i, the set of *out-neighbors* of i, $O_i = \{j : (i, j) \in L\} \subseteq K_i$, represents the set of neighbors to which i is sending traffic. Also, for any node i, the set of *in-neighbors* of i, $I_i = \{j : (j, i) \in L\} \subseteq K_i$, represents the set of neighbors from which i is receiving traffic. A transmission from node i reaches all of i's neighbors. Each node has a single transceiver. Thus, a node can not transmit and receive simultaneously. We assume *no capture*, i.e., node j can not receive any packet successfully if more than one of its neighbors are transmitting simultaneously. Therefore, a transmission on link $(i, j) \in L$ is successful if and only if no node in $K_j \cup \{j\} \setminus \{i\}$ transmits during the transmission on (i, j). We also assume, without loss of generality, that all the nodes share a single wireless channel of unit capacity.

B. Link Rate Expressions

In the following, the (slotted) Aloha protocol [1] is used to model the access control strategy in a random access wireless network. In the Aloha network, each node i transmits a packet with probability P_i in a slot. If i does not have an outgoing link, i.e., $O_i = \phi$, then $P_i = 0$. Once i decides to transmit in a slot, it selects a destination $j \in O_i$ with probability

$\frac{p_{(i,j)}}{P_i}$, where $\sum_{j\in O_i} p_{(i,j)} = P_i$. Therefore, in each slot, a packet is transmitted in link (i,j) with probability $p_{(i,j)}$. Let $\mathbf{p} = (p_{(i,j)}, (i,j) \in L)$ be the vector of transmission probabilities on all links, and let $\mathbf{P}_b$ denote the region defined by the boundary (feasibility) constraints on $\mathbf{p}$, i.e. $\mathbf{P}_b = \{\mathbf{p} : p_{(i,j)} \geq 0 \ \forall (i,j) \in L, \sum_{j\in O_i} p_{(i,j)} = P_i \leq 1 \ \forall i \in N\}$. Then the rate or the attainable throughput on link $l = (i,j)$, x_l, is

$$x_{(i,j)} = c_{ij}(\mathbf{p}) = p_{(i,j)}(1-P_j) \prod_{k\in K_j\setminus\{i\}} (1-P_k), \quad \mathbf{p}\in\mathbf{P}_b. \tag{1}$$

The term $(1-P_j)\prod_{k\in K_j\setminus\{i\}}(1-P_k)$ in (1) is the probability that a packet transmitted on link (i,j) is successfully received at j. Note that the rate on link (i,j) depends not only on the attempt probability on link (i,j), $p_{(i,j)}$, but also on the attempt probabilities of node j and its neighbors.

C. Problem Statement

We now consider the end-to-end proportionally fair rate control problem in the context of the multi-hop Aloha network model described above.

Let each session $s \in S$ be associated with a utility function $U_s : \Re^+ \to \Re$. Thus session s attains a utility $U_s(y_s)$ when it sends data at rate y_s satisfying $y_s \geq 0$. Specifically, we are interested in the proportionally fair rate control problem; therefore, the utility function U_s is chosen as the logarithmic function [7]. Note that the logarithmic function is increasing and strictly concave in its argument.

Note that the feasible rate allocations must satisfy the link capacity constraints, i.e., for any link (i,j) we have

$$\sum_{s\in S(i,j)} y_s \leq x_{(i,j)}. \tag{2}$$

The rate optimization problem is therefore formulated as

$$\begin{array}{llll} \mathbf{Q}: & \max & \sum_s \log(y_s), & \\ & \text{s.t.} & \sum_{s\in S(i,j)} y_s \leq x_{(i,j)} & \forall (i,j)\in L, \\ & & x_{(i,j)} = c_{ij}(\mathbf{p}) & \forall (i,j)\in L, \\ & & y_s \geq 0 & \forall s \in S, \\ & & \mathbf{p}\in\mathbf{P}_b. & \end{array} \tag{3}$$

The first and second sets of constraints ensure that the total session rates of traffic in a link cannot exceed the attainable throughput of the link, where $c_{ij}(\mathbf{p})$ is defined by (1). The third set of constraints ensure that the session rates are non-negative, and the fourth ensure that the attempt probabilities are feasible. The rate control question therefore represents a joint optimization problem which couples the link attempt probabilities at the link layer with the end-to-end session rates at the transport layer.

For simplicity of exposition and analysis, we assume that the link capacity constraints are linearly independent. In other words, if we define the $|L|\times|S|$ routing matrix $\mathbf{R}$ as

$$R(l,k) = \begin{cases} 1 & \text{if } k\in S(l), \\ 0 & \text{otherwise,} \end{cases} \tag{4}$$

then the rank of $\mathbf{R}$ is $|L|$. As an example, note that the stated assumption holds if there is at least one single-hop session across each link in the network.

D. Equivalent Convex Formulation

The end-to-end proportionally fair rate optimization problem in (3) appears to be a non-convex problem. However, the following lemma (proof in appendix A) states that the proportionally fair rate can be obtained by solving a convex optimization problem.

Lemma 1: The end-to-end proportionally fair rate control problem in a multi-hop random access network, as given by (3), is equivalent to the following convex program,

$$\begin{array}{ll} \max & \sum_{s\in S} z_s, \\ \text{s.t.} & \log\left(\sum_{s\in S(i,j)} e^{z_s}\right) - \log c_{ij}(\mathbf{p}) \leq 0 \ \ \forall (i,j)\in L, \\ & \mathbf{p}\in\mathbf{P}_b. \end{array} \tag{5}$$

In the above, z_s should be interpreted as the logarithm of the session rate y_s, i.e., $z_s = \log(y_s)$.

IV. Dual-Based Algorithm

A. Solution Approach

Instead of solving $\mathbf{Q}$ directly, we now consider a simplified version of the end-to-end proportionally fair rate optimization question where each link capacity is parameterized:

$$\begin{array}{llll} \hat{\mathbf{Q}}: & \max & \sum_{s\in S} \log(y_s), & \\ & \text{s.t.} & \sum_{s\in S(i,j)} y_s \leq x_{(i,j)} & \forall (i,j)\in L, \\ & & 0\leq y_s \leq 1 & \forall s\in S. \end{array} \tag{6}$$

In the above formulation, $x_{(i,j)}$, the rate on link (i,j), is assumed to be a given constant; however, the terms y_s, representing the end-to-end session rates, are variables whose values need to be determined optimally. Note that the constraints $y_s \leq 1$ are redundant constraints; these are introduced in order to ensure that the boundary constraints on the session rates represent a compact set, as required by our analysis. Let $\mathbf{Y}_b$ represent the region defined by these boundary constraints, i.e., if $\mathbf{y} = (y_s, s\in S)$ represents the vector of session rates, then $\mathbf{Y}_b = \{\mathbf{y} : 0\leq y_s \leq 1 \ \forall s \in S\}$.

Note that the optimum value in the parameterized problem $\hat{\mathbf{Q}}$ is a function of $\mathbf{x} = (x_{(i,j)} : (i,j)\in L)$. We define $\hat{U}(\mathbf{x})$ as the optimum value in $\hat{\mathbf{Q}}$, i.e.,

$$\hat{U}(\mathbf{x}) = \max_{\mathbf{y}\in\mathbf{Y}_b} \left\{ \sum_{s\in S} \log(y_s) \,\middle|\, \sum_{s\in S(i,j)} y_s \leq x_{(i,j)}, (i,j)\in L \right\}.$$

Since the vector of all link rates considered in $\mathbf{Q}$ is a function of the link attempt probabilities, we can define function $\tilde{U}(\mathbf{p}) = \hat{U}(\mathbf{c}(\mathbf{p}))$, where $\mathbf{c}(\mathbf{p}) = (c_{ij}(\mathbf{p}) : (i,j)\in L)$. Therefore problem $\mathbf{Q}$ can be rewritten as

$$\begin{array}{lll} \tilde{\mathbf{Q}}: & \max & \tilde{U}(\mathbf{p}), \\ & \text{s.t.} & \mathbf{p}\in\mathbf{P}_b. \end{array} \tag{7}$$

We solve the problem in (7) by updating the link attempt probabilities using the following equation

$$p_{(i,j)}^{(n+1)} = \left[p_{(i,j)}^{(n)} + \alpha \sum_{(r,t)\in L} \lambda_{(r,t)}^{*(n)} \frac{\partial c_{rt}}{\partial p_{(i,j)}}(\mathbf{p}^{(n)}) \right]_{\mathbf{P}_b}, \tag{8}$$

where n is the iteration number, α is the step size, and $[\cdot]_{\mathbf{P}_b}$ is the projection operator on the set $\mathbf{P}_b$ that ensures feasibility of the updated link attempt probabilities.

