Fair Bandwidth Sharing Algorithms based on Game Theory Frameworks for Wireless Ad-hoc Networks

Zuyuan Fang and Brahim Bensaou
Computer Science Department
The Hong Kong University of Science and Technology
Clear Water Bay, Hong Kong, PRC.
E-mail: {zyfang,brahim}@cs.ust.hk

Abstract— **This paper examines the theoretical aspects of bandwidth sharing in wireless, possibly mobile, ad-hoc networks (MANETs) through a game theoretic framework. It presents some applications to show how such a framework can be invoked to design efficient media access control protocols in a non-cooperative, self-organized, topology-blind environment as well as in environments where the competing nodes share some basic information to guide their choice of channel access policies. For this purpose, contentions between concurrent links in a MANET are represented by a conflict graph, and each maximal clique in the graph defines a contention context which in turn imposes a constraint on the share of bandwidth that the links in the clique can obtain. Using this approach the fair bandwidth allocation problem is modeled as a general utility based constrained maximization problem, called the system problem, which is shown to admit a unique solution that can only be obtained when global coordination between all links is possible. By using Lagrange relaxation and duality theory, both a non-cooperative and a cooperative game formulation of the problem are derived. The corresponding mathematical algorithms to solve the two games are also provided where there is no need for global information. Implementation issues of the algorithms are also considered. Finally, simulation results are presented to illustrate the effectiveness of the algorithms.**

Index Terms— **IEEE 802.11, Ad-hoc networks, Medium access control, Fairness, Backoff procedure, Game Theory.**

I. INTRODUCTION

Due to their ease of deployment, and a foreseeable wide range of both commercial and military applications, mobile ad-hoc networks (MANETs) are under extensive investigation in the research community. A MANET is a network where there are no centralized control units, and where any node can serve as both a host as well as a router. MANETs are usually self-organized and self-administrated networks where the nodes can move about arbitrarily. In a nutshell, the topology of MANETs is dynamic, and their architecture and control are usually totally distributed, which gives rise to many challenges in the design and implementation of protocols to operate such networks. At the base of these problems lies the fundamental problem of providing an efficient bandwidth sharing mechanism between the potentially competing nodes. Today, the so-called distributed coordination function (DCF) introduced in the wireless LAN standard IEEE 802.11 has gained wide popularity and use in most performance studies of routing protocols for MANETs. However, the DCF has been shown to suffer from the fairness problem, caused by the existence of hidden terminals and exacerbated by the adopted binary exponential backoff algorithm to resolve contention. Many schemes [1–6] have been proposed in the literature to overcome this problem and provide better bandwidth sharing between competing nodes. Some of these schemes [3–5] depend on the sharing of link information between nodes in the network, while others (e.g., [1], [6]) try to solve the issue of fairness by maintaining topology-transparency. In addition, some (e.g., [6]) are based on fully fledged scheduling to provide short term guarantees, while others use random access techniques with collision resolution to achieve long term guarantees.

In MANETs, the link level bandwidth plays an important role in QoS provisioning for end-to-end flows. If an end-to-end flow crosses several hops in the link layer, then the bandwidth that can be assigned to such flow is determined by the capacity of the bottleneck link. Traditionally, in order to provide QoS routing and be able to perform tasks such as admission control, an end-to-end flow's requested bandwidth is checked against the link layer bandwidth hop-by-hop to find a feasible and admissible path. Therefore, QoS routing relies on the ability of the system in *quantifying link layer bandwidth*. While this was not a major problem in traditional networks, it becomes challenging problem in MANETs due to the volatile nature of the network topology, and as a consequence to the variable capacity of link layer bandwidth. To illustrate this, consider the example in Fig. 1, and assume node F decides to establish an end-to-end flow to node D. Initially link 5 is not established since there is no traffic between node F and any of the other nodes. If this flow is to be established, the nodes need not only check the flow's required bandwidth against the available bandwidth on links 5, 2 and 3, but also whether establishing link 5 is possible at all: due to the nature of random access techniques such as CSMA/CA, link 5 would steal bandwidth from all its surrounding links and would eventually lead to violation of previously committed QoS guarantees on these links.

QoS routing algorithms for MANETs proposed in the literature (e.g., [7], [8]) sometimes directly use bandwidth as the metric to achieve QoS routing, and assume the link layer is

The work of Mr Zuyuan Fang was in part supported under Grant number AoE/E-01/99

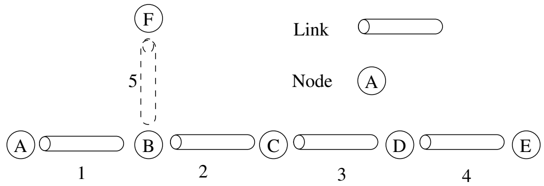


Fig. 1. Example of link layer wireless network

capable of providing such bandwidth without considering the complexity of these assumptions. In addition, QoS support frameworks and differentiated services frameworks such as INSIGNIA [9] also utilize hop-by-hop link layer bandwidth to check feasibility of routes and to reserve resources along the paths. Todays MAC schemes for MANETs are not capable of providing QoS. Therefore, it is very important to design techniques and tools to study the effects of bandwidth sharing principles on the QoS. Such tools will help guide the design of media access techniques that provide a certain level of link layer bandwidth guarantees.

The objective of link layer fairness schemes is to improve the bandwidth of links that face more drastic competition than some of their competitors. More specifically, given a fair bandwidth sharing principle, such as max-min fairness, proportional fairness and so on, link layer fairness is concerned with designing the mechanisms that share physical layer bandwidth among links optimally, where optimality is defined by the bandwidth sharing principle under consideration. It is important here to draw the readers' attention to the difference between this approach (link layer bandwidth sharing) and classic bandwidth sharing problems studied in the context of Internet communications using TCP. In the latter problems, link bandwidth is well defined and links are disjoint resources, in the former (i.e. wireless networks) physical layer bandwidth is spatially continuous. Therefore the constraints of the problem in wireless networks are more complex than those in their wired counterpart. In this paper we will concentrate our efforts on studying the problem of sharing physical layer bandwidth between links rather than sharing link layer bandwidth between end-to-end flows. Once the former problem is solved, this latter problem becomes the same as its counterpart in wired networks, which was studied extensively in the context of the Internet, as the flow control problem [10–12], the congestion control problem [13], [14] and the resource allocation problem [15]. Therefore, in the sequel, unless otherwise stated, when we refer to flows, links or link-flows, we are mainly concerned with link layer flows: that is one hop traffic flow between two nodes identified by the pair sender-receiver. For example in the sequel "link contention" and "flow contention" will bear the same meaning.

In this paper we will model the contention relations between link flows as a flow contention graph and based on the definition of a *feasible frequency vector*, the definition of *normalized clique capacity* is introduced and its property in a general flow contention graph is discussed. Based on this analysis, the fair bandwidth allocation problem is modeled as a concave utility maximization problem. With the help of classic tools provided by the well developed convex optimization theory, a non-cooperative game framework and a cooperative game framework are developed, and distributed algorithms are also derived from these frameworks. Existence and uniqueness of the equilibria reached by the games are demonstrated. In short, this paper proposes a unified framework for developing non-cooperative or cooperative distributed algorithms based on the problem formulation. The algorithms that can be derived from the non-cooperative game framework do not need to share local flow information between nodes, and each flow monitors the channel status by itself to decide the course of action. The algorithm derived from the cooperative game framework only needs flow information sharing within one hop. System wide fairness can be achieved under the control of these algorithms.

The remainder of the paper is organized as follows: Section II discusses the capacity of the flow contention graph and introduces the model for the bandwidth allocation problem. Section III introduces a non-cooperative game framework for the system, proves the existence of a unique equilibrium for the game, and proposes a distributed algorithm based on the framework. Section IV discusses the algorithm derived from solving the dual problem. Section V gives some simulation results for both the non-cooperative and cooperative game frameworks and numerical results for the cooperative game framework. Finally, Section VI concludes the paper.

