Cross-Layer Rate Control for End-to-End Proportional Fairness in Wireless Networks with Random Access*

Xin Wang
Department of Electrical, Computer and System Engineering
Rensselaer Polytechnic Institute
Troy, NY 12180, USA
wangx5@rpi.edu

Koushik Kar
Department of Electrical, Computer and System Engineering
Rensselaer Polytechnic Institute
Troy, NY 12180, USA
koushik@ecse.rpi.edu

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 non-linear optimization. Compared to its counterpart in a wired network where link capacities are assumed to be fixed, rate control in a multi-hop random access network is much more complex and requires joint optimization at both the transport layer and the link layer. 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 primal 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 to support our conclusions.

Categories and Subject Descriptors

C.2.4 [**Computer-Communication Networks**]: Distributed Systems

General Terms

Algorithms

Keywords

Cross-layer control, ad-hoc network, random access

*This work was supported in part by NSF grant ECS-0330203.

MobiHoc'05, May 25–27, 2005, Urbana-Champaign, Illinois, USA.

1. INTRODUCTION

The objective of rate control is generally to use all available bandwidth to the full while maintaining a certain "fairness" amongst the competing sessions in the network. In wired networks, the problem of rate control has been extensively researched, e.g., [5], [6]. It has been proved 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, the results on rate control in wired networks are not readily applicable to a wireless scenario.

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* [4] 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 the 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 larger time scale at the link layer and at a smaller time scale at the transport layer. We show that the dual-based algorithm essentially adjusts the link attempt probabilities in an ascent 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 simple transformations. More importantly, the transformed convex problem can be solved in a distributed manner. The primal-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, and the algorithm achieves a faster convergence rate compared to the dual approach.

The paper is organized as follows. Related work is discussed in Section 2. In Section 3, 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 4. We then consider the transformed convex program and discuss the primal-based algorithm in detail in Section 5. The two algorithms are compared in Section 6. Simulation results are presented in Section 7, and the paper is concluded in Section 8. All necessary proofs are provided in the appendix.

2. RELATED WORK

There are several existing works that address the problem of fair bandwidth sharing at link layer. In [9] Tassiulas *et al.* have proposed a centralized algorithm to attain max-min fair rate in certain ad-hoc networks. On the other hand, Nadagopal *et al.* [7] and Ozugur *et al.* [8] have proposed decentralized algorithms that try to achieve some fair rate allocations. Kar *et al.* [13] have considered proportional fairness problem in Aloha networks, both slotted and unslotted, and derived distributed strategies to achieve proportional fairness for single-hop flows in an Aloha network. In [14] Wang *et al.* provided 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, their results cannot be readily extended to the general rate control problem for end-to-end sessions in a multi-hop wireless network.

There are several recent works that address the question of cross-layer design in communication networks, especially in wireless networks. Related work that falls into this category includes cross-layer optimization in wireless networks [10], [11], and joint power and rate control in CDMA networks [12]. In [10], Johansson *et al.* consider the problem of finding jointly optimal end-to-end communication rates, routing, power allocation and transmission scheduling for wireless networks. However the approach is based on non-linear column generation which is difficult to implement in a distributed manner. In [11], Xiao *et al.* formulate the problem of simultaneous routing and resource allocation in wireless networks and propose distributed algorithms via dual decomposition. But 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 many cases. In [12], 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 [12] (we also use a result from [12] in our analysis), the problem considered and the approaches proposed in our work differ significantly from those in [12]. Unlike [12], we are interested in optimizing the transmission attempt probabilities at the lower layer, and not the transmission powers. Moreover, we propose both primal- and dual-based approaches, which work at the one and two timescales respectively; in contrast, the approach in [12] is a dual based approach which works at a single timescale.

3. FORMULATION

3.1 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 do not assume any capture, i.e., node j can not receive any packet successfully if more than one of its neighbors are transmitting simultaneously. A transmission on edge $(i, j) \in L$ 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.

3.2 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 edge, 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 edge (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 edges. Then, the rate or the attainable throughput on link $l = (i,j)$, x_l, is

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

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.

3.3 Problem Statement

We now consider the end-to-end proportionally fair rate control problem in a multi-hop Aloha network considered above.

Let each session $s \in S$ be associated with a utility function $U_s : \Re^+ \rightarrow \Re$. Thus session s attains a utility $U_s(y_s)$ when it transmits at rate y_s that satisfies $y_s \geq 0$. Specifically, we are interested in the proportionally fair rate control problem; therefore, the utility function U_s is chosen as the logarithmic function [5]. Note that the logarithmic function is increasing and strictly concave in its argument.

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

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

The rate optimization problem can therefore be formulated as

$$\begin{array}{lll} \mathbf{P}: & \max & \sum_s \log(y_s), \\ & \text{s.t.} & \sum_{s \in S(i,j)} y_s \leq x_{(i,j)} \quad \forall (i,j) \in L, \\ & & x_{(i,j)} = c_{ij}(\mathbf{p}) \quad \forall (i,j) \in L, \\ & & 0 \leq p_{(i,j)} \leq 1 \quad \forall (i,j) \in L, \\ & & P_i \leq 1 \quad \forall i \in N, \\ & & y_s \geq 0 \quad \forall s \in S. \end{array} \quad (3)$$

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. Note that the terms $c_{ij}(\mathbf{p})$ in the second set of constraints are defined by (1). The third and fourth sets of constraints come from the fact that the attempt probabilities are non-negative and cannot be greater than unity. The fifth set of constraints ensure that the session rates are non-negative.

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.

4. DUAL-BASED ALGORITHM

4.1 Solution Approach

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

$$\begin{array}{lll} \hat{\mathbf{P}}: & \max & \sum_s \log(y_s), \\ & \text{s.t.} & \sum_{s \in S(i,j)} y_s \leq x_{(i,j)} \quad \forall (i,j) \in L, \\ & & y_s \geq 0 \quad \forall s \in S. \end{array} \quad (4)$$

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 optimum value in the parameterized problem $\hat{\mathbf{P}}$ is a function on $\mathbf{x}$, where $\mathbf{x}$ is the vector of all link rates in the network, i.e. $\mathbf{x} = \{x_{(i,j)} : (i,j) \in L\}$. We define $\hat{U}(\mathbf{x})$ as the optimum value in $\hat{\mathbf{P}}$, i.e.,

$$\hat{U}(\mathbf{x}) = \max \left\{ \sum_s \log(y_s) \,\middle|\, \sum_{s \in S(i,j)} y_s \leq x_{(i,j)}, (i,j) \in L \right\} \quad (5)$$

