Throughput-Oriented MAC for Mobile Ad Hoc Networks: A Game-Theoretic Approach[⋆]

Fan Wang Ossama Younis Marwan Krunz[*]

Department of Electrical and Computer Engineering
University of Arizona, Tucson, AZ 85721

Abstract

The conservative nature of the 802.11 channel access scheme has instigated extensive research whose goal is to improve the spatial reuse and/or energy consumption of a mobile ad hoc network. Transmission power control (TPC) was shown to be effective in achieving this goal. Despite their demonstrated performance gains, previously proposed power-controlled channel access protocols often incur extra hardware cost (e.g., require multiple transceivers). Furthermore, they do not fully exploit the potential of power control due to the heuristic nature of power allocation. In this paper, we propose a distributed, single-channel MAC protocol (GMAC) that is inspired by game theory. In GMAC, each transmitter computes a utility function that maximizes the link's achievable throughput. The utility function includes a pricing factor that accounts for energy consumption. GMAC allows multiple potential transmitters to contend through an admission phase that enables them to determine the transmission powers that achieve the Nash Equilibrium (NE). Simulation results indicate that GMAC significantly improves the network throughput over the 802.11 scheme and over another single-channel power-controlled MAC protocol (POWMAC). These gains are achieved at no extra energy cost. Our results also indicate that GMAC performs best under high node densities and large data packet sizes.

Key words: Mobile ad hoc networks, MAC protocol, IEEE 802.11, game theory, power control, throughput.

[⋆] This work was supported by NSF under grants ANI-0095626, ANI-0325979, and ANI-0435490. Any opinions, findings, conclusions, and recommendations expressed in this material are those of the authors and do not reflect the views of NSF. An abridged version of this paper was presented at the *4th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt'06)*, 2006.

[*] Corresponding author.

Email addresses: wangfan@ece.arizona.edu (Fan Wang), younis@ece.arizona.edu (Ossama Younis), krunz@ece.arizona.edu (Marwan Krunz).

1 Introduction

Mobile ad hoc networks (MANETs) are self-organizing networks that provide an efficient solution when centralized control is infeasible (e.g., emergency and rescue operations, disaster-relief efforts, etc.). A key design objective in MANETs is to achieve high network throughput while maintaining energy-efficient wireless communications for mobile terminals [1,20]. To achieve this objective, efficient design of the MAC layer is necessary in order to resolve channel-contention and reduce packet collisions.

The "ad hoc" mode of the IEEE 802.11 standard [9] has so far been used as the *de facto* MAC protocol for MANETs. This standard is based on CSMA/CA with an optional RTS/CTS (request-to-send/clear-to-send) handshake to coordinate channel access and resolve contention. Several studies documented the inadequate performance of this protocol, which is attributed to its conservative treatment of potential interferers (the RTS/CTS packets are used to silence all overhearing terminals), its use of fixed transmission powers (TPs), and its inefficient handling of ACK packets (the exposed terminal problem). These problems are particularly acute in dense networks, where the transmitter-receiver distances are relatively small.

To overcome these problems, researchers considered the use of power control at the MAC layer as a way to improve spatial reuse (e.g., [14,25,16]) and/or reduce energy consumption (e.g., [3,11]). Our emphasis in this paper is on the first objective, i.e., to improve the network throughput. In throughput-oriented power-controlled MAC schemes, terminals broadcast some *collision avoidance information* (CAI) to neighboring terminals. This information is used to *bound* the transmission powers of potential future transmitters in the neighborhood. For example, CAI may include information about the maximum tolerable interference (MTI), defined as the amount of interference power that a receiver of a data or ACK packet is able to tolerate from one future interferer [16]. Future transmitters use the overheard MTI values along with other information (e.g., channel gains, load tolerance, etc.) to determine their TPs. This way, multiple interference-limited transmissions can take place concurrently in the vicinity of the same receiver.

Motivation and contributions. Despite their demonstrated performance gains, previously proposed TPC schemes suffer from several problems, including incompatibility with the 802.11 architecture, extra hardware, and lack of ACK protection. For instance, many of them offer dual-channel solutions, which often require two transceivers per terminal. In [16], the authors presented a single-channel, single-transceiver power-controlled MAC protocol called POWMAC, which allows for multiple concurrent interference-limited transmissions and provides protection for ACK packets. Simulations indicate that POWMAC achieves good performance improvement (up to 40%) over the classic (fixed-power) CSMA/CA. However, because it relies on heuristics for determining the MTI value (and hence the TPs), the protocol does not fully exploit the potential of power control and may sometimes unnecessarily silence some possible transmissions.

The following simple example illustrates the impact of heuristically setting the MTI in the POWMAC protocol. Consider a MANET of four terminals: A, B, C, and D, as depicted in Fig. 1, where P_{max} denotes the maximum transmission power. Let G_{ij} and d_{ij} denote, respectively, the channel gain and distance between any two terminals i and j. Suppose that terminals A and C wish to transmit to terminals B and D, respectively, and suppose that A succeeds in transmitting its RTS before C sends its RTS. Assume that $G_{AB} = G_{CD}$, and that A and C are within the maximum transmission range of each other. Let $\text{SNR}_{th} = 6$ dB be the required signal-to-noise ratio for both receivers. Assume a two-ray channel propagation model with a path loss factor of 4. For simplicity, we ignore the thermal noise. Clearly, the spatial reuse is maximized when both transmissions $A \rightarrow B$ and $C \rightarrow D$ are allowed to proceed concurrently. The necessary conditions for that are $\frac{G_{AB}p_A}{G_{CB}p_C} \geq \text{SNR}_{th}$ and $\frac{G_{CD}p_C}{G_{AD}p_A} \geq \text{SNR}_{th}$, where p_A and p_C are the transmission powers of terminals A and C, respectively. Combining these two inequalities, we get $\frac{G_{AD}}{G_{CD}} p_A \text{SNR}_{th} \leq p_C \leq \frac{G_{AB}}{G_{CB}} \frac{p_A}{\text{SNR}_{th}}$. Thus, it is necessary to have $\frac{G_{AD}}{G_{CD}} \text{SNR}_{th} \leq \frac{G_{AB}}{G_{CB}} \frac{1}{\text{SNR}_{th}}$. Since we assume $G_{AD} = G_{CB}$ and $G_{AB} = G_{CD}$, the above condition becomes:

$$\left(\frac{G_{AD}}{G_{AB}} \text{SNR}_{th}\right)^2 \leq 1$$

which leads to $\frac{G_{AD}}{G_{AB}} \leq \frac{1}{\text{SNR}_{th}}$. At $\text{SNR}_{th} = 6$ dB and a path loss factor of 4, the necessary condition for two concurrent transmissions reduces to $d_{AC} \geq 0.41 d_{AB}$.

Now suppose that the POWMAC protocol is to be applied to the same example network, with terminal A capturing the channel before terminal C. According to POWMAC, terminal A calculates the minimum power needed for correct reception, *heuristically* inflates this power by adding to it some "interference margin," and evenly distributes this margin among multiple future interfering terminals (whose number is not known in advance). Specifically, A's transmission power (p_A) will be inflated to allow for future interference from up to N_{AW} of B's neighbors. So, p_A will be chosen such that $\frac{G_{AB}p_A}{N_{AW}p_{MTI}} \geq \text{SNR}_{th}$, where p_{MTI} is the maximum tolerable interference a future interferer is allowed to add to receiver B. After p_{MTI} is set for link $A \rightarrow B$, when C's transmission power is computed, it must satisfy the following MTI requirement: $G_{CB}p_C \leq p_{MTI}$. In addition, C's power must satisfy the SNR requirement $\frac{G_{CD}p_C}{G_{AD}p_A} \geq \text{SNR}_{th}$. Combining these two inequalities, we get: $\frac{G_{AD}}{G_{CD}} \text{SNR}_{th} p_A \leq p_C \leq \frac{G_{AB}}{G_{CB}} \frac{1}{N_{AW}\text{SNR}_{th}} p_A$. Since we assume $G_{AD} = G_{CB}$ and $G_{AB} = G_{CD}$, the above condition reduces to $\frac{G_{AD}}{G_{AB}} \leq \frac{1}{\sqrt{N_{AW}}\text{SNR}_{th}}$. Given a typical N_{AW} value in POWMAC of 5, the necessary condition for two concurrent transmissions in POWMAC becomes $d_{AC} \geq 0.73 d_{AB}$. That is, if $0.41dB < d_{AC} < 0.73 d_{AB}$, then POWMAC allows only one of these two transmissions to proceed, even though the two can proceed simultaneously. This inefficiency is related to how the MTI is determined. Setting the value for this parameter is nontrivial. A value that is too large adds unnecessary interference in the network and wastes energy, while a value that is too small prevents some feasible concurrent transmissions.

To select the appropriate TPs that maximize the spatial reuse, we formulate the channel

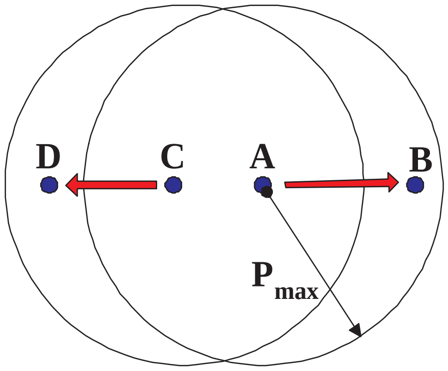


Fig. 1. Example of two transmissions that can proceed simultaneously if transmission powers are computed appropriately ($G_{AD} = G_{CB}$ and $G_{AB} = G_{CD}$).

contention problem as a non-cooperative power-control game and present a novel MAC protocol (GMAC) that implements this game in a distributed fashion. Game theory provides a powerful mathematical tool for decision-making among contending transmissions. It has been applied in the context of infrastructure-based (cellular) wireless networks (e.g, [23,4,10,26]). In these approaches, each user attempts to adjust his individual power in order to maximize a given utility function that may incorporate conflicting goals (e.g., signal quality and energy consumption). Though driven by similar game-theoretic analysis, our work is proposed for MANETs, and hence is quite different from its cellular counterparts in two aspects. First, in MANETs, multi-hop communication is common, whereas in cellular networks, terminals use a single hop to communicate with the base station (BS). Second, for uplink transmissions in cellular networks, interference occurs at the BS, which can be easily estimated. On the other hand, in MANETs, each reception will endure interference from all active transmissions in the network, particularly those within the receiver's proximity. For this reason, it is difficult for a terminal to estimate all possible sources of interference.

