

# Optimal Resource Allocation in Wireless Ad Hoc Networks: A Price-Based Approach

Yuan Xue, *Member, IEEE*, Baochun Li, *Senior Member, IEEE*, and Klara Nahrstedt, *Member, IEEE*

**Abstract**—The shared-medium multihop nature of wireless ad hoc networks poses fundamental challenges to the design of effective resource allocation algorithms that are *optimal* with respect to resource utilization and *fair* across different network flows. None of the existing resource allocation algorithms in wireless ad hoc networks have realistically considered end-to-end flows spanning multiple hops. Moreover, strategies proposed in wireline networks are not applicable in the context of wireless ad hoc networks, due to their unique characteristics of location-dependent contention. In this paper, we propose a new price-based resource allocation framework in wireless ad hoc networks to achieve *optimal* resource utilization and *fairness* among competing end-to-end flows. We build our pricing framework on the notion of *maximal cliques* in wireless ad hoc networks, as compared to individual links in traditional wide-area wireline networks. Based on such a price-based theoretical framework, we present a two-tier iterative algorithm. Distributed across wireless nodes, the algorithm converges to a global network optimum with respect to resource allocations. We further improve the algorithm toward asynchronous network settings and prove its convergence. Extensive simulations under a variety of network environments have been conducted to validate our theoretical claims.

**Index Terms**—Wireless communication, algorithm/protocol design and analysis, nonlinear programming.

## 1 INTRODUCTION

A wireless ad hoc network consists of a collection of wireless nodes without a fixed infrastructure. Each node in the network forwards packets for its peer nodes and each *end-to-end flow* traverses multiple hops of wireless links from a source to a destination. Compared with wireline networks, where flows only contend at the router that performs flow scheduling (contention in the time domain), the unique characteristics of multihop wireless networks show that flows also compete for shared channel if they are within the interference ranges of each other (contention in the spatial domain). This presents the problem of designing a topology-aware resource allocation algorithm that is both *optimal* with respect to resource utilization and *fair* across contending multihop flows.

In previous work, fair packet scheduling mechanisms have been proposed [1], [2], [3] and shown to perform effectively in providing fair shares among single-hop flows in wireless ad hoc networks and in balancing the trade-off between fairness and resource utilization. However, none of the previously proposed algorithms has considered end-to-end flows spanning multiple hops, which reflect the reality in wireless ad hoc networks. While these mechanisms may be sufficient for maintaining basic fairness properties among localized flows, they do not coordinate intraflow resource allocations between upstream and downstream hops of an end-to-end flow and, thus, will not be able to achieve global optimum with respect to resource utilization and fairness.

Due to the complexities of such intraflow coordinations, we are naturally led to a *price-based* strategy, where prices are computed as *signals* to reflect relations between resource demands and supplies and are used to coordinate the resource allocations at multiple hops. Previous research in wireline network pricing (e.g., [4], [5], [6]) has shown that pricing is effective as a means to arbitrate resource allocation. In these research results, a *shadow price* is associated with a wireline link to reflect relations between the traffic load of the link and its bandwidth capacity. A *utility* is associated with an end-to-end flow to reflect its resource requirement. Transmission rates are chosen to respond to the aggregated price signals along end-to-end flows such that the net benefits (the difference between utility and cost) of flows are maximized. It has been shown that [4], [5], at equilibrium, such a price-based strategy of resource allocation may achieve global optimum where resource is optimally utilized. Moreover, by choosing appropriate utilities, various fairness models can be achieved.

Unfortunately, there exist fundamental differences between multihop wireless ad hoc networks and traditional wireline networks, preventing verbatim applications of the existing pricing theories. In multihop wireless networks, flows that traverse the same geographical *vicinity* contend for the same wireless channel capacity. This is in sharp contrast with wireline networks, where only flows that traverse the same link contend for its capacity. When it comes to pricing, we may conveniently associate shadow prices with individual links in wireline networks to reflect their resource demand and supply. This is not feasible in wireless networks with the presence of location dependent contention. Due to the decentralized and self-organizing nature of ad hoc networks, the quest for a fully distributed and adaptive algorithm further exacerbates the problem.

In this paper, we address these unique characteristics of wireless ad hoc networks and follow a price-based strategy to allocate channel bandwidth to competing multihop flows. The fundamental question we seek to answer is: How much bandwidth should we allocate to each of the end-to-end

- Y. Xue is with the Department of Electrical Engineering and Computer Science, Vanderbilt University, VU Station B 351824, Nashville, TN 37235. E-mail: yuan.xue@vanderbilt.edu.
- K. Nahrstedt is with the Department of Computer Science, University of Illinois at Urbana-Champaign, 3101 Siebel Center for Computer Science, 201 N. Goodwin Avenue, Urbana, IL 61801. E-mail: klara@cs.uiuc.edu.
- B. Li is with the Department of Electrical and Computer Engineering, University of Toronto, 10 King's College Road, Toronto, Ontario M5S 3G4 Canada. E-mail: bli@eecg.toronto.edu.

Manuscript received 2 Oct. 2003; revised 5 June 2004; accepted 24 Nov. 2004; published online 16 Feb. 2006.

flows so that scarce resources in a wireless network may be optimally and fairly utilized? Toward this goal, our original contributions are two-fold. First, we build a pricing framework specifically tailored to the contention model of wireless networks and establish shadow prices based on the notion of *maximal cliques* in *wireless link contention graphs*, rather than individual links, as in wireline networks. In such a price-based theoretical framework, the price of an end-to-end multihop flow is the aggregate of prices of all its subflows, while the price of each of the subflows is the sum of shadow prices of *all* maximal cliques that it belongs to. With our new pricing framework, by choosing the appropriate utility functions, the optimality of resource allocations—in terms of both fairness and utilization—may be achieved by maximizing the aggregated utility across all flows. Second, we present a two-tier distributed algorithm to compute the bandwidth allocation for each of the end-to-end flows based on our price-based theoretical framework. The first tier of the algorithm constitutes an iterative algorithm that determines per-clique shadow prices and end-to-end flow resource allocations. We show that this algorithm converges to the unique equilibrium where the aggregated utility is maximized. The second tier of the algorithm constructs the maximal cliques in a distributed manner. To facilitate its deployment in practical network environments, the algorithm is further improved to accommodate asynchronous communications. We have performed extensive simulations under a variety of network settings and showed that our solution is practical for multihop wireless networks.

The remainder of this paper is organized as follows: We first present our price-based theoretical framework in wireless ad hoc networks (Section 2 and Section 3). We then proceed to design a two-tier decentralized algorithm in Section 4, which is further refined to accommodate asynchrony in Section 5. Finally, we evaluate the performance of our algorithm in a simulation-based study (Section 6), discuss related work (Section 7), and conclude the paper (Section 8).

## 2 RESOURCE CONSTRAINTS IN WIRELESS AD HOC NETWORKS

In this paper, we consider a wireless ad hoc network that consists of a set of nodes  $V$ . Each node  $i \in V$  has a transmission range  $d_{tx}$  and an interference range  $d_{int}$ , which can be larger than  $d_{tx}$ . Packet transmission in such a network is subject to location-dependent contention. There exist two models for packet transmission in wireless networks in the literature [7], generally referred to as the *protocol model* and the *physical model*. In the case of a single wireless channel, these two models are presented as follows:

1. *The Protocol Model*. In the protocol model, the transmission from node  $i$  to  $j$ , ( $i, j \in V$ ) is successful if 1) the distance between these two nodes  $d_{ij}$  satisfies  $d_{ij} < d_{tx}$ ; 2) any node  $k \in V$ , which is within the interference range of the receiving node  $j$ ,  $d_{kj} \leq d_{int}$ , is not transmitting. This model can be further refined toward the case of IEEE 802.11-style MAC protocols, where the sending node  $i$  is also required to be free of interference as it needs to receive the link layer acknowledgment from the receiving node  $j$ . Specifically, any node  $k \in V$ , which is within the interference range of the nodes  $i$  or  $j$  (i.e.,  $d_{ki} \leq d_{int}$  or  $d_{kj} \leq d_{int}$ ), is not transmitting.

2. *The Physical Model*. This model is directly related to the physical layer characteristics. The transmission from node  $i$  to  $j$  is successful if the signal-to-noise ratio at the node  $j$ ,  $SNR_{ij}$ , is not smaller than a minimum threshold:  $SNR_{ij} \geq SNR_{thresh}$ .

In this paper, we focus our attention on solving problems of resource allocation based on the protocol model, with particular interest in IEEE 802.11-style MAC protocols due to their popular deployment in realistic wireless systems. The problems of resource allocation under the physical model is beyond the scope of this paper and left as a future research direction. Under the protocol model, a wireless ad hoc network can be regarded as a *bidirectional* graph  $G = (V, E)$ .  $E \subseteq 2^V$  denotes the set of *wireless links* which are formed by nodes that are within the transmission range of each other. A wireless link  $e \in E$  is represented by its end nodes  $i$  and  $j$ , i.e.,  $e = \{i, j\}$ .

In such a network, there exists a set of *end-to-end flows*, denoted as  $F$ . Each flow  $f \in F$  goes through multiple hops in the network, passing a set of wireless links  $E(f)$ . A single-hop data transmission in the flow  $f$  along a particular wireless link is referred to as a *subflow* of  $f$ . Obviously, there may exist multiple subflows along the same wireless link. We use the notation  $S(S \subseteq E)$  to represent a set of wireless links in  $G$  such that each of the wireless links in  $S$  carries at least one subflow, i.e., the wireless link is not idle.

Based on the protocol model, flows in a wireless ad hoc network contend for shared resources in a location-dependent manner: Two subflows contend with each other if either the source or destination of one subflow is within the interference range ( $d_{int}$ ) of the source or destination of the other. Among a set of mutually contending subflows, only one of them may transmit at any given time. Thus, the aggregated rate of all subflows in such a set may not exceed the channel capacity. Formally, we consider a *wireless link contention graph* [3]  $G_c = (V_c, E_c)$ , in which vertices correspond to the wireless links (i.e.,  $V_c = S$ ), and there exists an edge between two vertices if the subflows along these two wireless links contend with each other.

In a graph, a complete subgraph is referred to as a *clique*. A *maximal clique* is defined as a clique that is not contained in any other cliques. In a wireless link contention graph, the vertices in a maximal clique represent a maximal set of mutually contending wireless links, along which at most one subflow may transmit at any given time.

We proceed to consider the problem of allocating rates to wireless links. We claim that a *rate allocation*  $\mathbf{y} = (y_s, s \in S)$  is *feasible* if there exists a collision-free transmission schedule that allocates  $y_s$  to  $s$ . Formally, if a rate allocation  $\mathbf{y} = (y_s, s \in S)$  is feasible, then the following condition is satisfied [2]:

$$\forall q \in Q, \sum_{s \in V(q)} y_s \leq C, \quad (1)$$

where  $Q$  is the set of all maximal cliques in  $G_c$  and  $C$  is the channel capacity. For a clique  $q$  in the wireless link contention graph  $G_c$ ,  $V(q) \subseteq S$  is the set of its vertices.

Equation (1) gives an upper bound on the rate allocations to the wireless links. In practice, however, such a bound may not be tight, especially with carrier-sensing-multiple-access-based wireless networks (such as IEEE 802.11). In this case, we introduce  $C_q$ , the *achievable* channel capacity at

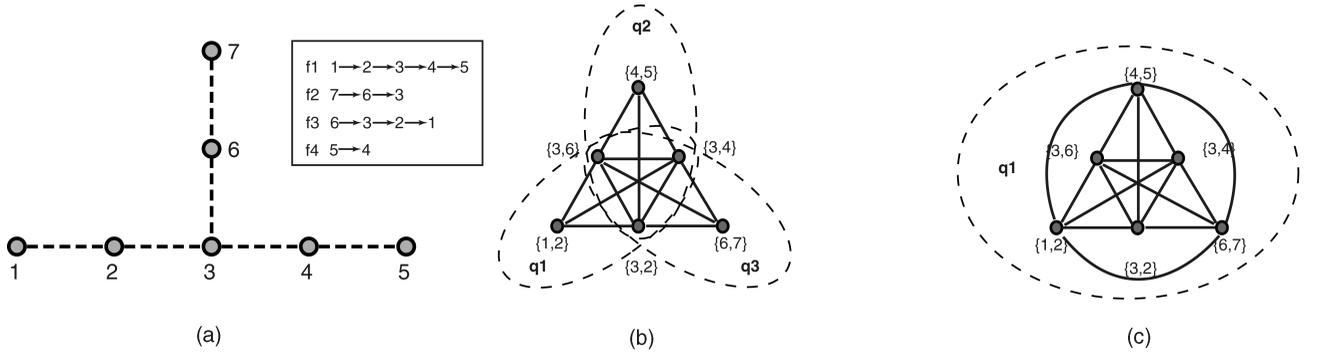


Fig. 1. Resource constraints in wireless ad hoc networks: an example. (a) Network topology. (b) Wireless link contention graph ( $d_{tx} = d_{int}$ ). (c) Wireless link contention graph ( $2d_{tx} = d_{int}$ ).

a clique  $q$ . More formally, if  $\sum_{s \in V(q)} y_s \leq C_q$ , then  $\mathbf{y} = (y_s, s \in S)$  is feasible. To this end, we observe that each maximal clique may be regarded as an independent channel resource unit with capacity  $C_q$ . It motivates the use of a *maximal clique* as a basic resource unit for pricing in wireless ad hoc networks, as compared to the notion of a *link* in wireline networks.

We now proceed to consider resource constraints on rate allocations among flows. To facilitate discussions, we define a clique-flow matrix  $\mathbf{R} = \{R_{qf}\}$ , where  $R_{qf} = |V(q) \cap E(f)|$  represents the number of subflows that flow  $f$  has in the clique  $q$ . If we treat a maximal clique as an independent resource, then the clique-flow matrix  $\mathbf{R}$  represents the “resource usage pattern” of each flow. Let the vector  $\mathbf{C} = (C_q, q \in Q)$  be the vector of achievable channel capacities in each of the cliques. In a wireless ad hoc network  $G = (V, E)$  with a set of flows  $F$ , there exists a feasible rate allocation  $\mathbf{x} = (x_f, f \in F)$ , if and only if  $\mathbf{R}\mathbf{x} \leq \mathbf{C}$ . This observation gives the constraints with respect to rate allocations to end-to-end flows in wireless ad hoc networks.