Since the vector of all link rates considered in $\mathbf{P}$ in turn is a function on 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 $\mathbf{P}$ can be rewritten as

$$\begin{array}{lll} \tilde{\mathbf{P}}: & \max & \tilde{U}(\mathbf{p}), \\ & \text{s.t.} & 0 \leq p_{(i,j)} \leq 1 \quad \forall (i,j) \in L, \\ & & P_i \leq 1 \quad \forall i \in N. \end{array} \quad (6)$$

We solve the problem in (6) by updating the link attempt probabilities using the following equation

$$p_{(i,j)}^{(n+1)} = p_{(i,j)}^{(n)} + \alpha \sum_{(s,t) \in L} \lambda_{(s,t)}^{*(n)} \frac{\partial c_{st}}{\partial p_{(i,j)}} (\mathbf{p}^{(n)}), \quad (7)$$

where α is the step size, $\frac{\partial c_{st}}{\partial p_{(i,j)}}$ is computed using the following formula

$$\frac{\partial c_{st}}{\partial p_{(i,j)}} = \begin{cases} (1 - P_t) \prod_{k \in K_t \setminus \{s\}} (1 - P_k) & \text{if } t = j \text{ and } s = i, \\ -p_{(s,t)} \prod_{k \in K_t \setminus \{s\}} (1 - P_k) & \text{if } t = i \text{ and } s \in I_t, \\ -p_{(s,t)}(1 - P_t) \prod_{k \in K_t \setminus \{s\}} (1 - P_k) & \text{if } t \in K_i \text{ and } s \in I_t \setminus \{i\}, \\ 0 & \text{otherwise}, \end{cases} \quad (8)$$

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

$$\boldsymbol{\lambda}^{(n)} = \arg \min_{\boldsymbol{\lambda} \geq 0} \max_{\mathbf{y}} L^{(n)}(\mathbf{y}, \boldsymbol{\lambda}). \quad (9)$$

In (9), $\boldsymbol{\lambda} = \left(\lambda_{(i,j)} : (i,j) \in L\right)$ is the vector of Lagrange multipliers for the capacity constraints on the wireless links, $\mathbf{y} = (y_s : s \in S)$ is the vector of the end-to-end session rates, and $L^{(n)}(\mathbf{y},\boldsymbol{\lambda})$ is the Lagrange function of $\hat{\mathbf{P}}$ when $\mathbf{x}=\mathbf{c}(\mathbf{p}^{(n)})$. Note that $L^{(n)}(\mathbf{y},\boldsymbol{\lambda})$ is given by

$$L^{(n)}(\mathbf{y},\boldsymbol{\lambda})=\sum_{s}\log(y_s)-\sum_{(i,j)}\lambda_{(i,j)}\left(\sum_{s\in S(i,j)}y_s-x_{(i,j)}^{(n)}\right) \tag{10}$$

We then solve $\mathbf{y}^{(n)}$ from $\hat{\mathbf{P}}$ when $\mathbf{x}=\mathbf{c}(\mathbf{p}^{(n)})$, i.e.,

$$\mathbf{y}^{(n)}=\arg\max\left\{\sum_{s}\log(y_s)\,\middle|\,\sum_{s\in S(i,j)}y_s\le c_{ij}(\mathbf{p}^{(n)}),(i,j)\in L\right\} \tag{11}$$

4.2 Convergence Analysis

We have the following theorem regarding to the convergence property of the dual-based approach.

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 with the iterative procedures stated in (7)-(11) when the step size is α. Then there exists an $\bar{\alpha}\in\Re^+$ such that for $\alpha<\bar{\alpha}$, the limit point of $\{\mathbf{p}^{(n)}(\alpha),\mathbf{y}^{(n)}(\alpha)\}$ is the globally optimal solution to the problem $\mathbf{P}$.*

Intuitively, the procedures from (7) to (10) adjust the link attempt probabilities in the gradient direction. Therefore the sequence of $\{\mathbf{p}^{(n)}(\alpha)\}$ converge to a local optimal point in $\tilde{\mathbf{P}}$ where the Karush-Kuhn-Tucker (KKT) conditions hold. Since $\mathbf{P}$ and $\tilde{\mathbf{P}}$ are equivalent, it can be shown that the KKT point of $\tilde{\mathbf{P}}$ actually gives to the KKT point in $\mathbf{P}$ if $\mathbf{y}$ is solved by (11). Therefore the procedures from (7) to (11) converge to the KKT point of $\mathbf{P}$. We further show that, although $\mathbf{P}$ appears to be non-convex, its KKT points are globally optimum.

4.3 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 $\mathbf{P}$.

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.

4.3.1 Flow Rate Control at the Transport Layer

The algorithm at the transport layer solves the rate control problem $\hat{\mathbf{P}}$. 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 algorithm stated in [6], 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 [6] 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 $\hat{\mathbf{P}}$ [2],[6]. Let $\mathbf{x}^{(n)}=\mathbf{c}(\mathbf{p}^{(n)})$ denote the link rate vector at iteration n. Then the Lagrangian in (10) at the nth iterative step can be rewritten as

$$L^{(n)}(\mathbf{y},\boldsymbol{\lambda})=\sum_{s\in S}(\log(y_s)-y_s\lambda^s)+\sum_{(i,j)\in L}\lambda_{(i,j)}x_{(i,j)}^{(n)}, \tag{12}$$

where

$$\lambda^s=\sum_{(i,j)\in L(s)}\lambda_{(i,j)}. \tag{13}$$

Note that the first term in (12) is separable in the session rates y_s. Therefore, the objective function for the dual problem of $\hat{\mathbf{P}}$ at $\mathbf{x}^{(n)}$ is

$$\begin{aligned} D^{(n)}(\boldsymbol{\lambda}) &= \max_{\mathbf{y}} L^{(n)}(\mathbf{y},\boldsymbol{\lambda}) \\ &= \sum_{s}B_s(\lambda^s)+\sum_{(i,j)}\lambda_{(i,j)}x_{(i,j)}^{(n)}, \end{aligned} \tag{14}$$

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

$$\min_{\boldsymbol{\lambda}\ge 0} D^{(n)}(\boldsymbol{\lambda}). \tag{15}$$

Since the logarithmic function is strictly concave and the constraints for rate allocations are linear, $\hat{\mathbf{P}}$ 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 $\hat{\mathbf{P}}$ achieves its optimum, denoted by $\boldsymbol{\lambda}^{*(n)}$, the corresponding y_s, denoted by $y_s^{*(n)}$, is the optimum solution to the primal problem.

Maximizing the dual function in (14) gives

$$y_s(\boldsymbol{\lambda})=\frac{1}{\lambda^s}, \tag{16}$$

where λ^s is given by (13).