The term $\frac{\partial c_{rt}}{\partial p_{(i,j)}}$ is computed using the following formula

$$\frac{\partial c_{rt}}{\partial p_{(i,j)}} = \begin{cases} (1-P_t) \prod_{k \in K_t \setminus \{r\}} (1-P_k) & \text{if } t=j \text{ and } r=i, \\ -p_{(r,t)} \prod_{k \in K_t \setminus \{r\}} (1-P_k) & \text{if } t=i \text{ and } r \in I_t, \\ -p_{(r,t)}(1-P_t) \prod_{k \in K_t \setminus \{r\}} (1-P_k) & \text{if } t \in K_i \text{ and } r \in I_t \setminus \{i\}, \\ 0 & \text{otherwise,} \end{cases} \tag{9}$$

and $\lambda^{*(n)}_{(i,j)}$ is the optimal solution to the dual problem of $\hat{\mathbf{Q}}$ when $\mathbf{x} = \mathbf{c}(\mathbf{p}^{(n)})$, i.e.,

$$\boldsymbol{\lambda}^{*(n)} = \arg\min_{\boldsymbol{\lambda} \geq 0} \max_{\mathbf{y} \in \mathbf{Y}_b} L^{(n)}(\mathbf{y}, \boldsymbol{\lambda}). \tag{10}$$

In (10), $\boldsymbol{\lambda} = \big(\lambda_{(i,j)} : (i,j) \in L\big)$ is the vector of dual variables for the capacity constraints on the wireless links, and $L^{(n)}(\mathbf{y}, \boldsymbol{\lambda})$ is the Lagrange function of $\hat{\mathbf{Q}}$ when $\mathbf{x} = \mathbf{c}(\mathbf{p}^{(n)})$. Note that $L^{(n)}(\mathbf{y}, \boldsymbol{\lambda})$ is given by

$$L^{(n)}(\mathbf{y}, \boldsymbol{\lambda}) = \sum_{s \in S} \log(y_s) - \sum_{(i,j) \in L} \lambda_{(i,j)} \left(\sum_{s \in S(i,j)} y_s - x^{(n)}_{(i,j)} \right), \tag{11}$$

where $x^{(n)}_{(i,j)} = c_{ij}(\mathbf{p}^{(n)})$ is the link rate value in the nth iterative step. We obtain $\mathbf{y}^{(n)}$ by solving $\hat{\mathbf{Q}}$ at $\mathbf{x} = \mathbf{c}(\mathbf{p}^{(n)})$, i.e.,

$$\mathbf{y}^{(n)} = \arg\max_{\mathbf{y} \in \mathbf{Y}_b} \left\{ \sum_{s \in S} \log(y_s) \,\middle|\, \sum_{s \in S(i,j)} y_s \leq c_{ij}(\mathbf{p}^{(n)}), (i,j) \in L \right\}. \tag{12}$$

B. Convergence Analysis

We have the following theorem regarding to the convergence property of the dual-based approach (proof in appendix B).

Theorem 1: Let $\{\mathbf{p}^{(n)}(\alpha), \mathbf{y}^{(n)}(\alpha)\}$ denote the sequence of vectors of link attempt probabilities and end-to-end session rates computed using the iterative procedures (8)-(12) with step size α. Then there exists an $\bar{\alpha} \in \Re^+$ such that for $0 < \alpha < \bar{\alpha}$, the limit point of $\{\mathbf{p}^{(n)}(\alpha), \mathbf{y}^{(n)}(\alpha)\}$ is the globally optimal solution to the problem $\mathbf{Q}$.

Intuitively, the update procedures in (8)-(11) adjust the link attempt probabilities in the gradient direction. Therefore the sequence of $\{\mathbf{p}^{(n)}(\alpha)\}$ converges to a local optimal point in $\tilde{\mathbf{Q}}$ where the Karush-Kuhn-Tucker (KKT) conditions hold. Since $\mathbf{Q}$ and $\tilde{\mathbf{Q}}$ are equivalent, it can be shown that the KKT point of $\tilde{\mathbf{Q}}$ yields a KKT point of $\mathbf{Q}$ if $\mathbf{y}$ is solved by (12). Therefore the procedures in (8)-(12) converge to the KKT point of $\mathbf{Q}$. We further show that, although $\mathbf{Q}$ appears to be non-convex, its KKT points are globally optimal.

C. Distributed Algorithm

In this section, we describe in detail a distributed implementation of the dual-based algorithm to solve the proportionally fair rate control problem $\mathbf{Q}$.

The algorithm works at both the transport layer and the link layer. Periodically, the attempt probabilities are updated at the link layer, using information on link prices and link attempt probabilities in a node's local neighborhood. Each time the attempt probabilities are updated, the algorithm works at the transport layer, where the optimal end-to-end session rates and optimal link prices (under the updated link rates) are computed by an iterative search. Therefore, the proposed algorithm works at the transport layer and the link layer at different time scales: it works at the link layer at a larger (longer) time scale, and at the transport layer at a smaller (shorter) time scale.

1) Flow Rate Control at the Transport Layer: The algorithm at the transport layer solves the rate control problem $\hat{\mathbf{Q}}$. When the link attempt probabilities have been updated, all link rates are computed accordingly. The algorithm at the transport layer is then executed, which solves essentially the same problem as the one in a wired network. In fact, the algorithm at the transport layer in our work is exactly the same as the one in [9], i.e., each source adjusts its session rate and each link adjusts its link price, in an iterative manner, until the optimal solutions are achieved. Note that the algorithm in [9] not only gives the optimal rates, but also the corresponding Lagrange multipliers (or optimal link prices).

We now state the procedures to solve the dual problem $\hat{\mathbf{Q}}$ [2],[9]. The Lagrangian in (11) at the nth iterative step can be rewritten as

$$L^{(n)}(\mathbf{y}, \boldsymbol{\lambda}) = \sum_{s \in S} (\log(y_s) - y_s \lambda^s) + \sum_{(i,j) \in L} \lambda_{(i,j)} x^{(n)}_{(i,j)}, \tag{13}$$

where

$$\lambda^s = \sum_{(i,j) \in L(s)} \lambda_{(i,j)}. \tag{14}$$

Note that the first term in (13) is separable in the session rates y_s. Therefore, the objective function for the dual problem of $\hat{\mathbf{Q}}$ at $\mathbf{x}^{(n)}$ is

$$\begin{aligned} D^{(n)}(\boldsymbol{\lambda}) &= \max_{\mathbf{y} \in \mathbf{Y}_b} L^{(n)}(\mathbf{y}, \boldsymbol{\lambda}) \\ &= \sum_{s \in S} B_s(\lambda^s) + \sum_{(i,j) \in L} \lambda_{(i,j)} x^{(n)}_{(i,j)}, \end{aligned} \tag{15}$$

where $B_s(\lambda^s) = \max_{0 \leq y_s \leq 1} (\log(y_s) - y_s \lambda^s)$. The dual problem is thus defined as

$$\min_{\boldsymbol{\lambda} \geq \mathbf{0}} D^{(n)}(\boldsymbol{\lambda}). \tag{16}$$

Since the logarithmic function is strictly concave and the link capacity constraints are linear, $\hat{\mathbf{Q}}$ is a convex program and hence has no duality gap. So at $\mathbf{x}^{(n)} = \mathbf{c}(\mathbf{p}^{(n)})$, when the dual problem of $\hat{\mathbf{Q}}$ achieves its optimum, the corresponding $\mathbf{y}$, obtained by maximizing the Lagrangian in (13), is the optimal solution to the primal problem $\hat{\mathbf{Q}}$. Note that for any $\boldsymbol{\lambda}$, the corresponding $\mathbf{y}$ that maximizes the Lagrangian is obtained as

$$y_s(\boldsymbol{\lambda}) = \min(\frac{1}{\lambda^s}, 1), \tag{17}$$

where λ^s is given by (14).

The dual problem can then be solved using gradient projection method, where the Lagrange multipliers are adjusted in the direction opposite to the gradient $\nabla D^{(n)}(\lambda)$:

$$\begin{aligned}\lambda_{(i,j)}^{(k+1)} &= \left[\lambda_{(i,j)}^{(k)} - \gamma \frac{\partial D^{(n)}}{\partial \lambda_{(i,j)}}(\boldsymbol{\lambda}^{(k)})\right]_+ \\ &= \left[\lambda_{(i,j)}^{(k)} + \gamma(y^{(i,j)}(\boldsymbol{\lambda}^{(k)}) - x_{(i,j)}^{(n)})\right]_+,\end{aligned} \tag{18}$$

where k is the iteration number, $\gamma > 0$ is the step size, $[z]_+ = \max\{z, 0\}$, $\boldsymbol{\lambda}^{(k)} = (\lambda_{(i,j)}^{(k)} : (i,j) \in L)$, and $y^{(i,j)}(\boldsymbol{\lambda}) = \sum_{s \in S(i,j)} y_s(\boldsymbol{\lambda})$ is the aggregate session rates at link (i,j).