Unlike previous work on game-theoretic MAC design, GMAC does not rely on any correlations in the offered traffic. It makes power control decisions on a per packet basis. Instead of making power decisions in parallel, as in previous work, each receiver uses its knowledge of previously scheduled transmissions in its vicinity to decide whether to join this schedule or not (this will be explained in Section 4).

GMAC supports a CSMA/CA-like access mechanism for distributing TPs to contending transmitters, hence enabling more concurrent transmissions than in previously proposed techniques. The protocol is asynchronous, completely distributed, and uses only a single channel for both data and control packets. This ensures hardware compatibility with the 802.11 scheme. We also introduce a linear pricing function to obtain Pareto improvements in the achieved NE solution (this will be explained in Section 3).

System model. We consider a MANET with the following capabilities:

- The channel gain is stationary for the transmission duration of a few control packets and one data packet. This assumption holds for typical mobility patterns and transmission rates.

- Channel gains between any two terminals are symmetric. This is the underlying assumption in any RTS/CTS-based protocol, including the IEEE 802.11 scheme.
- Each terminal is capable of estimating the channel gains from the received signal strength of control packets, which are transmitted at power P_{max}.

Organization. The rest of the paper is organized as follows. In Section 2, we briefly discuss TPC schemes for MANETs and outline previous work on the application of game-theoretic power control analysis. Section 3 introduces our game-theoretic formulation. Based on the analysis in Section 3, we present our GMAC protocol design in Section 4. In Section 5, we provide simulation results and insights on the performance of GMAC. Finally, we draw our conclusions in Section 6 and propose some extensions for future work.

2 Related Work

TPC protocols proposed in the literature focused on either reducing energy consumption (e.g., [3,11]), or increasing network throughput (e.g., [14,16,18,25,27]). For example, Agarwal et al. [3] assumed that each terminal possesses a fixed number of power levels, and proposed a mechanism that uses the minimum required power level for transmission. A similar approach was proposed in [11] to reduce energy consumption in networks with non-uniformly distributed terminals. These approaches reduce the overall energy consumption, but achieve comparable throughput to that of the 802.11 scheme.

Throughput-oriented TPC protocols include PCMA [14], PCDC [15], and POWMAC [16] (see [12] for a complete overview of various TPC protocols in MANETs). The work in this paper belongs to this class of protocols. In PCMA [14], the receiver advertises its interference margin by sending busy-tone pulses over a dedicated control channel. The use of a control channel along with a busy-tone scheme was proposed in [25], where the sender transmits the data packets and busy tones at a reduced power, while the receiver transmits its busy tones at the maximum possible power. The PCDC protocol [15] uses two frequency-separated channels for data and control packets, allowing for interference-limited concurrent transmissions in the vicinity. In contrast to the protocols above, the POWMAC protocol [16] uses a single channel for both data and control packets. Data packets are transmitted after several RTS/CTS exchanges take place. This enables the scheduling of multiple concurrent transmissions in the same vicinity, provided that a certain interference margin is not exceeded at each transmitter-receiver pair.

Historically, TPC was used in cellular networks to improve various performance metrics, including throughput and energy efficiency. Some protocols (e.g., [23,10,26,4]) used game theory to compute the transmission powers of terminals in a single cell. In [23], Saraydar et al. proposed a game-theoretic power control algorithm for data transmissions in cellular networks. The number of efficient bits transmitted per energy unit was used as the utility function. By using a linear pricing function for the transmission power, Pareto improvement of the game was achieved. Ji and Huang [10] formulated a game for uplink power control in cellular networks. Their game's utility is a decreasing concave function of the transmission power and an increasing concave function of the signal-to-interference ratio. An iterative algorithm that searches for the equilibrium solution was introduced and analyzed under different scenarios. The same framework was adapted in [26] by using a different sigmoid utility function. Alpcan et al. [4] adopted as a cost function the difference between a pricing function based on power consumption and a utility function based on Shannon capacity. They also proposed two iterative algorithms to obtain the NE and proved that these algorithms converge under certain conditions. All these game-theoretic protocols consider infrastructure-based cellular networks. Extending game theory to MANETs is challenging since such networks typically lack a centralized infrastructure. Very recently, there has been an increasing interest in using game theory to compute the transmission powers in MANETS [8,5]. In these works, individual terminals use some feedback (e.g., received SNR) to update their power levels. Then, they perform a few iterations in which the transmission power is selected and refined at each terminal until an optimal power allocation vector is determined. This approach implicitly assumes that the set of active contenders and their relative channel gains remain static over several data packet durations. Such assumptions may not be valid in MANETs, as terminals may join and leave the network frequently and channel conditions may rapidly change.

3 Formulation of the Power-Control Game

In this section, we formulate the power-control game and use it to select the links that can be activated concurrently within a neighborhood. We then provide our rationale for selecting an appropriate value for the pricing factor of the game's utility function. We consider a general multi-hop ad hoc network, where the mobile terminals contend for the channel whenever they have data to send. Our GMAC protocol, presented in Section 4, intelligently uses this analysis to enable concurrent transmissions across the entire network.

3.1 Utility Function

We first define an appropriate utility function for the power-controlled channel access game. Our goal is to maximize the overall network throughput while preventing terminals from unnecessarily using high power. We use Shannon's capacity (maximum achievable rate) as the figure of merit. Applying Shannon's formula implies that the system is rate-adaptive, which is achieved in practice through adaptive coding/modulation at the physical layer [1] . Even with rate adaptivity, we still require the received SNR to be above

[1] Some approaches incorporate the modulation scheme in the utility function [23].

a fixed threshold (SNR$_{th}$), so that "packet capture" can take place. When the SNR is greater than SNR$_{th}$, the achievable rate increases with the SNR according to Shannon's formula (which approximates the more practical but less tractable staircase function the characterizes the rate-vs-SNR relationship of typical multi-rate systems).

Accordingly, the utility function for an active link i is defined as (similar to [4]):

$$u_i(p_i, \mathbf{p}_{-i}) = \ln(1 + \gamma_i) - \alpha_i p_i, \;\; i = 1, 2..., n \tag{1}$$

where p_i is the transmission power of i's transmitter (to be computed), $\mathbf{p}_{-i} \triangleq [p_1, ..., p_{i-1}, p_{i+1}, ..., p_n]$ is a vector that represents the transmission powers of all links other than i, γ_i is the received SINR at the receiver, and α_i is the pricing factor. The SINR is given by:

$$\gamma_i = \frac{h_{ii} p_i}{\sum_{j \neq i} h_{ji} p_j + \sigma^2} \tag{2}$$

where h_{ji} denotes the channel gain between the *transmitter* of link j and the *receiver* of link i, and σ^2 is the measured noise-plus-interference power. Note that, in general, $h_{ji} \neq h_{ij}$.

The second term in (1) is a linear pricing function that represents the "price" for consuming a specific amount of power. Each terminal selects its transmission power such that its own utility function is maximized. This results in a standard non-cooperative game [7] of the following form:

$$\max_{p_i} \;\; u_i(p_i, \mathbf{p}_{-i}), \text{for all } i = 1, 2, \ldots, n \tag{3}$$

subject to the constraint:

$$C_1 : p_i \in S_i \triangleq [0, P_{max}]. \tag{4}$$

We assume that SNR$_{th}$ and P_{max} are the same for all terminals. The solution to the above game is the one that achieves the NE. Previous work [23] showed that the NE for the game in (3) and (4) exists if the following two conditions are satisfied:

(1) S_i is a nonempty and convex subset of some Euclidean space; and
(2) u_i is a continuous and quasi-concave function in p_i.

The first condition is readily satisfied. To show that the second condition is also satisfied, we take the second-order partial derivative of u_i with respect to p_i:

$$\frac{\partial^2 u_i}{\partial p_i^2} = -\frac{h_{ii}^2}{(h_{ii} p_i + \sum_{j \neq i} h_{ji} p_j + \sigma^2)^2}. \tag{5}$$

Since $\frac{\partial^2 u_i}{\partial p_i^2} < 0$, the second condition is satisfied. Therefore, the NE exists.

However, the existence of the NE does not guarantee that the SNR threshold is satisfied for all links. Thus, we impose another constraint:

$$C_2 : \gamma_i \geq \text{SNR}_{th}, \text{for all } i = 1, 2, \ldots, n. \tag{6}$$

In GMAC, if C_2 is not satisfied for some links, we do not allow these links to proceed. As we explain later, we handle this situation during the admission control phase. A terminal competing for the channel will compute the NE powers. If these powers are feasible (i.e., C_1 and C_2 are both satisfied), the terminal will decide whether or not to proceed concurrently with previously scheduled transmissions. This decision-making process is made serially by terminals during a contention period known as the *access window* (AW) (described in Section 4).