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

$$\begin{aligned} \lambda_{(i,j)}^{(n+1)} &= \left[\lambda_{(i,j)}^{(n)}-\gamma\frac{\partial D^{(n)}}{\partial\lambda_{(i,j)}}(\boldsymbol{\lambda}^{(n)})\right]^+ \\ &= \left[\lambda_{(i,j)}^{(n)}+\gamma(y^{(i,j)}(\boldsymbol{\lambda}^{(n)})-x_{(i,j)}^{(n)})\right]^+, \end{aligned} \tag{17}$$

where $\gamma>0$ is the step size, $[z]^+=\max\{z,0\}$, $\boldsymbol{\lambda}^{(n)}=(\lambda_{(i,j)}^{(n)}:(i,j)\in L)$, 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:

1. For each link $(i,j)\in L$,
   (a) Compute the new price $\lambda_{(i,j)}^{(n+1)}$ using (17).
   (b) Communicate new price $\lambda_{(i,j)}^{(n+1)}$ to the sources of all sessions that use link (i,j).
2. For each session $s\in S$,
   (a) Receive from the network the sum of the prices of links on s's path, and calculate λ^s using (13).
   (b) Compute the new rate $y_s^{(n+1)}$ using (16).

(c) Communicate new rate $y_s^{(n+1)}$ to all links (i,j) on s's path.

3. Repeat Steps 1 and 2 until the session rates and link prices converge.

4.3.2 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 (7). 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 (8), note that the partial derivative $\frac{\partial c_{st}}{\partial p_{(i,j)}}$ 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.

4.3.3 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:

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

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

3. Once the iterative procedure in step 2 has converged, update the link attempt probabilities using (7) and (8).

4. Increment n by 1. Repeat steps 2 and 3 until the link attempt probabilities have converged.

5. PRIMAL-BASED ALGORITHM

5.1 Equivalent Convex Formulation

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

THEOREM 2. *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 programming problem*

$$\begin{array}{lll} \max & \sum_{s\in S} z_s, & \\ s.t. & \log\left(\sum_{s\in S(i,j)} e^{z_s}\right) - \log p_{(i,j)} - \log(1-P_j) & \\ & -\sum_{k\in K_j\setminus\{i\}} \log(1-P_k) \le 0 & \forall (i,j)\in L, \\ & 0 \le p_{(i,j)} \le 1 & \forall (i,j)\in L, \\ & P_i \le 1 & \forall i \in N. \end{array} \qquad (18)$$

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

5.2 Solution Approach

Let $\mathbf{z} = (z_s, s \in S)$, and $\mathbf{w} = (\mathbf{p}, \mathbf{z})$. We then define $\tilde{U}_s(\mathbf{w}) = z_s$ for end user $s \in S$, and $g_l(\mathbf{w}) = \log\left(\sum_{s\in S(i,j)} e^{z_s}\right) - \log p_{(i,j)} - \log(1-P_j) - \sum_{k\in K_j\setminus\{i\}} \log(1-P_k)$ for link $l = (i,j) \in L$. Let $\mathbf{W}$ represent the region in which $0 \le p_{(i,j)} \le 1$ for any link $(i,j) \in L$ and $P_i \le 1$ for any node $i \in N$. The problem in (18) can be rewritten as

$$\begin{array}{lll} \max & \sum_{s\in S} \tilde{U}_s(\mathbf{w}), & \\ \text{s.t.} & g_l(\mathbf{w}) \le 0 & l \in L, \\ & \mathbf{w} \in \mathbf{W}. & \end{array} \qquad (19)$$

Instead of solving the constrained convex programming problem in (19) directly, we consider the following optimization problem

$$\begin{array}{ll} \max & \sum_{s\in S} \tilde{U}_s(\mathbf{w}) - \kappa \sum_{l\in L} \max\{0, g_l(\mathbf{w})\}, \\ \text{s.t.} & \mathbf{w} \in \mathbf{W}, \end{array} \qquad (20)$$

where κ, the "penalty scaling factor", is a positive constant. Comparing (19) with (20), we see that the only difference is that the constraint for each link $l \in L$, $g_l(\mathbf{w})$, has been transferred to the objective function in (20). It is worth noting that the term $\kappa \max\{0, g_l(\mathbf{w})\}$ can be interpreted as the penalty associated with the violation of the capacity constraint of link l.

Let $\tilde{\mathbf{W}}^*$ be the set of optimal solutions to (20). It follows from Theorem 4.2 of [3] that, there exists $\tilde{A}$ such that the set of optimal solutions to (19) coincides with $\tilde{\mathbf{W}}^*$ for any $\kappa \ge \tilde{A}$. Therefore we can solve (20) to obtain the optimal solutions of (19).

We now present the subgradient method to solve (20) in an iterative manner. Let $p_{(i,j)}^{(n)}$ and $z_s^{(n)}$ respective denote the values of $p_{(i,j)}$ and z_s at the n iterative step, and let $\mathbf{p}^{(n)} = (p_{(i,j)}^{(n)}, (i,j) \in L)$. Let $\mathbf{x}^{(n)} = \mathbf{c}(\mathbf{p}^{(n)})$ denote the link rate vector at iteration n. For each link $(i,j) \in L$, define the "link congestion indicator" for link (i,j) at the nth iteration, $\epsilon_{(i,j)}^{(n)}$, as

$$\epsilon_{(i,j)}^{(n)} = \begin{cases} 0 & \text{if } \sum_{s\in S(i,j)} e^{z_s^{(n)}} \le x_{(i,j)}^{(n)}, \\ 1 & \text{otherwise.} \end{cases} \qquad (21)$$

Let γ_n be the step size at the nth iteration. Then z_s is updated as

$$z_s^{(n+1)} = z_s^{(n)} + \gamma_n \left(1 - \kappa \sum_{(i,j)\in L(s)} \frac{\epsilon_{(i,j)}^{(n)} e^{z_s^{(n)}}}{\sum_{r\in S(i,j)} e^{z_r^{(n)}}}\right) \qquad (22)$$

and the attempt probability on link (i,j), $p_{(i,j)}$, is updated as

$$p_{(i,j)}^{(n+1)} = p_{(i,j)}^{(n)} - \gamma_n \kappa \sum_{(s,t)\in L} \frac{\epsilon_{(s,t)}^{(n)}}{x_{(s,t)}^{(n)}} \cdot \frac{\partial c_{st}}{\partial p_{(i,j)}}(\mathbf{p}^{(n)}), \qquad (23)$$

where $\frac{\partial c_{st}}{\partial p_{(i,j)}}$ is defined by (8).

Note that since e^{z_s} $(= y_s)$ is interpreted as the rate of session s, therefore $\frac{e^{z_s}}{\sum_{r\in S(i,j)} e^{z_r}}$ in (22) can be interpreted as the fraction of the overall traffic on link (i,j) contributed by session s.

In (23), $\frac{\partial c_{st}}{\partial p_{(i,j)}}(\mathbf{p}^{(n)})$ depicts how the attempt probability on link (i,j) impacts the rate on link (s,t); from (23), note that this impact is weighted by the inverse of the rate on that link.