II. Model and Problem Formulation

In general, when studying the problem of fair bandwidth allocation two approaches are available: the first one consists of finding a schedule that achieves a given fairness principle (without knowing the capacity of the network) and the throughput achieved by such a schedule is taken as the performance metric to measure how good the scheduling is; the second approach consists of starting by estimating the bandwidth then sharing this bandwidth between the nodes subject to fairness constraints. In this paper we adopt the latter approach to formulate the problem however when solving the problem we revert to the former approach. Therefore in this section, we first discuss the issues of network bandwidth estimation, then based on this discussion, formulate the fair bandwidth sharing problem in MANETs as a constrained maximization problem.

A. Flow Contention Graph and Clique Capacity

In this paper, mobile nodes are assumed to possess only one transceiver and cannot send and receive simultaneously. Nodes mutually interfere with each other. Under these assumptions, a MANET can be modeled as an un-directed graph, in which the vertices represent the stations in the network, and the edges between vertices represent data flows between these two stations (i.e., an established link). From this so-called node graph, we can construct another graph that captures all the contention relations between the links of the network. In this flow contention graph, a vertex represents an active flow/link,

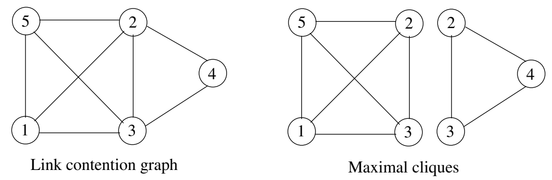


Fig. 2. Link contention graph and its decomposition in maximal cliques

and an edge between two vertices denotes wireless proximity between two links: two links contend with each other when either the sender or the receiver of one is within interference range of the sender or the receiver of the other. In this paper we do not take into account capture effect. An accurate flow contention graph could be constructed based on the SIR model as proposed in [16], which may be very difficult in reality. In practice, if we want to construct the flow contention graph, we can assume two nodes contend with each other if they are within each other's carrier sense range.

Fig. 2 shows the link-flow contention graph that corresponds to the node graph of Fig. 1. In a flow contention graph, flows in the same maximal clique cannot transmit simultaneously. For example, in Fig. 2, flows 1, 2, 3, and 5 are in the same clique, and cannot transmit simultaneously. For the same reason, flows 2, 3 and 4 cannot transmit simultaneously. However, flows 4 and 5 can be activated at the same time, since they belong to different cliques. To capture these constraints on simultaneous transmissions, a flow contention graph can be decomposed into a set of maximal cliques. Each clique stands for a contention context, therefore it can be treated as a "channel resource" [1], [17]. Flows in the same clique share the capacity of the clique. A flow may belong to several cliques. The principle is that one flow can succeed in transmission if and only if all flows that share at least one clique with this flow do not transmit. Examining the problem from another viewpoint, the flows that form an independent set of a flow contention graph can transmit simultaneously. For flow contention graph G, let $\mathcal{I}$ be the family of all independent sets of this graph. Schedule S can be defined as an infinite sequence of independent sets, $I_1, I_2, \cdots, I_k, \cdots$, where $I_k \in \mathcal{I}$. The frequency of flow i in schedule S is then defined as [18]:

$$f_i = lim_{t\to\infty}\frac{\sum_{k=1}^{t} S(i,k)}{t},$$

where $S(i,k)$ is an indicator function such that, $S(i,k)=1$ if $i \in I_k$, and $S(i,k)=0$ otherwise. For a flow contention graph with N flows, a vector of frequencies $\hat{f} = (f_1, \cdots, f_N)$ is feasible if there exists a schedule S such that the frequency of the ith flow in schedule S is at least f_i [17]. The frequency of flow i can be treated as the normalized bandwidth allocated to this flow in schedule S. Based on the definition of feasibility of a frequency vector, we can introduce the definition of normalized clique capacity as follows:

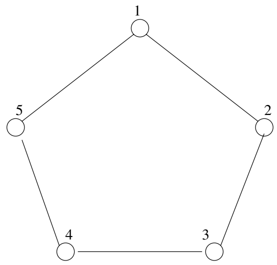
Definition 2.1: For a feasible vector of frequencies, the sum of all the fair shares (frequencies) allocated to the flows in one clique is defined as the normalized capacity of the clique.

However, given a vector of frequencies $\hat{f}$, verifying its feasibility is a hard task. Since, according to the definition of feasibility previously, we need to find a schedule that achieves $\hat{f}$. It turns out that for a perfect graph [1], the definition of clique feasibility is sufficient to justify the feasibility of the frequency vector. A vector of frequencies $(f_1, \cdots, f_N)$ is *clique feasible* for flow contention graph G if $\sum_{i\in C} f_i \le 1$ for all cliques C in graph G. This definition shows that for a perfect flow contention graph, the capacity of a clique can be normalized to 1. However, this is not true for general graphs (including non-perfect graphs). For example, in Fig. 3, if the capacity for each clique is normalized to 1, then for the ring graph of size 5 (a non-perfect graph), each flow should obtain a capacity of $\frac{1}{2}$ according to the max-min fairness allocation criterion [3]. However, scheduling the links according to a max-min schedule on the ring allocates only a capacity of $\frac{2}{5}$ to each link. Therefore clique feasibility is only valid for perfect graphs.

For a general flow contention graph, Theorem 5, proved in [17], states a general definition of feasibility (*graph feasibility*) of a frequency vector. The theorem states that a feasible vector of frequencies must be a convex combination of the characteristic vectors of all independent sets of the flow contention graph. This theorem shows that the set of all feasible frequencies is a closed, convex and compact set. In addition, it is shown in [17] that *clique feasibility* is equivalent to *graph feasibility* if and only if the graph is perfect. According to the *Strong Perfect Graph Conjecture* stated by Claude Berge in the 60s, and which has been recently proved as a theorem by Chudnovsty [19], a graph is perfect if and only if it has no induced subgraph that is isomorphic to an odd cycle of length at least 5 without chords (in graph theory terminology, an odd hole), or the complement of such a cycle. Therefore if there are odd holes in a flow contention graph, the capacity of any clique that includes edges of an odd hole (or the components that can be reduced to an odd hole) should be reduced (i.e., cannot be normalized to 1). Let us take the contention graph in Fig. 4 as an example, the graph is non-perfect as it contains an odd hole of size 5. The maximum clique size is 3 and all vertices belong to cliques of degree 3. Therefore, clique feasibility states that each node is assigned a normalized capacity of $\frac{1}{3}$. However, we can show that no vertex can obtain a normalized capacity of $\frac{1}{3}$ under the max-min allocation, despite the fact that no node is part of a maximal clique of size more than 3. The max-min fairness schedule in Fig. 4(b) shows that the capacity of each maximal clique should be reduced to $\frac{6}{7}$.

In general, it is very hard to tell whether a graph is perfect or not. Moreover, one can easily prove that global topology information is required to perform such classification (an odd hole can span the whole network). A practical algorithm for ad hoc networks should depend at most on local information up to a few hops. Hence, when we need to know the capacity of a clique, we do not try to justify whether a graph is perfect or not. A systematic reduction of the capacity of

[1]A graph is perfect if for all its induced subgraphs the size of the maximum clique is equal to the chromatic number.