3.2 Computing the NE

To find the NE of the game, we construct and analyze the players' *best response functions* [19]. The best response of link i is the transmission power that maximizes its utility function and satisfies the constraint C_1. The power that maximizes i's utility function can be obtained by equating the first-order derivative of u_i to zero:

$$\frac{\partial u_i}{\partial p_i} = \frac{h_{ii}}{h_{ii}p_i + \sum_{j \neq i} h_{ji}p_j + \sigma^2} - \alpha_i = 0. \tag{7}$$

Therefore,

$$p_i = \frac{1}{\alpha_i} - \frac{\sum_{j \neq i} h_{ji}p_j + \sigma^2}{h_{ii}}. \tag{8}$$

The computed p_i may or may not satisfy the constraint C_1. Thus, the best response function of link i, p_i^*, is given by:

$$p_i^* = \begin{cases} 0, & \text{if } p_i \leq 0 \\ p_i, & \text{if } 0 < p_i < P_{max} \\ P_{max}, & \text{if } p_i \geq P_{max} \end{cases} \tag{9}$$

We now use the best response function to obtain the NE. Consider the simple example in Fig. 1 and assume that α_i is fixed to $1/P_{max}$ (the reason for such a selection will be explained later). Fig. 2 represents the best response functions for the links A-B and C-D. The cross-point of these two functions is the NE. As stated in Section 1, the achieved NE satisfies the SNR constraint as long as $d_{AC} \geq 0.41 d_{AB}$. If d_{AC} gets smaller, the slope of the best response function becomes flatter, and so the NE power will eventually not satisfy the constraint C_2.

In general, if we have n links, the NE is the cross-point of the best response hyper-planes of all the links. By rearranging the terms in (8) and writing n simultaneous equations, (8) can be expressed in a matrix form as:

$$\mathbf{HP}^* = \mathbf{G} \tag{10}$$

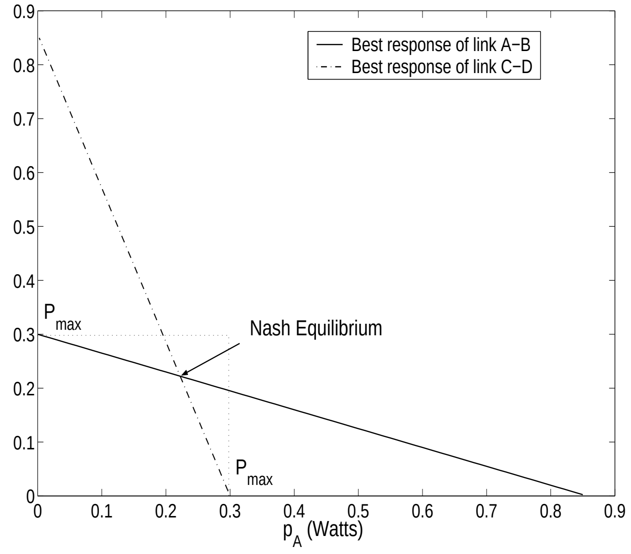


Fig. 2. Best response functions for links A-B and C-D.

where $\mathbf{H} = [h_{ij}]_{i,j}$ is an $n \times n$ matrix representing the channel gains between the transmitter/receiver pairs. Note that the matrix $\mathbf{H}$ is, in general, asymmetric. $\mathbf{G} = [g_1, g_2, ..., g_n]^T$ is an $n \times 1$ vector with $g_i \triangleq \frac{h_{ii}}{\alpha_i} - \sigma^2$. If the constraint C_2 is satisfied for all links, then $\mathbf{P}^* = [p_1^*, p_2^*, ..., p_n^*]$ is the NE solution, and can be calculated using:

$$\mathbf{P}^* = \mathbf{H}^{-1}\mathbf{G}. \tag{11}$$

Equation 11 is valid only when $\mathbf{H}$ is invertible, which is often the case because the elements of $\mathbf{H}$ (the channel gains) are, in general, independent random variables. As discussed in Section 4.2.3, in the unlikely event that $\mathbf{H}$ becomes non-invertible upon the inclusion of a link i, the GMAC protocol handles this situation conservatively by simply dropping link i from the set of links to be concurrently activated.

We use (8) to reformulate the constraint C_2 as a lower bound on p_i as follows:

$$p_i \geq \frac{1}{\alpha_i} \frac{\mathrm{SNR}_{th}}{1 + \mathrm{SNR}_{th}}. \tag{12}$$

Therefore, the constraints on p_i become:

$$P_{min} \triangleq \frac{\mathrm{SNR}_{th}}{1 + \mathrm{SNR}_{th}} P_{max} \leq p_i \leq P_{max}. \tag{13}$$

With α_i set to $1/P_{max}$, the computed $p_i, i = 1, 2, \ldots$, are guaranteed not to exceed P_{max} (based on (8)). It is possible, however, that one or more of the computed powers may fall below P_{min}. If this happens, one of several possible approaches can be taken to remedy the situation, depending on whether the computation is done when a link is first being considered for admission or if this is the final power computation that is performed by the first scheduled receiver (denoted as the *master receiver*). In the first case, a link is trying to admit itself, but the resulting NE leads to an infeasible set of powers. Accordingly, this link will not be allowed to transmit, i.e., the corresponding receiver will send a negative CTS (NCTS) packet [2] . In the second case, the master receiver checks the admission rule sequentially according to the orders by which links were admitted. For example, if the computed $\mathbf{P}^*$ is feasible (i.e., admission rule is satisfied) for link 1, links $\{1, 2\}$, ..., and links $\{1, 2, \ldots, L-1\}$, but is infeasible for links $\{1, 2, \ldots, L\}$, then the master receiver will drop link L (assigning it zero transmission power). For this reason, a selfish link has no incentive to violate the admission rule.

It should be noted that the fixed lower bound P_{min} is derived from the coupled (SNR) constraint, and the derivation is valid only when the TPs are set to their best response values. Such a bound would no longer be valid if we were to set the TP of some infeasible link j (with $p_j < P_{min}$ before power adjustment) to P_{min}, since in this case the lower bounds on the TPs of other feasible links (given by $\left(\frac{\sum_{j \neq i} h_{ji} p_j + \sigma^2}{h_{ii}}\right) \mathrm{SNR}_{th}$ for link i) are now functions of the various TPs, and any of these power-dependent bounds may be violated as a consequence of increasing p_j to P_{min}. The following example illustrates the situation. Consider a scenario with two links. Suppose that the computed TP of link 2 (p_2) was found to be less than P_{min}. If we set p_2 to P_{min} and recompute the NE for both links, we will likely end up with a higher value for p_1 than in the first NE computation (because link 1 will need to increase its power to overcome the extra interference due to increasing p_2). The newly computed p_1 may not make it possible for link 2 (with $p_2 = P_{min}$) to satisfy its SNR constraint. In other words, setting p_2 to P_{min} and recomputing the NE does not guarantee that the resulting NE will satisfy the SNR constraint (which depends on the TPs of all links). For this reason, if the first NE computation produces $p_2 < P_{min}$, we simply drop link 2 from the considered links.

Note that our approach to find NE is different from that proposed by Rosen [22]. Rosen's framework requires the strategy set to be nonempty, which is not always satisfied in our setting. This is why we are sometimes forced to drop some links to be able to find a solution for a subset of the competing ones. Another difference is that Rosen's algorithm runs a number of iterations to reach the NE, which typically requires execution time that exceeds the milli-second granularity required for decisions at the MAC layer. We also note that the uniqueness of the NE in our formulation is not a concern because we are interested in finding a feasible solution that allows the competing links to proceed simultaneously.

[2] An alternative to sending the NCTS is not to send back a regular CTS. However, in common CSMA/CA protocols, not receiving the CTS usually triggers a retransmission of the RTS, which would be unnecessary in our case.

3.3 Selecting the Pricing Factor

Our previous analysis was carried out with α_i set to $1/P_{max}$ for all i. In this section, we give insight into the rationale behind this choice. We later study the impact of other choices on the achievable network throughput and energy consumption.

In principle, the pricing factor is intended to drive the NE solution towards a Pareto optimum. Computing the "optimal" value for the pricing factor on a per-link basis requires global information about channel gains, which is not available in our distributed setup. Instead, we derive a range of values for α_i (upper and lower bounds) that is necessary for the existence of a feasible NE and that at the same time limits the energy consumption at the given terminal. We show that $\alpha_i = 1/P_{max}$ falls within this range; at the same time, such a simple, node-independent setting enables us to convert the coupled (SNR) constraint into an easier-to-handle power-bounding constraint.

Proposition 1 *If there exists a feasible power assignment (i.e., one that satisfies both C_1 and C_2) for the transmitters to proceed concurrently, then α_i has to be upper bounded by $\frac{\ln(1+SNR_{th})}{SNR_{th}} y_i$, where $y_i \triangleq \frac{h_{ii}}{\sum_{j \neq i} h_{ji} p_j + \sigma^2}$.*

Proof. See Appendix. □

Fig. 3 illustrates the best response of link i as a function of α_i, with SNR$_{th}$ = 6 dB, P_{max} = 0.03 W, and y_i = 300 W^{-1}. The figure shows that if the pricing factor α_i is set too large, then link i will not be activated.

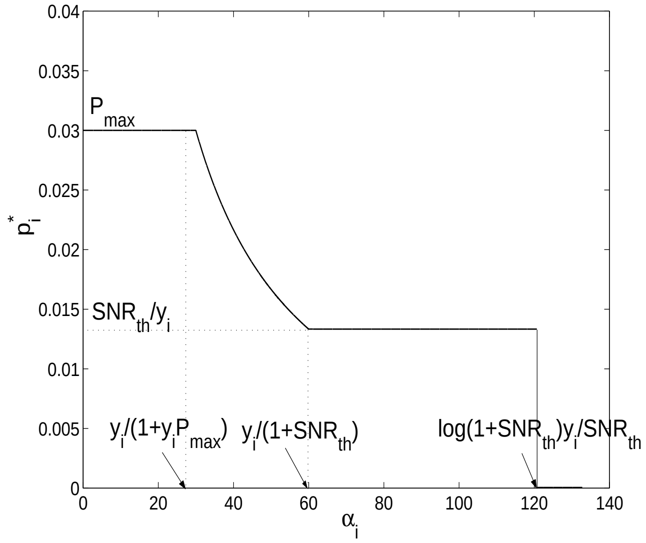


Fig. 3. Best response of link i as a function of α_i.

The goal of the pricing term in the utility function is to prevent terminals from using excessive powers. If the pricing factor is too small, then every terminal will use the largest possible power for transmitting. This will eventually result in high interference on other transmissions. Therefore, α_i will also need to be lower bounded to reduce interference and energy consumption.