We present an example to illustrate the concepts and notations defined so far. Fig. 1a shows the topology of the network, as well as its ongoing flows. The corresponding wireless link contention graph is shown in Fig. 1b, where the interference range is the same as transmission range ( $d_{int} = d_{tx}$ ), and in Fig. 1c, where the interference range is twice as large as the transmission range ( $d_{int} = 2 \cdot d_{tx}$ ). In this example, there are four end-to-end flows

$$\begin{aligned} f_1 &= \{\{1, 2\}, \{2, 3\}, \{3, 4\}, \{4, 5\}\}, \\ f_2 &= \{\{7, 6\}, \{6, 3\}\}, \\ f_3 &= \{\{6, 3\}, \{3, 2\}, \{2, 1\}\}, \text{ and} \\ f_4 &= \{\{5, 4\}\}. \end{aligned}$$

As such, in Fig. 1b, there are three maximal cliques in the contention graph:

$$\begin{aligned} q_1 &= \{\{1, 2\}, \{3, 2\}, \{3, 4\}, \{3, 6\}\}, \\ q_2 &= \{\{3, 2\}, \{3, 4\}, \{4, 5\}, \{3, 6\}\}, \text{ and} \\ q_3 &= \{\{3, 2\}, \{3, 4\}, \{3, 6\}, \{6, 7\}\}; \end{aligned}$$

in Fig. 1c, where  $d_{int} = 2 \cdot d_{tx}$ , there is only one maximal clique

$$q_1 = \{\{1, 2\}, \{3, 2\}, \{3, 4\}, \{3, 6\}, \{4, 5\}, \{6, 7\}\}.$$

We use  $y_{ij}$  to denote the aggregated rate of *all* subflows along wireless link  $\{i, j\}$ . For example,  $y_{12} = x_1 + x_3$ ,  $y_{36} = x_2 + x_3$ . In each clique, the aggregated rate may not

exceed the corresponding channel capacity. In particular, when  $d_{int} = d_{tx}$ ,

$$\begin{aligned} y_{12} + y_{32} + y_{34} + y_{36} &\leq C_1, \\ y_{32} + y_{34} + y_{45} + y_{36} &\leq C_2, \text{ and} \\ y_{32} + y_{34} + y_{36} + y_{67} &\leq C_3. \end{aligned}$$

When  $d_{int} = 2 \cdot d_{tx}$ ,  $y_{12} + y_{32} + y_{34} + y_{36} + y_{45} + y_{67} \leq C'_1$ .

When it comes to end-to-end flow rate allocation, the resource constraint imposed by shared wireless channels is as follows: When  $d_{int} = d_{tx}$ ,

$$\begin{pmatrix} 3 & 1 & 3 & 0 \\ 3 & 1 & 2 & 1 \\ 2 & 2 & 2 & 0 \end{pmatrix} \mathbf{x} \leq \mathbf{C}.$$

And, when  $d_{int} = 2 \cdot d_{tx}$ ,

$$(4 \ 2 \ 3 \ 1) \mathbf{x} \leq \mathbf{C}'.$$

In summary, the unique characteristics of location-dependent contention in wireless ad hoc networks imply a fundamentally different resource model compared to the case of wireline networks. In wireline networks, the capacity of a *link* represents the constraint on flows contending for its bandwidth, which is independent from other links. However, in the case of wireless ad hoc networks, the capacity of a wireless link is interrelated with other wireless links in its vicinity. Such a fundamental difference calls for a new treatment with respect to the models of resource constraints and allocations in wireless networks. Our original contribution toward this direction is one of the highlights of this paper.

### 3 PRICE-BASED THEORETICAL FRAMEWORK IN WIRELESS AD HOC NETWORKS

We now formally present our new pricing framework based on previous observations with respect to resource constraints in wireless ad hoc networks.

#### 3.1 Primal Problem: Optimal Resource Allocation

We associate each end-to-end flow  $f \in F$  with a *utility function*  $U_f(x_f) : \mathbb{R}_+ \rightarrow \mathbb{R}_+$ , which represents the degree of satisfaction of its corresponding end user. Moreover, we make the following assumptions about  $U_f$ :

- **A1.** On the interval  $I_f = [m_f, M_f]$ , the utility function,  $U_f$ , is increasing, strictly concave, and twice continuously differentiable.

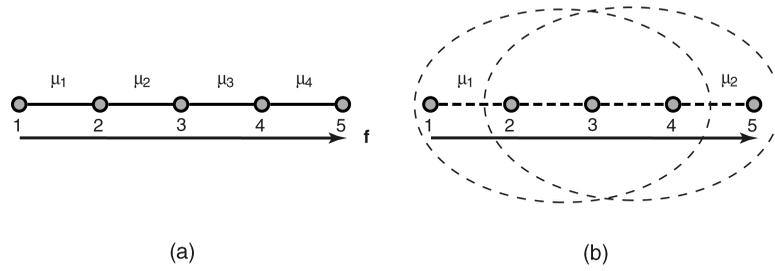


Fig. 2. Price-based framework: a comparison. (a) Wireline network. (b) Wireless ad hoc network.

- **A2.** The curvature of  $U_f$  is bounded away from zero on  $I_f$ :  $-U_f''(x_f) \geq 1/\kappa_f > 0$ .
- **A3.**  $U_f$  is additive so that the aggregated utility of rate allocation  $x = (x_f, f \in F)$  is  $\sum_{f \in F} U_f(x_f)$ .

We investigate the problem of optimal rate allocation in the sense of maximizing the aggregated utility function, which is also referred to as the *social welfare* in the literature. Such an objective achieves Pareto optimality with respect to the resource utilization and also realizes different fairness models—including proportional and max-min fairness [6]—when appropriate utility functions are specified. We argue that the problem of optimal resource allocation in wireless ad hoc networks may be formulated as the following nonlinear optimization problem (*primal problem*):

$$\mathbf{P} : \text{maximize } \sum_{f \in F} U_f(x_f) \quad (2)$$

$$\text{subject to } R\mathbf{x} \leq \mathbf{C} \quad (3)$$

$$\text{over } x_f \in I_f. \quad (4)$$

The objective function in (2) of the optimization problem maximizes the aggregated utility of all flows. The constraint of the optimization problem (3) is the resource constraint from the shared wireless channel, as discussed in Section 2. By optimizing toward such an objective, both *optimal resource utilization* and *fair resource allocations* may be achieved among *end-to-end* flows spanning multiple hops.

### 3.2 Dual Problem: Clique-Based Pricing Framework

We proceed to study how the solution to the problem  $\mathbf{P}$  may be derived, so that optimal resource allocation in terms of both utilization and fairness may be achieved. By Assumption A1, the objective function of  $\mathbf{P}$  in (2) is differentiable and strictly concave. In addition, the feasible region of the optimization problem in (3) and (4) is convex and compact. By nonlinear optimization theory, there exists a maximizing value of argument  $x$  for the above optimization problem. Let us consider the Lagrangian form of the optimization problem  $\mathbf{P}$ :

$$L(\mathbf{x}; \boldsymbol{\mu}) = \sum_{f \in F} (U_f(x_f) - x_f \sum_{q \in Q} \mu_q R_{qf}) + \sum_{q \in Q} \mu_q C_q, \quad (5)$$

where  $\boldsymbol{\mu} = (\mu_q, q \in Q)$  is a vector of Lagrange multipliers.

Now, we seek a decentralized solution where knowledge of utility functions of all flows is not needed. The key to decentralization is to investigate its dual problem and to decompose the problem via pricing. Let us first consider the dual problem  $\mathbf{D}$  of the primal problem  $\mathbf{P}$  as follows:

$$\mathbf{D} : \min_{\boldsymbol{\mu} \geq 0} D(\boldsymbol{\mu}), \quad (6)$$

where

$$D(\boldsymbol{\mu}) = \max_{x_f \in I_f} L(\mathbf{x}; \boldsymbol{\mu}) \\ = \sum_{f \in F} \max_{x_f \in I_f} \left( U_f(x_f) - x_f \sum_{q: E(f) \cap V(q) \neq \emptyset} \mu_q R_{qf} \right) + \sum_{q \in Q} \mu_q C_q. \quad (7)$$

Let us also define

$$\lambda_f = \sum_{q: E(f) \cap V(q) \neq \emptyset} \mu_q R_{qf}. \quad (8)$$

In this equation, the Lagrange multipliers  $\mu_q$  may be interpreted as the implied cost of a unit flow accessing the channel in the maximal clique  $q$ . More straightforwardly,  $\mu_q$  is the *shadow price of the clique*  $q$ .  $\lambda_f$ , on the other hand, may be interpreted as the *shadow price of the flow*  $f$ . From (8), we may observe that flow  $f$  needs to pay for *all the maximal cliques* that it traverses. For each clique, the price to pay is the product of the number of wireless links that  $f$  traverses in this clique and the shadow price of the clique. Alternatively, since

$$\lambda_f = \sum_{q: E(f) \cap V(q) \neq \emptyset} \mu_q R_{qf} = \sum_{s: s \in E(f)} \sum_{q: s \in V(q)} \mu_q, \quad (9)$$

the shadow price of a flow is also the aggregated price of all its subflows. For each subflow, its price is the aggregated price of all the maximal cliques that it belongs to.

We illustrate these observations with an example, shown in Fig. 2. The wireline network shown in Fig. 2a has a chain topology consisting of four links, associated with prices  $\mu_1, \mu_2, \mu_3, \mu_4$ . In this case, the price of the flow  $f$  is  $\lambda_f = \sum_{l=1}^4 \mu_l$ . In comparison, though the wireless ad hoc network in Fig. 2b (in this example,  $d_{int} = d_{tx}$ ) has the same topology, its maximal cliques  $q_1 = \{\{1, 2\}, \{2, 3\}, \{3, 4\}\}$  and  $q_2 = \{\{2, 3\}, \{3, 4\}, \{4, 5\}\}$  are, in effect, its units for resource allocations. Let the shadow prices of these two cliques be  $\mu_1$  and  $\mu_2$ . The price of flow  $f$  that traverses these two cliques is given by  $\lambda_f = 3\mu_1 + 3\mu_2$ , which is the sum of the product of the number of subflows of  $f$  in each clique and the shadow price of this clique. Alternatively, the price can also be written as  $\lambda_f = \mu_1 + (\mu_1 + \mu_2) + (\mu_1 + \mu_2) + \mu_2$ , which is the sum of the prices of its subflows. The price of each subflow is the aggregated price of all the maximal cliques that it belongs to.

## 4 TWO-TIER PRICE-BASED ALGORITHM FOR RESOURCE ALLOCATIONS

With an objective of promoting theory to practice, we proceed to present a decentralized two-tier algorithm based on the clique-based theoretical pricing framework that we have presented. The objective of the algorithm is to achieve

TABLE 1  
First Tier: The Iterative Algorithm

<b>Clique Price Update</b> (by clique $q$ ): At times $t = 1, 2, \dots$	
1	Receive rates $x_f(t)$ from all flows $f$ where $E(f) \cap V(q) \neq \emptyset$
2	Update price $\mu_q(t+1) = [\mu_q(t) - \gamma(C_q - \sum_{f: E(f) \cap V(q) \neq \emptyset} x_f(t) R_{qf})]^+$
3	Send $\mu_q(t+1)$ to all flows $f$ where $E(f) \cap V(q) \neq \emptyset$
<b>Rate Update</b> (by flow $f$ ): At times $t = 1, 2, \dots$	
4	Receive channel prices $\mu_q(t)$ from all cliques $q$ where $E(f) \cap V(q) \neq \emptyset$
5	Calculate $\lambda_f(t) = \sum_{q: E(f) \cap V(q) \neq \emptyset} \mu_q(t) R_{qf}$
6	Adjust rate $x_f(t+1) = x_f(\lambda_f(t))$
7	Send $x_f(t+1)$ to all cliques $q$ where $E(f) \cap V(q) \neq \emptyset$ .

optimal resource allocation in wireless ad hoc networks. In the first tier, we design an iterative algorithm that determines per-clique prices and flow rate allocations. In the second tier, we present a decentralized algorithm to construct maximal cliques. Finally, we discuss the implementation choices to integrate these two tiers.

#### 4.1 First Tier: Per-Clique Price Calculation

Treating cliques as units of resource allocation, we first present an iterative algorithm that solves the problem **P**. The iterative algorithm we propose applies the gradient projection method to the dual problem **D**.

Let

$$\phi_f(x_f) = U_f(x_f) - \lambda_f x_f. \quad (10)$$

As  $\lambda_f$  is the shadow price of the flow  $f$ ,  $\phi_f(x_f)$  may be considered as the net benefit of the flow  $f$ , which is the difference between its utility and its cost. By Assumption A1,  $\phi_f(x_f)$  is strictly concave and twice continuously differentiable. Therefore, a unique maximizer of  $\phi_f(x_f)$  exists when

$$\frac{d\phi_f(x_f)}{dx_f} = U'_f(x_f) - \lambda_f = 0. \quad (11)$$

We define such a maximizer as follows:

$$x_f(\lambda_f) = \arg \max_{x_f \in I_f} \{\phi_f(x_f)\}. \quad (12)$$

Obviously,  $x_f(\lambda_f) = [U_f^{-1}(\lambda_f)]_{m_f}^{M_f}$ . Here,  $x_f(\lambda_f)$  is generally referred to as the *demand function*, which reflects the optimal rate for flow  $f$ , where its net benefit is maximized with a flow price of  $\lambda_f$ .