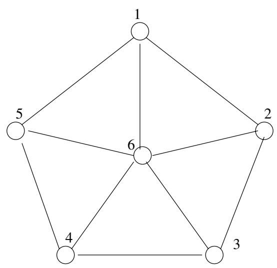


(a) Contention graph

Rounds \ Flow	1	2	3	4	5
1	X		X		
2		X		X	
3			X		X
4	X			X	
5		X			X

(b) Max-min fair schedule

Fig. 3. Odd hole and its max-min fair schedule



(a) Flow contention graph

Rounds \ Flow	1	2	3	4	5	6
1	X		X			
2		X		X		
3			X		X	
4	X			X		
5		X			X	
6						X
7						X

(b) Max-min fair schedule

Fig. 4. Imperfect graph and its max-min schedule

all cliques leads to a tight lower bound on the frequencies, guarantees a minimum throughput to each link, and has the virtue of requiring very low complexity. On the other hand the throughput of the network would be reduced by up to 1/3 in the worst case. Determining exactly by how much we should reduce the capacity is out of the scope of this paper. Notwithstanding this, using an early result of Claude Shannon on edge colorability in graphs [20] cited also in [18] one can show that if the capacity is reduced to 2/3 then clique feasibility is sufficient to ensure feasibility in general graphs without having to check the existence of odd holes. This bound however becomes loose as the degree of the graph increases (for example a clique of 101 vertices is colorable with 101 colors while the bound gives 150). As we will see later we do not need to know the exact capacity of the cliques when we adopt the non-cooperative game theoretic approach to solve the problem.

B. Problem Formulation

We can formulate the bandwidth allocation problem in MANETs based on the flow contention graph. For flow contention graph G, assume the number of flows in the graph is N. The set of flows is denoted as $\mathcal{N}=\{1, \cdots, N\}$. The rate for flow i is defined as $x_i, i = 1, \cdots, N$. The set of maximal cliques in G is denoted as $\mathcal{M}=\{1, \cdots, M\}$. The capacity of clique j is defined as $c_j, j \in \mathcal{M}$. One flow may belong to several maximal cliques. These relations of belonging can be described by matrix A as follows:

$$a_{j,i} = \begin{cases} 1, & \text{if flow } i \text{ belongs to clique } j \ , \ i \in \mathcal{N} \\ 0, & \text{if flow } i \text{ does not belong to clique } j \ , \ j \in \mathcal{M} \end{cases}$$

The capacity constraints of the flows can therefore be defined as:

$$\boldsymbol{Ax} \leq \boldsymbol{C} \tag{1}$$

where $\boldsymbol{x} = (x_1, \cdots, x_N)$ is the flow rate column vector and $\boldsymbol{C} = (c_1, \cdots, c_M)$ is the clique capacity vector.

In addition, flow rates must take non-negative values:

$$x_i \geq 0, \qquad i = 1, \cdots, N \tag{2}$$

The set of flow rate vectors Ω that satisfy conditions (1) and (2) is called a feasible set. The objective of the resource allocation is to find a feasible flow rate vector that satisfies some performance requirement, such as the fairness property. According to [14], the concavity of a utility function can guarantee fairness: if $x^1 > x^2$ and $\phi(x)$ is a strictly concave function, then we have $\phi'(x^2) > \phi'(x^1)$. When the system wants to maximize the value of $\phi(x)$, a small value of x will be favored. Therefore, if a strictly concave utility function of the flow rates is defined and maximized, then the flow with the smaller rate will be favored, which implies fairness. Following this principle, defining a strictly concave utility function $f_i(x_i)$ for each flow, the objective of fair resource allocation in a MANET can be described as:

$$\text{maximize} \qquad \sum_{i=1}^{N} w_i f_i(x_i) \tag{3}$$

where $w_i(> 0)$ is a positive constant. The objective of introducing $w_i(> 0)$ is to provide weighted fairness or service differentiation. In general, each flow can use a different utility function. However, if the flows choose the same utility function, then w_i can be treated as the user's willingness to pay [10]. According to (3), the flow with the larger value w_i will be favored, thus it can be used to achieve service differentiation. In other words, this results in a weighted fairness, where w_i serves as the weight of flow i.

Equations (1),(2) and (3) form the system model of the resource allocation problem in MANETs.

C. System Problem

Based on the above analysis, if we assume $Q(i)$ to be the set of cliques that include flow i, and $S(j)$ be the set of flows that form clique j, the resource allocation problem can be formulated as a constrained maximization problem, which is called system problem (primal problem) $\boldsymbol{P}$ defined as follows:

$$\begin{aligned} \boldsymbol{P}: \qquad & \max_{x_i} \qquad \sum_i w_i f_i(x_i) \\ & \text{subject to:} \quad \sum_{i \in S(j)} x_i \leq c_j, \qquad j = 1, \cdots, M \\ & \qquad\qquad\qquad x_i \geq 0, \qquad i = 1, \cdots, N \end{aligned} \tag{4}$$

To analyze this problem, let us first consider the feasible set Ω. Since the constraints are linear inequalities, and flow rates are non-negative and upper bounded by the capacity of the clique, it can be shown that Ω is a nonempty, convex and compact set. This is consistent with Theorem 5 proved in [17], which confirms the correctness of the modeling. Since the feasible set Ω is convex and compact, and by construction, the objective function $f_i(x_i)$ is strictly concave, there is a unique maximizer for the maximization problem [21]. However, though the objective function is separable in x_i, solving this problem requires coordination of possibly all flows, which is not practical in MANETs. In order to derive practical distributed algorithms, the relaxation of the primal and dual problem of P are considered instead.

In this formulation, the utility function plays an important role in the model, since it determines the targeted fairness objective. Let us consider some popular concave utility functions that have been proposed in the literature. As discussed in [14], a series of concave utility functions can be defined in terms of flow rate x as:

$$f_\alpha(x) = \begin{cases} logx, & \text{if } \alpha = 1 \\ (1-\alpha)^{-1}x^{1-\alpha}, & \text{otherwise} \end{cases} \tag{5}$$

The fairness property is determined by the parameter α. The flow rate allocation will approach the system optimal fairness as $\alpha \to 0$, to the proportional fairness as $\alpha \to 1$, to the harmonic mean fairness as $\alpha \to 2$, and to the max-min fairness as $\alpha \to \infty$.

III. Non-cooperative Game Framework

We have shown that system problem $\boldsymbol{P}$ has a unique solution in its feasible set. However, in order to obtain this solution, cooperation between all flows is required. This will incur high communication overheads, which will reduce much of the available bandwidth of the system. According to the theory of convex optimization, by considering the Lagrange relaxation and dual problem, distributed iterative algorithms can be derived to solve the system problem.

In this section, the Lagrange relaxation of the system problem is defined. According to the definition, each flow maximizes its own utility function in a selfish way. Thus this problem is further modeled as a non-cooperative game. It is shown that under the control of the derived distributed algorithm, the system will converge to the unique Equilibrium point.

A. Framework

Now consider the Lagrangian of system problem P, we have:

$$L(\boldsymbol{x}, \boldsymbol{\lambda}) = \sum_i w_i f_i(x_i) + \boldsymbol{\lambda}^t(\boldsymbol{C} - \boldsymbol{A}\boldsymbol{x}), \tag{6}$$

where $\boldsymbol{\lambda}$ is a vector of Lagrange multipliers.

Based on the Lagrangian defined in Equation (6), the Lagrange relaxation of system problem $\boldsymbol{P}$ can be defined as follows:

$$\boldsymbol{Q}: \qquad max_{x_i} \sum_i V_i(x_i) \tag{7}$$

where

$$V_i(x_i) = w_i f_i(x_i) - \sum_{j \in Q(i)} \int_0^{x_i} \lambda_j(v)dv, \tag{8}$$

and $\lambda_j = p_j(\sum_{i \in S(j)} x_i)$ is defined as a function of the total flow rates on clique j, which is a nonnegative, continuous, convex and increasing function.

Problem $\boldsymbol{Q}$ can be modeled as a non-cooperative game (the basic concepts of game theory can be found in [22]), since each flow maximizes its own payoff $V_i(x_i)$ in a selfish way, we refer to it as the MAC game. In this game, the players are the flows (actually the sender nodes of the links). The strategy space for a flow is the range of the flow rate, which is determined by the capacity of the cliques. The strategy for flow i can be defined as $U^i = \{x_i | 0 < x_i <= C^i_{max}\}$, where C^i_{max} is the largest capacity of cliques that include flow i. The strategy space for the game is $U = U^1 \times \cdots \times U^N$, which is equivalent to the feasible set Ω. The payoff function is the function $V_i(x_i)$, which is a mapping from $U^1 \times \cdots \times U^N$ to $\mathbb{R}$.