Proposition 2 *To prevent terminals from using excessive powers, α_i has to be lower bounded by $\frac{SNR_{th}}{1+SNR_{th}} \frac{1}{P_{max}}$.*

Proof. See Appendix. □

Note that Proposition 2 provides a loose lower bound on α_i. If $\frac{\mathrm{SNR}_{th}}{1+\mathrm{SNR}_{th}} \frac{1}{P_{max}} \leq \alpha_i \leq \frac{y_i}{1+y_i P_{max}}$, then $p_i^* = P_{max}$. From the previous two propositions, we select α_i as follows.

Proposition 3 *For $SNR_{th} \geq 3$ dB, selecting $\alpha_i = \frac{1}{P_{max}}$ satisfies the bounds in Propositions 1 and 2, and hence facilitates achieving the NE.*

Proof. See Appendix. □

Note that the SNR regime addressed in Proposition 3 covers many practical non-spreading scenarios.

The effect of the pricing factor on the system throughput and energy consumption will be demonstrated in Section 5.

4 Proposed GMAC Protocol

In this section, we design a distributed, game-theory-inspired MAC protocol (GMAC) for improving the spatial throughput in a MANET. Terminals within a neighborhood use the previously presented game-theoretic approach to contend for the channel and compute their transmission powers that achieve the NE (note that multiple neighborhoods may exist simultaneously). We first describe our system model and assumptions, and then provide details of the GMAC protocol.

4.1 Protocol Overview

Unlike the IEEE 802.11 scheme, GMAC does not use RTS/CTS control packets to silence neighboring terminals. Instead, these packets are used to broadcast interference and channel-gain information that can be used by overhearing terminals to decide the feasibility of concurrent transmissions and their corresponding transmission powers. To assert their intentions to transmit, terminals exchange control packets over a certain time duration, referred to as the access window (AW). The AW allows several pairs of neighboring terminals to exchange their control packets so that data transmissions can proceed concurrently (see Fig. 4). Using an AW for contention was originally proposed in the MACA-P protocol [2] and was later integrated into the design of POWMAC [16]. In GMAC, we exploit the AW differently, as will be explained in Section 4.2.

The AW consists of several fixed-duration access slots, whose number is adjusted dynamically according to the network load. The transmission powers of contending terminals are not assigned until the AW is over. This mitigates the effect of heuristically presetting the tolerable interference (as in POWMAC [16]) and results in better use of the network capacity.

GMAC protects ACK packets by sending them sequentially at power P_{max} after data transmissions are completed. The order in which ACKs are transmitted corresponds to the order in which the corresponding terminals appear in the AW.

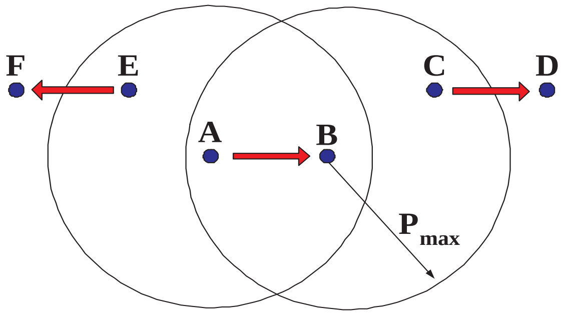


Fig. 4. An example of three concurrent transmissions.

The protocol allows several *clusters* (regions where multiple links contend for channel access) to be formed dynamically in a multi-hop network [3] . The transmitters in the cluster will not be aware of their final data packet transmission powers until announced by the "cluster head." The purpose of this design is to maximize the spatial throughput in each cluster in order to improve the throughput over the entire network.

4.2 Operational Details

In GMAC, a potential transmitter that senses a free channel and is not aware of any scheduled transmissions is considered a *master sender*. The target receiver of a master sender is called a *master receiver*. We refer to a link that involves a master sender and its receiver as a *master link*. All other communicating terminals in the same vicinity of a master link are *slave terminals*. The region formed by the transmission range of a master receiver constitutes a cluster in which the master receiver is its head. All terminals inside this cluster other than the master sender are called *in-cluster* slave terminals. Such terminals can hear and correctly decode the master receiver's CTS packet. Slave terminals that are outside the master receiver's cluster but are within the master sender's transmission range are called *out-cluster* slave terminals. Below, we describe the operation of each type of terminals. Throughout, we assume that α_i is fixed to $\frac{1}{P_{max}}$.

[3] Note that no actual clustering takes place.

4.2.1 Master sender

Consider a master sender A that has a data packet to transmit to another terminal B. If A does not sense the carrier for a randomly selected backoff duration, it sends an RTS at power P_{max}. This RTS packet includes the backoff duration that preceded the sending of the RTS, the MAC data packet size, and the remaining number of slots (N_{AW}) in the AW (how N_{AW} is determined will be explained in Section 4.3). For now, we assume that the MAC data packet size is fixed for all transmissions. We later discuss the handling of variable-packet sizes. Upon receiving the RTS, the master receiver B estimates the channel gain h_{AB} based on the received power. It then estimates the total noise-plus-interference at the time the RTS is received. Since there is no way to precisely predict out-of-range interference in a distributed environment, receiver B inflates the estimated interference by a certain margin to account for future out-of-range interference. A similar approach is often used in cellular networks for an analogous problem [21]. We use σ^2 to represent the inflated value. The same estimation and inflation procedures are repeated by all receivers. Terminal B then calculates the power p_A^* that maximizes the utility function in (1) assuming that only one transmission ($A \rightarrow B$) will take place in B's cluster. Accordingly,

$$p_A^* = \frac{1}{\alpha_i} - \frac{\sigma^2}{h_{AB}}. \tag{14}$$

If $p_A^* < P_{min}$, then the existing interference at receiver B is too high and the transmission should not be allowed to proceed. In this case, terminal B will respond with an NCTS, informing A that it cannot proceed with its transmission. On the other hand, if $p_A^* \geq P_{min}$, B will send back a CTS containing the values of h_{AB}, N_{AW}, and the data packet size. It should be noted that the computed p_A^* at this point is not necessarily the transmission power that A will eventually use to send its data packet. The final transmission power used by A will not be decided until the end of all negotiations in the AW.

Upon receiving B's CTS, terminal A replies back with a DTS (decide-to-send) packet that includes the channel gain h_{AB}. The DTS is needed to inform out-cluster slave terminals about the success of the RTS/CTS exchange between A and B. The 3-way (RTS/CTS/DTS) handshake is depicted in Fig. 5.

4.2.2 In-cluster slave terminals

Consider an in-cluster slave terminal, say terminal C in Fig. 4. Suppose that C overhears B's CTS packet and has a data packet to send. It selects an idle slot in the ongoing AW and backs off for a random duration of time. If no carrier is sensed, it sends an RTS packet at power P_{max}. Terminal C's RTS will include h_{AB}, obtained from the previous RTS/CTS/DTS exchange between A and B. It will also include h_{CB} to be used by receiver D. When D receives the RTS, it first calculates h_{CD}. If D previously overheard A's DTS, it may have already computed h_{AD}; otherwise, h_{AD} is set to zero (this means that D is out of the maximum transmission range of A). Terminal D then calculates the NE power vector $\mathbf{P}^* = [p^*_{AB} \; p^*_{CD}]$, assuming that two transmissions $A \rightarrow B$ and $C \rightarrow D$ will take place simultaneously. If the computed powers are feasible, i.e., satisfy the feasibility condition in (13), D sends back a CTS that includes h_{CD} and h_{AD}. Otherwise, if either p^*_{AB} or p^*_{CD} is infeasible, or if the matrix $\mathbf{H}$ is non-invertible, D sends back an NCTS. Upon receiving a CTS from D, terminal C sends a DTS that includes h_{CD} and h_{AD}. This channel gain information will be later used by the master receiver to compute the final transmission powers for all transmitters within its cluster. If more transmissions are to be scheduled following the RTS/CTS/DTS exchange between C and D, the same procedure is repeated.

In general, the RTS of any transmitter contains the channel gains between that transmitter and all receivers previously scheduled in the same AW. Each receiver needs to make an admission decision by calculating the NE power vector $\mathbf{P}^*$, assuming that all previously scheduled links and its own link will proceed concurrently. This receiver will then send back a CTS if a feasible solution exists. The CTS contains all the channel gains between the current receiver and the transmitters of all previously scheduled links. Finally, the transmitter sends a DTS that announces the channel gains included in the CTS to be used by the master receiver.

The above serialized admission phase is necessary to allow nodes that are in the vicinities of slave nodes but are not in the vicinity of the master receiver to determine whether they can create new clusters or not. Note that it is natural to adopt a non-cooperative game-theoretic approach during this phase, as each contending link operates separately from other links using only local information about links in its receiver's vicinity. Iterative power control, typically used in game-theoretic approaches, is not used here due to its complexity, which involves performing intra-packet power adjustments.

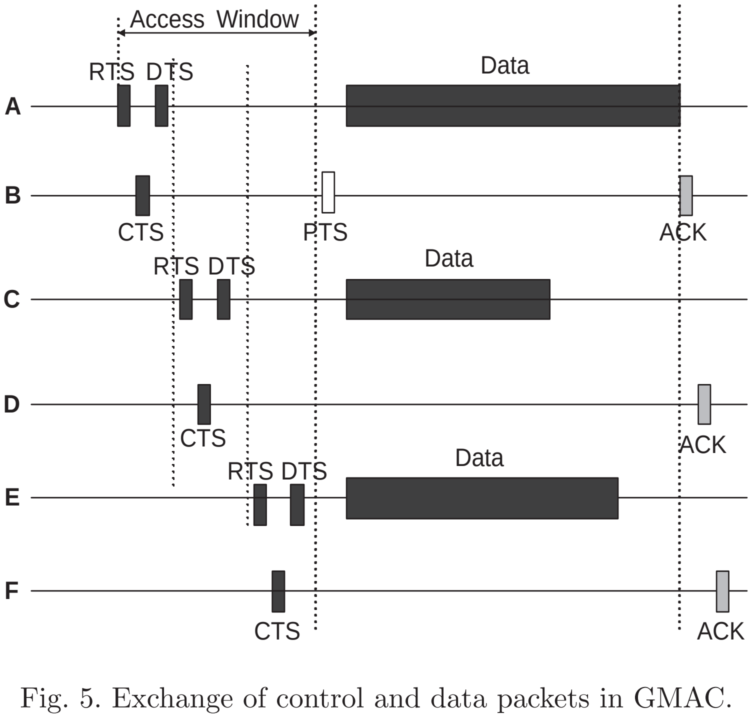


Fig. 5. Exchange of control and data packets in GMAC.