Now, we solve the dual problem **D** using the gradient projection method [8]. In this method,  $\mu$  is adjusted in the opposite direction to the gradient  $\nabla D(\mu)$ :

$$\mu_q(t+1) = \left[ \mu_q(t) - \gamma \frac{\partial D(\mu(t))}{\partial \mu_q} \right]^+, \quad (13)$$

where  $\gamma$  is the step size.  $D(\mu)$  is continuously differentiable since  $U_f$  is strictly concave [8]. Thus, based on (7), the  $q$ -dimension of the gradient is given as follows:

$$\frac{\partial D(\mu)}{\partial \mu_q} = C_q - \sum_{f: E(f) \cap V(q) \neq \emptyset} x_f(\lambda_f) R_{qf}. \quad (14)$$

Equation (14) gives the difference between the resource capacity  $C_q$  and its load demand  $\sum_{f: E(f) \cap V(q) \neq \emptyset} x_f(\lambda_f) R_{qf}$ , which are the rates of all flows that pass this clique multiplied by the number of subflows they have in this clique. Substituting (14) into (13), we have

$$\mu_q(t+1) = \left[ \mu_q(t) - \gamma \left( C_q - \sum_{f: E(f) \cap V(q) \neq \emptyset} x_f(\lambda_f(t)) R_{qf} \right) \right]^+. \quad (15)$$

Equation (15) reflects the law of supply and demand. If the demand for bandwidth at clique  $q$  exceeds its supply  $C_q$ , the resource constraint is violated. Thus, the clique price  $\mu_q$  is increased. Otherwise,  $\mu_q$  is reduced.

We summarize the first-tier iterative algorithm in Table 1, where clique  $q$  and flow  $f$  are considered as abstract entities capable of computing and communicating. In the second tier of the algorithm, details of such entities will be presented.

We are now in a position to show the properties of this iterative algorithm. Let us define  $Y(f) = \sum_{q \in Q} R_{qf}$ , which leads to the definition of  $\bar{Y} = \max_{f \in F} Y(f)$  as, intuitively speaking, the “length” of the “longest” path. We further define  $Z(q) = \sum_{f \in F} R_{qf}$ , leading to the definition of  $\bar{Z} = \max_{q \in Q} Z(q)$  as the number of subflows at the most “congested” clique. Let  $\bar{\kappa} = \max_{f \in F} \kappa_f$ , where  $\kappa_f$  is the bound on the curvature of  $U_f(\cdot)$  (see Assumption A2).

**Theorem 1.** Assume that  $0 < \gamma < 2/\bar{\kappa}\bar{Y}\bar{Z}$ , starting from any initial rates  $\mathbf{x}(0)$  ( $x_f \in I_f$ ) and prices  $\mu(0) \geq 0$ , every limit point  $(\mathbf{x}^*, \mu^*)$  of the sequence  $(\mathbf{x}(t), \mu(t))$  generated by the algorithm in Table 1 is primal-dual optimal.

The detailed proof of this theorem is given in our technical report [9].

#### 4.2 Second Tier: Decentralized Clique Construction

The first tier of the algorithm treats maximal cliques as entities that are able to perform the communication and computation tasks. Obviously, these tasks need to be performed by the network nodes that constitute the maximal clique. As a starting point, a decentralized algorithm to construct maximal cliques is required. Since the existing maximal clique construction algorithms are centralized in nature [10], they cannot be directly applied here. Nevertheless, the unique graphical properties of the

wireless link contention graph may have the potential to facilitate efficient clique construction. Hereafter, we present a decentralized maximal clique construction algorithm that explores the characteristics of wireless link contention graphs. In this algorithm, the network topology is decomposed into overlapping subgraphs and maximal cliques are constructed based only on local topological information within each of the subgraphs. Since only wireless links that are geographically close to each other will form an edge in the wireless link contention graph, the communication and computational overhead is significantly reduced.

We first show the implication of topological characteristics of wireless link contention graphs when it comes to constructing maximal cliques. Let us denote the maximal clique that contains wireless link  $s \in S$  as  $q(s)$  (i.e.,  $s \in V(q)$ ) and the set of all maximal cliques that contain the wireless link  $s$  as  $Q(s) = \{q(s)\}$ . We now give the subgraph of  $G$  on which  $Q(s)$  can be constructed. To facilitate discussions, we introduce the following terms.

**Definition 1 (Neighbor Sets).** *The wireless link neighbor set  $\mathbb{LN}(e)$  of a wireless link  $e \in E$  is defined as*

$$\mathbb{LN}(e) = \{e' | e \cap e' \neq \emptyset, e' \in E\}.$$

Similarly, the wireless link  $k$ -neighbor set  $\mathbb{LN}^k(e)$  of  $e$  is defined by induction: 1)  $\mathbb{LN}^1(e) = \mathbb{LN}(e)$  and 2)  $\mathbb{LN}^k(e) = \mathbb{LN}(\mathbb{LN}^{k-1}(e))$  for  $k > 1$ . For  $s \in S \subseteq E$ , we further define  $\mathbb{SN}^k(s) = \mathbb{LN}^k(s) \cap S$ .

**Theorem 2.** *Let graph  $G_c[V_c(s)]$  be an induced subgraph of  $G_c$  with  $V_c(s) = \mathbb{SN}^2(s) \subseteq V_c$ . Then,  $G_c[V_c(s)]$  contains sufficient and necessary topological information to construct  $Q(s)$ , when  $d_{int} = d_{tx}$ . And,  $G_c[V_c(s)]$  contains necessary topological information to construct  $Q(s)$ , when  $d_{int} > d_{tx}$ . Let graph  $G(s)$  be  $G(s) = (V(s), E(s))$  with  $E(s) = \mathbb{LS}^3(s)$  and  $V(s) = \{i | \exists s \text{ such that } i \in s \text{ and } s \in \mathbb{SN}^2(s)\}$ .  $G(s)$  is a subgraph  $G$ , and  $G(s)$  contains sufficient and necessary topological information to construct  $G_c[V_c(s)]$ .*

**Proof.** When  $d_{int} = d_{tx}$  by the definitions of the wireless link contention graph and clique, it is obvious that  $\cup_{q \in Q(s)} V(q) = \mathbb{SN}^2(s)$ . This shows that  $G_c[V_c(s)]$  contains sufficient and necessary topological information to construct  $Q(s)$ . Also, for any pair of  $s'', s' \in \mathbb{SN}^2(s)$ , we need to know whether they contend with each other to determine whether they are connected in  $G_c[V_c(s)]$ . Apparently,  $\mathbb{LN}^3(s)$  contains all the topological information to construct  $G_c[V_c(s)]$ . When  $d_{int} > d_{tx}$ , there may exist wireless links which interfere with each other, yet are not connected by any other wireless links in the network topology. Thus,  $G_c[V_c(s)]$  only contains necessary topological information.  $\square$

For wireless link  $s \in S$ , one of its vertices will be selected as its *delegation node*, denoted as  $v(s)$ . The delegation node  $v(s)$  constructs all maximal cliques  $q \in Q(s)$  by applying the Bierstone algorithm [10] on graph  $G_c[V_c(s)]$ .

We consider an example shown in Fig. 3. When  $d_{int} = d_{tx}$ , the set of wireless links

$$\mathbb{SN}^2(\{1, 2\}) = \{\{8, 3\}, \{3, 1\}, \{1, 2\}, \{2, 5\}, \{5, 14\}, \{6, 11\}, \{6, 12\}\}$$

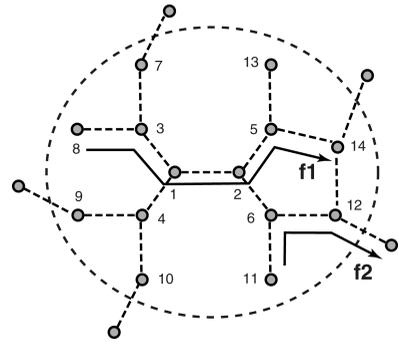


Fig. 3. Example of decentralized clique construction.

represents all the vertices that appear in  $G_c[V_c(\{1, 2\})]$ . To construct all the maximal cliques  $Q(\{1, 2\})$ , the algorithm also needs to determine which wireless links in  $\mathbb{SN}^2(\{1, 2\})$  contend with each other. For example, in Fig. 3, whether subflow  $\{5, 14\}$  contends with  $\{6, 12\}$  needs to be known to determine whether they are within the same clique. This implies that the knowledge of the wireless link  $\{12, 14\}$  needs to be known by the algorithm for correct clique construction. Thus,

$$\mathbb{LN}^3(\{1, 2\}) = \mathbb{SN}^2(\{1, 2\}) \cup \{12, 14\}$$

needs to be known. When  $d_{int} > d_{tx}$ , the network topology graph does not have sufficient information to infer all the interferences among wireless links. In this case, the clique construction algorithm only provides an approximation solution. For practical deployment, it will work with the measurement-based bandwidth estimation technique presented in Section 6, which takes into account the interferences among wireless communications.

### 4.3 Two-Tier Algorithm: Integration Choices

In the first tier of the algorithm, maximal clique  $q$  is considered as an entity that is able to perform the following tasks:

1. Obtain the aggregated rate  $\sum_{f: E(f) \cap V(q) \neq \emptyset} x_f R_{qf}$  within it.
2. Compute the clique-based shadow price  $\mu_q$ .
3. Communicate the price  $\mu_q$  to the flows that traverse through.

After presenting the decentralized clique construction algorithm, we now proceed to discuss how these tasks are distributed to the network nodes that constitute the maximal clique. There are two implementation choices.

In implementation 1, one delegation node in clique  $q$  serves as a master that performs the task of price calculation, denoted as  $v(q)$ . At time  $t$ , each delegation node  $v(s)$  collects the rate of flow  $f$  which passes it (i.e.,  $s \in E(f)$ ), computes rate  $y_s$  at wireless link  $s$  according to  $y_s(t) = \sum_{f: s \in E(f)} x_f(t)$  and, sends it to the master nodes  $v(q)$  of all cliques  $q$  which  $s$  belongs to (i.e.,  $s \in V(q)$ ). The master node  $v(q)$  then computes the new price  $\mu_q(t+1)$  of clique  $q$  according to

$$\mu_q(t+1) = \left[ \mu_q(t) - \gamma \left( C_q - \sum_{s: s \in V(q)} y_s(t) \right) \right]^+ \quad (16)$$

and distributes it to the other delegation nodes  $v(s)$  within clique  $q$  (i.e.,  $s \in V(q)$ ). After obtaining the updated clique price  $\mu_q(t+1)$ ,  $v(s)$  computes a per hop price  $\lambda_s(t+1)$  according to  $\lambda_s(t+1) = \sum_{q:s \in V(q)} \mu_q(t+1)$  for each flow  $f$  that satisfies  $s \in E(f)$ , then sends  $\lambda_s(t+1)$  to the source of  $f$ .

For flow  $f$ , its source node performs the task of rate update. When the source node receives the per hop prices  $\lambda_s(t)$ , it computes the flow price  $\lambda_f(t)$  according to  $\lambda_f(t) = \sum_{s:s \in E(f)} \lambda_s(t)$  and adjusts the rate  $x_f$  according to  $x_f(t+1) = x_f(\lambda_f(t))$ . It then notifies  $v(s)$  ( $s \in E(f)$ ) of  $x_f(t+1)$ .

In implementation 2, every delegation node in a clique performs price calculation. In particular, it differs from implementation 1 in the following aspects. At time  $t$ , each delegation node  $v(s)$  broadcasts the rate information  $y_s$  to the other delegation nodes  $v(s')$  that satisfy  $s' \in \mathcal{SN}^2(s)$ . As such, each delegation node  $v(s)$  can compute the price of clique  $q$  ( $s \in V(q)$ ) independently. Let us denote the price of clique  $q$  at node  $v(s)$  as  $\mu_{q(v(s))}$ .  $\mu_{q(v(s))}$  is calculated as follows:

$$\mu_{q(v(s))}(t+1) = \left[ \mu_{q(v(s))}(t) - \gamma \left( C_q - \sum_{s:s \in V(q)} y_s(t) \right) \right]^+ \quad (17)$$

Node  $v(s)$  then directly computes and communicates  $\lambda_s(t+1)$  to the source of  $f$ , which satisfies  $s \in E(f)$ .

## 5 TOWARD ASYNCHRONY: IMPROVING THE TWO-TIER ALGORITHM

Our two-tier algorithm assumes that updates at the sources and the relaying nodes are synchronized to occur at times  $t = 1, 2, \dots$ . In realistic ad hoc network environments, however, such synchronization is difficult to achieve. In this section, we improve the algorithm to an asynchronous setting, where sending rates and clique prices are updated at different times at different nodes.

First, we briefly introduce the asynchronous model that will be used for our algorithm in the context of implementation I. Let  $T = \{0, 1, 2, \dots\}$  be the set of time instances at which either rates or prices are updated. In particular, we define

1.  $T_q \subseteq T$ —the set of time instances at which master node  $v(q)$  updates  $\mu_q$ .
2.  $T_s^\lambda \subseteq T$ —the set of time instances at which delegation node  $v(s)$  updates  $\lambda_s$ .
3.  $T_f \subseteq T$ —the set of time instances at which the source of flow  $f$  updates  $x_f$ .
4.  $T_s^y \subseteq T$ —the set of time instances at which delegation node  $v(s)$  updates  $y_s$ .

The asynchronous model further makes the following assumption.

- **A4. (Partial Asynchronism).** There exists a positive integer  $B$  such that:
  1. For every flow  $f$ , clique  $q$ , and wireless link  $s$ , the time between consecutive updates is bounded by  $B$  for both price and rate updates.
  2. One-way communication delay between any two nodes is at most  $B$  time units.

This partial asynchronism model is first discussed in [11] and is then adopted by Low and Lapsley in the context of wireline networks [5]. Now, we improve our two-tier

resource allocation algorithm and analyze its convergence under this asynchronous model.

In the asynchronous environment, node  $v(q)$ , which updates the price  $\mu_q(t)$  at time  $t \in T_q$ , may not have the knowledge of rate information  $y_s(t)$ . Instead, it only knows a sequence of recent rate updates,  $y_s((\tau_s^q)^1)$ ,  $y_s((\tau_s^q)^2)$ ,  $\dots$ , that satisfy  $t - B \leq (\tau_s^q)^1 \leq (\tau_s^q)^2 \leq \dots \leq t$ . Thus, node  $v(q)$  estimates the rate  $\hat{y}_s^q(t)$  by using a weighted average of recent values as follows:

$$\hat{y}_s^q(t) = \sum_{t'=t-B}^t \alpha_s^q(t', t) y_s(t') \quad \text{with} \quad \sum_{t'=t-B}^t \alpha_s^q(t', t) = 1. \quad (18)$$

Further, node  $v(q)$  computes the price of clique  $q$  according to the following, which is essentially (15) with the load  $\sum_{f:E(f) \cap V(q) \neq \emptyset} x_f(\lambda_f(t)) R_{qf}$  replaced by its estimation  $\hat{y}_s^q(t)$ .

$$\mu_q(t+1) = \left[ \mu_q(t) - \gamma \left( C_q - \sum_{s:s \in V(q)} \hat{y}_s^q(t) \right) \right]^+, \forall t \in T_q. \quad (19)$$

At all times  $t \notin T_q$ ,  $\mu_q$  is unchanged, i.e.,  $\mu_q(t+1) = \mu_q(t)$ .