5.3 Convergence Analysis

We now provide the convergence analysis for the iterative procedures stated in (22) and (23). Note that the link attempt probabilities in (23) and the logarithmic value of the session rate z_s are updated using the subgradient method [3].

Denote $\mathbf{W}^*$ as the set of optimal solutions of the problem given in (18). 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 $\Phi_r(\mathbf{S})$ be the set of all points whose distance from $\mathbf{S}$ is at most r for any compact set $\mathbf{S}$, i.e. $\Phi_r(\mathbf{S}) = \{\mathbf{w} : \rho(\mathbf{w}, \mathbf{S}) \leq r\}$. From the convergence results for the subgradient method [3], we have the following theorems.

THEOREM 3. *Let $\{\mathbf{w}^{(n)}\}$ denote the sequence of vectors defined by the iterative procedure stated in (22)-(23). If the step sizes satisfy the following criteria*

$$\lim_{n\to\infty} \gamma_n = 0, \qquad \sum_{n=0}^{\infty} \gamma_n = \infty, \tag{24}$$

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

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

.

Theorem 3 states that the distance of the vector of link attempt probabilities and session rates from the set of optimal solutions decrease to zero if the step sizes satisfy the constraints in (24). If the step sizes are constant, we have slightly weaker convergent results.

THEOREM 4. *Let $\{\mathbf{w}^{(n)}(\gamma)\}$ denote the sequence of vectors defined by the iterative procedure stated in (22)-(23) with $\gamma_n = \gamma, \forall n$. Then, there exists a $\tilde{A} < \infty$ and a function $r(\gamma)$ such that $\lim_{\gamma\to 0+} r(\gamma) = 0$, and for all $\kappa > \tilde{A}$,*

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

Theorem 4 states that for a constant step size, the vector of link attempt probabilities and session rates converges to a neighborhood around the optimum, and the size of this neighborhood becomes arbitrarily small with decreasing step-size.

5.4 Distributed Algorithm

We now state formally the update procedures for the primal-based algorithm. Let $\kappa > \tilde{A}$ be a positive constant, and γ_n be the step size at the nth iteration. The primal-based algorithm is summarized as follows:

1. Set $n = 0$. For any link $(i, j) \in L$, choose initial attempt probabilities satisfying $0 < p_{(i,j)}^{(0)} < 1$. For any session s, choose an initial session rate satisfying $y_s^{(0)} > 0$, and set $z_s^{(0)} = \log y_s^{(0)}$.

2. For each link $(i, j) \in L$, compute the link congestion indicator $\epsilon_{(i,j)}^{(n)}$.

3. Update all link attempt probabilities using (23).

4. For each session $s \in S$, update z_s, the (logarithmic) utility of the session using (22). The new session rate is then obtained as $y_s^{(n+1)} = e^{z_s^{(n+1)}}$.

5. Increment n by 1. Repeat steps 2-4 until the link attempt probabilities and session rates have converged.

6. COMPARISON OF THE ALGORITHMS

The dual-based algorithm and the primal-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. It is worth noting that the convergence process at the transport layer (inner loop) can be time consuming in some cases, and this may slow down the overall convergence.

In contrast, the primal-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, the main advantage of the primal-based algorithm is that it avoids embedded loops.

7. SIMULATION INVESTIGATION

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

7.1 Simple Network Scenario

In this section, a simple network scenario is considered. The network topology is shown in Fig. 1. It is seen that there are three nodes, A, B, and C, and two links, 0 and 1, in this scenario. Three end-to-end sessions, namely, f_0, f_1,

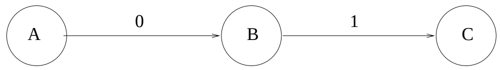

Figure 1: A simple network.

and f_2 are setup in this scenario. The source, the sink, and the path for each session are shown in Table 1.

Table 1: The Source, Sink, and Path of the Sessions.

Session	Source Node	Sink Node	Links on the Path
f_0	A	B	0
f_1	B	C	1
f_2	A	C	0, 1

To investigate the performance of the proposed algorithm, we compare the globally optimum solutions solved by Matlab, the solutions given by the dual-based algorithm, and the solutions given by the primal-based algorithm. The results are presented in Table 2. The comparisons show that both algorithms converge to the optimum accurately.

Table 2: The Optimum Results and the Solutions Given by the Distributed Algorithms.

Variables	p_0	p_1	x_0	x_1
optimum solutions	1	0.5	0.5	0.5
dual-based algorithm	1	0.5004	0.4996	0.4996
primal-based algorithm	1	0.4999	0.4999	0.5001
Variables	y_0	y_1	y_2	U^*
optimum solutions	0.3333	0.3333	0.1667	-3.9890
dual-based algorithm	0.3337	0.3329	0.1667	-3.9889
primal-based algorithm	0.3139	0.3139	0.1860	-3.9889

7.1.1 Dual-Based Algorithm

In Fig. 4, we demonstrate the convergence of the link attempt probabilities, link capacities, session rates and aggregate utility, as a function of the number of link layer updates (outer loop iterations). The step size for attempt probability adjustment at the link layer is set to 5×10^{-4}. It can be seen from the plots that after 60 iterations at the link layer, all the variables are well within 5% of their optimal values.

7.1.2 Primal-Based Algorithm

In this simulation, the step size is set to 1.5×10^{-6}. Fig. 5, which shows the attempt probabilities, link rates, session rates, and the aggregate utility in 500 iterations, demonstrates the convergence of the primal algorithm in the simple scenario considered.

7.2 Ad-hoc Network Scenario

A median-sized ad-hoc network scenario is considered in this section. The network is composed of 6 nodes and 8 links. Its topologies is shown in Fig. 2.

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 3.

The globally optimum solutions given by Matlab, the solutions given by the dual-based algorithm and the solutions given by the primal-based algorithm are presented in Table 4. Through comparison, it can be seen that, in this ad-hoc network, both algorithms achieve the globally optimum solutions.

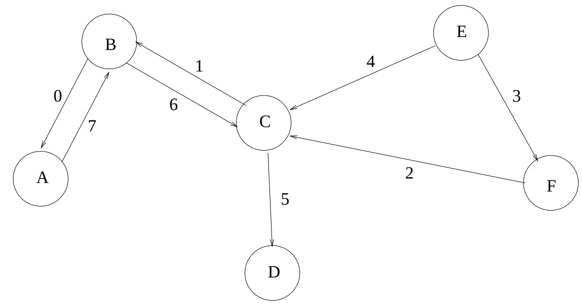

Figure 2: An ad-hoc network.

Table 3: The Source, Sink, and Path of the Flows.

Flow	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

Table 4: The Optimum Results and the Solutions Given by the Distributed Algorithms.