A non-cooperative game settles at a so-called Nash equilibrium if one exists. In the MAC game, a vector of strategies $u^* \in U$ is called a Nash equilibrium if no player can increase its payoff by adjusting its strategy unilaterally. The Nash equilibrium point is important in practical terms because if one exists, then the game will converge to the equilibrium point and thus the stability of the system is guaranteed. As for the MAC game, we can prove the following result:

Theorem 3.1: The non-cooperative MAC game admits a unique Nash equilibrium in its pure strategy space.

The proof of Theorem 3.1 is given in the Appendix.

B. Distributed Algorithm Based on Non-cooperative Game Framework

The non-cooperative MAC game admits a unique Nash equilibrium. To reach this equilibrium, each player can change its strategy at a rate proportional to the gradient of its payoff function with respect to its strategy, subject to constraints [23]. Based on this principle, a distributed algorithm can be described as follows:

$$\frac{dx_i}{dt} = \theta_1 - \theta_2 \frac{1}{f_i'(x_i)} \sum_{j \in Q(i)} \lambda_j(t) \tag{9}$$

where

$$\lambda_j(t) = p_j(\sum_{i \in S(j)} x_i(t)) \tag{10}$$

The algorithm described by Equations (9) and (10) is a generalization of the algorithm presented in [12]. Actually if we let $f_i(x_i) = logx_i$, then this algorithm reduces to that described by (5) and (6) in [12]. Intuitively, the algorithm requires that (the node controlling) the link increases its rate proportionally to w_i and decreases it proportionally to $\sum_{j \in Q(i)} \lambda_j(t)$. Here θ_1 is the increase factor, and θ_2 is the decrease factor. As discussed in [12], the term $\lambda_j(t)$ in (9) and (10) can be thought of as if resource j generates a continuous stream of feedback signals at rate $p_j(y)$ when the total flow passing through resource j is y. It can also be explained as defining the price for resource j is $p_j(y)$ when the total flow is y. In short, $\sum_{j \in Q(i)} \lambda_j(t)$ reflects the network status perceived by flow i.

In the rate control problem or in the routing problem for a wired network, the network status can be explicitly obtained from network switches or routers or measured by the end hosts. However, in order to develop topology-transparent algorithms for ad hoc networks, the network status can only be measured by the hosts. By adopting similar techniques as in [12], [13], it can be proved that the system under the control of the distributed algorithm defined by (9) and (10) is globally stable, and the algorithm will lead the system to the unique equilibrium point. In addition, the algorithm can be generally treated as an additive increase multiplicative decrease scheme. Therefore the convergence speed is relatively high, and is determined by the two control parameters.

1) Implementation Issues: Equation (9) describes a general framework for developing distributed algorithms for MANETs. Intuitively, the framework tells us that each flow should adjust its flow rate based on the channel status it perceives.

As mentioned, we are more interested in the link level fairness in MANETs. In a real implementation, the sender of each link is responsible of controlling the rate on the link. Therefore each sending node is associated with a flow rate control parameter for each link. In contention based MAC protocols, the size of the contention window can be chosen as the control parameter. Furthermore, according to (9), the key components of the algorithm include:

- *Utility Function:* The utility function is determined by the objective of the fairness requirement. There is a trade-off between the fairness and the channel utilization – viz., the globally fairer is the scheme, the lower the channel utilization is. Besides the series of concave utility function defined by (5), other types of concave utility functions may also be adopted [24].
- *Global Parameters:* The increase factor θ_1 and the decrease factor θ_2 are system-wide parameters. The choices of the value of θ_1 and θ_2 are determined by both the adopted utility function and the type of feedback signals we can obtain or infer from the network. In general, they depend on the type of network under consideration and should be chosen by simulations.
- *Feedback Signals:* A mechanism is necessary in order to obtain the network status. To implement topology-blind and totally distributed algorithms, in which flows are not allowed to exchange any information with each other, the network status can only be measured or estimated by mobile hosts. For example, this can be the collisions encountered by flow i (which has been used in [1]), the service time or the queuing delay [14] for a packet, or the number of successful transmissions during a period of time as in [24]. In other words, the feedback signals imply the contention status of flow i with all its competitors.

As we can see, in order to implement these algorithms, the capacity of cliques is not required to be known in advance. The fairness objective of capacity allocation is approached in a selfish way.

Since the derived algorithms do not depend on any information sharing, they are suitable for MANETs with highly dynamic topologies. If the topology remains static, and the algorithm converges, the fairness objective can be achieved. When there are nodes joining or leaving the network frequently, the equilibrium point keeps changing, and the algorithm will track this point. The transient time is determined by the convergence speed of the algorithm, as well as the frequency of topology changes.

Algorithm 1 Distributed Fair MAC

States:
t^i_{packet}, current packet length.
t^i_{unit}, 4 way handshake duration. This is already available
in 802.11 in the outgoing NAV calculation
k_i, the time interval in terms t^i_{unit}.
t^i_p, current frame time interval as seen by node i.
n_i, the number of successful transmission.
win_i, contention window.
Events:
Init:
$t^i_{packet} = 0$ and $k_i = 1$.
$t^i_{unit} = t_{RTS} + t_{CTS} + t_{ACK} + 3 * t_{SIFS} +$
$t_{DIFS} + t^i_{packet}$.
$t^i_p = k_i * t^i_{unit}$.
Arm timer TP with time t^i_p.
Channel Access:
Use current contention window win_i to access channel.
If (successfully received ACK packet)
$n_i = n_i + 1$.
Update t^i_{packet}.
Update t^i_{unit}.
Timer Event: (upon expiry of TP)
if $(n_i > 1)$
if ($win_i == CW_{max}$)
$k_i = k_i - n_i + 1$; $t^i_p = k_i * t^i_{unit}$
else $win_i = 2 * win_i$;
else
if ($n_i == 0$)
if ($win_i == CW_{min}$)
$k_i = k_i + 1; t^i_p = k_i * t^i_{unit}$
else $win_i = win_i/2$;
endif
endif
$n_i = 0$, and $t^i_p = k_i * t^i_{unit}$.
Arm a new timer event TP with time t^i_p.

2) FMAC as An Example: Following the non-cooperative game framework, a family of practical distributed algorithms can be developed for MANETs. There are already some protocols proposed in the literature that fit within this framework. For example, a Proportional Fair Contention Resolution (PFCR) algorithm has been proposed in [1]. In PFCR, the objective function is the logarithm utility function, the channel status is measured by collisions perceived by a flow. Simulations show that PFCR achieves good proportional fairness. In this part, we take the FMAC [24] that fits to the non-cooperative game framework (refer to [24] for details) as an example to illustrate the effectiveness and correctness of the framework.

As discussed, the utility function can be a general concave function. In the FMAC, assume the flow rate for flow i is x_i in a time interval of length t_i, the objective function is defined as $-(x_i \times t_i - 1)^2$, which is a general concave utility function. Intuitively, the objective is trying to let each flow transmit exactly one packet in a time interval t_i whose length changes with the load of the network or the contention context. The number of transmissions in time interval t_i serves as the feedback signal and can be measured by each flow. Each flow adjusts either its contention window or the time interval and tries to approach the optimal value (0) of the utility function. The algorithm is shown in Algorithm 1.

IV. Cooperative Game Framework

By considering the dual problem of the system problem, a duality-based distributed algorithm is derived in this section. In order to solve the dual problem, each flow should construct its local flow contention graph by collecting flow information up to one hop away, and decompose this local contention graph into a set of maximal cliques. Since in this scheme, every flow obtains information about all its competitors, flows tend to cooperate with each other. This fits into a cooperative-game framework.