Similarly, to compute the per hop price  $\lambda_s(t)$  at time  $t \in T_s^\lambda$ , node  $v(s)$  estimates the clique price  $\hat{\mu}_q^s(t)$  according to

$$\hat{\mu}_q^s(t) = \sum_{t'=t-B}^t \alpha_q^s(t', t) \mu_q(t') \quad \text{with} \quad \sum_{t'=t-B}^t \alpha_q^s(t', t) = 1 \quad (20)$$

and calculates per hop price according to

$$\lambda_s(t+1) = \sum_{q:s \in V(q)} \hat{\mu}_q^s(t), \forall t \in T_s^\lambda. \quad (21)$$

At time  $t \notin T_s^\lambda$ ,  $\lambda_s$  is unchanged.

At time  $t \in T_f$ , the source of  $f$  estimates its flow price according to  $\hat{\lambda}_f(t) = \sum_{s:s \in E(f)} \hat{\lambda}_s^f(t)$ , where

$$\hat{\lambda}_s^f(t) = \sum_{t'=t-B}^t \alpha_f^s(t', t) \lambda_s(t') \quad \text{with} \quad \sum_{t'=t-B}^t \alpha_f^s(t', t) = 1, \quad (22)$$

and computes its rate according to  $x_f(t+1) = x_f(\hat{\lambda}_f(t))$ ,  $\forall t \in T_f$ . At time  $t \notin T_f$ ,  $x_f$  is unchanged.

At time  $t \in T_s^y$ , node  $v(s)$  calculates  $y_s$  as  $y_s(t+1) = \sum_{f:s \in E(f)} \hat{x}_f^s(t)$ , where

$$\hat{x}_f^s(t) = \sum_{t'=t-B}^t \alpha_f^s(t', t) x_f(t') \quad \text{with} \quad \sum_{t'=t-B}^t \alpha_f^s(t', t) = 1. \quad (23)$$

At time  $t \notin T_s^y$ ,  $y_s(t+1) = y_s(t)$ .

In this algorithm, the elements of  $T$  can be viewed as the indices of the sequence of physical times at which updates to either prices or rates occur. The sets  $T_f$ ,  $T_q$ ,  $T_s^\lambda$ ,  $T_s^y$  as well as the physical times they represent need not be known to any other nodes since their knowledge is not required in the price and rate computation. Thus, there is no requirement for synchronizing the local clocks at different nodes. We are able to show that, under Assumption A4, our resource allocation algorithm converges to global optimum even in asynchronous environments. Our main result is formally presented in the following theorem.

**Theorem 3.** Assume that the step size  $\gamma$  is sufficiently small. Then, starting from any initial rate  $x(0)$  ( $x_f \in I_f$ ) and prices  $\mu(0) \geq 0$ , every limit point  $(x^*, \mu^*)$  of the sequence  $(x(t), \mu(t))$  generated by the asynchronous price-based resource allocation algorithm is primal-dual optimal.

The detailed proof of this theorem is given in our technical report [9].

The improvements on implementation 2 toward asynchrony is similar to implementation 1. The only difference is that it does not need the communication between delegation nodes and master node of a clique for its price update as the clique prices are computed independently at delegation nodes. We show via simulation that the asynchronous algorithm under implementation 2 closely matches the global optimum at equilibrium, if the step size  $\gamma$  is sufficiently small, and the initial prices  $\mu_{q(v(s))}(0)$  at different delegation nodes  $v(s)$  are the same for a clique  $q$ .

## 6 EMPIRICAL STUDIES

In this section, we present deployment issues of our price-based resource allocation algorithm in realistic wireless networking environments and evaluate its performance in an empirical study involving a set of simulation environments.

### 6.1 Simulation Environments

We study the price-based resource allocation algorithm in three different simulation environments. The first environment, referred to as *synsim* for convenience, assumes bounded communication delay and synchronized message updates. The second environment, referred to as *asynsim* for convenience, considers the asynchronous environments in wireless ad hoc networks. In both environments, we assume that the transmission range is the same as the interference range, both of which are 250m. We further assume that the MAC layer scheduling is *ideal* in the sense that it can achieve the wireless channel capacity of 2 Mbps and the routing algorithm selects the shortest path. The third environment, referred to as *realsim*, considers realistic wireless networking environments. *Realsim*, implemented using the ns-2 simulator, adopts the two-ray ground reflection model as the radio propagation model and uses IEEE 802.11 DCF as the MAC protocol. The transmission range in *realsim* is smaller than the interference range, which are 250m and 550m, respectively. The data transmission rate in *realsim* is 1 Mbps. With respect to routing, the AODV routing protocol [12] is used in *realsim*. In all simulation environments, the utility function  $U_f(x_f) = \ln(x_f)$  is used, which enforces proportional fairness.

### 6.2 Deployment Issues in Realistic Wireless Environments

Realistic physical and MAC layers in wireless ad hoc networks—that are reflected in *realsim*—presents several challenges to deploy our price-based resource allocation algorithm. First, the achievable channel capacity varies at different contention regions (cliques) depending on the MAC protocol. It is usually much smaller than the ideal channel capacity and cannot be known a priori. Dynamically estimating the achievable channel capacity at different contention regions is a critical problem to deploy our algorithm in realistic wireless environments. Second, the two-tier decentralized clique construction and price calculation algorithm requires communication among nodes, which may introduce additional overhead to the network. Designing an efficient communication protocol that still ensures appropriate algorithm convergence is also a challenging problem. To address these challenges, we present two deployment techniques: measurement-based bandwidth estimation and lightweight communication protocol.

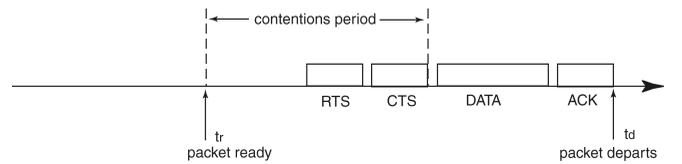


Fig. 4. Measurement-based bandwidth estimation.

#### 6.2.1 Measurement-Based Bandwidth Estimation

The measurement-based bandwidth estimation is based on the approach presented in [13]. It measures the *achievable* bandwidth of each wireless link based on its historical data transmission results.

As shown in Fig. 4, under the IEEE 802.11 MAC protocol, at time  $t_r$ , when a packet from a particular wireless link becomes the head-of-line packet (i.e., the first packet waiting to be transmitted), we claim that the packet is *ready*. At time  $t_d$ , when the link layer acknowledgment is received, the packet *departs*. The transmission delay of this packet is then given as  $t_d - t_r$ , which includes a contention period. The contention period indicates the channel bandwidth used by packet transmissions of other wireless links within the contention region. The achievable bandwidth observed by this wireless link is then calculated as  $\frac{z}{t_d - t_r}$ , where  $z$  is the size of the packet. To achieve more accurate measurement results, we use a window of  $w$  packets to conduct the bandwidth estimation, i.e., the bandwidth is estimated as  $\frac{w \cdot z}{\sum_{i=1}^w t_d^i - t_r^i}$ . The measurement-based bandwidth estimation takes into account the effect of physical layer interference and the inefficiency of MAC protocols, as it is based on the *scheduling results* of packet transmissions.

#### 6.2.2 Lightweight Communication Protocol

To calculate the price of each clique, only its gradient (i.e., the difference between achievable capacity and traffic demand) needs to be known. Based on this observation, each wireless link calculates its local gradient by monitoring its achievable bandwidth and its traffic load. Instead of communicating both load and bandwidth information, only the gradient information is sent along with the connectivity information to construct cliques and compute their prices. To achieve low overhead communication, the information is sent via piggybacking. First, the local gradient information of each wireless link is piggybacked onto the data packets of the flows passing by to notify the downstream nodes. Second, working with the AODV routing protocol, the connectivity and local gradient information is also piggybacked onto HELLO packets and sent at a certain time interval. Based on Theorem 2, end nodes of subflow  $s$  cache the information within  $\mathbb{LS}^3(s)$  and transmit information within  $\mathbb{LS}^2(s)$  to their neighboring nodes. The prices are also piggybacked onto data packets so that the destination of a flow can notify its source via FEEDBACK packets. Such protocol provides an asynchronous information update for price calculation and communication. As we have shown in Theorem 3, the price-based algorithm converges to the global optimum even in such asynchronous environments.

Using the above deployment techniques, we have implemented *realsim* in ns-2. As shown in Fig. 5a, the price-based resource allocation algorithm is implemented as several components at different levels in ns-2. At the MAC level, the

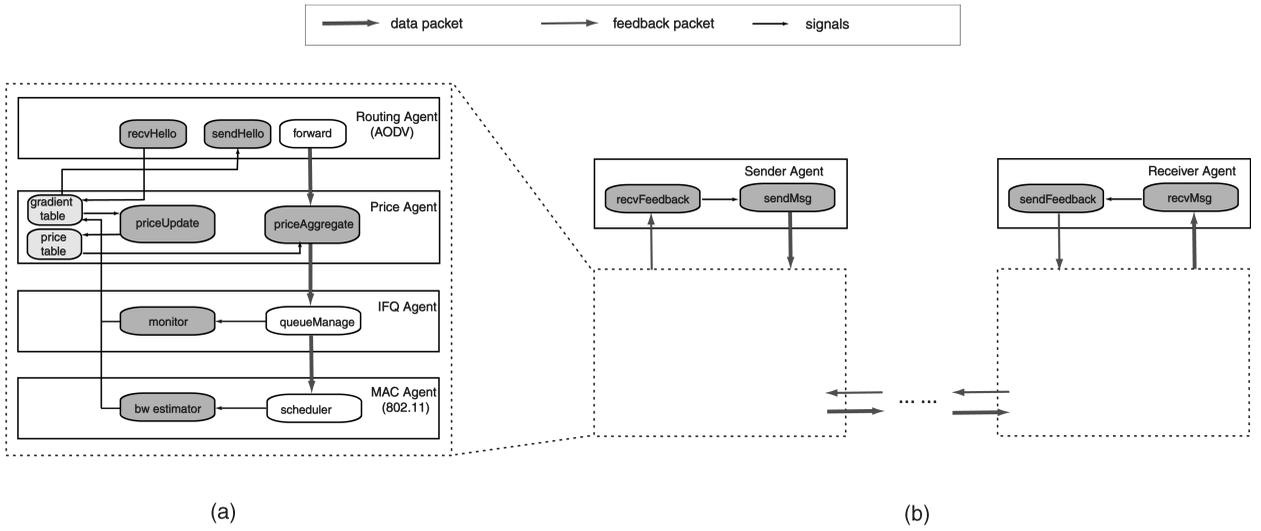


Fig. 5. Implementation in ns-2 simulator.

bandwidth estimator measures the local achievable bandwidth to each neighboring node. At the interface queue level, the monitor observes the backlogged traffic to each neighboring node. Working with the bandwidth estimator, the monitor generates the local gradient for each wireless link to its neighbors. At the routing level, HELLO messages of the AODV routing protocol communicate the gradient information to its neighbors. The local gradient information, together with the gradient information received from HELLO messages, is maintained in a gradient cache table. The changes at the gradient cache table trigger the price update component, which reads the gradient information and calculates the clique prices. These clique prices are maintained at a price cache table. The price aggregation component receives data packets from the routing layer. Depending on the data packet's next hop, the price aggregation component calculates the per hop price and adds it to the aggregated price from the upstream hops. At the end nodes, the receiving component retrieves the aggregated price information from the data packets and sends back FEEDBACK packets if it observes a price change. Upon receiving FEEDBACK packets, the sending component adjusts its sending rate based on the new price via the rate update algorithm, as shown in Fig. 5b.

### 6.3 Convergence

We first study the convergence behavior of our price-based resource allocation algorithm under different simulation environments and identify the factors that affect this procedure.

#### 6.3.1 Convergence Speed

With appropriately tuned step sizes, we first evaluate how rapidly our algorithm converges to the global optimum. We simulate the algorithm on chain topologies from four hops to 10 hops in *synsim*. As an example, the four-hop chain topology and its traffic pattern is shown in Fig. 6. In all the experiments, the initial values of sending rates are 2 Mbps and the initial prices are 2. The results are shown in Table 2 along with their corresponding step sizes, which are tuned to ensure the most efficient convergence.<sup>1</sup> From these results,

1. The termination criteria in *synsim* are  $|x_f(t) - x_f^*| \leq \epsilon$  for all  $f \in F$  and  $|\mu_q(t) - \mu_q^*| \leq \epsilon$  for all  $q \in Q$  with  $\epsilon = 10^{-4}$ .

we observe that the best step sizes and the convergence time apparently correspond to the scale of the network. In particular, the larger the number of cliques through which the longest flow passes (i.e.,  $\bar{Y}$ ), the smaller the step sizes and the more iterations are required for convergence.

We have also carried out this experiment with different initial settings of prices and rates. We observe that the algorithm always converges, regardless of the initial settings. In particular, the sending rates always converge to a unique optimum, regardless of the initial rates, and the prices may converge to different values—all of which are dual optimal—if different initial prices are used. This is because, at equilibrium, only the flow price  $\lambda_f^*$  is constrained by  $U_f'(x_f^*) = \lambda_f^*$  and different price vectors  $\mu^*$  may lead to the same value of  $\lambda_f^*$ .