4.2.3 Final computation of transmission powers

After the master receiver (cluster head) receives information about all scheduled in-cluster transmissions, it computes the NE power values for all transmitters using (11). In the unlikely event that $\mathbf{H}$ becomes non-invertible, the master receiver drops the link that causes this non-invertibility from the set of links to be concurrently activated. Because of the unlikeliness of such a scenario, the impact of dropping a link on the overall spatial reuse is negligible (in fact, we have not seen this scenario in any of our simulation experiments). If the computed power vector is infeasible, the master receiver checks the admission rule sequentially according to the orders by which links were admitted, and drops the link which results in an infeasible power vector. This is to ensure that a selfish link has no incentive to violate the admission rule.

The cluster head will then broadcast a power-to-send (PTS) packet, informing all in-cluster transmitters (including the master sender) of the final powers to be used for transmitting their data packets. It is worth noting that by the time the final TPs are determined, the master receiver will have enough information about in-cluster terminals to enable it to optimize their TPs with respect to a *global* utility function (e.g., cooperatively maximize the sum of their utilities). Yet, we do not pursue such a strategy, primarily because the resulting optimization problem is non-convex (the non-convexity stems from the TP variables in the denominator of the log term in the utility function), making it infeasible to determine the optimal solution. Instead of determining a heuristic (sub-optimal) solution for a non-convex global utility function, we opt to compute an optimal solution for the convex local utilities. Note that in our setup any feasible solution (i.e., one that admits all contending links) is acceptable, so a globally optimal solution is not actually needed.

4.2.4 Out-cluster slave terminals

The data transmissions of out-cluster slave transmitters do not add significant interference to the master receiver. Therefore, the strategy that we propose for such terminals is different from the one for in-cluster slave terminals. An out-cluster transmitter, say terminal E in Fig. 4, only needs to compute its own utility without worrying about the interference it causes to the master receiver. Terminal E assumes that transmitters of previously scheduled links will use the maximum transmission power (P_{max}) for their transmissions. It computes its transmission power accordingly. After a randomly selected backoff period, terminal E sends an RTS packet. The RTS does not need to include the channel gain information as before. However, it needs to specify that A's transmission has been scheduled. Receiver F assumes that transmitter A will transmit at power P_{max} and the received power p_{AF} is considered background noise. Thus, the transmission power that E should use can be computed as follows:

$$p^*_{EF} = \frac{1}{\alpha_i} - \frac{\sigma^2 + h_{AF}P_{max}}{h_{EF}}. \tag{15}$$

If the computed p^*_{EF} is feasible, terminal F will send back a CTS. The power p^*_{EF} should be included in this CTS. After receiving this CTS, terminal E will send a DTS that includes p^*_{EF}. This p^*_{EF} is the final data transmission power of terminal E in this scenario.

4.2.5 ACK transmissions

By overhearing the CTS/DTS packets, each terminal in the vicinity of scheduled transmissions will have knowledge of the time required for the schedule to be completed. Such a terminal will not contend for the channel until the data and ACK packets of the overheard schedule are completely transmitted. For scheduled transmissions, ACK packets will be serially transmitted using P_{max} after all data transmissions are completed. As previously mentioned, the order of the ACK transmissions will conform to the order of their schedules in the AW. Therefore, ACK transmissions within a cluster do not interfere with each other.

4.2.6 Contention resolution

We adopt a similar approach to the Proportionally Fair Contention Resolution (PFCR) algorithm proposed in [17]. The PFCR algorithm uses a *persistent* mechanism for contention resolution instead of the *backoff* mechanism used in the IEEE 802.11 scheme. It operates as follows. Each terminal can be in one of three possible states: NO-CONTEND, CONTEND and TRANSMIT. When a terminal i has a packet to send, it has to wait until the start of the next available slot in the AW. It then senses the channel. If no carrier is sensed, terminal i will switch its state from NO-CONTEND to CONTEND with probability p_i (how p_i is updated is explained below). Terminal i then chooses a random wait time B_i that is uniformly distributed in the interval $[0, B]$, where B is a fixed system parameter. If terminal i senses a busy channel during this wait time B_i, it switches its state from CONTEND to NO-CONTEND, and p_i is updated as $p_i = (1-\beta)p_i + \gamma$, where β and γ are system parameters. Otherwise, the state of terminal i is switched to TRANSMIT, which means that i can now send RTS packets to try to acquire the channel. In the mean time, p_i is updated as $p_i = p_i + \gamma$. The persistent approach was shown to be more robust and efficient than the backoff approach [17]. It also ensures proportional fairness among users.

Since B is a system parameter and is known by all the terminals in the network, using the PFCR algorithm ensures that the size of the access slot (AS) is fixed. As shown in Fig. 6, the size of each AS is B + ‖RTS‖+‖CTS‖+‖DTS‖ + 3 SIFS, where SIFS denotes the short interframe spacing between successive control packets.

Because B_i is uniformly distributed in $[0, B]$, transmitter i will have to include this B_i in its RTS packet. As a result, all i's neighbors will be aware of the start and duration of the AS. In other words, all terminals receiving this RTS will be locally synchronized. Similarly, receiver j will also need to include this B_i in its CTS packet.

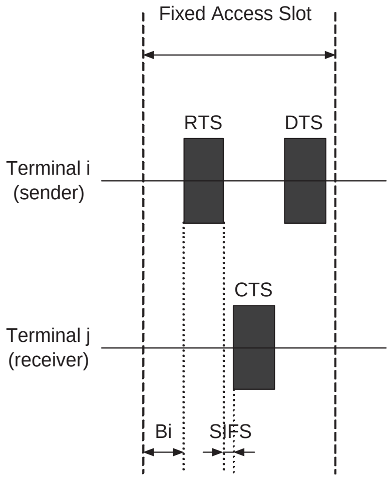


Fig. 6. Access slot in the GMAC protocol.

4.3 Computing the AW Size

Each terminal i will store an initial value for the AW, which is to be used if i acts as a master sender. As described below, the AW size of i will be updated (increased or decreased) according to the load in i's vicinity. The choice of the AW size should aim at maximizing the number of concurrent transmissions, without wasting network resources. GMAC adopts an AW adaptation approach similar to that of POWMAC [16]. After each data transmission, every terminal checks the number of concurrent transmissions in its vicinity. If this number is less than a prespecified fraction (say $\delta\%$) of the AW size, then the current AW size is too large for the present load. In this case, the AW size is decreased by one. Similarly, if this number is larger than $\delta\%$ of the current AW size, then AW size will be increased by one to possibly allow for more concurrent transmissions.

4.4 Terminals Within Two Master Clusters

In a multi-hop network, there may be cases where a slave terminal, say C, resides within two different clusters. This situation is exemplified in Fig. 7, where the transmissions $A \rightarrow B$ and $A' \rightarrow B'$ are scheduled master links that have not started their data transmissions yet. The AWs of the two master receivers may not be synchronized, as shown in Fig. 8. If terminal C wishes to transmit, it has two options. One option is to wait until the two master links finish their data transmissions. Another option is to determine whether the misalignment between the two AWs is less than the maximum backoff window (B). If so, then terminal C can send its RTS and compete for the channel. The problem with the second option is that even if the misalignment is smaller than B, allowing $C \rightarrow D$ to proceed will reduce the possibility of admitting future transmissions in both of these two clusters. So this option is only beneficial under low traffic conditions, where the possibility of having future transmissions is low so that the situation depicted in Fig. 7 plays a big role in system throughput. In our design, we adopt the first option.

4.5 Fairness Issues

At first, one might think that the non-uniqueness of our NE solution leads to unfairness in the operation of the protocol. However, this is not the case, because a prospective link automatically drops itself out if no feasible solution is found when considering links that have previously contended for the channel during the current access window. So the earlier a link captures an access slot, the better are its chances of being admitted. Because each contending sender must back off randomly before sending its RTS packet in a given access slot of a given access window (this also includes the master sender), nodes with packets to transmit have the same likelihood of capturing that slot. Whichever transmitting node captures the slot is guaranteed to proceed if its NE solution (considering itself and previously scheduled transmissions) is feasible. Accordingly, the protocol is as fair as the classic CSMA/CA scheme.

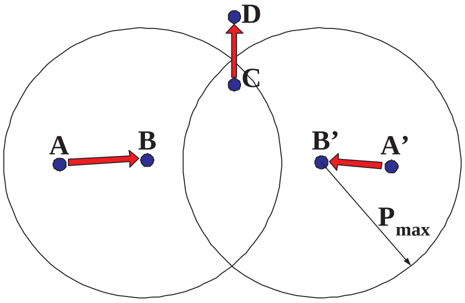


Fig. 7. Example of a slave terminal that falls in two concurrent clusters.

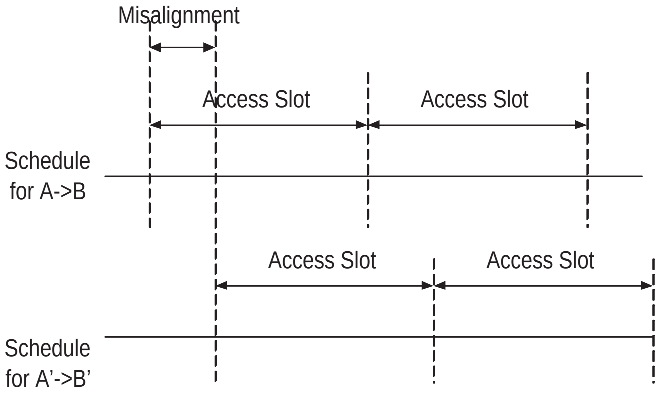


Fig. 8. Two unsynchronized schedules (based on the topology in Fig. 7).