A. Distributed Algorithm Based on the Dual Problem

The Lagrangian of system problem P defined by Equation (6) has a similar form as that in [11]. Following the analysis methodology provided in [11], the dual problem of the system problem can be defined as:

$$\boldsymbol{D}: \qquad min_{\lambda \geq 0} d(\lambda)$$

where

$$d(\lambda) = \sum_i max(w_i f_i(x_i) - x_i \sum_{j \in Q(i)} \lambda_j) + \sum_j \lambda_j c_j \quad (11)$$

In (11), the term λ_j can be assumed to be the cost per unit of bandwidth in clique $j, j \in \mathcal{M}$. When the total offered flow rate in clique j is $\sum_{i \in S(j)} x_i$, the cost of the bandwidth in clique j is $\lambda_j(\sum_{i \in S(j)} x_i)$. Alternatively, assuming a fixed price λ_j, the optimal value of the flow rate can be computed by solving the following problem for flow i:

$$max_{x_i}(w_i f_i(x_i) - x_i \sum_{j \in Q(i)} \lambda_j). \quad (12)$$

Since function $f_i(x_i)$ is strictly concave, the unique maximizer for problem (12) exists, and can be computed as:

$$x_i^* = f_i^{'-1}(\frac{\sum_{j \in Q(i)} \lambda_j}{w_i}). \quad (13)$$

The goal of any distributed algorithm based on the dual problem is to search for the optimal value of λ_j for problem $\mathbf{D}$. A vector of flow rates can be computed by each link using (13) independently for a given value of $\boldsymbol{\lambda}$. According to the duality theory, the obtained vector of flow rates is the optimal value of the primal system problem. The motivation of solving the dual problem is that this can be done distributively on each clique, given the local flow information, and hence a global state information exchange is avoided.

1) The Gradient Projection Algorithm: In [11], a distributed algorithm based on the gradient projection method has been proposed to solve the dual problem. If we treat the clique as the link and the flow as the source in the flow control problem, a similar distributed algorithm can also be derived. This algorithm is called the Cooperative Game Framework based algorithm (CGF), which should include two components: the link algorithm and the clique algorithm. However, since unlike in wired networks where the router can execute the algorithm on behalf of its links (equivalent to cliques) the clique is just an abstract concept that has no real existence or control entity. Therefore the clique algorithm should also be taken in charge by the link flows that belong to the clique (i.e. distributively). The detail of the CGF algorithm is shown in Algorithm 2.

According to Algorithm 2, Equation (14) is used for a clique to calculate the new price. Intuitively, it implies the basic requirement of supply and consume: if the total offered flow rate is less (respectively more) than the capacity of the clique, the price decreases (respectively increases).

It can be proved that under the appropriate value of a step size, for any initial feasible flow rate x_0 and price λ_j^0, any accumulation point (x^*, λ^*) generated by the algorithm is primal-dual optimal. This conclusion shows that Algorithm 2 will lead the system to the primal-dual optimal point, which is unique. Therefore, the system is globally stable. Define $\bar{Q} = max_{i \in \mathcal{N}} |Q(i)|$ as the largest number of cliques that contain the same link. Denote $\bar{S} = max_{j \in \mathcal{M}} |S(j)|$ as the maximal size of cliques. Let $\bar{\delta}$ be the upper bound of function $-f''(x)$, then the range of the step size can be defined as in [11]:

$$0 < \gamma < \frac{2}{\bar{\delta}\bar{Q}\bar{S}},$$

In addition, the step size determines the convergence speed: the larger the value of the step size, the faster the convergence of the algorithm.

Algorithm 2 Cooperative Game Framework based Algorithm

The algorithm is executed by each flow (the sender node) round by round, and can be described in the following steps:

1) Initially, flow i chooses a feasible flow rate $x_i(0)$.
2) Flow i collects its local flow information, constructs its local flow contention graph, and decomposes it into a set of cliques $Q(i)$.
3) Initial price $\lambda_j(0)$ is set for each clique in $Q(i)$. The initial price is a global parameter of the system.
4) In round k, flow i calculates a new flow rate according to Equation (13).
5) Flow i disseminates the new flow rate information to all contending flows in one hop.
6) In round $k+1$, clique j calculates a new price according to Equation (14).

$$\lambda_j(k+1) = max\left(0, \lambda_j(k) + \gamma(\sum_{i \in S(j)} x_i^* - c_j)\right) \quad (14)$$

where γ is the step size, c_j is the capacity of clique j, $\sum_{i \in S(j)} x_i^*$ is the total flow rate of clique j in the previous round.

7) If the flow contention graph has changed (e.g., due to mobility), go back to (2), otherwise go back to step (4).

2) Implementation Issues: In order to implement this algorithm practically, the following issues need to be considered:

- To be able to calculate the new price for a clique, each flow needs to exchange the flow rate information with all its contending flows. This can be done by periodically broadcasting the flow information to all of the one-hop neighbors. Each broadcast costs only one packet per link. Eventually, since the wireless channel is a broadcast environment, a gross time granularity implementation may require nodes to periodically broadcast a single packet that carries information about all the flow rates under the node's control.
- The local flow contention graph is reconstructed whenever the topology of the flow contention graph changes.
- The algorithm is synchronous. This can be done by associating a round number with the flow information in the broadcast message.
- As discussed before, the clique capacity should be reduced by a factor for all cliques to avoid the problem of non schedulability (in the case of the max-min fairness).

Note that though the algorithm depends on the information of the local flows, the algorithm can adapt to the dynamics of the topology. Since if the local information is not correct or complete, the calculated flow rate would not be optimal. Then either collisions increase, or bandwidth utilization decreases. However, if the running time of the algorithm in one round is smaller than the time scale in which topology changes occur, which is a very reasonable assumption when considering QoS, then the flow rate will track the optimal value.

In a real implementation, the local flow information collection algorithm is very important to the convergence speed of the algorithm. The step size determines in how many rounds the algorithm will converge provided no changes occur in the topology. However, in each round, since the flow rate information needs to be disseminated, the time to finish this procedure determines how long the round is. In order to accelerate the convergence speed, it is suggested to use dual channel protocols where an under-loaded (high priority) control channel can be used to collect flow information and a data channel is used to vehicle data packets.

V. Simulation and Numerical Results

Both the FMAC and CGF algorithm have been implemented in NS2 [25]. In this section, some simulation results are presented for these two algorithms. The numerical results are also introduced to illustrate the effect of system parameters on the CGF algorithm.

In order to investigate the performance of the CGF algorithm, firstly the simulation and numerical analysis are conducted on a simple scenario, which is shown in Fig. 5. In addition, simulations for both the CGF and FMAC algorithm are also conducted on a more realistic topology scenario generated randomly. In the random scenario, 20 flows are uniformly distributed in a square area of $1000 \times 1000m^2$. The flow contention graph of the random scenario is shown in Fig. 6. In our simulations, the channel bandwidth is set to $11Mbps$. Each source generates constant bit traffic with the same rate R. The packet size is fixed and set as $1024byte$. All simulations are run to $400s$, which is long enough to measure the performance of the algorithms.

A. Cooperative Game Framework

For the CGF algorithm, the utility function introduced in (5) is adopted. In our implementation, in order to make sure the flow contention graph is consistent, the carrier sense range is set to be the same as the transmission range, and the flow contention graph is constructed accordingly. The dual channel MAC is emulated over a single channel, by assigning a higher priority to control messages. This is different from adopting a pure dual channel and is expected to produce lower performance than a pure dual channel model. The normalized capacity of cliques have been reduced to 0.6, and all flows use the same value of $w_i (= 1)$.

1) Numerical Results: In the numerical analysis, the topology of the network is assumed to be static, and the local flow contention graph has been constructed for each flow and remains stable. For the simple scenario, Fig. 7(a) shows the allocated normalized capacity against the time for different step size. As we can see, the system converges under the control of the CGF algorithm. It also shows that the larger the step size, the faster the convergence. In Fig. 7(b), different utility functions are used. The system converges under the control of the algorithms with different utility functions. In the equilibrium, the flow in the most competitive environment (flow 2) are allocated more capacity under the harmonic mean fairness ($\alpha = 2$) than in the proportional fairness ($\alpha = 1$). This verifies the correctness of the algorithm.