The rate control algorithm at the transport layer is summarized as follows when the link rates $\mathbf{x}^{(n)}$ are given:

i. For each link $(i,j) \in L$,
   a) Compute the new price $\lambda_{(i,j)}^{(k+1)}$ using (18).
   b) Communicate new price $\lambda_{(i,j)}^{(k+1)}$ to the sources of all sessions that use link (i,j).

ii. For each session $s \in S$,
   a) Receive from the network the sum of the prices of links on s's path, λ^s (see (14)).
   b) Compute the new rate $y_s^{(k+1)}$ using (17).
   c) Communicate new rate $y_s^{(k+1)}$ to all links (i,j) on s's path.

iii. Repeat Steps i and ii until the session rates and link prices converge.

2) Attempt Probability Adjustment at the Link Layer: When the optimal session rates have been achieved at the given link rates, the proposed algorithm will work at the link layer to update the attempt probabilities using (8)-(9). The main purpose of the link attempt probabilities adjustment is to change the wireless link rates and ensures that the bottleneck link capacities are increased so that the total system utility can be improved further.

From (9), note that the partial derivative $\frac{\partial c_{rt}}{\partial p_{(i,j)}}$ is nonzero only for a link whose sink t is either node i or a node in the neighborhood of i. Therefore, the attempt probability of link (i,j) can be updated using only the link prices and attempt probabilities of the links within a two-hop neighborhood of (i,j), i.e., the link attempt probabilities can be updated using only local information.

3) Implementation of the Dual-Based Algorithm: The dual-based algorithm for end-to-end proportionally fair rate allocations in random access networks can be summarized as follows:

i. Set $n = 0$. For any link $(i,j) \in L$, choose initial attempt probabilities satisfying $0 < p_{(i,j)}^{(0)} < 1$.

ii. Compute the optimal link price and flow rates in a distributed manner using the rate control algorithm at the transport layer, assuming fixed link rates.

iii. Once the iterative procedure in step ii has converged, update the link attempt probabilities using (8) and (9).

iv. Increment n by 1. Repeat steps ii and iii until the link attempt probabilities have converged.

V. Penalty-Based Algorithm

A. Penalty-Based Approach

The penalty-based algorithm is based on the equivalent convex program introduced in Section III-D.

Let $\mathbf{z} = (z_s, s \in S)$, $\mathbf{w} = (\mathbf{p}, \mathbf{z})$. Define $\bar{U}_s(\mathbf{w}) = z_s$ for end-user $s \in S$, and define $g_l(\mathbf{w}) = \log\left(\sum_{s \in S(i,j)} e^{z_s}\right) - \log c_{ij}(\mathbf{p})$ for link $l = (i,j) \in L$. Note that the terms $g_l(\cdot)$ and its derivatives become unbounded at the boundary points, i.e., when $p_{(i,j)} \to 0$, $P_i \to 1$, or $z_s \to -\infty$. The approach we propose next requires bounded gradients (subgradients), and we impose this condition by optimizing over a set $\bar{\mathbf{W}}$ which excludes the boundary region, and is defined as $\bar{\mathbf{W}} = \{(\mathbf{p}, \mathbf{z}) : \mathbf{p} \in \bar{\mathbf{P}}_b, \mathbf{z} \in \bar{\mathbf{Z}}_b\}$, where $\bar{\mathbf{P}}_b = \{\mathbf{p} : p_{(i,j)} \geq \epsilon \ \forall (i,j) \in L, P_i = \sum_{j \in O_i} p_{(i,j)}, P_i \leq 1 - \epsilon \ \forall i \in N\}$, and $\bar{\mathbf{Z}}_b = \{\mathbf{z} : z_s \geq -M \ \forall s \in S\}$, and M is a large positive constant.

Note that we can make $\bar{\mathbf{W}}$ as close to the original feasibility region as we want, by making ϵ sufficiently small, and M sufficiently large.

The problem posed in (5) under these additional restrictions can be written as

$$\bar{\mathbf{Q}}: \quad \begin{array}{ll} \max & \sum_{s \in S} \bar{U}_s(\mathbf{w}), \\ \text{s.t.} & g_l(\mathbf{w}) \leq 0 \qquad \forall l \in L, \\ & \mathbf{w} \in \bar{\mathbf{W}}. \end{array} \tag{19}$$

It is worth noting that $\bar{U}_s(\mathbf{w})$ is a concave (linear) function and $g_l(\mathbf{w})$ is a convex function.

A penalty function should impose a positive penalty at infeasible points and no penalty at feasible points. A suitable *penalty function* β for the constraints in $\bar{\mathbf{Q}}$ is defined by

$$\beta(\mathbf{w}) = \sum_{l \in L} \phi(g_l(\mathbf{w})) \tag{20}$$

where ϕ is a continuous function satisfying the following:

$$\phi(s) = 0 \quad \text{if } s \leq 0 \quad \text{and} \quad \phi(s) > 0 \quad \text{if } s > 0. \tag{21}$$

In the following, the function ϕ is of the form

$$\phi(s) = [\max\{0, s\}]^m. \tag{22}$$

where m is a positive integer. The *auxiliary function* for $\bar{\mathbf{Q}}$ is then defined by $\sum_{s \in S} \bar{U}_s(\mathbf{w}) - \kappa\beta(\mathbf{w})$, where κ is the "penalty scaling factor".

Instead of solving $\bar{\mathbf{Q}}$ directly, the penalty-based approach attempts to solve the following problem [4]:

$$\max_{\mathbf{w} \in \bar{\mathbf{W}}} \{\sum_{s \in S} \bar{U}_s(\mathbf{w}) - \kappa\beta(\mathbf{w})\} = \theta(\kappa). \tag{23}$$

Note that $\theta(\kappa)$ is actually an unconstrained optimization problem. Since $\bar{U}_s(\mathbf{w})$ is concave while $\beta(\mathbf{w})$ is convex (when ϕ is defined by (22)), $\theta(\kappa)$ can be computed by standard convex programming techniques. Note that the differentiability of ϕ depends on the value of m, which dictates whether we need to use a subgradient-based method or a gradient-based method. In the following, we present both these approaches, and argue that they have different properties.

B. Subgradient Method when $m = 1$

If $m = 1$, $\theta(\kappa)$ is defined as

$$\theta(\kappa) = \max_{\mathbf{w} \in \bar{\mathbf{W}}} \left\{ \sum_{s \in S} \bar{U}_s(\mathbf{w}) - \kappa \sum_{l \in L} \max\{0, g_l(\mathbf{w})\} \right\}. \quad (24)$$

It can be shown that there exists a scalar constant $\bar{A}$ such that the set of optimal solutions of $\bar{\mathbf{Q}}$ coincides with the optimal solutions of $\theta(\kappa)$ for any $\kappa > \bar{A}$. It is also worth noting that the objective of $\theta(\kappa)$ is not a smooth function of $\mathbf{w}$, and a subgradient method is applied to iteratively search for the optimal solution of $\theta(\kappa)$.

1) Subgradient Method: We now present the subgradient method to solve $\theta(\kappa)$ in an iterative manner. Let $p_{(i,j)}^{(n)}$ and $z_s^{(n)}$ respectively denote the values of $p_{(i,j)}$ and z_s at the nth iterative step, and let $\mathbf{p}^{(n)} = (p_{(i,j)}^{(n)}, (i,j) \in L)$. Let γ_n be the step size at the nth iteration. The subgradient projection method can be summarized as follows:

$$p_{(i,j)}^{(n+1)} = \left[p_{(i,j)}^{(n)} + \gamma_n \xi_{(i,j)}^{(n)} \right]_{\bar{\mathbf{P}}_b}, \quad (25)$$

$$z_s^{(n+1)} = \left[z_s^{(n)} + \gamma_n \chi_s^{(n)} \right]_{\bar{\mathbf{Z}}_b}, \quad (26)$$

where $[\cdot]_{\bar{\mathbf{P}}_b}$ and $[\cdot]_{\bar{\mathbf{Z}}_b}$ are projection operators. Here, $\xi_{(i,j)}^{(n)}$ and $\chi_s^{(n)}$ are subgradients of the objective function of $\theta(\kappa)$ (cf. (23)) with respect to $p_{(i,j)}$ and z_s respectively, at the nth iteration, and are defined as

$$\xi_{(i,j)}^{(n)} = \kappa \sum_{(r,t) \in L} \frac{\varepsilon_{(r,t)}^{(n)}}{x_{(r,t)}^{(n)}} \cdot \frac{\partial c_{rt}}{\partial p_{(i,j)}}(\mathbf{p}^{(n)}), \quad (27)$$

$$\chi_s^{(n)} = 1 - \kappa \sum_{(i,j) \in L(s)} \frac{\varepsilon_{(i,j)}^{(n)} e^{z_s^{(n)}}}{\sum_{v \in S(i,j)} e^{z_v^{(n)}}}. \quad (28)$$