4.6 Protocol Overhead

We now analyze the overhead of the GMAC protocol and compare it with the 802.11 scheme. Let S_c and S_d denote the sizes of the control and data packet, respectively. The corresponding transmission durations are $\frac{S_c}{R^*}$ and $\frac{S_d}{R^*}$, where R^* denotes the transmission rate that corresponds to SNR$_{th}$. For simplicity, we ignore the small guarding interval (SIFS) in our calculations. Assume that the size of the AW in GMAC is N slots and the average number of concurrent transmissions is m, where $1 < m \leq N$. The total time it takes to transmit m data packets according to GMAC is $T_{GMAC} = N(3S_c/R^* + B) + (S_c + S_d + mS_c)/R^*$. In the 802.11 scheme, one RTS and one CTS packets are exchanged ahead of each data packet. Therefore, the total time needed to transmit m data packets is $T_{802.11} = m(2S_c + S_d + S_c)/R^*$.

Accordingly, GMAC outperforms the 802.11 protocol if $T_{GMAC} < T_{802.11}$. The throughput improvement factor of GMAC over the 802.11 protocol is defined as $K \overset{\text{def}}{=} \frac{T_{802.11}}{T_{GMAC}}$. With some manipulation, K can be expressed as:

$$K = \frac{3mS_c/S_d + m}{(3N + m + 1)S_c/S_d + 1 + N\frac{B}{S_d/R^*}} \tag{16}$$

K becomes larger as the ratio S_c/S_d decreases. As S_c/S_d goes to zero, the throughput improvement approaches $K = m$. Note that B is much smaller than the data packet duration S_d/R^*. In order to have $K > 1$, we need:

$$\frac{S_c}{S_d} < \frac{m - 1}{3N - 2m + 1} - \frac{N}{3N - 2m + 1}\frac{B}{S_d/R^*}. \tag{17}$$

We will further study the relationship between K and the ratio (S_c/S_d) in Section 5.

The overhead of GMAC is similar to that of POWMAC [16]. In order for GMAC to outperform POWMAC, we need the average number of concurrent transmissions in GMAC to be larger than that of POWMAC. It was shown through the example in Section 1 that this can happen. We later investigate the throughput improvements of GMAC and POWMAC through simulations.

4.7 Variable-size Packet Transmissions

Throughout the paper, we assumed that S_d is fixed. We now outline two approaches to handle variable-packet sizes via intelligent scheduling (The details of these approaches are discussed in [24]). Both approaches are based on sending multiple data packets back-to-back during a fixed interval, which we refer to as a *data window*. The first approach employs intelligent scheduling to embed packets from different links in order to maximize the number of admitted links. The second approach allows gap-filling using packets from already admitted links with backlogged transmission queues. The second approach is less complex. However, the first approach is more attractive for interactive applications, where bandwidth sharing among different participants is more desirable than just boosting the network throughput.

Table 1
Simulation parameters

Data Packet Size	2 KB
Transmission rate	1 Mbps
SINR threshold	6 dB
Maximum transmission power	15 dBm
Maximum transmission range	750 meters
Maximum carrier-sense range	1500 meters
Path loss factor	4
Number of terminals	100

5 Performance Evaluation

We now evaluate the performance of the GMAC protocol and compare it with two single-channel protocols: POWMAC [16] and IEEE 802.11. Our performance metrics are the network goodput (number of successfully received bytes per time unit), the number of concurrent transmissions, and the total energy consumption per delivered packet (accounting for both data and control packet transmissions). We show the performance under different node densities, traffic models, packet sizes, and pricing factors.

5.1 Simulation Setup

We conduct simulations using the CSIM simulation package [13]. Our simulator captures the behavior of the physical and MAC layers of a wireless network. At the physical layer, each node estimates the total interference and checks the received SNR to determine whether the packet is correctly received or not. For the MAC layer, the simulator implements the 802.11 scheme, POWMAC, and our proposed GMAC.

The simulation parameters are provided in Table 1. These parameters correspond to the Cisco Aironet Series 350 hardware specifications [6]. We use the two-ray propagation model with a path loss factor of 4. The network area is divided into 10×10 equal-sized squares, one for each terminal. The location of each terminal is randomly assigned within its square. Each terminal generates packets according to a Poisson process with rate λ. We conduct experiments for “single-neighborhood” and “multi-neighborhood” configurations, as described below.

5.2 Single-Neighborhood Network Configuration

In some scenarios, such as computers in a conference room, each terminal is within the maximum transmission range of all other terminals. In other words, all the exchanged control packets, which are sent using P_{max}, can be received by each terminal. We model this scenario in our simulations by taking the field size to be $500m \times 500m$. For each generated packet, the destination is randomly selected.

Fig. 9 depicts the network goodput versus λ for the three examined protocols. The figure shows that GMAC achieves about 80% improvement in network goodput over the 802.11 scheme, and about 40% improvement over POWMAC. Fig. 10 depicts a histogram of the number of concurrent transmissions (m) for both POWMAC and GMAC. It is clear that GMAC achieves a larger number of concurrent transmissions than POWMAC for all values of $m \geq 2$. This explains the goodput difference between the two protocols. Note that in this configuration, the 802.11 scheme allows for only one transmission at one time.

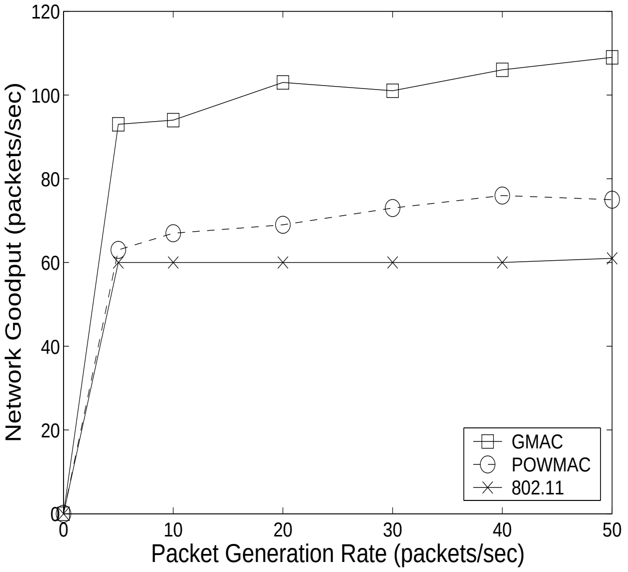


Fig. 9. Network goodput vs. traffic load for the single-neighborhood configuration.

5.3 Multi-Neighborhood Network Configuration with a Fixed Field Size

We now examine a more general ad hoc network, whereby terminals can be out of range from each other, leading to hidden-terminal problems. Specifically, we place 100 terminals within a square area of length 1500 meters. The square is split into 100 smaller squares, one for each terminal. The location of a terminal within each small square is randomized. For each generated packet, the destination is randomly selected from the one-hop neighbors of the sender. Fig. 11(a) depicts the network goodput versus λ. It shows that GMAC can achieve up to 70% increase in goodput over 802.11 and up to 25% increase over POWMAC. This improvement is due to the increase in the number of concurrent

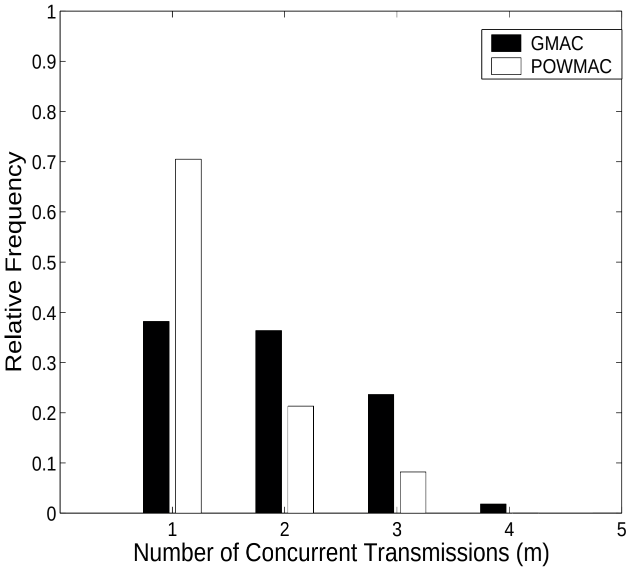
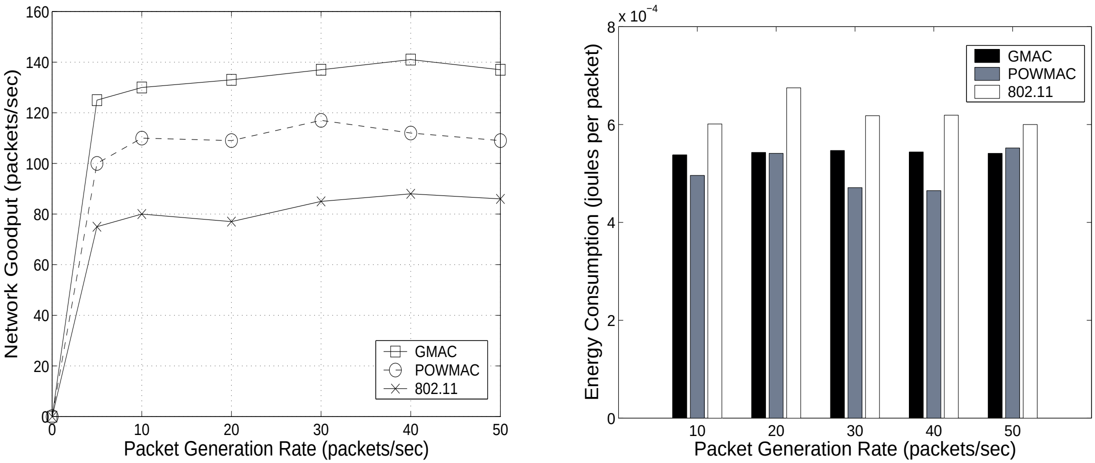


Fig. 10. Histogram of concurrent transmissions (single neighborhood).

transmissions. Fig. 11(b) depicts the energy consumption versus λ for the three protocols. It is clear that the energy consumption associated with GMAC is comparable to that of POWMAC and 802.11.