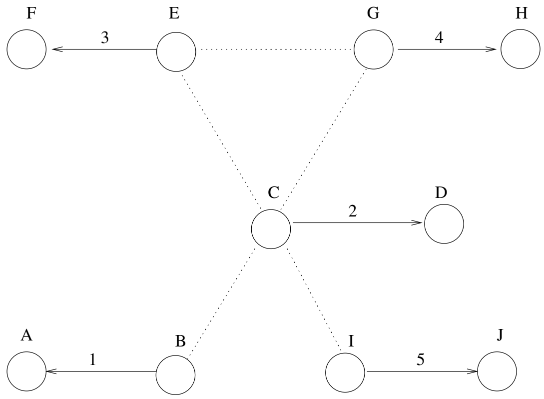


Fig. 5. The network topology of the simple scenario

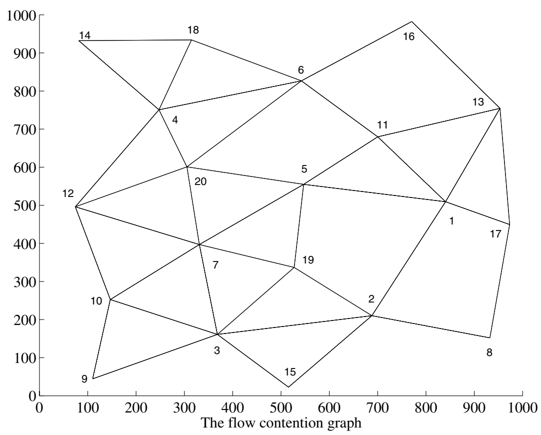
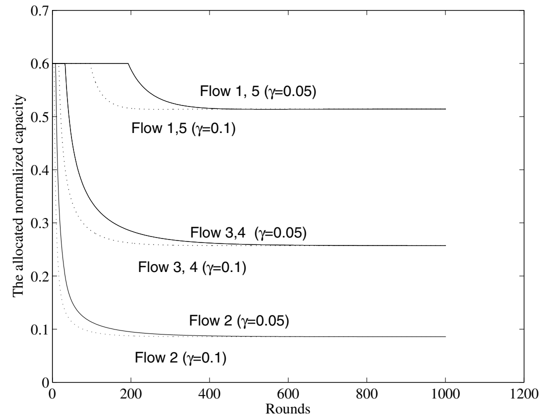
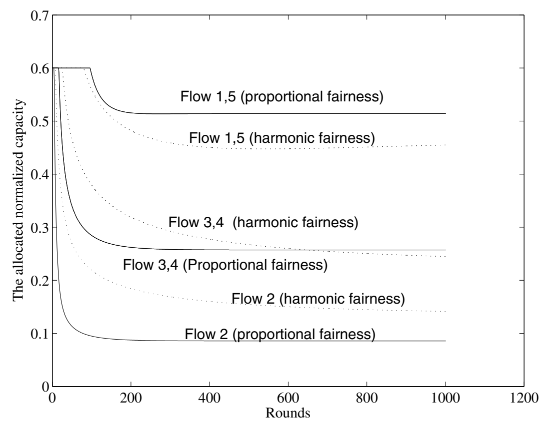


Fig. 6. The flow contention graph of the random scenario



(a) Proportional fairness with different step values



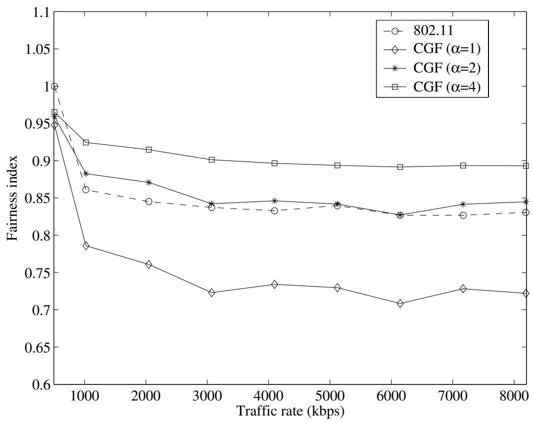
(b) Comparison of proportional fairness and harmonic fairness

Fig. 7. Numerical results for the CGF algorithm (simple scenario)

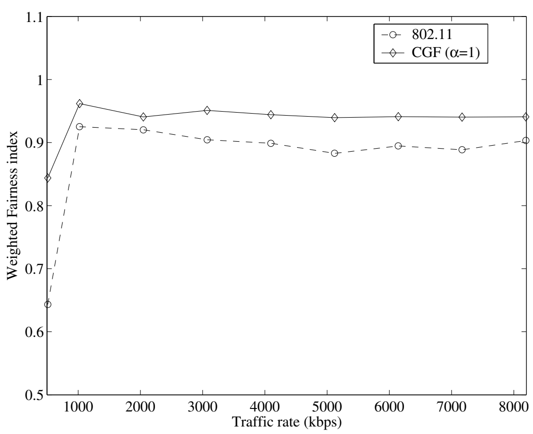
2) Simulation Results: The simulation results for the simple scenario are shown in Fig. 8 and 9. Fig. 8(a) shows the global fairness index [26] of the network under the control of the DCF and CGF with different fairness objectives (by adjusting parameter α). It is shown that when the channel is not saturated (flow rate is low), there is no fairness problem with the network. When the traffic is high, the network faces the fairness problem. The CGF algorithm can achieve the fairness objectives. As seen from the figure, the larger the value of α, the globally fairer the algorithm is. Note that when $\alpha = 1$, the fairness index of the CGF is smaller than that of 802.11 DCF. However, if we calculated the proportional weighted fairness index [2], as shown in Fig. 8(b), the system is fairer under the control of the CGF algorithm. This shows that the system indeed achieves the proportional fairness when α is set to 1. Use the same approach, we can show that the system can achieve the harmonic mean fairness when α is set to 2 and so on. Fig. 8(c) shows the global fairness index evolves with the time. As we can see the global fairness index converges, which implicates the convergence of the system under the control of both algorithms. Fig. 9(a) shows individual average throughput. It can be seen that flow 2, which faces the most competitive environment, achieves more throughput under the control of the CGF algorithm. Fig. 9(b) shows while the CGF achieves the fairness objectives, it does not sacrifice too much aggregate throughput.

B. Non-cooperative Game Framework

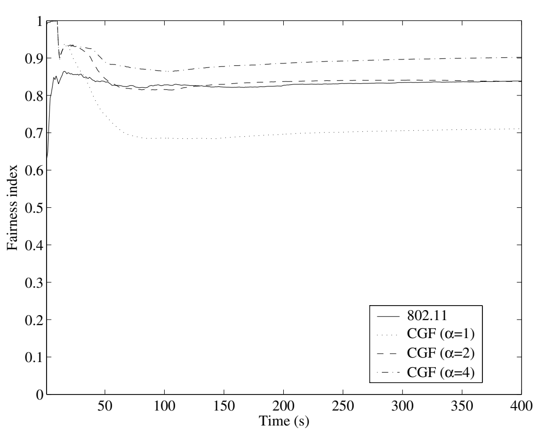
Fig. 10 shows the simulation results for the random scenario under the control of the FMAC algorithm compared with the DCF and CGF algorithm. It can be seen from Fig. 10(a) that the FMAC algorithm achieves much better global fairness compared with the CGF and DCF. However it loses much aggregate throughput, which is shown in Fig. 10(b). This shows that there is a tradeoff between the fairness objective and the aggregate throughput. Since the CGF algorithm aims to achieve local proportional fairness, it still maintains larger aggregate throughput, while meeting its fairness objective. The global fairness index used is close to the max-min fairness and therefore it is normal that when α increases the global fairness increases as well. Fig. 11 shows how the fairness index evolves with the time, and verifies that the system converges under the control of the algorithms.