Note that $\frac{\partial c_{rt}}{\partial p_{(i,j)}}$ is defined by (9). Also note that $\varepsilon_{(i,j)}^{(n)}$ is the "link congestion indicator" for link $(i,j) \in L$ at the nth iteration, and is defined as

$$\varepsilon_{(i,j)}^{(n)} = \begin{cases} 0 & \text{if } \sum_{s \in S(i,j)} e^{z_s^{(n)}} \leq x_{(i,j)}^{(n)}, \\ 1 & \text{otherwise.} \end{cases} \quad (29)$$

Since e^{z_s} ($= y_s$) is interpreted as the rate of session s, $\frac{e^{z_s}}{\sum_{v \in S(i,j)} e^{z_v}}$ in (28) can be interpreted as the fraction of the overall traffic on link (i,j) contributed by session s.

In (27), $\frac{\partial c_{rt}}{\partial p_{(i,j)}}(\mathbf{p}^{(n)})$ depicts how the attempt probability on link (i,j) impacts the rate on link (r,t); from (27), note that this impact is weighted by the inverse of the rate on that link.

2) Convergence Analysis: We now provide the convergence analysis for the iterative procedures stated in (25)-(26).

Denote $\bar{\mathbf{W}}^*$ as the set of optimal solutions of $\bar{\mathbf{Q}}$. Let $\rho(\mathbf{w}, \mathbf{S}) = \min_{\mathbf{w}' \in \mathbf{S}} ||\mathbf{w} - \mathbf{w}'||$ denote the Euclidean distance of a point $\mathbf{w}$ from any set $\mathbf{S}$. Let $\Phi_\delta(\mathbf{S})$ be the set of all points whose distance from $\mathbf{S}$ is at most δ for any compact set $\mathbf{S}$, i.e. $\Phi_\delta(\mathbf{S}) = \{\mathbf{w} : \rho(\mathbf{w}, \mathbf{S}) \leq \delta\}$. From the convergence results for the subgradient method [3], we have the following results.

Theorem 2: Let $\{\mathbf{w}^{(n)}\}$ denote the sequence of vectors defined by the iterative procedure stated in (25)-(26). If the step sizes satisfy the following criteria

$$\lim_{n \to \infty} \gamma_n = 0, \qquad \sum_{n=0}^{\infty} \gamma_n = \infty, \quad (30)$$

then there exists a $\bar{A} < \infty$, such that for all $\kappa > \bar{A}$,

$$\lim_{n \to \infty} \rho\left(\mathbf{w}^{(n)}, \bar{\mathbf{W}}^*\right) = 0.$$

Theorem 2 states that the distance of the vector of link attempt probabilities and session rates from the set of optimal solutions decreases to zero if the step sizes satisfy the constraints in (30). If the step sizes are constant, we have slightly weaker convergence results.

Corollary 1: Let $\{\mathbf{w}^{(n)}(\gamma)\}$ denote the sequence of vectors defined by the iterative procedure stated in (25)-(26) with $\gamma_n = \gamma \ \forall n$. Then, there exists a $\bar{A} < \infty$ and a function $\delta(\gamma)$ such that $\lim_{\gamma \to 0+} \delta(\gamma) = 0$, and for all $\kappa > \bar{A}$,

$$\lim_{n \to \infty} \rho\left(\mathbf{w}^{(n)}(\gamma), \Phi_{\delta(\gamma)}(\bar{\mathbf{W}}^*)\right) = 0.$$

Loosely speaking, Corollary 1 states that for a constant step size, the vector of link attempt probabilities and session rates "converges to a neighborhood" around the optimum. Moreover, the size of this neighborhood becomes arbitrarily small with decreasing step-size.

3) Distributed Algorithm: We now formally state the subgradient update procedures for the penalty-based algorithm when $m = 1$. The penalty-based algorithm is summarized as follows:

i. For any link $(i,j) \in L$, choose initial attempt probabilities satisfying $0 < p_{(i,j)} < 1$. For any session s, choose an initial session rate satisfying $y_s > 0$, and set $z_s = \log y_s$.
ii. For link $(i,j) \in L$, update link congestion indicator $\varepsilon_{(i,j)}^{(n)}$ using (29).
iii. Update link attempt probabilities using (25) and (27).
iv. For session $s \in S$, update z_s, the (logarithmic) utility of the session, using (26) and (28). The new session rate is then obtained as $y_s = e^{z_s}$.
v. Repeat steps ii-iv until the link attempt probabilities and session rates have converged.

C. Gradient Method when $m \geq 2$

If $m \geq 2$, optimal solutions of $\theta(\kappa)$ do not give the exact solutions to $\bar{\mathbf{Q}}$. However, from Theorem in 9.2.2 of [4], we have

$$\max_{\mathbf{w} \in \bar{\mathbf{W}}} \left\{ \sum_{s \in S} \bar{U}_s(\mathbf{w}) \colon g_l(\mathbf{w}) \leq 0 \right\} = \lim_{\kappa \to \infty} \theta(\kappa).$$

Therefore, by making κ is sufficiently large, $\theta(\kappa)$ can be made to represent $\bar{\mathbf{Q}}$ closely. Also note that the penalty function is smooth when $m \geq 2$, and hence the gradient method is directly applicable.

1) Gradient Method: The gradient projection method updates $p_{(i,j)}$ and z_s as follows:

$$p_{(i,j)}^{(n+1)} = \left[p_{(i,j)}^{(n)} + \gamma \xi_{(i,j)}^{(n)}\right]_{\bar{\mathbf{P}}_b}, \tag{31}$$

$$z_s^{(n+1)} = \left[z_s^{(n)} + \gamma \chi_s^{(n)}\right]_{\bar{\mathbf{Z}}_b}, \tag{32}$$

where γ is the (constant) step size. Here, $\xi_{(i,j)}^{(n)}$ and $\chi_s^{(n)}$ are the gradients of the objective function of $\theta(\kappa)$ (cf. (23)) with respect to $p_{(i,j)}$ and z_s respectively, at the nth iterative step, and are defined as

$$\xi_{(i,j)}^{(n)} = m\kappa \sum_{(r,t)\in L} \left[\max\left\{0, g_{(i,j)}(\mathbf{w})\right\}\right]^{m-1} \cdot \frac{\varepsilon_{(r,t)}^{(n)}}{x_{(r,t)}^{(n)}} \cdot \frac{\partial c_{rt}(\mathbf{p}^{(n)})}{\partial p_{(i,j)}}, \tag{33}$$

$$\chi_s^{(n)} = 1 - m\kappa \sum_{(i,j)\in L(s)} \frac{\left[\max\left\{0, g_{(i,j)}(\mathbf{w})\right\}\right]^{m-1} \varepsilon_{(i,j)}^{(n)} e^{z_s^{(n)}}}{\sum_{v\in S(i,j)} e^{z_v^{(n)}}}, \tag{34}$$

where $\frac{\partial c_{rt}}{\partial p_{(i,j)}}$ and $\varepsilon_{(i,j)}^{(n)}$ are defined in (9) and (29), respectively.

2) Convergence Analysis: It can be verified that the gradients defined in (33)-(34) satisfy the Lipschitz continuity condition [2]. Since the objective in $\theta(\kappa)$ is a concave function, there exists a $\bar{\gamma}_\kappa \in \Re^+$ such that for any $0 < \gamma < \bar{\gamma}_\kappa$, the sequence stated in (31)-(32) converges to the globally optimal solution of $\theta(\kappa)$.

VI. Comparison of the Algorithms

The dual-based algorithm and the penalty-based algorithm solve the proportionally fair rate control problem using different procedures, and from a practical viewpoint, each algorithm has certain advantages over the other.

In the dual-based algorithm, the separation between the transport layer and the link layer is better maintained. The link rates are updated at the link layer and the session rates are adjusted at the transport layer. The cross-layer cooperation between the transport layer and the link layer lies in the fact that, the link layer adjusts link probabilities using the link prices computed by the transport layer, and the transport layer adjusts its session rates using the link rates computed by the link layer. Note that the dual-based algorithm has embedded loops. In the inner loop (in a smaller time scale), the transport layer searches for the session rates and link prices, and in the outer loop (in a larger time scale), the link layer adjusts the link attempt probabilities and updates the link rates. The dual-based algorithm converges to the optimal solutions when the link layer chooses the 'right' link attempt probabilities (and hence the 'right' link rates) such that the bottlenecks are optimally 'shuffled' around in the network, and the transport layer finds the optimal session rates for these 'right' link rates.