(a) Network goodput vs. λ

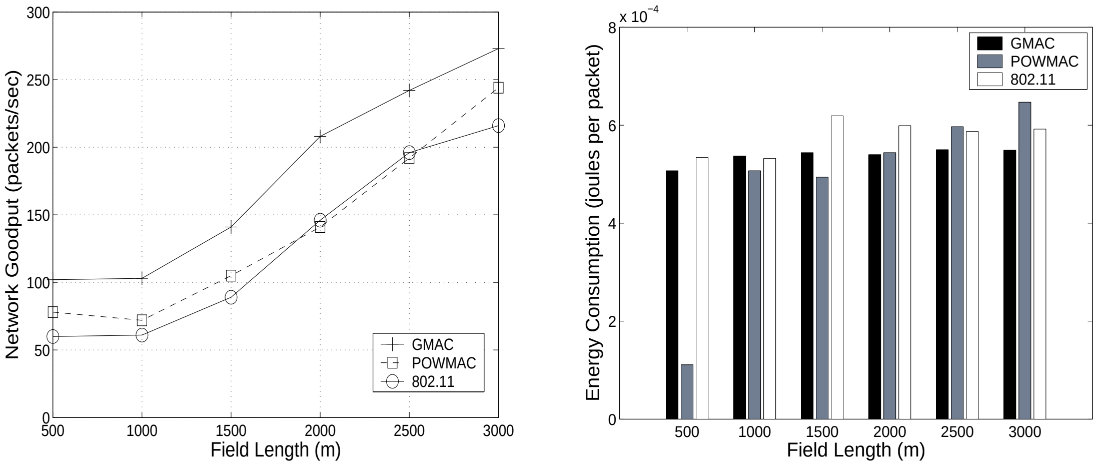
(b) Energy consumption vs. λ

Fig. 11. Performance for a multi-neighborhood configuration under a fixed field size.

5.4 Multi-Neighborhood Network Configuration with a Variable Field Size

In this scenario, we vary the length of the square field while fixing the number of terminals (i.e., we vary the node density). The packet generation rate is fixed at 40 packets/sec. The achieved goodput is shown in Fig. 12(a). In this scenario, GMAC shows consistent goodput improvement over both 802.11 scheme and POWMAC, especially under high densities. At the same time, the energy consumption of GMAC is comparable to both the

802.11 scheme and POWMAC, as shown in Fig. 12(b).



(a) Network goodput vs. field size

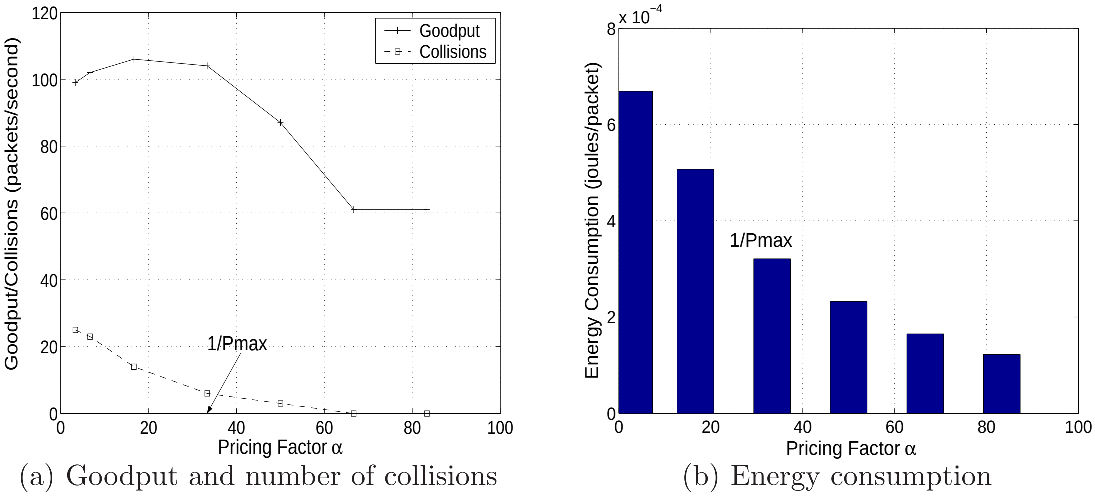
(b) Energy consumption vs. field size

Fig. 12. Performance as a function of node density (field size varied).

5.5 *Effect of the Pricing Factor* α_i

Optimal selection of α_i relies on the channel conditions and the transmission powers of all the links. For optimal performance results, every link i needs to compute the optimal value of its α_i online (i.e., on a per packet basis). This is difficult since information about other transmission powers is not available to link i during the contention phase. Our analysis in Section 3 showed that $\alpha_i = \frac{1}{P_{max}}$ facilitates achieving NE. However, using a fixed value for α_i may not result in optimal performance. To demonstrate the effect of α_i on the performance, we vary α_i and report the achieved goodput and total energy consumption. Note that in our analysis, we showed that a good choice of α_i is in the range $\frac{\mathrm{SNR}_{th}}{1+\mathrm{SNR}_{th}} \frac{1}{P_{max}}$ to $\frac{\log(1+\mathrm{SNR}_{th})}{\mathrm{SNR}_{th}} y_i$. For $P_{max} = 15$ dBm and $\mathrm{SNR}_{th} = 6$ dB, the lower bound corresponds to 25.3, and the upper bound is a function of y_i. So by varying α_i from 0 to 100, we try to cover the range of all interesting α_i selections and find the best α_i. For this experiment, we set $\lambda = 40$ packets/sec and take the field size to be 500m×500m.

Fig. 13(a) indicates that the overall system goodput is maximized when α_i is slightly smaller than $1/P_{max}$. However, as α_i decreases, the number of collisions increases, i.e., the protocol becomes more aggressive. This consequently results in an increase in the total energy consumption, as shown in Fig. 13(b). Intuitively, a smaller α_i allows more links to proceed concurrently, but this does not necessarily guarantee the maximum network goodput because more collisions may occur. Although $\alpha_i = 1/P_{max}$ does not maximize the system goodput, it provides a reasonable tradeoff between throughput and energy consumption. Adaptive setting of α_i is left for future work.



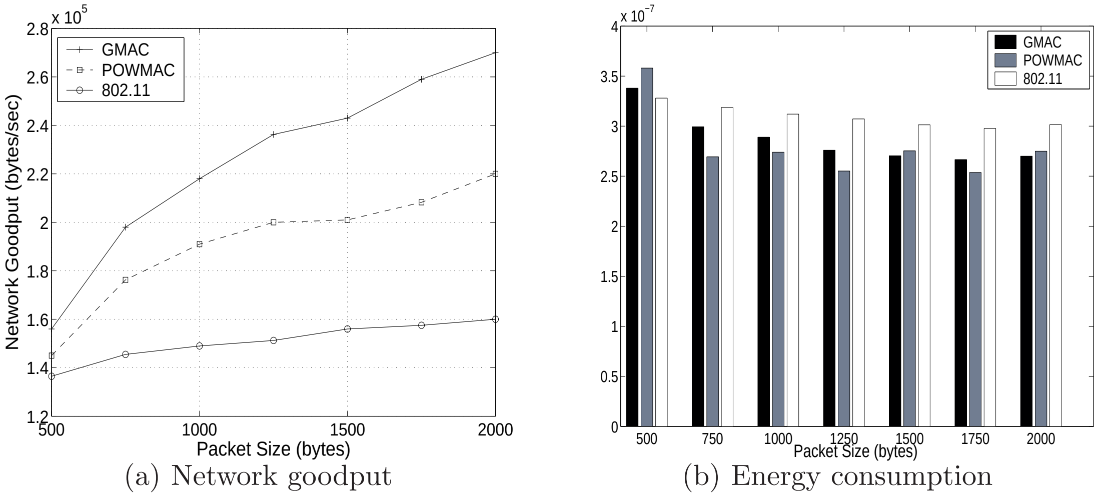
(a) Goodput and number of collisions

(b) Energy consumption

Fig. 13. Effect of α_i on the performance of GMAC.

5.6 Effect of Packet Size

We now study the effect of the data packet size on the network goodput. In this experiment, the field size is fixed to 1500m×1500m. We set $\lambda = 20$ packets/sec and $S_c = 20$ bytes. Fig. 14(a) shows the improvement in network goodput for GMAC as the data packet size increases. Although both POWMAC and GMAC depict the same trend, GMAC outperforms POWMAC by allowing more concurrent transmissions to take place. Fig. 14(b) shows the total energy consumption for different packet sizes and illustrates that more energy per byte is consumed for smaller packet sizes. This is not surprising since the percentage of energy spent in transmitting control packets increases when data packet sizes decrease.



(a) Network goodput

(b) Energy consumption

Fig. 14. Performance under different packet sizes.

5.7 Comparison with Centralized Scheduling

In this section, we show that even though GMAC can significantly improve goodput, its scheduling of transmissions is still suboptimal when compared to an optimal scheduling performed by a *centralized* controller. The controller is assumed to be aware of all the terminals and requests, and can completely regulate all the traffic. It uses exhaustive search to select the largest possible number of feasible transmissions and allow them to proceed concurrently.