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

7.2.1 Dual-Based Algorithm

In this simulation, the step size for the attempt probability adjustment at the link layer is still set to 5×10^{-4}. Fig. 6 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. From the plots it can be seen that all the variables are within 10% of their globally optimum values after 300 iterations at the link layer.

Recall that the dual-based algorithm is implemented at the link layer and the transport layer 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 (link layer and the transport layer combined) versus the number of iterations at the link layer. In the simulation, the algorithm at the transport layer terminates when the change in the link price is less than 1 and the change in the session rates is less than 10^{-3}.

From the figure, 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 median sized network should range from a few seconds to a few minutes.

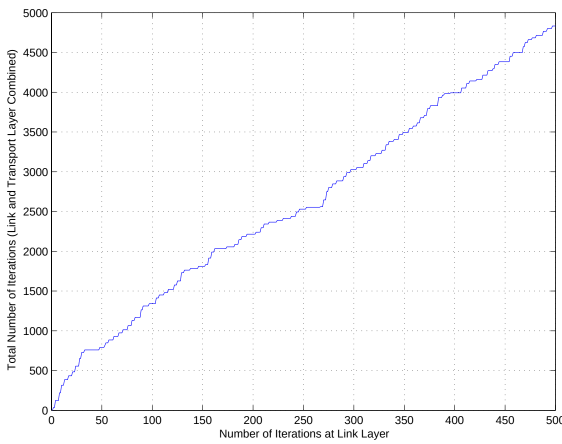

Figure 3: The total number of iterations vs. the number of iterations at the link layer.

7.2.2 Primal-Based Algorithm

In this simulation, we still set the step size to 1.5×10^{-6} as we do in simulating the simple network scenario. Fig. 7 show how the link attempt probabilities, link rates, session rates, and the aggregate utility converge when the primal-based algorithm is used. It can be seen from the plots that after about 2000 iterations, all the variables are within 10% of their globally optimum 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. 7 that there is an obvious thickening of the computed link attempt probabilities and link capacities, meaning that the computed values do not exactly converge to the optimal values, but fluctuates around them. Recall that in Section 5.3 we have argued 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 link the one observed in the plots of link attempt probabilities and link capacities in Fig. 7. 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 flow 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.

8. CONCLUSIONS

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 much more difficult than its wired network counterpart.

In this paper, 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.

9. REFERENCES

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

[5] 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.

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

[7] 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.

[8] 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.

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

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

[11] L. Xiao, M. Johnasson, S. Boyd, "Simultaneous Routing and Resource Allocation via Dual Decomposition", *IEEE Transactions on*

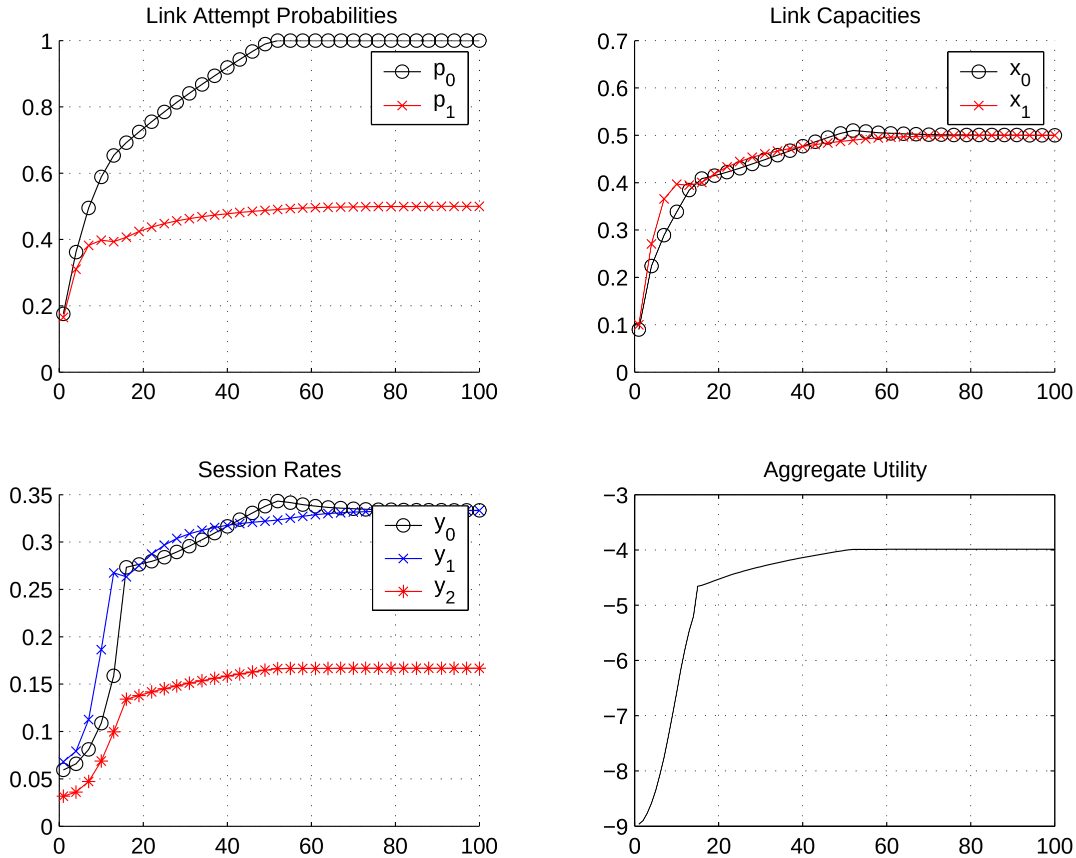

Figure 4: The link attempt probabilities, link rates, session rates, and the aggregate utility when the dual-based algorithm is used. (The x axis denotes the number of iterations at the link layer.)

Communications, Vol. 52, No. 7, pages 1136-1144, July 2004.

[12] 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.

[13] 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.

[14] 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

Since proof of Theorem 1 uses Theorem 2, we will prove Theorem 2 first.

A. PROOF OF THEOREM 2

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

$$\begin{aligned} &\log\left(\textstyle\sum_{s \in S(i,j)} e^{z_s}\right) - \log(p_{(i,j)}) \\ &- \log(1-P_j) - \textstyle\sum_{k \in K_j \setminus \{i\}} (1-P_k) \le 0 \end{aligned} \quad (25)$$

It is worth noting that $\log\left(\sum_{s \in S(i,j)} e^{z_s}\right)$ is a convex function for z_s (see the proof in [12]), and $\log p_{(i,j)}$ is concave in $p_{(i,j)}$. Also note that $\log(1-P_k)$ are concave functions of $\mathbf{p}$. It then follows that the set of constraints in (25) is convex.