#### 6.3.2 Convergence in Asynchronous Environments

With *asynsim*, we evaluate the convergence behavior of our algorithm in asynchronous environments. We first show the convergence behavior of implementation 1. Recall that, in asynchronous models, recent updates are averaged to accommodate delayed and out-of-order messages. In the simulation, we adopt a moving average method to specify the weight parameters. In particular, let  $0 \leq (\tau)^1 \leq (\tau)^2 \leq \dots \leq (\tau)^m$  be the time instances of the received updates, then  $\alpha_s^q$  is specified as

$$\alpha_s^q(t', t) = 1 \text{ if } m = 1, t' = (\tau)^1, \quad (24)$$

$$\alpha_s^q(t', t) = (1 - \alpha)^{m-j} \cdot \alpha \quad (25)$$

$$\text{if } m > 1, t' = (\tau)^j, j = 1, 2, \dots, m.$$

$\alpha_s^f$ ,  $\alpha_f^s$ , and  $\alpha_q^s$  are defined in the same way.  $\alpha$  is a unified parameter that represents the weight of the history in estimation. When  $\alpha = 0$ , only the most recently received

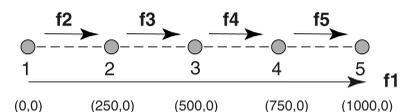


Fig. 6. Four-hop chain topology.

TABLE 2  
Number of Iterations Toward Convergence

topology (hops)	4	5	6	7	8	9	10
step size ( $\gamma$ )	1	0.8	0.75	0.62	0.6	0.6	0.5
number of iterations to converge	18	23	42	46	55	96	102

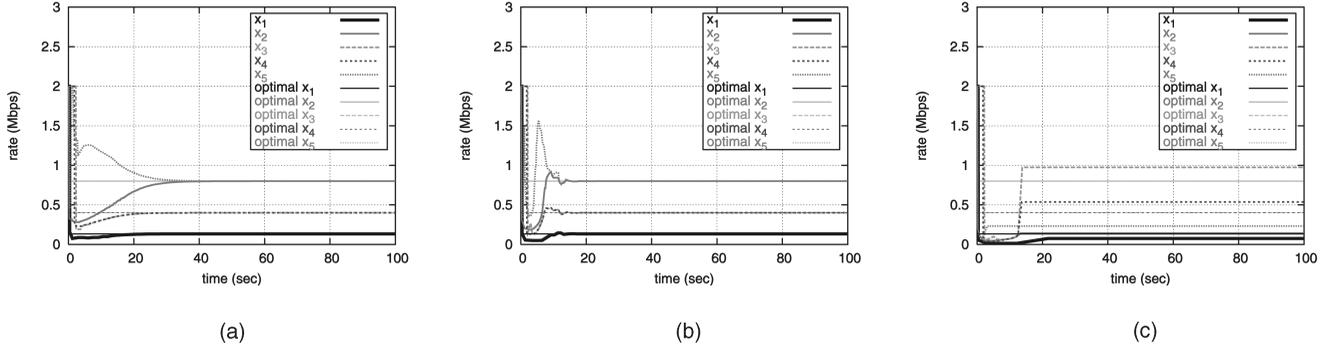


Fig. 7. Asynchronous experiments of implementation 1 with different step sizes  $\gamma$  ( $\alpha = 0$ ). (a)  $\gamma = 0.01$ . (b)  $\gamma = 0.05$ . (c)  $\gamma = 0.5$ .

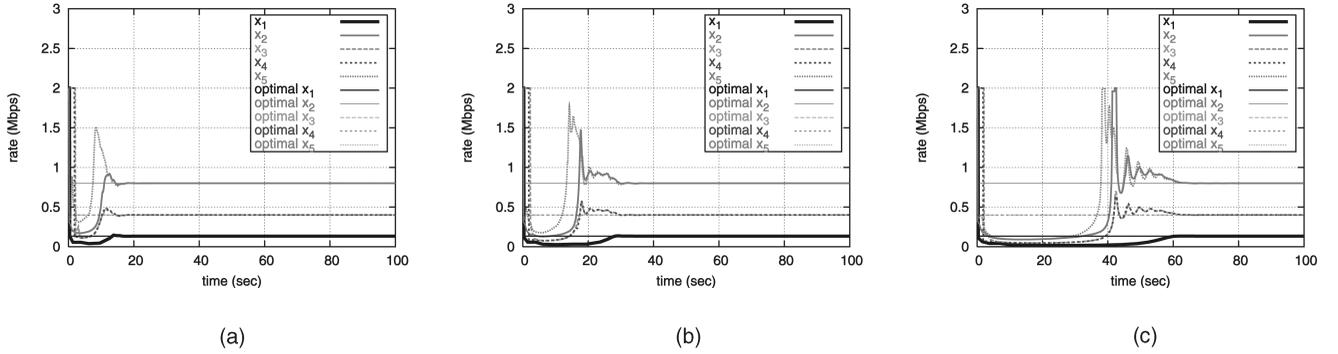


Fig. 8. Asynchronous experiments of implementation 1 with different weights  $\alpha$  ( $\gamma = 0.05$ ). (a)  $\alpha = 0.1$ . (b)  $\alpha = 0.4$ . (c)  $\alpha = 0.6$ .

update is used for estimation. We evaluate the impact of  $\alpha$  and step size  $\gamma$  on the convergence. We conduct the experiments on a 4-hop chain topology (Fig. 6) using different values of  $\gamma$  and  $\alpha$ . The experimental results are shown in Fig. 7 and Fig. 8, respectively. From the results, we have the following observations: First, at equilibrium, with sufficient small step size ( $\gamma \leq 0.05$ ), independent of the choices of  $\alpha$ , rate allocation in asynchronous environments achieves the global optimum as in synchronous settings. This validates the theoretical claim in Theorem 3. Second, the value of the step size that ensures the convergence and optimal rate allocation is much smaller than the synchronous case. For example, as shown in Fig. 7, the resource allocation does not converge to the optimum when the step size  $\gamma = 0.5$ , while, in synchronous settings, the algorithm converges to optimum when  $\gamma = 1$ . Further, the value of  $\alpha$  affects the convergence speed, with a larger  $\alpha$  leading to a longer convergence time. This means that, in implementation 1,  $\alpha = 0$  can ensure the fastest convergence to the global optimum.

We now repeat these experiments using implementation 2, with step size  $\gamma = 0.05$ . The results are shown in Fig. 9. From these results, we have the following observations: First, the equilibrium rate allocation closely matches the optimal values. Second, the value of  $\alpha$  affects the convergence speed and how close the equilibrium rate allocation matches the optimum. The reason behind this observation can be intuitively explained as follows: In implementation 2,

different delegation nodes have different rate estimations for clique price calculation, depending on the value of  $\alpha$ . Although the clique price changes will converge to zero at each individual node, the difference between clique prices at these nodes does not. Such a difference may vary with the value of  $\alpha$ . Yet, no matter what value  $\alpha$  is set to, the equilibrium rate allocation always closely matches the optimum in the simulations. This is because nodes within a clique have relatively small communication delays, hence small rate estimation differences.

### 6.3.3 Impact of Measurement Window Size on Convergence

Now, we study the convergence behavior of our algorithm with bandwidth estimation and evaluate the impact of measurement parameters in *realsim*. We experiment with different measurement window sizes  $w$  on the four-hop chain topology (Fig. 6). The results are shown in Fig. 10. From the results, we have the following observation: The algorithm converges faster in the case of  $w = 5$  than the case of  $w = 20$  because a smaller measurement window gives faster feedback on the channel condition. On the other hand, too small  $w$  results in imprecise measurement. This leads to slight fluctuations at the equilibrium, as shown in Fig. 10 in the case of  $w = 5$ . In what follows, we use  $w = 20$  as the default

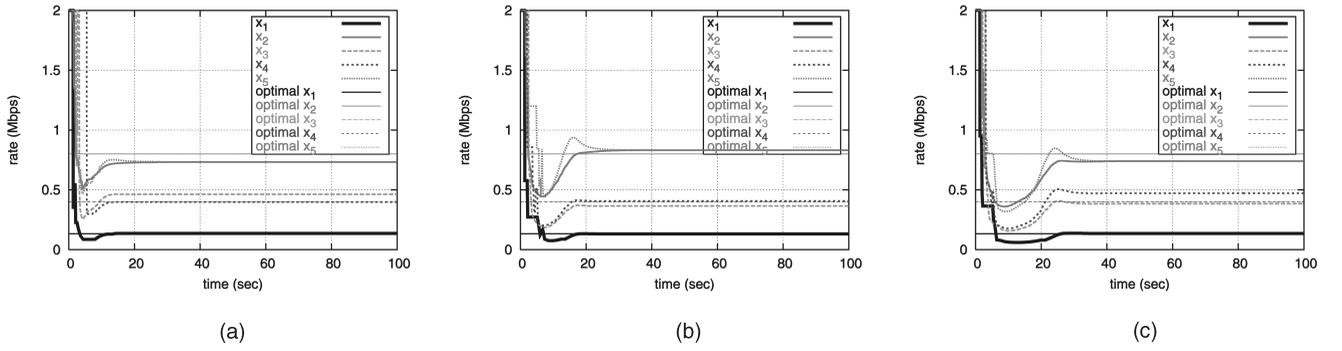


Fig. 9. Asynchronous experiments with implementation 2. (a)  $\alpha = 0.01$ . (b)  $\alpha = 0.4$ . (c)  $\alpha = 0.6$ .

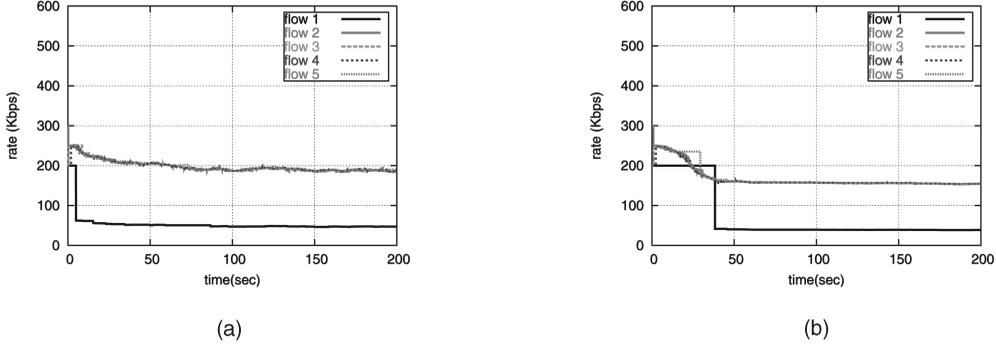


Fig. 10. Convergence with bandwidth estimation. (a)  $w = 5$ . (b)  $w = 20$ .

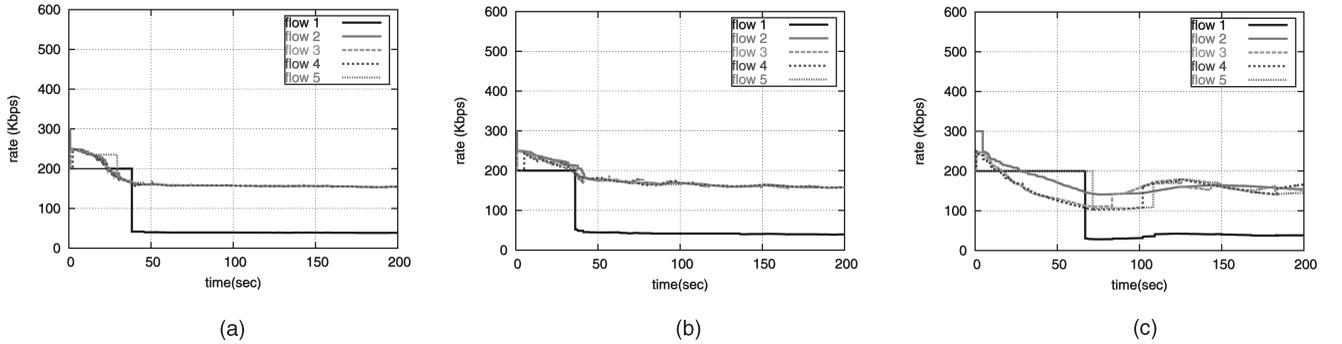


Fig. 11. Convergence under the piggyback-based lightweight communication protocol. (a) hello interval = 1 second. (b) hello interval = 10 seconds. (c) hello interval = 20 seconds.

measurement window size because it gives stable and precise measurement results with acceptable convergence speed.

### 6.3.4 Impact of HELLO Interval on Convergence

We experiment with different lengths of HELLO intervals and evaluate its impact on convergence. As shown in Fig. 11, large HELLO intervals (e.g., interval = 20 seconds) increase the convergence time and may cause small fluctuations. The results also show that, in a static environment, a HELLO interval of less than 10 seconds can ensure convergence with satisfactory speed and can achieve stability at equilibrium. Obviously, different lengths of HELLO intervals also affect the overhead. Long HELLO intervals can significantly reduce the overhead. In the following simulations, we use 1 second as the default length of the HELLO interval.

### 6.3.5 Convergence in Random Networks

Now, we study the convergence behavior of our algorithm with respect to both transmission rate and throughput in a

randomly generated wireless network, as shown in Fig. 12a. This network consists of 30 nodes deployed over a  $600 \times 600 m^2$  region. In the experiment, five flows are established between five different pairs of nodes to start at different time instants. Fig. 12b and Fig. 12c plot the instantaneous transmission rate and throughput of each flow, respectively. From these results, we have the following observations: 1) Our algorithm converges with satisfactory speed even in relatively large scale networks (e.g., 30 node over a  $600 \times 600$  region) and 2) our algorithm converges when traffic dynamically joins the network.

## 6.4 Impact of Realistic Wireless Interference

In the next set of experiments, we are interested in studying the impact of realistic wireless interference on our algorithm.

### 6.4.1 Special Scenarios

First, using *realsim*, we perform our experiments in a set of special network topologies: the *hidden terminal* scenario, the *exposed terminal* scenario, and the *race condition* scenario.