(a) Global fairness index

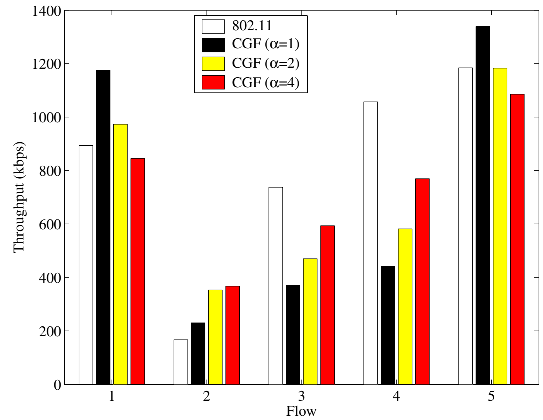


(b) Proportional weighted fairness index

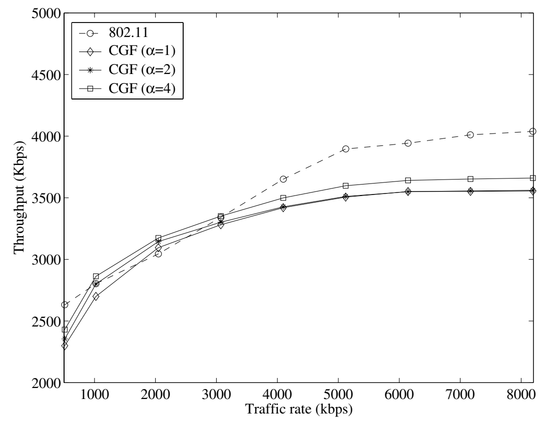


(c) Convergence of the global fairness index ($R = 8192kbps$)

Fig. 8. Fairness index (simple scenario)



(a) Individual average throughput ($R = 8192kbps$)



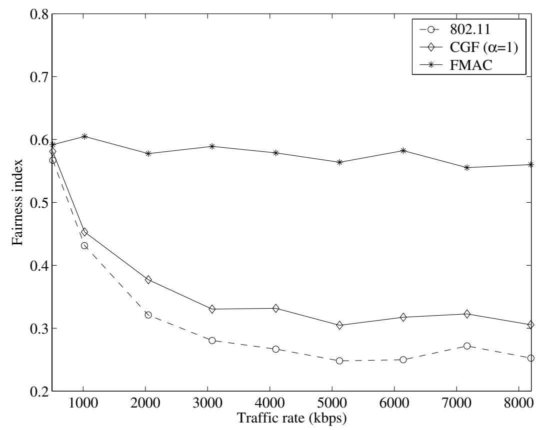
(b) Aggregate throughput

Fig. 9. Throughput (simple scenario)

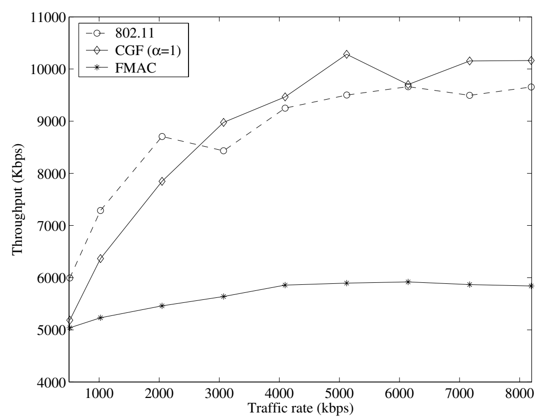
Note that the DCF is not fair because the protocol does not adopt a concave utility function approach. In the DCF, the action of the binary exponential backoff is to reset the contention window upon each success and to double it on failure. The DCF behavior can be modeled by a game where each node tries to maximize its own rate, which is a monotonic non concave function. Therefore the node that is the most immune to interference (hidden from its competitors) always achieves a much higher rate than those of the others.

VI. Conclusions

In this paper, we have studied the problem of fair bandwidth sharing between nodes at the link level in MANETs through a game theoretic approach. A contention graph has been adopted to capture the contention relations between links, and each maximal clique is treated as a "channel resource". The bandwidth allocation problem has been modeled as a constrained maximization problem, the primal problem. To guarantee that the maximizer of such problem achieves the fairness property,



(a) Global fairness index



(b) Aggregate throughput

Fig. 10. System performance (random scenario)

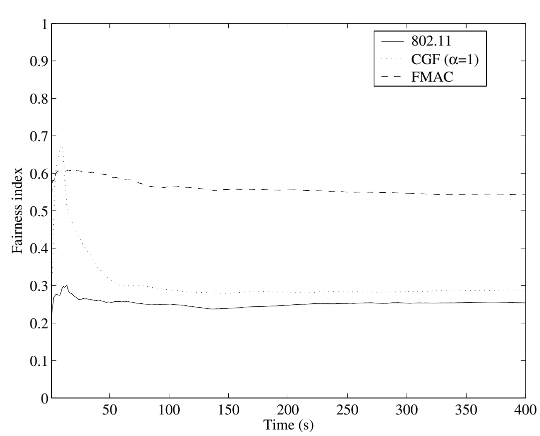


Fig. 11. Convergence of the global fairness index (random scenario, $R = 8192kbps$)

the utility function has been chosen to be a strictly concave function. To solve the primal directly, system wide cooperation may be needed, which is not desirable. Therefore, by considering the relaxation of the primal, namely, the unconstrained problem, a non-cooperative game framework has been derived. It is proved that the non-cooperative game has a unique Nash equilibrium. A distributed algorithm based on the non-cooperative game framework has been proposed. It is shown in the paper that the algorithm will lead the system to this unique Nash Equilibrium. Another alternative, by considering the dual problem, a distributed algorithm has been proposed. This algorithm requires some local cooperation which translates into the exchanges of link information between nodes up to one hop in the flow contention graph. It has been shown that the algorithm will converge to the optimal point of the primal problem. Some examples of MAC protocols drawn from the open literature have been used to show how they fit within one of the two frameworks, and both simulations and numerical results are given to illustrate the effectiveness and correctness of the two frameworks.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their valuable comments to improve the paper and Professor Tamer Basar for his kind help on the topic of game theory.

REFERENCES

[1] T. Nandagopal, T. E. Kim, X. Gao, and V. Bharghavan, "Achieving MAC Layer Fairness in Wireless Packet Networks," in *ACM MOBICOM*, 2000, pp. 87–98.
[2] N.H. Vaidya, P. Bahl, and S. Gupta, "Distributed fair scheduling in a wireless LAN," in *ACM MOBICOM*, 2000, pp. 167–178.
[3] X. L. Huang and B. Bensaou, "On Max-Min Fairness and Scheduling in Wireless Ad-Hoc Networks: Analytical Framework and Implementation," in *ACM MobiHOC*, 2001, pp. 221–231.
[4] H. Luo and S. Lu, "A Topology Independent Fair Queueing Model in Ad Hoc Wireless Networks," in *In Proceedings of IEEE ICNP'00*, Osaka, Japan, Aug. 2000.
[5] L. Bao and J. J. Garcia-Luna-Aceces, "A New Approach to Channel Access Scheduling for Ad Hoc Networks," in *SIGMOBILE: ACM Special Interest Group on Mobility of Systems, Users, Data and Computing*, 2001, pp. 210–221.
[6] S. Basagni, D. Bruschi, and I. Chlamtac, "A mobility-transparent deterministic broadcast mechanism for ad hoc networks," *IEEE/ACM Transactions on Networking*, vol. 7, no. 6, pp. 799–807, December 1999.
[7] S. Chen and K. Nahrstedt, "Distributed Quality-of-Service Routing in AD Hoc Networks," *IEEE JSAC*, vol. 17, no. 8, pp. 1488–1505, August 1999.
[8] M. K. Chan and B. Bensaou, "A novel admission control and qos routing protocol for ad hoc networks," in *IEEE LANMAN*, 2002.
[9] S. B. Lee, G. S. Ahn, X. Zhang, and A. T. Campbell, "INSIGNIA: An IP-Based Quality of Service Framework for Mobile Ad Hoc Networks," in *Proc of 5th International Workshop on Mobile Multimedia Communications (MoMuC,98)*, Berlin, Germany, Oct. 1998.
[10] F. Kelly, "Charging and rate control for elastic traffic," *European Transactions on Telecommunications*, vol. 8, pp. 33–37, January 1997.
[11] S. H. Low and D. E. Lapsley, "Optimization Flow Control–I: Basic Algorithm and Convergence," *IEEE/ACM Transactions on Networking*, vol. 7, no. 6, pp. 861–874, Dec. 1999.
[12] F. P. Kelly, A. Maulloo, and D. Tan, "Rate control for communication networks: Shadow prices, proportional fairness and stability," *Journal of Operation Research Society*, vol. 49, no. 3, pp. 237–252, Mar. 1998.
[13] T. Alpcan and T. Basar, "A game-theoretic framework for congestion control in general topology networks," in *Proc. 41st IEEE Conference on Decision and Control*, Dec. 2002, Las Vegas, Nevada, USA.