Therefore the problem in (18), which is equivalent to (3), is a convex programming problem. □

B. PROOF OUTLINE OF THEOREM 1

For simplicity of exposition, we assume that the capacity constraints in $\hat{\mathbf{P}}$ are not degenerate for any $\mathbf{x}$, i.e., if we define the $|L| \times |S|$ routing matrix $\mathbf{R}$ as

$$R(l,k) = \begin{cases} 1 & \text{if } k \in S(l), \\ 0 & \text{otherwise,} \end{cases} \quad (26)$$

and $|L| > |S|$, then any $|S|$ columns of $\mathbf{R}$ are linearly independent.

Denote the optimal Lagrange multipliers as $\boldsymbol{\lambda}^*$ when $\hat{\mathbf{P}}$ is parameterized at $\bar{\mathbf{x}}$. According to the *Sensitivity Theorem*[2], we obtain that

$$\nabla_{\mathbf{x}} \hat{U}(\bar{\mathbf{x}}) = \boldsymbol{\lambda}^* \quad (27)$$

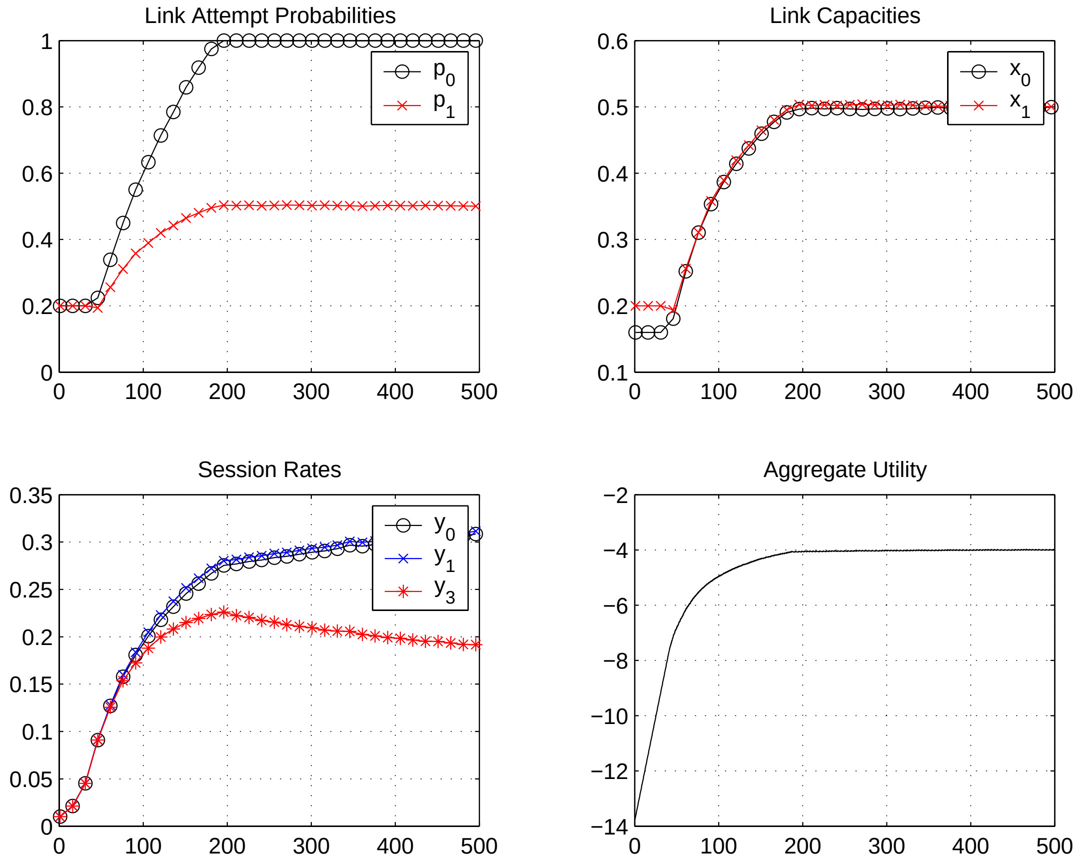

Figure 5: The link attempt probabilities, link rates, session rates, and the aggregate utility when the primal-based algorithm is used. (The x axis denotes the number of iterations at the link/transport layer.)

and there exists an open sphere $\mathfrak{B}$ centered at $\bar{\mathbf{x}}$ such that for every $\mathbf{x} \in \mathfrak{B}$ there is an $\boldsymbol{\lambda}(\mathbf{x})$ which are the associated Lagrange multipliers. Furthermore, $\boldsymbol{\lambda}(\cdot)$ is continuous differential in $\mathfrak{B}$ and $\boldsymbol{\lambda}(\bar{x}) = \boldsymbol{\lambda}^*$.

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 1. *If we define*

$$\bar{d}_{ij} = \sum_{(s,t) \in L} \lambda^*_{(s,t)} \frac{\partial c_{st}}{\partial p_{(i,j)}}(\bar{\mathbf{p}}) \tag{28}$$

where $\boldsymbol{\lambda}^*$ is the vector of Lagrange multipliers for $\hat{\mathbf{P}}$ at $\mathbf{x} = \mathbf{c}(\bar{\mathbf{p}})$, and denote $\bar{\mathbf{d}} = (\bar{d}_{ij} : (i,j) \in L)$, 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 holds true here, and therefore there exists an $\bar{\alpha} \in \mathfrak{R}^+$ such that for $\alpha < \bar{\alpha}$, the sequence $\{\mathbf{p}^{(n)}(\alpha)\}$, which is generated using the procedures stated in (7) - (10), converges to a local optimum point of $\tilde{\mathbf{P}}$.

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

Lemma 2. *Denote $\mathbf{p}^*$ as the local optimum point for $\tilde{\mathbf{P}}$, and $\tilde{\mathbf{y}}^*$ is solved from (11), i.e.,*

$$\mathbf{y}^* = \arg\max \left\{ \sum_s \log(y_s) \,\middle|\, \sum_{s \in S(i,j)} y_s \le c_{ij}(\mathbf{p}^*), (i,j) \in L \right\} \tag{29}$$

then $(\mathbf{p}^*, \tilde{\mathbf{y}}^*)$ is the local optimum point for $\mathbf{P}$.

From Lemma 2 we conclude that, the stationary point $\mathbf{p}^*$ of the sequence $\{\mathbf{p}^{(n)}(\alpha)\}$, which is generated using the procedures stated in (7) to (10), and the corresponding $\mathbf{y}^*$, which is calculated using (11), constitute a local optimum for $\mathbf{P}$.