In contrast, the penalty-based algorithm shows lesser modularity than the dual-based algorithm. At the transport layer, a session updates its rate based on its contribution to the traffic at the congested links on its path. At the same time, the link layer updates its attempt probabilities by considering how the neighboring congested links will be affected. Thus, in this case, updates occur at the transport layer and at the link layer at the same time scale, and information is exchanged between different layers at a faster time-scale than that in the dual-based algorithm. However, note that the penalty-based algorithm does not have any embedded loops.

In the next section, we evaluate and compare the performance of the two approaches through simulation experiments.

VII. Simulation Investigation

In this section, we investigate the performance of the two distributed algorithms, i.e., the dual-based algorithm and the penalty-based algorithm, in providing end-to-end rate control in multi-hop random access networks. Simulation results for a simple ad-hoc network topology is shown below; simulations carried out on various other network topologies/scenarios confirm that both algorithms achieve the globally optimal solutions.

The network that we consider is composed of 6 nodes and 8 links, and is shown in Fig. 1.

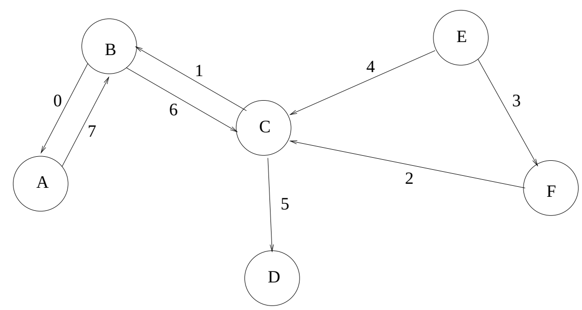

Fig. 1. An ad-hoc network.

Three end-to-end sessions, namely, f_0, f_1, and f_2 are setup in this network. The source, the sink, and the path of the three sessions are shown in Table I.

TABLE I
The Source, Sink, and Path of the Sessions.

Session	Source Node	Sink Node	Links on the Path
f_0	E	A	3, 2, 1, 0
f_1	E	D	4, 5
f_2	A	D	7, 6, 5

The globally optimal solutions given by Matlab, and the solutions obtained through synchronous updates of the dual-based and the penalty-based algorithms are presented in Table II. Through comparison, it can be seen that, in this ad-hoc network, both algorithms achieve the globally optimal solutions.

Dual-Based Algorithm: In this simulation, the step size for the attempt probability adjustment at the link layer is set to 5×10^{-4}. Fig. 2 shows how link attempt probabilities, link rates, session rates, and the aggregate utility converge with each iteration in the link layer when the dual-based algorithm is adopted. In the figure, the link attempt probabilities and link capacities of only links 1, 3, 5, 7 are shown; the corresponding quantities for links 0, 2, 4, 6 show a similar trend. From the plots and Table II, it can be seen that all the variables are within 10% of their globally optimal values after 300 iterations at the link layer.

TABLE II

THE OPTIMAL RESULTS AND THE SOLUTIONS GIVEN BY THE DISTRIBUTED ALGORITHMS.

Variables	p_0	p_1	p_2	p_3
optimal solutions	0.06475	0.1003	0.2102	0.09548
dual-based algorithm	0.0688	0.1019	0.2099	0.1040
penalty-based algorithm	0.0649	0.0943	0.2054	0.0898
Variables	p_4	p_5	p_6	p_7
optimal solutions	0.3488	0.2103	0.2898	0.1971
dual-based algorithm	0.3314	0.2063	0.2677	0.1913
penalty-based algorithm	0.3584	0.2133	0.2925	0.2101
Variables	x_0	x_1	x_2	x_3
optimal solutions	0.05198	0.05198	0.05198	0.05198
dual-based algorithm	0.0556	0.0552	0.0552	0.0543
penalty-based algorithm	0.0537	0.0478	0.0496	0.0524
Variables	x_4	x_5	x_6	x_7
optimal solutions	0.1226	0.2103	0.0877	0.0877
dual-based algorithm	0.1206	0.2031	0.0832	0.0881
penalty-based algorithm	0.1266	0.2133	0.0888	0.0934
Variables	y_0	y_1	y_2	U^*
optimal solutions	0.05198	0.1226	0.0877	-7.4897
dual-based algorithm	0.0543	0.1198	0.0832	-7.5187
penalty-based algorithm	0.0478	0.1266	0.0878	-7.5329

Recall that the dual-based algorithm is implemented at the link and the transport layers at different time scales. At each link layer iteration, when the link attempt probabilities have been adjusted, the algorithm then works at the transport layer to compute the optimal session rates and link prices by an iterative search. Therefore the complexity of the algorithm should be estimated by the number of iterations at both the link layer and the transport layer. Fig. 3 plots the total number of iterations (the link and the transport layers combined) versus the number of iterations at the link layer. In the simulation, the algorithm at the transport layer terminates when the variation in the session rates is less than 10^{-3}. From the figure and Table II, it can be seen that for 300 iterations at the link layer (when all the variables are within 10% of their globally optimal values), the total number of iterations at the link layer and the transport layer combined is about 3000. Therefore there are roughly 10 iterations at the transport layer for each rate update in the link layer. Note that each iteration at the transport layer needs end-to-end communication, and therefore requires at least one RTT (which is typically in the order of msecs to tens of msecs). Therefore, assuming that the iterations at the transport layer occur once every few RTTs, the overall convergence time of the algorithm for a medium sized network should range from a few seconds to a few minutes.

Penalty-Based Algorithm: In this simulation, we set the step size to 1.5×10^{-6}. Let us first consider the case of $m = 1$, when the subgradient method is used for the iterative search. Fig. 4 shows how the link attempt probabilities, link rates, session rates, and the aggregate utility converge when the penalty-based algorithm is used. It can be seen from the plots that after about 2500 iterations, all the variables are quite close to their globally optimal values. Note that the total number of iterations for this algorithm is in the same order as that of the dual-based algorithm. In this case too, each iteration requires end-to-end communication and therefore would require at least one RTT. The overall convergence time is expected to range from a few seconds to a few minutes.

Note in Fig. 4 that there is an obvious thickening of

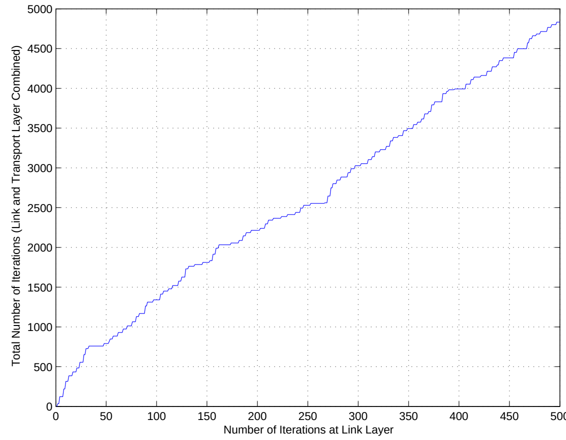

Fig. 3. The total number of iterations (link and transport layers combined) vs. the number of iterations at the link layer.

the computed link attempt probabilities and link capacities, implying that the computed values do not exactly converge to the optimal values, but fluctuates around them. Recall from Section V-B that we need step sizes close to zero in order to guarantee exact convergence. If the step size is a constant, but small, as in this case, then we can only guarantee that our algorithm achieves solutions that are close-to-optimal. When the total traffic is close to the link rate, the link congestion indicator fluctuates between 0 and 1, as can be expected from intuition. This causes the fluctuations in the link attempt probabilities and rates like those observed in Fig. 4. Smaller step sizes cause smaller fluctuations, but also result in lower convergence speeds. Thus the choice of the step size is a trade-off between the convergence speed and the magnitude of fluctuations. In this case, the step size has been chosen appropriately, based on this trade-off. In practice, a session could choose large step sizes initially, to ensure fast convergence; subsequently, the step sizes can be reduced once the rate starts fluctuating around the same mean value.

Now let us consider the case of $m = 2$, when the gradient method is used for the iterative search. Fig. 5 shows how the link attempt probabilities, link rates, session rates, and the aggregate utility converge in this case, in a scenario similar to the one considered above. Comparing with Fig. 4 (the $m = 1$ case), we see that the fluctuations is considerably reduced in the $m = 2$ case. Unlike the subgradient, the gradient is a "smooth" function; therefore, the gradient method results is smooth variations of the variables. Note however, this reduction in fluctuations could come as the cost of slower convergence and lesser accuracy; a careful comparison of the results obtained in this case also supports this fact.