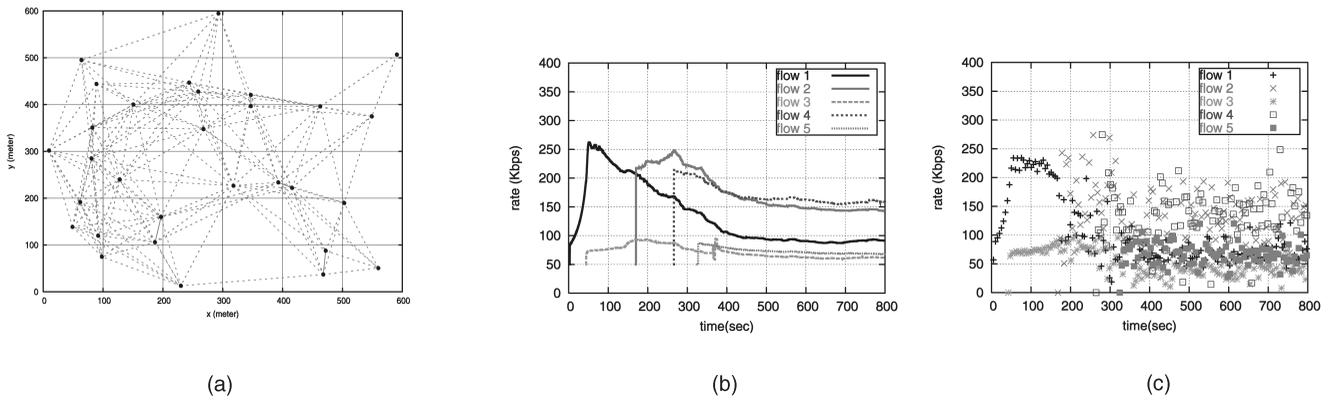


Fig. 12. Convergence in random networks. (a) Random topology. (b) Transmission rate. (c) Throughput.

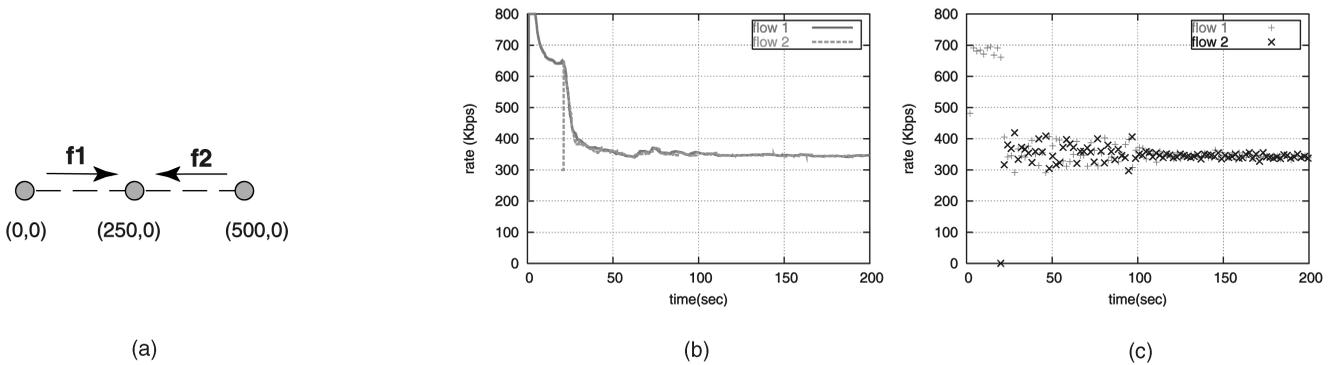


Fig. 13. Convergence in the hidden terminal scenario 1. (a) Hidden terminal topology. (b) Transmission rate. (c) Throughput.

- Hidden terminal.* Fig. 13 shows one example of the hidden terminal scenario, as well as experimental results on the convergence of the transmission rate and the throughput of our algorithm. From the results, we observe that the algorithm performs as expected: At equilibrium, two flows share the resource fairly. The result is obvious because the sending nodes of both flows are able to obtain the information from each other, thus correctly constructing the clique and calculating its price. In contrast, we also show the performance of our algorithm over a different hidden terminal scenario, as shown in Fig. 14. In this scenario, the sending nodes of the two flows are unable to communicate, though their transmissions still interfere with each other. Thus, each wireless link treats itself as the only link within the clique, though the correct clique construction should consist of both wireless links. In this case, the price of a clique relies on the gradient of one wireless link, which is in turn calculated based on the bandwidth estimation at either node 2 or node 3. Node 2 can sense the interference from node 3, when it sends FEEDBACK packets to node 1. Similarly, node 3 can sense the interference from node 2, when it sends the data packets. But, due to their asymmetric traffic loads, their bandwidth estimation results are different. As a result, the rate allocation of these two flows is not fair at equilibrium.
- Exposed terminal.* Due to the coordination of RTS/CTS at the MAC layer, the sending nodes are able to communicate with each other to exchange necessary

load and topology information. Fig. 15 shows that the algorithm performs correctly in the exposed terminal scenario.

- Race condition.* The result under the race condition is shown in Fig. 16. We observe that the performance of our algorithm under the race condition is different from the hidden terminal scenario in its delayed convergence. This is because flow  $f_1$  is unable to capture the wireless channel at the beginning due to the unawareness of the RTS/CTS signals from the transmission of the other flow and the unfair backoff of IEEE 802.11. Once it gets the chance to transmit, the load change will be detected by flow  $f_2$  via bandwidth estimation, which in turn leads to the price increase of this contention region and the rate decrease of flow  $f_2$ . Via the communication between their receiving nodes, these two flows share the same view of the network condition and the price, thus converging to an equilibrium, where their transmission rates are the same. Although they have approximately the same throughput at equilibrium, a slight difference and fluctuation can still be observed on their instantaneous throughput, especially compared with the results in the hidden terminal scenario in Fig. 13. This is caused by the imprecise measurement under the race condition: IEEE 802.11 has short-term unfairness in its scheduling even when both flows send at the same rate, which is achievable. This problem is rooted at the MAC protocol and cannot be resolved by our algorithm. However, as we may observe from the

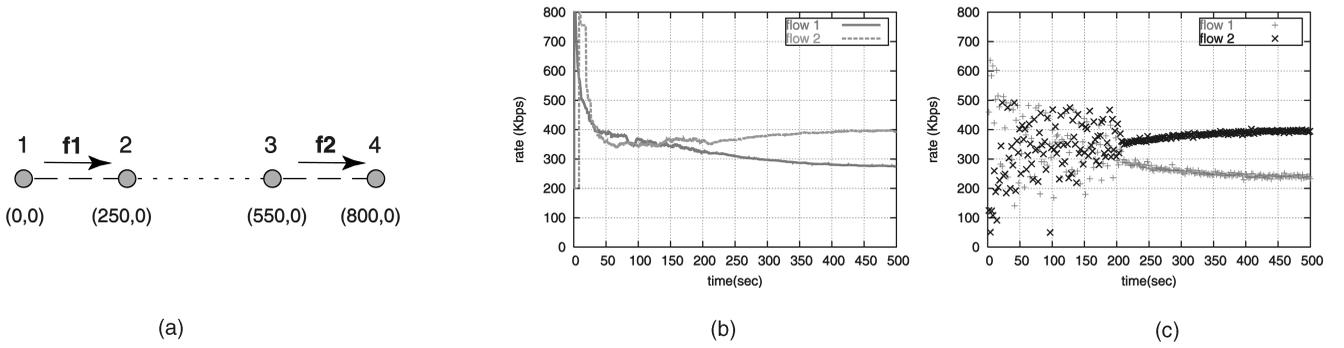


Fig. 14. Convergence in the hidden terminal scenario 2. (a) Hidden terminal topology. (b) Transmission rate. (c) Throughput.

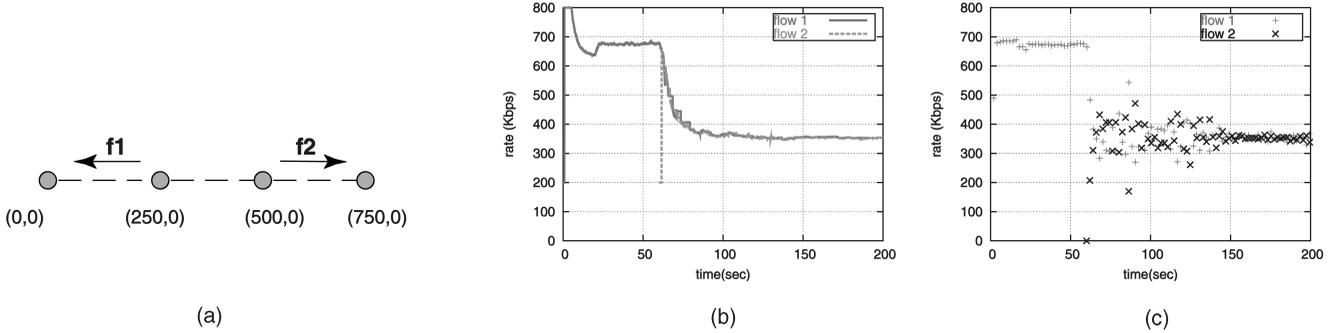


Fig. 15. Convergence in the exposed terminal scenario. (a) Exposed terminal topology. (b) Transmission rate. (c) Throughput.

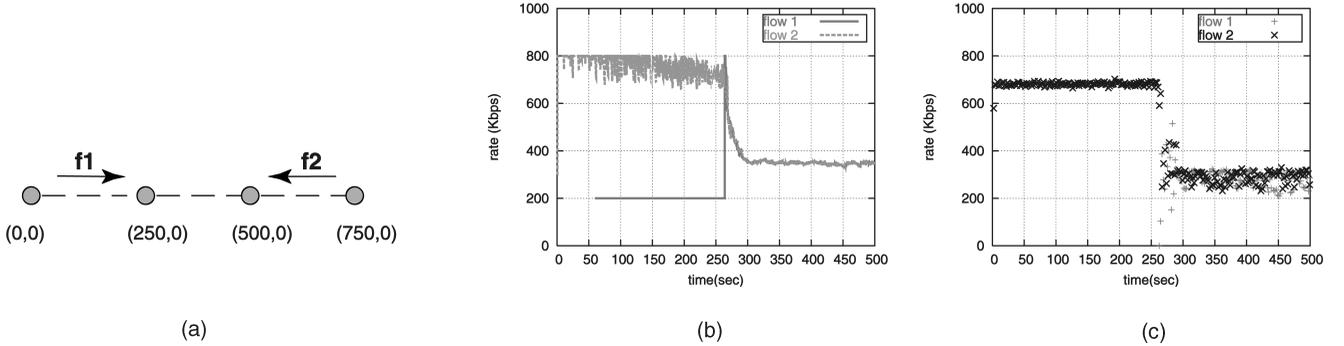


Fig. 16. Convergence in the race condition scenario. (a) Race condition topology. (b) Transmission rate. (c) Throughput.

results, the long term fairness can be guaranteed at the equilibrium via our price-based algorithm.

#### 6.4.2 Comparison Studies

To further illustrate the meaning of contention region and the impact of interference, we compare the equilibrium resource allocations of an ad hoc network with a wireline network of the same topology and two ad hoc networks with different interference ranges.

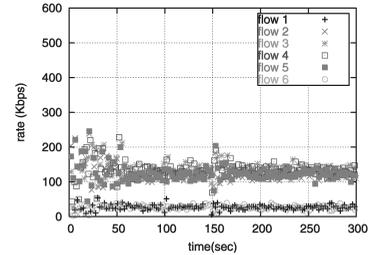
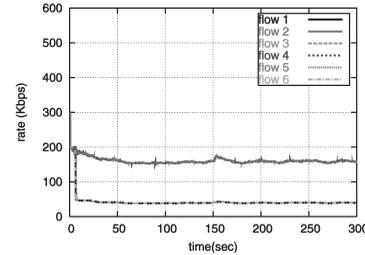
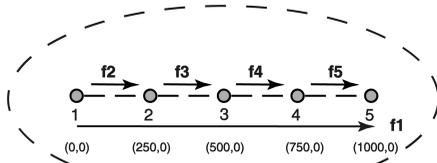
First, the rate allocation and the equilibrium prices of the wireline network and the wireless ad hoc network with a four-hop chain topology are compared under *synsim*. The cliques of the ad hoc network under *synsim* are the same as in Fig. 2. The results are listed in Table 3. From these results, we have the following observations: First, the rate allocated to each flow in the ad hoc network is less than the rate allocated to the corresponding flow in wireline networks. The difference lies in their different definitions of contention regions. In the wireline network, a wireline link represents a contention region, whose capacity is the link bandwidth. In the ad hoc network, a wireless link is no longer a contention

region. Instead, the set of wireless links, formally represented by a clique, constitutes the contention region and shares the channel capacity. Thus, with the same capacity of the wireless channel and the wireline link, the throughput of the ad hoc network is lower than that of the wireline network. Second, in the wireline network, the rates of all single-hop flows are the same. In the ad hoc network, the rates of these flows are different. The reason is that, in the wireline network, flows  $f_2$  through  $f_5$  enjoy the same amount of resources; while, in the ad hoc network, due to location-dependent contention,  $f_3$  suffers higher contention than  $f_2$ . This is also reflected through the prices that  $f_2$  and  $f_3$  need to pay. For  $f_2$ , the price is  $\lambda_2 = \mu_1$ , which equals to 1.25 at equilibrium, while the price for  $f_3$  is  $\lambda_3 = \mu_1 + \mu_2$ , which equals to 2.5. Third, in both networks, the equilibrium rate allocations for flows with different lengths are different. This is actually the result of proportional fairness. In particular, the longer the flow, the less the rate allocated. This observation is natural from the perspective of maximizing the aggregated utility. When the utility functions of all flows

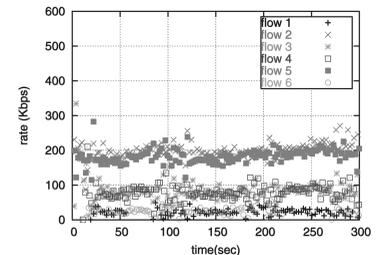
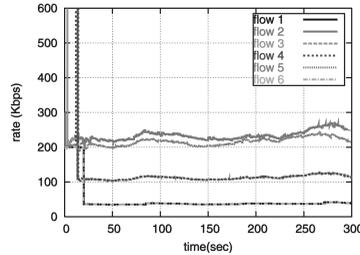
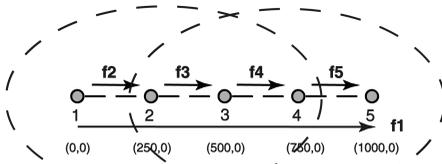
TABLE 3  
Rate Allocations and Equilibrium Prices in Different Networks

	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$\mu_1$	$\mu_2$	$\mu_3$	$\mu_4$
wireline network	0.4	1.6	1.6	1.6	1.6	0.625	0.625	0.625	0.625
ad hoc network	0.133	0.8	0.4	0.4	0.8	1.25	1.25	N/A	N/A

(I) interference range = 550 m , transmission range = 250 m



(II) interference range = transmission range = 250 m



(a)

(b)

(c)

Fig. 17. Comparison of different interference ranges. (a) Topology and clique. (b) Transmission rate. (c) Throughput.

are the same, long flows consume more resources for an unit of utility increase. Hence, the short flows are favored.