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

Lemma 3. *If y^*_s, $s = 1, ..., S$ and $p^*_{(i,j)}$, $(i,j) \in L$ satisfy the first order necessary condition for optimality for the nonlinear program in (3), then $z^*_s = \log(y^*_s)$, $s = 1, ..., S$ and $p^*_{(i,j)}$, $(i,j) \in L$ will satisfy the first order necessary condition for optimality for the convex program in (18). Conversely, if z^*_s, $s = 1, ..., S$ and $p^*_{(i,j)}$, $(i,j) \in L$ satisfy the*

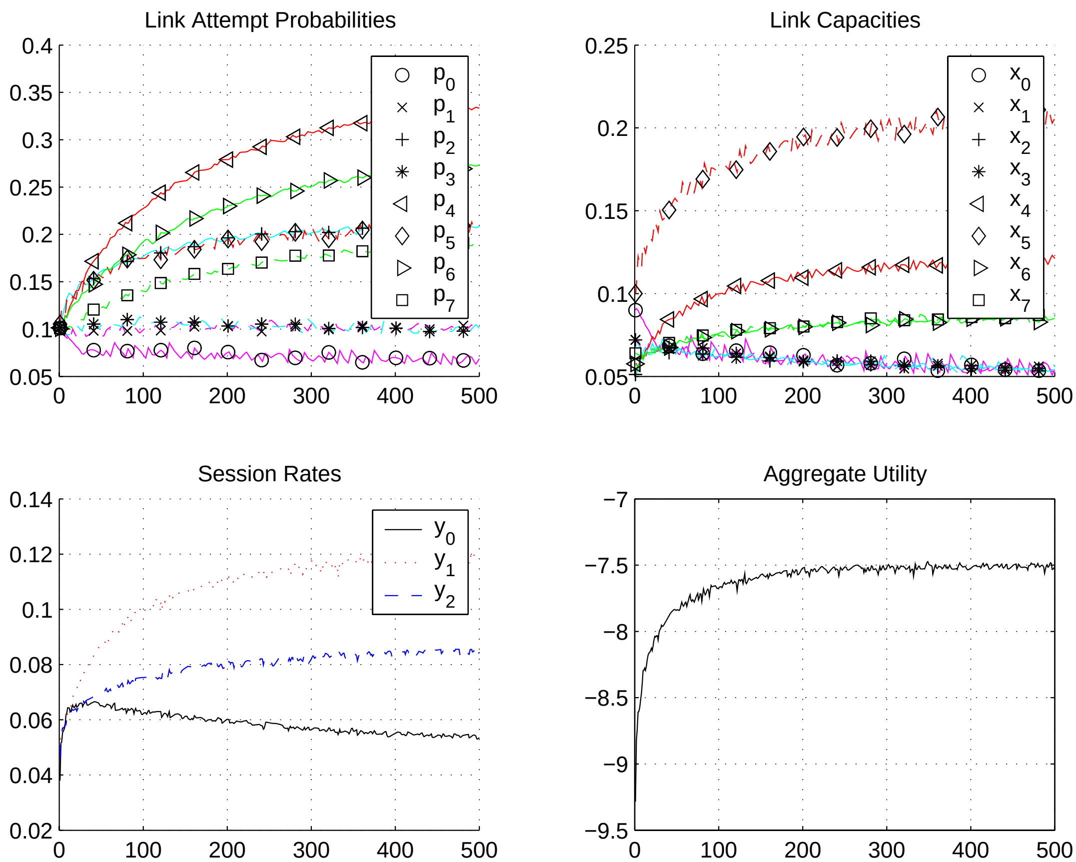

Figure 6: The link attempt probabilities, link rates, session rates, and the aggregate utility when the dual-based algorithm is used. (The x axis denotes the number of iterations at the link layer.)

first order necessary condition for optimality for the nonlinear program in (18), then $y_s^* = e^{z_s^*}$, $s = 1, ..., S$ *and* $p^*_{(i,j)}$, $(i,j) \in L$ *will satisfy the first order necessary condition for optimality for the convex program in (3).*

PROOF. 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, then there exists $u^*_{(i,j)} \geq 0$ for $(i,j) \in I$ such that

$$\left(-\nabla U + \sum_{(i,j) \in I} u^*_{(i,j)} \nabla g_{(i,j)} \right) \Bigg|_{\mathbf{y}^*, \mathbf{p}^*} = 0 \qquad (30)$$

where

$$\begin{aligned} g_{(i,j)} &= \textstyle\sum_{s \in S(i,j)} y_s - x_{(i,j)} \\ &= \textstyle\sum_{s \in S(i,j)} y_s - p_{(i,j)}(1 - P_j) \prod_{k \in K_j \setminus \{i\}} (1 - P_k) \end{aligned}$$

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

Therefore, for y_s, we have

$$\begin{aligned} & \left(-\frac{\partial U}{\partial y_s} + \sum_{(i,j) \in I} u^*_{(i,j)} \frac{\partial g_{(i,j)}}{\partial y_s} \right) \Bigg|_{\mathbf{y}^*, \mathbf{p}^*} \\ = & \left(-\frac{1}{y_s} + \sum_{(i,j) \in I \cap L(s)} u^*_{(i,j)} \right) \Bigg|_{\mathbf{y}^*, \mathbf{p}^*} = 0 \end{aligned} \qquad (31)$$

and for $p_{(s,t)}$, we have

$$\begin{aligned} & \left(-\tfrac{\partial U}{\partial p_{(i,j)}} + \textstyle\sum_{(i,j) \in I} u^*_{(i,j)} \tfrac{\partial g_{(i,j)}}{\partial p_{(s,t)}} \right) \Bigg|_{\mathbf{y}^*, \mathbf{p}^*} \\ = & \textstyle\sum_{(i,j) \in I} u^*_{(i,j)} \tfrac{\partial x_{(i,j)}}{\partial p_{(s,t)}} \Bigg|_{\mathbf{y}^*, \mathbf{p}^*} = 0 \end{aligned} \qquad (32)$$

For the nonlinear program in (18), denote

$$\begin{aligned} \tilde{g}_{(i,j)} &= \log \left(\textstyle\sum_{s \in S(i,j)} e^{z_s} \right) - \log x_{(i,j)} \\ &= \log \left(\textstyle\sum_{s \in S(i,j)} e^{z_s} \right) - \log p_{(i,j)} - \log(1 - P_j) \\ &\quad - \textstyle\sum_{k \in K_j \setminus \{i\}} (1 - P_k) \end{aligned}$$

and denote $\tilde{I} = \{(i,j) : \tilde{g}_{(i,j)}|_{\mathbf{y}^*, \mathbf{p}^*} = 0,\ (i,j) \in L\}$. Obviously $I = \tilde{I}$. we take $\tilde{u}^*_{(i,j)} = u^*_{(i,j)} c_{ij}(\mathbf{p}^*)$ for all (i,j) in $\tilde{I}$. Note that $\tilde{I} = I$ and hence $g_{(i,j)}|_{\mathbf{y}^*, \mathbf{p}^*} = 0$ for any (i,j) in $\tilde{I}$. It then follows

$$c_{ij}(\mathbf{p}^*) = p^*_{(i,j)}(1 - P_j^*) \prod_{k \in K_j \setminus \{i\}} (1 - P_k^*) = \sum_{l \in S(i,j)} e^{z_l^*} \qquad (33)$$