VIII. CONCLUSION

In this paper, we address the end-to-end proportionally fair rate control problem in a multi-hop random access network with a general network topology. In wireless networks, the feasible rate region is a complex, non-separable function of the link attempt probabilities. Therefore the optimal rate control problem in wireless networks is much more difficult than its wired network counterpart.

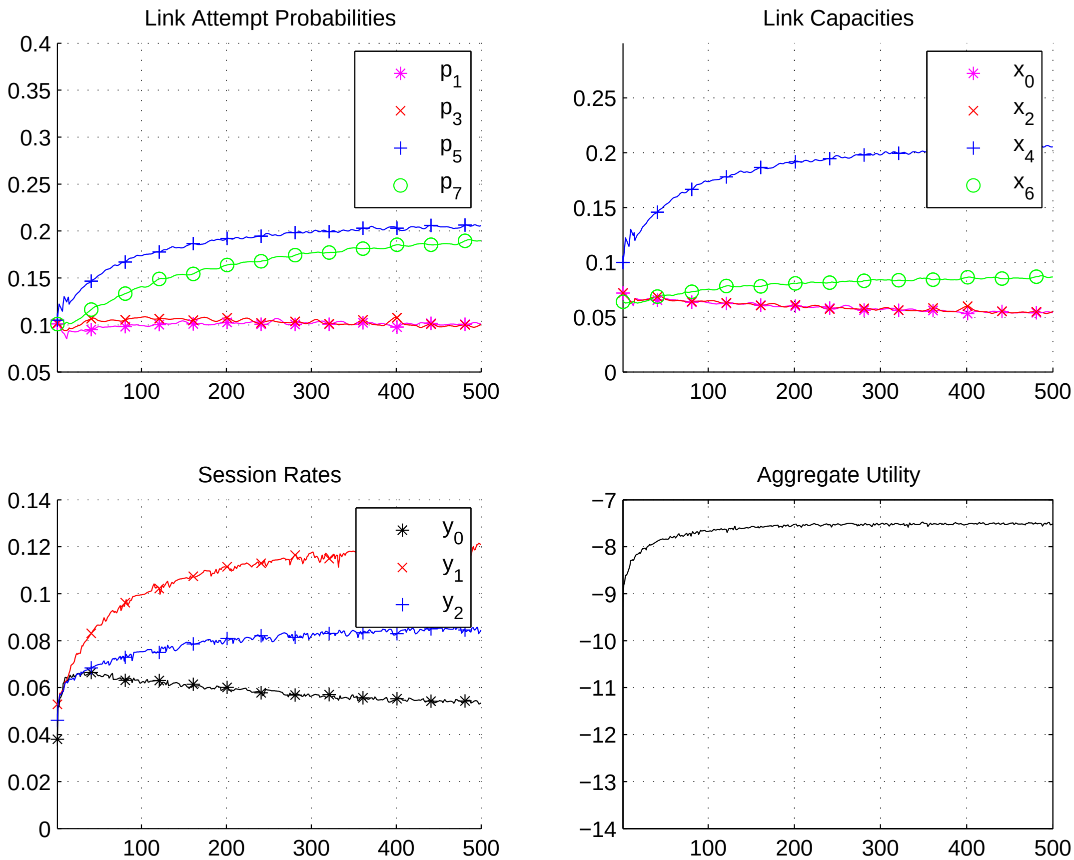

Fig. 2. The link attempt probabilities, link rates, session rates, and the aggregate utility when the *dual-based* algorithm is used. (The x axis denotes the number of iterations at the link layer.)

We formulate the end-to-end rate control problem in random access networks as an optimization problem, and propose two cross-layer algorithms to solve the problem, both of which can be implemented in a distributed manner. Using nonlinear optimization techniques, we prove that both algorithms converge to the global optimum. Simulation results under various network scenarios also support our analytical observations.

REFERENCES

[1] D. Bertsekas and R. Gallager, *Data Networks*, Prentice Hall, 1992.

[2] D. P. Bertsekas, *Nonlinear Programming*, Athena Scientific, 1999.

[3] N. Z. Shor, *Minimization Methods for Non-differentiable Functions*, Springer-Verlag, 1985.

[4] M. S. Bazaraa, H. D. Sherali, C.M. Shetty, *Nonlinear Programming: Theory And Algorithms*, Wiley, 1993.

[5] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University, 2004.

[6] F. P. Kelly, "Charging and Rate Control for Elastic Traffic", *European Transactions on Telecommunications*, vol. 8, no. 1, 1997, pp. 33-37.

[7] F. Kelly, A. Maulloo, D. Tan, "Rate Control for Communication Networks: Shadow Prices, Proportional Fairness and Stability", *Journal of Operations Research Society*, vol. 49, no. 3, 1998, pp. 237-252.

[8] X. Lin and N. B. Shroff, "Joint Rate Control and Scheduling in Multihop Wireless Networks", *Proc. 43rd IEEE Conference on Decision and Control*, Paradise Island, Bahamas, December 2004.

[9] S. Low, D. E. Lapsley, "Optimization Flow Control, I: Basic Algorithm and Convergence", *IEEE/ACM Transactions on Networking*, vol. 7, no. 6, December 1999.

[10] T. Nandagopal, T.-E. Kim, X. Gao, V. Bharghavan, "Achieving MAC Layer Fairness in Wireless Packet Networks", in *Proc. ACM/IEEE MobiCom '00*, pp. 8798, Boston, USA, August 2000.

[11] T. Ozugur, M. Naghshineh, P. Kermani, C. M. Olsen, B. Rezvani, J. A. Copeland, "Balanced Media Access Methods for Wireless Networks", in *Proc. ACM/IEEE MobiCom '98*, Dallas, USA, October 1998.

[12] L. Tassiulas, S. Sarkar, "Maxmin Fair Scheduling in Wireless Networks", in *Proceedings of Infocom 2002*, pp. 763-772, New York, USA, June 2002.

[13] M. Johnasson, L. Xiao, "Cross-layer optimization of wireless networks using nonlinear column generation", accept by *IEEE Transactions on Wireless Communications*.

[14] L. Xiao, M. Johnasson, S. Boyd, "Simultaneous Routing and Resource Allocation via Dual Decomposition", *IEEE Transactions on Communications*, Vol. 52, No. 7, pages 1136-1144, July 2004.

[15] M. Chiang, "To Layer or Not To Layer: Balancing Transport and Physical Layers in Wireless Multihop Networks", in *Proceedings of Infocom 2004*, Hong Kong, China, March 2004.

[16] K. Kar, S. Sarkar, L. Tassiulas, "Achieving Proportionally Fair Rates using Local Information in Multi-hop Aloha Networks", *Proceedings of Annual Allerton Conference*, Urbana-Champaign, USA, October 2003.

[17] X. Wang, K. Kar, "Distributed Algorithms for Max-min Fair Rate Allocation in Aloha Networks", *Proceedings of Annual Allerton Conference*, *Urbana-Champaign*, USA, October 2004.

APPENDIX

A. Proof of Lemma 1

Proof: If we denote $z_s = \log y_s$, then the objective in (3) can be rewritten as $U = \sum_{s \in S} z_s$, which is still a concave function. Since the logarithmic function is strictly increasing, each link constraint in (3) can then be rewritten as

$$\log\left(\sum_{s \in S(i,j)} e^{z_s}\right) - \log(c_{ij}(\mathbf{p})) \le 0. \tag{35}$$