[14] J. Mo and J. Walrand, "Fair end-to-end window-based congestion control," *IEEE/ACM Transactions on Networking*, vol. 8, no. 8, pp. 556–567, Oct. 2000.
[15] H. Yaiche, R. R. Mazumdar, and C. Rosenberg, "A game theoretic framework for bandwidth allocation and pricing in broadband networks," *IEEE/ACM Transactions on Networking*, vol. 8, no. 5, pp. 667–678, Oct. 2000.
[16] J. Gronkvist and A. Hansson, "Comparison between graph-based and interference-based stdma scheduling," in *MobiHOC 2001*, 2001, pp. 255,258.
[17] A. Bar-noy, A. Mayer, B. Schieber, and M. Sudan, "Guaranteeing fair service to persistent dependent tasks," *SIAM J. COMPUT*, vol. 27, no. 4, pp. 1168–1189, August 1998.
[18] B. Hajeck and G. Sasaki, "Link scheduling in polynomial time," *IEEE Transactions on Information Theory*, vol. 34, no. 5, pp. 910–917, Sep. 1988.
[19] G. Cornuejols, "The strong perfect graph conjecture," in *International Congress of Mathematicians*, 2002, Bejing.
[20] C. E. Shannon, "A theorem on coloring the lines of a network," *J. Math. Phys.*, vol. 28, pp. 148–151, 1949.
[21] S. G. Nash and A. Sofer, *Linear and Nonlinear Programming*, McGRAW-HILL International Editions, 1996.
[22] T. Basar and G. J. Olsder, *Dynamic Noncooperative Game Theory*, Acsdemic Press, Harcort Brace Company, 1995, Second Edition.
[23] J. B. Rosen, "Existence and uniqueness of equilibrium points for concave n-person games," *Econometrica*, vol. 33, pp. 520–534, Jul. 1965.
[24] Z. Fang and B. Bensaou, "A Novel Topology-blind Fair Medium Access Control Algorithm for Wireless LAN and Ad Hoc Networks," in *ICC2003*, 2003.
[25] "The network simulator -ns-2," http://www.isi.edu/nsnam/ns/,2.1b9a.
[26] R. Jain, W. Hawe, and D. Chiu, "A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer Systems," Tech. Rep., DEC-TR-301, Sep., 26 1984.

APPENDIX

The following results follow the proof methodologies provided in [13], [23].

Lemma 1: The feasible set Ω is a nonempty, convex and compact set.

Proof: It has been shown that the feasible set of the flow rate vectors can be described as:

$$\Omega = \{\boldsymbol{x} | \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{C}, \boldsymbol{x} \geq 0\}$$

Define the maximal size (the number of flows in the clique) of a clique as $\bar{S}$, the minimal capacity of the clique as C_{min}. If each flow chooses the flow rate $\frac{C_{min}}{\bar{S}}$, and let $\bar{\boldsymbol{x}} = (\frac{C_{min}}{\bar{S}}, \cdots, \frac{C_{min}}{\bar{S}})$. It is obvious that $\bar{\boldsymbol{x}}$ is a feasible flow rate vector. This shows that Ω is an nonempty set. Assume the maximal capacity of cliques is C_{max}. The flow rate of each flow can not exceed C_{max}. Therefore, the set Ω can be bounded. Now Assume $\boldsymbol{x^1}$ and $\boldsymbol{x^2}$ are two feasible flow rate vectors, we have

$$\lambda \boldsymbol{x^1} + (1-\lambda)\boldsymbol{x^2} \leq A(\lambda \boldsymbol{x^1} + (1-\lambda)\boldsymbol{x^2}) \leq C$$

This result shows that Ω is a convex set. In conclusion, we can see that Ω is nonempty, and it is a convex and compact set. ■

With the help of Lemma (1), we can prove the following result.

Theorem 1.1: The non-cooperative game MAC admits a unique Nash equilibrium in its pure strategy space.

Proof: Lemma (1) shows that the strategy space for the MAC game is a nonempty, convex and compact set. Consider the objective function $V_i(x_i)$ for player i, we have:

$$V_i''(x_i) = w_i f_i''(x_i) - \sum_{j \in Q(i)} \lambda_j' \tag{15}$$

Since $f_i(x_i)$ is strictly concave, we have $f_i''(x_i) < 0$. And λ_j is a convex function, therefore $\lambda_j' \geq 0$. Based on these two conditions, we can see from Equation (15) that $V_i''(x_i) < 0$, hence $V_i(x_i)$ is a strictly concave function. According to Theorem 1 in [23], these conditions are sufficient to insure that the non-cooperative game admits a Nash Equilibrium in its pure strategy space.

Now define a weighted nonnegative sum of the function $V_i(\boldsymbol{x})$ as

$$\sigma(\boldsymbol{x}, \boldsymbol{r}) = \sum_{i=1}^{N} r_i V_i(\boldsymbol{x}), \quad r_i \geq 0 \tag{16}$$

The pseudo-gradient of $\sigma(\boldsymbol{x}, \boldsymbol{r})$ can be defined as

$$g(\boldsymbol{x}, \boldsymbol{r}) = \begin{bmatrix} r_1 \bigtriangledown V_1(\boldsymbol{x}) \\ r_2 \bigtriangledown V_2(\boldsymbol{x}) \\ \vdots \\ r_N \bigtriangledown V_N(\boldsymbol{x}) \end{bmatrix}$$

Denote $B_{ij} = r_i \frac{\partial^2 V_(\boldsymbol{x})}{\partial x_i \partial x_j}$. The Jacobian with respect to x of $g(\boldsymbol{x}, \boldsymbol{r})$ can be computed as:

$$\mathbf{G} \begin{pmatrix} B_{11} & B_{12} & \dots & B_{1N} \\ B_{21} & B_{22} & \dots & B_{2N} \\ \vdots & \ddots & & \vdots \\ B_{N1} & B_{N2} & \dots & B_{NN} \end{pmatrix}$$

where

$$B_{ij} = \begin{cases} r_i(w_i f_i''(x_i) - \sum_{k \in Q(i)} \lambda_k') < 0 & j = i \\ -r_i \sum_{k \in P} \frac{\partial \lambda_k}{\partial x_j} < 0 & j \neq i, P \neq \emptyset \\ 0 & j \neq i, P = \emptyset \end{cases}$$

where $P = Q(i) \cap Q(j)$. It can be shown that matrix $\mathbf{G}$ can be decomposed as the sum of a negative definite diagonal matrix and a set of semi-negative definite matrices as in [13]. Therefore $\mathbf{G}$ is negative definite. It follows that $\mathbf{G} + \mathbf{G^t}$ is also negative definite for positive vector $\boldsymbol{r}(r_i > 0)$. According to Theorem 6 in [23], $\sigma(\boldsymbol{x}, \boldsymbol{r})$ is diagonally strictly concave. Therefore according to Theorem 2 in [23], the equilibrium point of the MAC game with respect to $\boldsymbol{x}$ is unique. ■