To further illustrate the impact of interference, we compare the resource allocation on two ad hoc networks with different interference ranges. The results are shown in Fig. 17b and Fig. 17c. We observe that the resource allocations are different for two networks. The reason behind this observation is that different interference ranges lead to different contention regions as shown in Fig. 17a. When the interference range is 550m, the network only consists of one contention region. On the other hand, when interference range is 250m, there are two overlapping contention regions in the network.

## 6.5 Overhead

Now, we evaluate the overhead of our algorithm under different mobility degrees. In the simulation, 30 mobile nodes are randomly deployed on a  $600 \times 600 m^2$  network. They move according to the random waypoint mobility model with an average node speed of 20m/s. The pause time interval is varied in the experiment. For each simulation, the results are averaged over 10 randomly generated mobile scenarios with the same pause time interval.

Fig. 18a plots the normalized overall packet overhead, which is the ratio between the number of nondata packets and the data packets delivered at each hop. The overall overhead includes the FEEDBACK packets sent by the receiver of each flow and the AODV routing packets, which

include HELLO packets that carry the price calculation information. We compare the overhead of our algorithm with the overhead of the TCP protocol<sup>2</sup> running over AODV. From the results, we observe that our algorithm has lower packet overhead than TCP. This is mainly because our price-based resource allocation algorithm generates fewer FEEDBACK packets than the ACK packets of TCP. Fig. 18b plots the ratio between the number of FEEDBACK packets sent at the receivers of the flows and the data packets that they receive. We observe that fewer FEEDBACK packets are generated with lower mobility. This is because, at the equilibrium where the price is unchanged, no FEEDBACK packet needs to be sent. Moreover, because the lightweight communication protocol uses packet piggybacking as its information delivery method, it does not introduce many additional control packets (AODV packets) at the routing layer into the network.

We proceed to study the overhead of our algorithm in bits per second. From Fig. 18d, we observe that the overhead of our algorithm in bits per second is also comparable to TCP over AODV, although our algorithm uses larger AODV HELLO packets.

To further reduce the overhead, we introduce a set of  $k$ -hop heuristic algorithms. In these heuristics, end nodes of subflow  $s$  cache the information within  $\mathbf{LS}^k(s)$  ( $k \leq 3$ ) and transmit information within  $\mathbf{LS}^{k'}(s)$  ( $k' = k - 1$ ) to their

2. TCP is also considered as a form of resource allocation for end-to-end flows in existing literature [4].

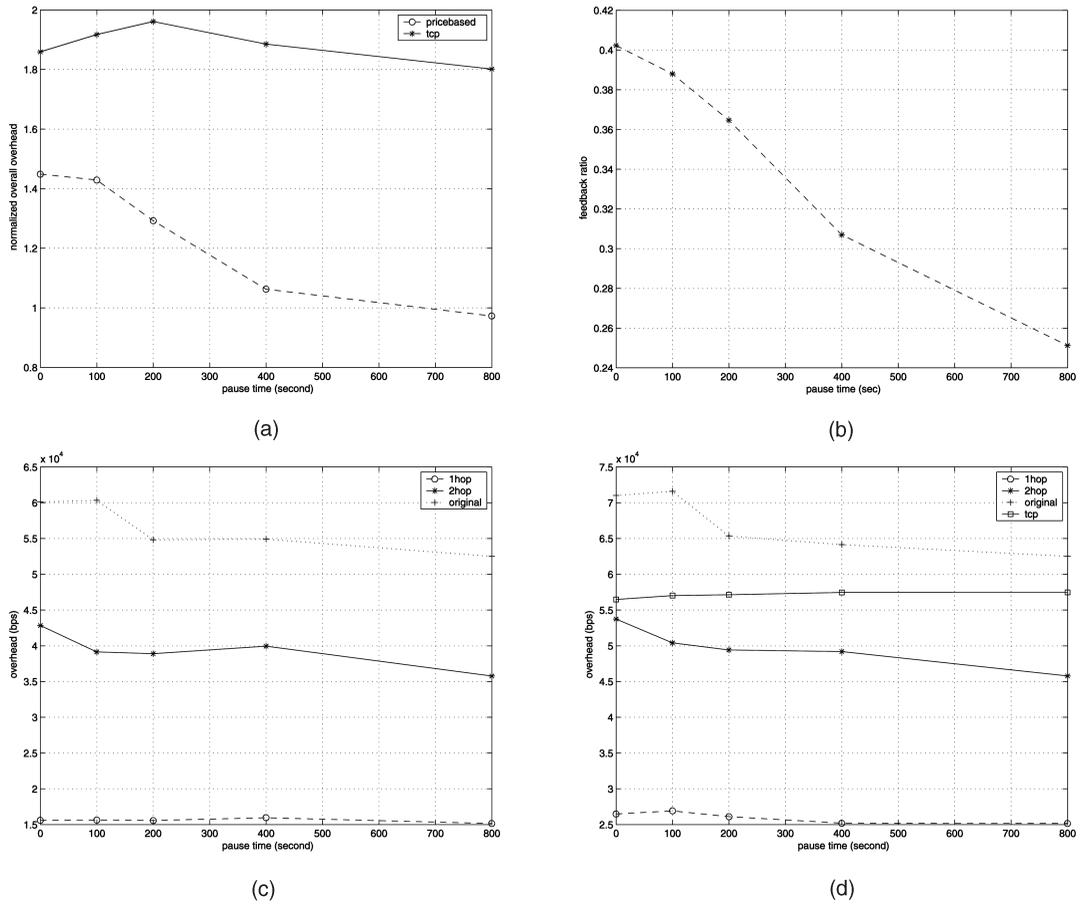


Fig. 18. Overhead. (a) Normalized overall packet overhead. (b) Normalized FEEDBACK packet overhead. (c) HELLO packet overhead in bits per second. (d) Overall overhead in bits per second.

neighbor nodes. Based on such partial knowledge of the network topology and load, cliques are constructed and their prices are computed approximately.

The result in Fig. 18c shows the overhead in bits per second incurred by HELLO packets when heuristics with different information propagation ranges are used. Fig. 18d compares the overall overhead in bits per second of the original price-based resource allocation algorithm with its heuristics. From the results, we observe that the two-hop heuristic has similar overhead as TCP over AODV, while one-hop heuristic has even smaller overhead in bits per second than TCP.

We further study the performance of these heuristics in terms of rate allocation. First, we study the aggregated utilities achieved by different heuristics and compare them with the original algorithm and TCP. In the experiment, we consider wireless ad hoc networks at two scales. At the smaller scale, the networks have 20 nodes deployed over a  $500 \times 500 m^2$ . At the larger scale, the networks have 30 nodes deployed over a  $600 \times 600 m^2$  region. For each network scale, 10 topologies are randomly generated. The original algorithm and its heuristics are simulated and compared with TCP over the same topology. The results are shown in Fig. 19a and Fig. 19b, respectively, corresponding to each network scale.

From these results, we have the following observations: First, the price-based resource allocation algorithm and its heuristics all outperform TCP in terms of aggregated utility.

In networks with a smaller scale, the performance of the heuristics closely matches the original algorithm. This observation is because, in networks with smaller scales, the hop counts between any two nodes are small. Thus, heuristics with smaller information propagation ranges are sufficient to communicate the information for clique construction and price calculation. In some topologies, one-hop heuristic can provide even better performance than the original algorithm due to its lower overhead. In networks with larger scales, the two-hop heuristic and the original algorithm give better performance than the one-hop heuristic owing to more precise clique construction. Moreover, two-hop heuristic can sometimes outperform the original algorithm due to its lower communication overhead.

Larger aggregated utility indicates more fair resource allocation and better resource utilization. To further understand the results in terms of aggregated utility in Fig. 19, we plot in Fig. 20 the throughput of each flow from the simulation result of one randomly generated  $600 \times 600 m^2$  network. The result clearly reflects the fairness improvement achieved by our price-based resource allocation algorithms in comparison with TCP.

## 6.6 Impact of Mobility

In this section, we study the behavior of our algorithm over mobile ad hoc networks. In particular, we seek to find the threshold of mobility where the convergence speed of the algorithm is not fast enough to ensure a sufficient portion of

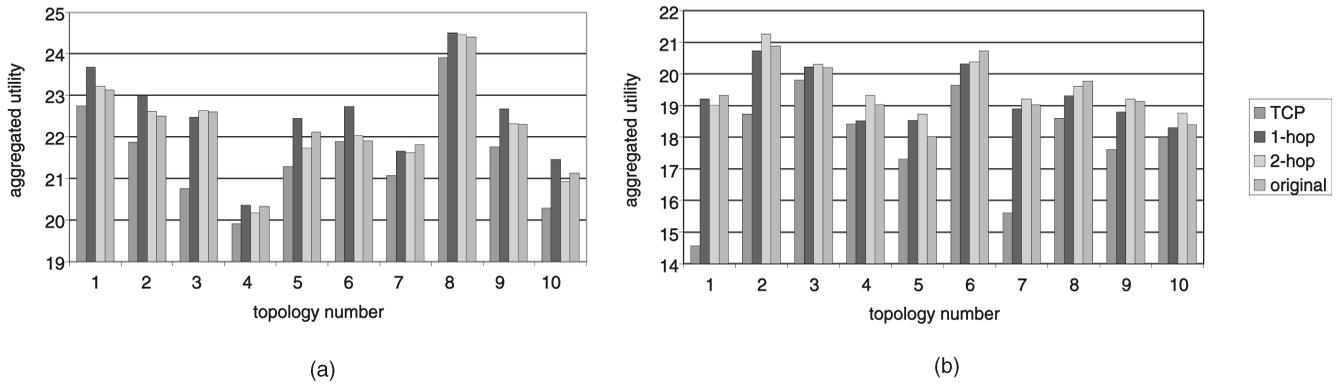


Fig. 19. Comparison of price-based resource algorithms and TCP over randomly generated networks. (a) 500m\*500m network. (b) 600m\*600m network.

time at equilibrium. We experiment on a specially designed mobile scenario, as shown in Fig. 21a. In this mobile scenario, node 1 is moving between location (150, 1000) and (150, 500) observing the random waypoint model, and node 2 is moving between (150, 500) and (150, 0). During the simulation, these two nodes will serve as the relaying nodes for flow  $f_1$  in an interwoven fashion. The convergence behavior and the throughput under different node speeds and pause time intervals are plotted in Fig. 21. From the figure, we can observe that the new convergence occurs after broken routes are reestablished. The results show that the algorithm converges and stays at equilibrium for a large portion of time when the node moves at 10m/s without pause. When the node speed increases to 20m/s, the flow spends approximately the same amount of time during the convergence and at the equilibrium. Further increasing the node speed under this scenario may result in insufficient amount of equilibrium time where resource is optimally allocated. Obviously, if the pause time interval is increased, the algorithm can support higher node speeds. This is illustrated in the figure, when node speeds are 50m/s with the pause time interval as 20 seconds.

To study the impact of mobility on our algorithm over random networks, we organized the simulated network scenarios into mobility patterns. Each mobility pattern, generated randomly, specifies a sequence of movements. Within each mobility pattern, the mobility index specifies the average node speed and pause time of each mobile scenario. For example, under one mobility pattern, if, with index 1, a node takes time  $t$  to move from location  $a$  to location  $b$ , then, with index 2, this node will take  $2 \times t$  to traverse this distance. Our experiment ranges from mobility index 1, which

corresponds to an average node speed of 100m/s and pause time interval of 10 seconds, to mobility index 6, which corresponds to an average node speed of 16.67m/s and pause time interval of 60 seconds. Fig. 22a plots the aggregated utility of our algorithm with varied mobility indices under four different mobility patterns. To better understand the performance indicated by the aggregated utility, Fig. 22b plots the throughput of each flow under pattern 1 with varied mobility indices. From these results, we observe that the difference from mobility index 4, which corresponds to an average node speed of 25m/s and pause time interval of 40 seconds, to mobility index 6 is quite small. Moreover, even in highly mobile environments such as the ones indicated by mobility indices smaller than 3, the performance of our algorithm still degrades reasonably with the increased mobility.

## 7 RELATED WORK

We evaluate and highlight our original contributions in light of previous related work.

The problem of optimal and fair resource allocation has been extensively studied in the context of wireline networks. Among these works, pricing has been shown to be an effective approach to achieve distributed solution for rate allocation (e.g., [4], [5], [14]). The role of price in our work is similar to [4], [5], which reflects the relation of the demand and the supply of resources. Nevertheless, the fundamental differences in contention models between ad hoc and wireline networks deserve a fresh treatment to this topic. As we have emphasized, these resource allocation strategies employed in the wireline network may not be applied directly in the context of ad hoc networks due to the unique characteristics of the shared wireless channel.

A collection of papers have studied the use of pricing in the context of wireless networks (e.g., [15], [16]). In these works, pricing has been used as a mechanism for optimal distributed power control. In comparison, our work is toward different objectives and in different wireless environments. For example, we study rate allocation in multihop wireless networks with time-slotted MAC, while most of the work in this group study base-station-based single-hop wireless networks with CDMA. In addition, Liao et al. [17] use price as an incentive for service class allocation in wireless LAN. Their solution, however, is applicable in scenarios where centralized management is readily available.

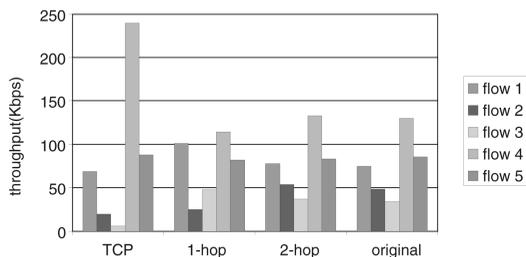


Fig. 20. Throughput comparison over a randomly generated network with 30 nodes deployed over a  $600 \times 600 m^2$  region.