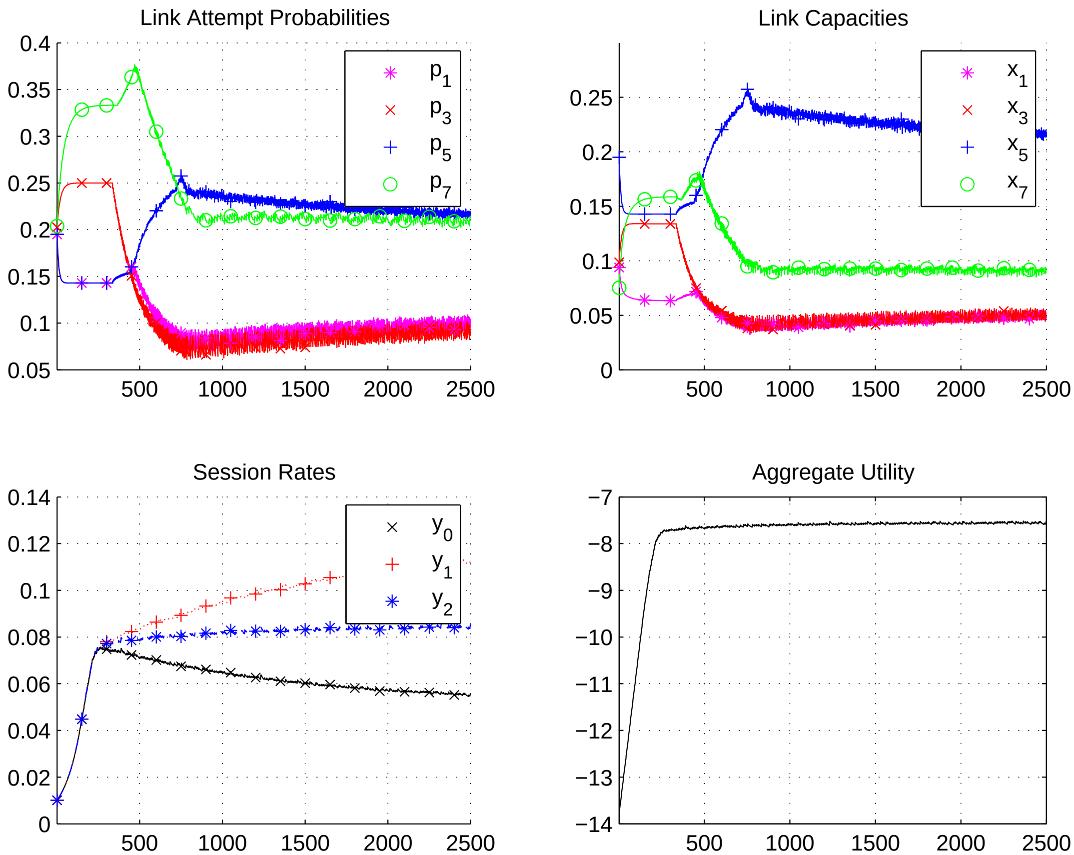

Fig. 4. The link attempt probabilities, link rates, session rates, and the aggregate utility when the *penalty-based* algorithm is used, with $m = 1$. (The x axis denotes the total number of iterations at the link and the transport layers.)

It is worth noting that $\log\left(\sum_{s\in S(i,j)} e^{z_s}\right)$ is a convex function for z_s (for proof, see [5] or [15]), and $\log(c_{ij}(\mathbf{p}))$ is concave in $\mathbf{p}$ (for proof, see [16]). It then follows that the set of constraints in (35) is convex.

Therefore the problem in (5), which is equivalent to (3), is a convex programming problem. ■

B. Proof Outline of Theorem 1

Let $\boldsymbol{\lambda}(\bar{\mathbf{x}})$ denote the optimal dual variables (Lagrange multipliers) of $\hat{\mathbf{Q}}$ parameterized at $\bar{\mathbf{x}}$. Then $\boldsymbol{\lambda}(\cdot)$ is continuously differentiable, and according to the *Sensitivity Theorem* [2],

$$\nabla_{\mathbf{x}} \hat{U}(\bar{\mathbf{x}}) = \boldsymbol{\lambda}(\bar{\mathbf{x}}). \tag{36}$$

Therefore $\hat{U}$ is total differentiable in $\mathbf{x}$, and the differential is

$$d\hat{U} = \sum_{l\in L} \frac{\partial \hat{U}}{\partial x_l} dx_l = \sum_{l\in L} \lambda_l(\mathbf{x}) dx_l.$$

Since $\tilde{U}(\mathbf{p}) = \hat{U}(\mathbf{c}(\mathbf{p}))$, and since $\mathbf{c}(\mathbf{p})$ is total differentiable in $\mathbf{p}$, it follows that $\tilde{U}(\mathbf{p})$ is total differentiable in $\mathbf{p}$. Therefore we have the following property.

Lemma 2: Let $\bar{\mathbf{d}} = (\bar{d}_{ij} : (i,j) \in L)$ be defined as

$$\bar{d}_{ij} = \sum_{(r,t)\in L} \bar{\lambda}_{(r,t)} \frac{\partial c_{rt}}{\partial p_{(i,j)}}(\bar{\mathbf{p}}), \tag{37}$$

where $\bar{\boldsymbol{\lambda}}$ is the vector of Lagrange multipliers of $\hat{\mathbf{Q}}$ at $\mathbf{x} = \mathbf{c}(\bar{\mathbf{p}})$. Then $\bar{\mathbf{d}}$ is the gradient direction of $\tilde{U}(\mathbf{p})$ at $\bar{\mathbf{p}}$, i.e., $\nabla_{\mathbf{p}} \tilde{U}(\bar{\mathbf{p}}) = \bar{\mathbf{d}}$.

It can be verified that the Lipschitz continuity condition [2] holds here. Thus there exists an $\bar{\alpha} \in \Re^+$ such that for $0 < \alpha < \bar{\alpha}$, the sequence $\{\mathbf{p}^{(n)}(\alpha)\}$ generated using the procedures stated in (8)-(11), which update the link attempt probabilities in the gradient direction, converge to a local optimum point of $\tilde{\mathbf{Q}}$.

Since $\tilde{\mathbf{Q}}$ and $\mathbf{Q}$ are equivalent, a local optimum point of $\tilde{\mathbf{Q}}$ corresponds to a local optimum point of $\mathbf{Q}$. Therefore the following property holds true.

Lemma 3: Denote $\mathbf{p}^*$ as the local optimum point of $\tilde{\mathbf{Q}}$, and $\mathbf{y}^*$ is obtained using (12), i.e.,

$$\mathbf{y}^* = \arg\max \left\{ \sum_{s\in S} \log(y_s) \,\middle|\, \sum_{s\in S(i,j)} y_s \le c_{ij}(\mathbf{p}^*), (i,j) \in L \right\}, \tag{38}$$

then $(\mathbf{p}^*, \mathbf{y}^*)$ is the local optimum point of $\mathbf{Q}$.

Note that $\mathbf{y}$ obtained by solving (12) is continuous in $\mathbf{p}$. From Lemma 3 we conclude that, the stationary point $\mathbf{p}^*$ of the sequence $\{\mathbf{p}^{(n)}(\alpha)\}$, which is generated using the procedures stated in (8)-(11), and the corresponding $\mathbf{y}^*$, which is calculated using (12), constitute a local optimum of $\mathbf{Q}$.

To show that the dual-based algorithm actually converges to the globally optimal values, we need the following property.

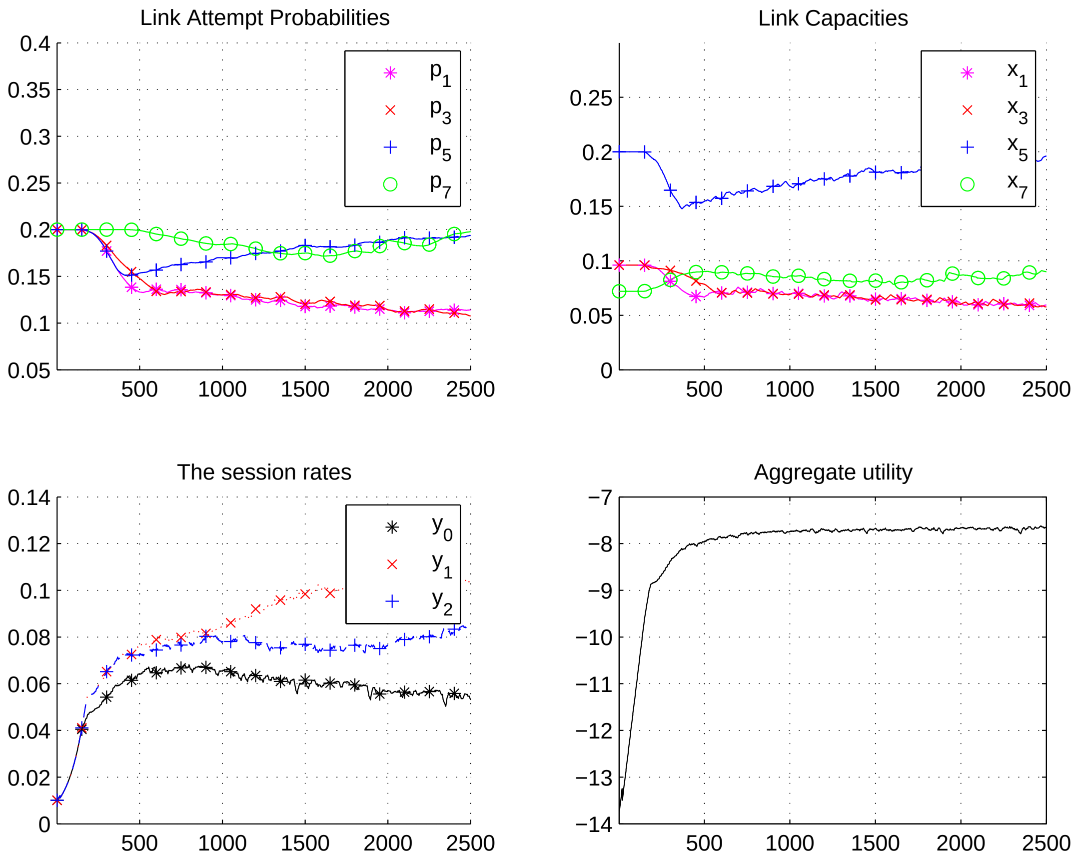

Fig. 5. The link attempt probabilities, link rates, session rates, and the aggregate utility when the *penalty-based* algorithm is used, with $m = 2$. (The x axis denotes the total number of iterations at the link and the transport layers.)