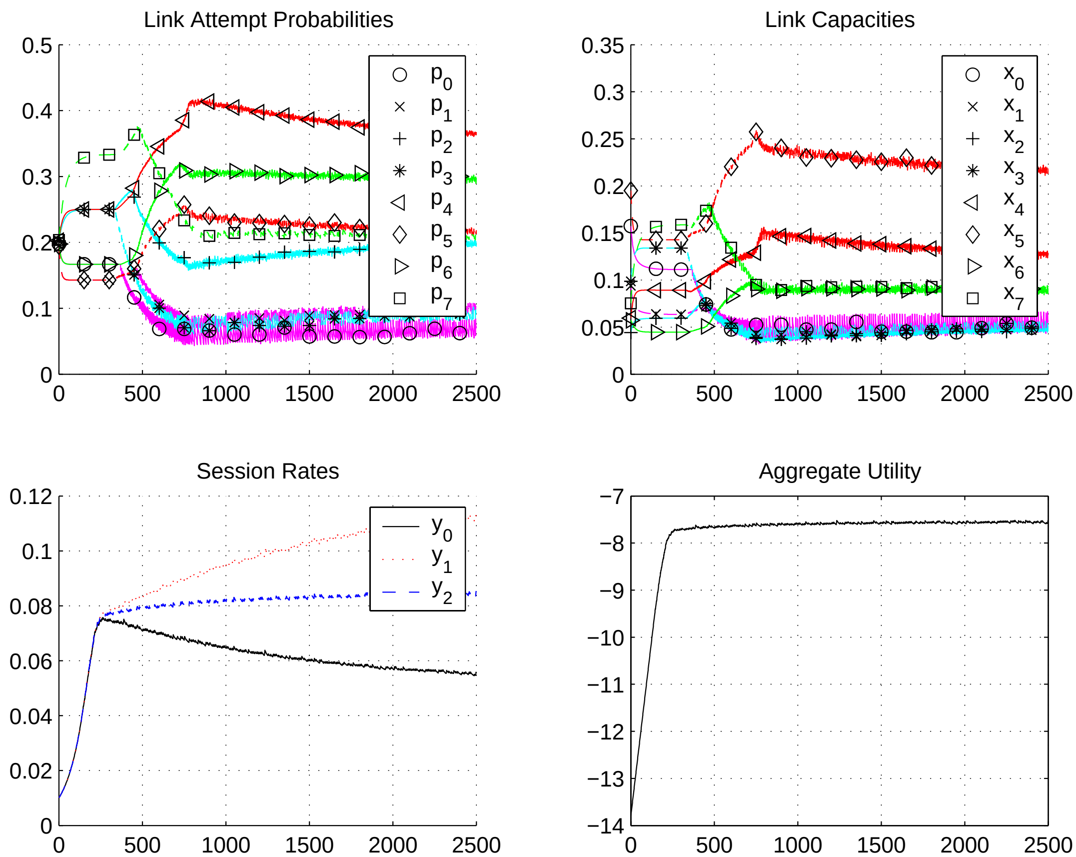

Figure 7: The link attempt probabilities, link rates, session rates, and the aggregate utility when the primal-based algorithm is used. (The x axis denotes the number of iterations at the link/transport layer.)

For z_s, we have

$$\begin{aligned} & \left(-\frac{\partial U}{\partial z_s} + \sum_{(i,j)\in\tilde{I}} \tilde{u}^*_{(i,j)} \frac{\partial \tilde{g}_{(i,j)}}{\partial z_s}\right)\Bigg|_{\mathbf{y}^*,\mathbf{p}^*} \\ = & \left(-1 + \textstyle\sum_{(i,j)\in\tilde{I}\cap L(s)} \tilde{u}^*_{(i,j)} \dfrac{e^{z_s}}{\sum_{l\in S(i,j)} e^{z_l}}\right)\Bigg|_{\mathbf{y}^*,\mathbf{p}^*} \\ = & \left(-1 + \textstyle\sum_{(i,j)\in I\cap L(s)} u^*_{(i,j)} e^{z_s}\right)\Big|_{\mathbf{y}^*,\mathbf{p}^*} \\ = & \left(-1 + \textstyle\sum_{(i,j)\in I\cap L(s)} u^*_{(i,j)} y_s\right)\Big|_{\mathbf{y}^*,\mathbf{p}^*} \\ = & \; y_s \left(-\frac{1}{y_s} + \sum_{(i,j)\in I\cap L(s)} u^*_{(i,j)}\right)\Bigg|_{\mathbf{y}^*,\mathbf{p}^*} = 0 \end{aligned} \tag{34}$$

For $p_{(s,t)}$, we have

$$\begin{aligned} & \left(-\frac{\partial U}{\partial p_{(i,j)}} + \sum_{(i,j)\in I} \tilde{u}^*_{(i,j)} \frac{\partial g_{(i,j)}}{\partial p_{(s,t)}}\right)\Bigg|_{\mathbf{y}^*,\mathbf{p}^*} \\ = & \left(\textstyle\sum_{(i,j)\in I} \tilde{u}^*_{(i,j)} \dfrac{1}{c_{ij}(\mathbf{p})} \dfrac{\partial c_{ij}}{\partial p_{(s,t)}}\right)\Bigg|_{\mathbf{y}^*,\mathbf{p}^*} \\ = & \left(\textstyle\sum_{(i,j)\in I} u^*_{(i,j)} \dfrac{\partial c_{ij}}{\partial p_{(s,t)}}\right)\Bigg|_{\mathbf{y}^*,\mathbf{p}^*} = 0 \end{aligned} \tag{35}$$

Therefore we have

$$\left(-\nabla U + \sum_{(i,j)\in\tilde{I}} \tilde{u}^*_{(i,j)} \nabla \tilde{g}_{(i,j)}\right)\Bigg|_{\mathbf{y}^*,\mathbf{p}^*} = 0 \tag{36}$$

i.e. $\mathbf{z}^*$ and $\mathbf{p}^*$ satisfy the KKT condition for the optimization problem in (18).

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

Note that the optimization problem in (18) is a convex programming problem and hence its KKT point is globally optimum. Since a point satisfies the KKT condition of (18) if and only if the corresponding point satisfies the KKT condition of $\mathbf{P}$ (Lemma 3), and since a point in (18) yields the same objective value as its corresponding point does in $\mathbf{P}$ (Theorem 2), it immediately follows that the KKT point in $\mathbf{P}$ is actually globally optimum. Therefore the dual-based algorithm converges to the globally optimum solutions.