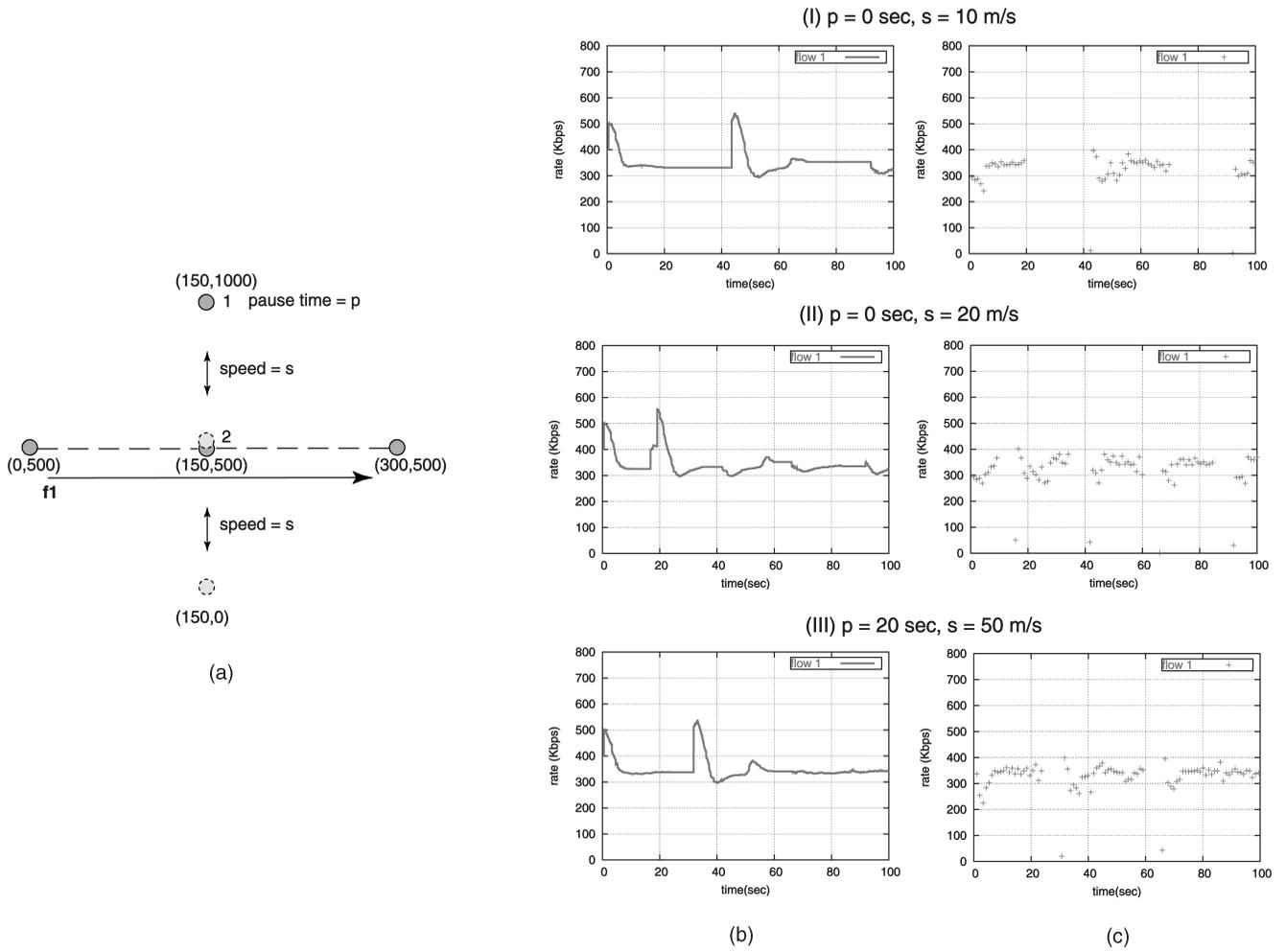


Fig. 21. Impact of node mobility. (a) Mobile scenario. (b) Transmission rate. (c) Throughput.

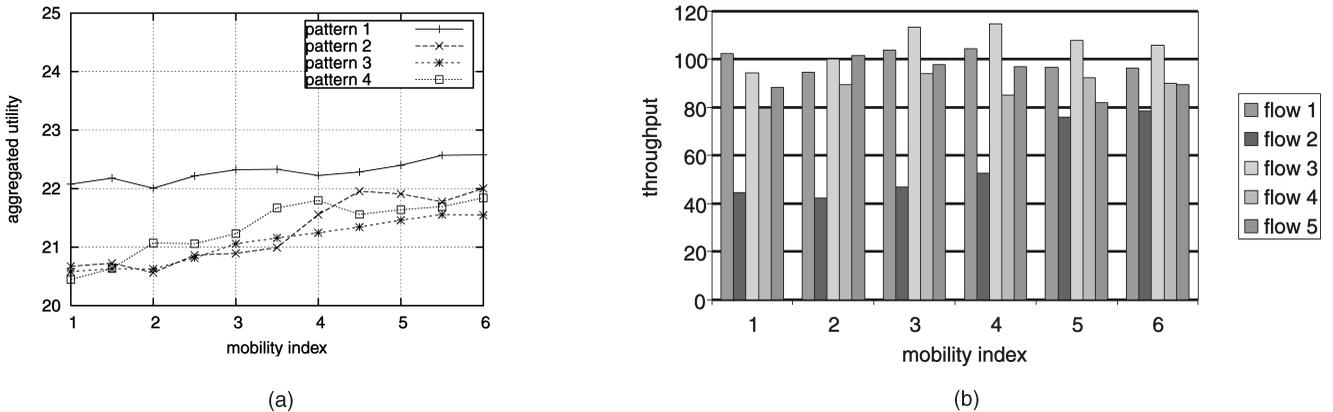


Fig. 22. Impact of node mobility. (a) Aggregated utility under different patterns. (b) Throughput of each flow under pattern 1.

There also exists work to use pricing as incentives to encourage packet relays in wireless ad hoc networks (e.g., [18], [19]). Our work is fundamentally different from these results in the following aspects. First, in [18], a simplified wireless ad hoc network model is used, where each node  $i$  in the network has a capacity of  $C_i$ , which is independent from other nodes. We have shown that such a network model is not able to correctly characterize the unique characteristics of location dependent contention in ad hoc networks. Second, in [18], [19], a user is assumed to have limited transmission

resources and the role of pricing is to provide adequate user incentives to forward packets for other users. The goal of optimal price setting at each node is to maximize its net benefit. In contrast, the role of pricing in our work is to regulate channel access and to provide globally optimal resource allocation in the sense of maximizing aggregated utility.

Resource allocation, using MAC-layer fair scheduling for single-hop MAC layer flows, has also been studied in wireless ad hoc networks [1], [2], [3]. In comparison, we address end-to-end multihop flows. It can be shown that

fair resource allocation among single-hop flows may not be optimal for multihop flows due to the unawareness of bottlenecks and lack of coordination among upstream and downstream hops. Moreover, global optimal resource allocation among multihop flows cannot be completely reached only by MAC-layer scheduling, which is only based on local information. Price is needed as a signal to coordinate the global resource allocation. Finally, we argue that our proposed solution for end-to-end flows is complementary to any MAC-layer solutions and can be implemented based on them.

## 8 CONCLUDING REMARKS

In this paper, we have presented a novel price-based resource allocation algorithm based on an analytical pricing model that is specifically designed for the unique characteristics of multihop wireless ad hoc networks. The original contribution incorporated in the pricing model is the association of shadow prices with the *maximal cliques* in the contention graph model, rather than with individual links as in wireline networks. Based on insights brought forth by such strategies, the algorithms proposed are fully distributed and arbitrate the contention among end-to-end *multihop* flows with respect to fair resource allocation. The validity of our claims is supported by both theoretical studies and extensive simulation results. To the best of our knowledge, there does not exist any previous work that addresses the problem of enforcing fairness among multihop flows in wireless ad hoc networks, especially when a price-based approach is utilized to design fully distributed algorithms to achieve this goal.

## ACKNOWLEDGMENTS

This research was supported by the US Office of Naval Research MURI NAVY CU 37515-6281 grant and Vodafone Fellowship. Any opinions, findings, and conclusions are those of the authors and do not necessarily reflect the views of the above agencies.

## REFERENCES

- [1] T. Nandagopal, T.-E. Kim, X. Gao, and V. Bharghavan, "Achieving MAC Layer Fairness in Wireless Packet Networks," *Proc. ACM Mobicom Conf.*, pp. 87-98, 2000.
- [2] L. Tassiulas and S. Sarkar, "Maxmin Fair Scheduling in Wireless Networks," *Proc. INFOCOM Conf.*, pp. 763-772, 2002.
- [3] H. Luo, S. Lu, and V. Bharghavan, "A New Model for Packet Scheduling in Multihop Wireless Networks," *Proc. ACM Mobicom Conf.*, pp. 76-86, 2000.
- [4] F.P. Kelly, A.K. Maulloo, and D.K.H. Tan, "Rate Control in Communication Networks: Shadow Prices, Proportional Fairness and Stability," *J. Operational Research Soc.*, vol. 49, pp. 237-252, 1998.
- [5] S.H. Low and D.E. Lapsley, "Optimization Flow Control: Basic Algorithm and Convergence," *IEEE/ACM Trans. Networking*, vol. 7, no. 6, pp. 861-874, 1999.
- [6] F.P. Kelly, "Charging and Rate Control for Elastic Traffic," *European Trans. Telecomm.*, vol. 8, pp. 33-37, 1997.
- [7] P. Gupta and P.R. Kumar, "The Capacity of Wireless Networks," *IEEE Trans. Information Theory*, vol. 46, no. 2, pp. 388-404, Mar. 2000.
- [8] D. Bertsekas, *Nonlinear Programming*. Athena Scientific, 1999.
- [9] Y. Xue, B. Li, and K. Nahrstedt, "Optimal Resource Allocation in Wireless Ad Hoc Networks: A Price-Based Approach," technical report UIUCDCS-R-2004-2505, Univ. of Illinois at Urbana-Champaign, Dec. 2004.
- [10] J.G. Augustson and J. Minker, "An Analysis of Some Graph Theoretical Cluster Techniques," *J. ACM*, vol. 17, no. 4, pp. 571-586, 1970.

- [11] D. Bertsekas and J. Tsitsiklis, *Parallel and Distributed Computation*. Prentice-Hall, 1989.
- [12] C. Perkins and E. Royer, "Ad-Hoc On-Demand Distance Vector Routing," *Proc. Second IEEE Workshop Mobile Computing Systems and Applications (WMCSA '99)*, 1999.
- [13] S.H. Shah, K. Chen, and K. Nahrstedt, "Dynamic Bandwidth Management for Single-Hop Ad Hoc Wireless Networks," *ACM/Kluwer Mobile Networks and Applications (MONET)*, vol. 10, no. 1, 2005.
- [14] R.J. La and V. Anantharam, "Utility-Based Rate Control in the Internet for Elastic Traffic," *IEEE/ACM Trans. Networking*, vol. 10, no. 2, pp. 272-286, 2002.
- [15] T.M. Heikkinen, "On Congestion Pricing in a Wireless Network," *Wireless Networks*, vol. 8, no. 4, pp. 347-354, 2002.
- [16] D. Julian, M. Chiang, D. O'Neill, and S. Boyd, "QoS and Fairness Constrained Convex Optimization of Resource Allocation for Wireless Cellular and Ad Hoc Networks," *Proc. IEEE INFOCOM Conf.*, pp. 477-486, June 2002.
- [17] R. Liao, R. Wouhaybi, and A. Campbell, "Incentive Engineering in Wireless LAN Based Access Networks," *Proc. 10th IEEE Int'l Conf. Network Protocols (ICNP)*, Nov. 2002.
- [18] Y. Qiu and P. Marbach, "Bandwidth Allocation in Ad-Hoc Networks: A Price-Based Approach," *Proc. INFOCOM Conf.*, 2003.
- [19] L. Buttyan and J.P. Hubaux, "Stimulating Cooperation in Self-Organizing Mobile Ad Hoc Networks," *ACM/Kluwer Mobile Networks and Applications (MONET)*, vol. 8, no. 5, Oct. 2003.



Yuan Xue received the BS degree in Computer Science from the Harbin Institute of Technology, China, in 1998 and the MS and PhD degrees in Computer Science from the University of Illinois at Urbana-Champaign in 2002 and 2005. Currently, she is an assistant professor in the Department of Electrical Engineering and Computer Science at Vanderbilt University. She is a recipient of the Vodafone fellowship. Her research interests include wireless and sensor networks, QoS support, and network security. She is a member of the IEEE.



Baochun Li received the BEng degree in 1995 from the Department of Computer Science and Technology, Tsinghua University, China, and the MS and PhD degrees in 1997 and 2000 from the Department of Computer Science, University of Illinois at Urbana-Champaign. Since 2000, he has been with the Department of Electrical and Computer Engineering at the University of Toronto, where he is currently an associate professor. He has held the Nortel Networks Junior Chair in Network Architecture and Services since October 2003 and the Bell University Laboratories Endowed Chair in Computer Engineering since August 2005. In 2000, he was the recipient of the IEEE Communications Society Leonard G. Abraham Award in the Field of Communications Systems. His research interests include application-level Quality of Service provisioning, wireless, and overlay networks. He is a senior member of IEEE and a member of ACM.



Klara Nahrstedt received the AB and MSc degrees in mathematics from Humboldt University, Berlin, Germany, and the PhD degree in computer science from the University of Pennsylvania. She is a professor in the Computer Science Department at the University of Illinois at Urbana-Champaign, where she does research on Quality of Service (QoS) aware systems with an emphasis on end-to-end resource management, routing, and middleware issues for distributed multimedia systems. She is the coauthor of the widely used multimedia book *Multimedia: Computing, Communications and Applications* (Prentice Hall) and the recipient of the Early US National Science Foundation Career Award, the Junior Xerox Award, the IEEE Communication Society Leonard Abraham Award for Research Achievements, and the Ralph and Catherine Fisher Professorship Chair. Since June 2001, she has served as the editor-in-chief of the *ACM/Springer Multimedia Systems Journal*. She is a member of the IEEE.