Lemma 4: If $(\mathbf{y}^*, \mathbf{p}^*)$ satisfies the first order necessary condition for optimality for the nonlinear program in (3), then $(\mathbf{z}^*, \mathbf{p}^*)$, where $z_s^* = \log(y_s^*)$, $s \in S$, will satisfy the first order necessary condition for optimality for the convex program in (5). Conversely, if $(\mathbf{z}^*, \mathbf{p}^*)$ satisfies the first order necessary condition for optimality for the problem in (5), then $(\mathbf{y}^*, \mathbf{p}^*)$, where $y_s^* = e^{z_s^*}$, $s \in S$, will satisfy the first order necessary condition for the problem in (3).

Proof: For simplicity of exposition, we outline the proof without considering the boundary constraints $\mathbf{p} \in \mathbf{P}_b$; the analysis can however be easily extended to incorporate those constraints. Let $g_{(i,j)}(\mathbf{y}, \mathbf{p}) = \sum_{s \in S(i,j)} y_s - c_{ij}(\mathbf{p})$, and $\tilde{g}_{(i,j)}(\mathbf{z}, \mathbf{p}) = \log(\sum_{s \in S(i,j)} e^{z_s}) - \log c_{ij}(\mathbf{p})$ represent the capacity constraint for link $(i, j) \in L$ in the problems (3) and (5), respectively. Denote $\mathbf{y}^* = (y_s^*, s \in S)$, $\mathbf{z}^* = (z_s^* = \log(y_s^*), s \in S)$, and $\mathbf{p}^* = (p_{(i,j)}^*, (i, j) \in L)$. Let U be the objective function; then $U = \sum_{s \in S} \log(y_s) = \sum_{s \in S} z_s$.

If $\mathbf{y}^*$ and $\mathbf{p}^*$ satisfy the first order necessary condition in (3), then there exists $u_{(i,j)}^* \geq 0$ for $(i, j) \in I$ such that

$$\left(-\nabla U + \sum_{(i,j) \in I} u_{(i,j)}^* \nabla g_{(i,j)} \right) \Bigg|_{\mathbf{y}^*, \mathbf{p}^*} = 0, \qquad (39)$$

where $I = \{(i, j) \in L : g_{(i,j)}|_{\mathbf{y}^*, \mathbf{p}^*} = 0\}$.

Therefore, for any $s \in S$, we have

$$\begin{aligned} &\left(-\frac{\partial U}{\partial y_s} + \sum_{(i,j) \in I} u_{(i,j)}^* \frac{\partial g_{(i,j)}}{\partial y_s} \right) \Bigg|_{\mathbf{y}^*, \mathbf{p}^*} \\ = &\left(-\frac{1}{y_s} + \sum_{(i,j) \in I \cap L(s)} u_{(i,j)}^* \right) \Bigg|_{\mathbf{y}^*, \mathbf{p}^*} = 0, \end{aligned} \qquad (40)$$

and for any $(r, t) \in L$, we have

$$\begin{aligned} &\left(-\frac{\partial U}{\partial p_{(r,t)}} + \sum_{(i,j) \in I} u_{(i,j)}^* \frac{\partial g_{(i,j)}}{\partial p_{(r,t)}} \right) \Bigg|_{\mathbf{y}^*, \mathbf{p}^*} \\ = &\sum_{(i,j) \in I} u_{(i,j)}^* \frac{\partial c_{ij}}{\partial p_{(r,t)}} \Bigg|_{\mathbf{y}^*, \mathbf{p}^*} = 0. \end{aligned} \qquad (41)$$

Let $\tilde{I} = \{(i, j) \in L : \tilde{g}_{(i,j)}|_{\mathbf{y}^*, \mathbf{p}^*} = 0$. Obviously $I = \tilde{I}$, and therefore $g_{(i,j)}|_{\mathbf{y}^*, \mathbf{p}^*} = 0$ for any (i, j) in $\tilde{I}$. We take $\tilde{u}_{(i,j)}^* = u_{(i,j)}^* c_{ij}(\mathbf{p}^*) = u_{(i,j)}^* \sum_{v \in S(i,j)} e^{z_v^*}$ for all $(i, j) \in \tilde{I}$.

Then, for any $s \in S$, we have

$$\begin{aligned}
&\left.\left(-\frac{\partial U}{\partial z_s} + \sum_{(i,j)\in\tilde{I}} \tilde{u}^*_{(i,j)} \frac{\partial \tilde{g}_{(i,j)}}{\partial z_s}\right)\right|_{\mathbf{z}^*,\mathbf{p}^*} \\
= \quad &\left.\left(-1 + \textstyle\sum_{(i,j)\in\tilde{I}\cap L(s)} \tilde{u}^*_{(i,j)} \frac{e^{z_s}}{\sum_{v\in S(i,j)} e^{z_v}}\right)\right|_{\mathbf{z}^*,\mathbf{p}^*} \\
= \quad &\left.\left(-1 + \textstyle\sum_{(i,j)\in I\cap L(s)} u^*_{(i,j)} e^{z_s}\right)\right|_{\mathbf{z}^*,\mathbf{p}^*} \\
= \quad &\left.\left(-1 + \textstyle\sum_{(i,j)\in I\cap L(s)} u^*_{(i,j)} y_s\right)\right|_{\mathbf{y}^*,\mathbf{p}^*} \\
= \quad & y_s \left.\left(-\frac{1}{y_s} + \sum_{(i,j)\in I\cap L(s)} u^*_{(i,j)}\right)\right|_{\mathbf{y}^*,\mathbf{p}^*} = 0,
\end{aligned} \tag{42}$$

where the last equality is due to (40).

Also, for any $(r,t) \in L$, we have

$$\begin{aligned}
&\left.\left(-\frac{\partial U}{\partial p_{(r,t)}} + \sum_{(i,j)\in\tilde{I}} \tilde{u}^*_{(i,j)} \frac{\partial \tilde{g}_{(i,j)}}{\partial p_{(r,t)}}\right)\right|_{\mathbf{z}^*,\mathbf{p}^*} \\
= \quad &\left.\left(\textstyle\sum_{(i,j)\in I} \tilde{u}^*_{(i,j)} \frac{1}{c_{ij}(\mathbf{p})} \frac{\partial c_{ij}}{\partial p_{(r,t)}}\right)\right|_{\mathbf{z}^*,\mathbf{p}^*} \\
= \quad &\left.\left(\textstyle\sum_{(i,j)\in I} u^*_{(i,j)} \frac{\partial c_{ij}}{\partial p_{(r,t)}}\right)\right|_{\mathbf{y}^*,\mathbf{p}^*} = 0,
\end{aligned} \tag{43}$$

where the last equality is due to (41).

Therefore we have

$$\left.\left(-\nabla U + \sum_{(i,j)\in\tilde{I}} \tilde{u}^*_{(i,j)} \nabla \tilde{g}_{(i,j)}\right)\right|_{\mathbf{z}^*,\mathbf{p}^*} = 0, \tag{44}$$

i.e., $(\mathbf{z}^*, \mathbf{p}^*)$ satisfies the KKT condition for the optimization problem in (5).

Using the same line of analysis, we can prove the converse result. This completes the proof. ■

Note that the optimization problem in (5) is a convex programming problem and hence its KKT point is globally optimal. Since a point satisfies the KKT condition of (5) if and only if the corresponding point satisfies the KKT condition of $\mathbf{Q}$ (Lemma 4), and since a point in (5) yields the same objective value as its corresponding point does in $\mathbf{Q}$ (Lemma 1), it immediately follows that the KKT point in $\mathbf{Q}$ is actually globally optimal. Therefore the dual-based algorithm converges to the globally optimal solutions.