In this simulation experiment, 10 terminals are randomly placed in a single-neighborhood network configuration of size 500m×500m. The resulting goodput is shown in Fig. 15. The difference in goodput between the optimal solution and GMAC is due to three main reasons. First, GMAC does not have global knowledge of the entire network (e.g., channel gains between all contending terminals) in order to schedule the transmissions accordingly. Second, instead of using a complex optimization method, GMAC uses a low-complexity, game-theory-motivated heuristic algorithm to achieve a feasible solution. The heuristic nature of the algorithm lies in how terminals are admitted into the AW and when the final power decisions are made. Finally, the exchange of control packets wastes a proportion of system resources and consequently reduces goodput. Nonetheless, for a distributed solution, GMAC performs reasonably well in dense networks and can improve the throughput over conventional schemes by up to 80%.

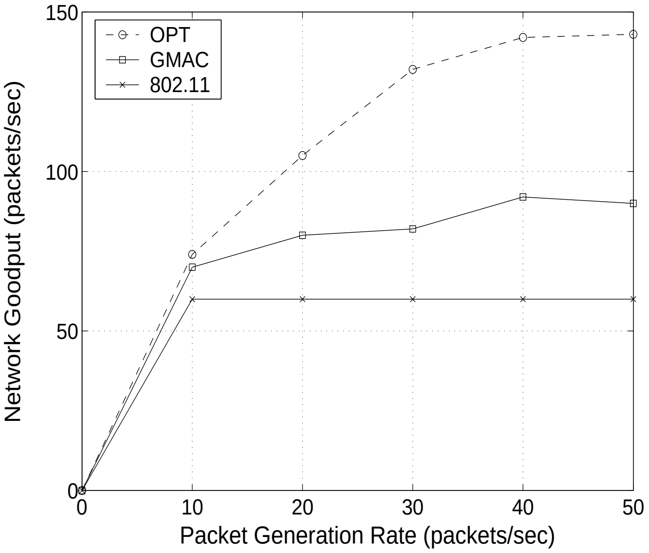
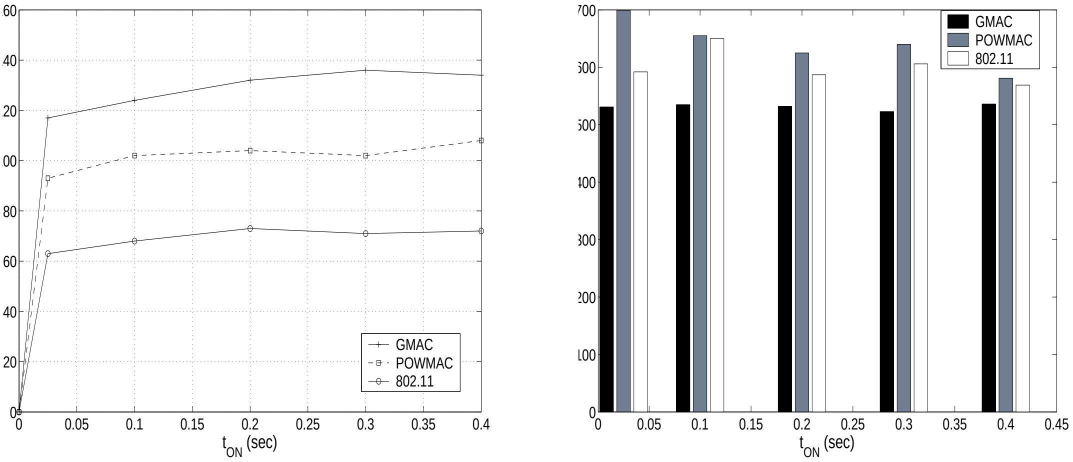


Fig. 15. Comparison with optimal centralized scheduling.

5.8 Effect of Traffic Model

To study the effect of the traffic model, we replace the Poisson traffic generator by an Interrupted Poisson Process (IPP) that is characterized by an alternating ON/OFF pattern. Each terminal has two states: ON and OFF. The time spent in the ON state is exponentially distributed with parameter t_{ON} and the time in the OFF state is exponentially distributed with parameter t_{OFF}. During the ON period, packets are generated according to a Poisson process with parameter λ. We fix $\lambda = 40$ packets/second and $t_{OFF} = 0.1$ sec, and vary t_{ON}. The field size is 1500m×1500m, and the number of nodes is set to 25. The system goodput is shown in Fig. 16(a). The figure shows that GMAC outperforms 802.11 protocol and POWMAC in all scenarios. Fig. 16(b) shows the total energy consumption and demonstrates that the three protocols consume comparable amounts of energy.



(a) Network goodput

(b) Energy consumption

Fig. 16. Performance under IPP traffic generation model.

6 Conclusions

In this paper, we proposed a game-theoretic power control MAC protocol (GMAC) for improving the throughput of a MANET. GMAC uses a single channel for both data and control packets. It allows each user to determine whether or not it is feasible to transmit concurrently with previously scheduled transmissions. GMAC enables multiple transmissions to proceed concurrently by computing the NE powers for all contending transmitters.

We compared the performance of GMAC with the IEEE 802.11 scheme and the POWMAC scheme. Our simulation results show that GMAC significantly improves the network goodput over both schemes. In some scenarios, the network goodput under GMAC was 80% (40%) larger than that of the 802.11 (POWMAC) scheme. GMAC also maintains comparable energy consumption to both POWMAC and the 802.11 scheme.

References

[1] http://www.ietf.org/html.charters/manet-charter.html.

[2] A. Acharya, A. Misra, and S. Bansal. MACA-P: A MAC for concurrent transmissions in multi-hop wireless networks. In *Proceeding of the First IEEE PerCom 2003 Conference*, volume 2, pages 59–66, March 2003.

[3] S. Agarwal, S. Krishnamurthy, R. Katz, and S. Dao. Distributed power control in ad-hoc wireless networks. In *Proceedings of the International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pages 59–66, 2001.

[4] T. Alpcan, T. Basar, R. Srikant, and E. Altman. CDMA uplink power control as a noncooperative game. *Wireless Networks*, 8(6):659–670, November 2002.

[5] Q. Chen and Z. Niu. A game-theoretical power and rate control for wireless ad hoc networks with step-up price. *IEICE Transactions on Communications*, E88-B(9):3515–3523, September 2005.

[6] The Cisco Aironet 350 Series of wireless LAN, http://www.cisco.com/warp/public/cc/pd/witc/ao350ap.

[7] D. Fudenberg and J. Tirole. *Game Theory.* The MIT Press, 1991.

[8] J. Huang, R. Berry, and M. Honig. A game theoretic analysis of distributed power control for spread spectrum ad hoc networks. In *Proceedings of 2005 IEEE International Symposium on Information Theory*, September 2005.

[9] International Standard ISO/IEC 8802-11. *ANSI/IEEE Std 802.11.* Part 11: wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications.

[10] H. Ji and C.-Y. Huang. Non-cooperative uplink power control in cellular radio systems. *Wireless Networks*, 4(3):233–240, 1998.

[11] V. Kawadia and P. R. Kumar. Power control and clustering in ad hoc networks. In *Proceedings of the IEEE INFOCOM Conference*, April 2003.

[12] M. Krunz, A. Muqattash, and S.-J. Lee. Transmission power control in wireless ad hoc networks: Challenges, solutions, and open issues. *IEEE Network*, 18(5):8–14, September-October 2004.

[13] Mesquite Software Incorporation, http://www.mesquite.com.

[14] J. Monks, V. Bharghavan, and W.-M. Hwu. A power controlled multiple access protocol for wireless packet networks. In *Proceedings of the IEEE INFOCOM Conference*, pages 219–228, 2001.

[15] A. Muqattash and M. Krunz. A distributed transmission power control protocol for mobile ad hoc networks. *IEEE Transaction on Mobile Computing*, 3(2):113–128, April-June 2004.

[16] A. Muqattash and M. Krunz. POWMAC: A single-channel power-control protocol for throughput enhancement in wireless ad hoc networks. *IEEE Journal on Selected Areas in Communications*, 23(5):1067–1084, May 2005.

[17] T. Nandagopal, T. Kim, X. Gao, and V. Bharghavan. Achieving MAC layer fairness in wireless packet networks. In *Proceedings of the ACM MobiCom Conference*, pages 87–98, 2000.

[18] S. H. R. Naqvi and L. Patnaik. A distributed channel access protocol for ad hoc networks with feedback power control. *IEEE Transaction on Mobile Computing*, 5(10):1448 – 1459, Oct. 2006.

[19] M. J. Osborne. *An Introduction to Game Theory.* Oxford University Press, 2004.

[20] R. Ramanathan and J. Redi. A brief overview of ad hoc networks: Challenges and directions. *IEEE Communications Magazine*, 40(5):20–22, May 2002.

[21] T. S. Rappaport and L. B. Milstein. Effects of radio propagation path loss on DS-CDMA cellular frequency reuse efficiency for the reverse channel. *IEEE Transactions on Vehicular Technology*, 41(3):231–242, 1992.

[22] J. B. Rosen. Existence and uniqueness of equilibrium points for concave N-person games. *Econometrica 33 (3) (1965).*

[23] C. U. Saraydar, N. B. Mandayam, and D. J. Goodman. Efficient power control via pricing in wireless data networks. *IEEE Transactions on Communications*, 50(2):291 – 303, February 2002.

[24] F. Wang, O. Younis, and M. Krunz. Throughput-oriented MAC for mobile ad hoc networks with variable packet sizes. In *Proceedings of the IEEE SECON 2006 Conference*, Reston, Virginia, Sep. 2006.

[25] S.-L. Wu, Y.-C. Tseng, and J.-P. Sheu. Intelligent medium access for mobile ad hoc networks with busy tones and power control. *IEEE Journal on Selected Areas in Communications*, 18(9):1647–1657, 2000.

[26] M. Xiao, N. B. Shroff, and E. K. P. Chong. Utility-based power control in cellular wireless systems. In *Proceedings of the IEEE INFOCOM Conference*, volume 1, pages 412 – 421, April 2001.

[27] J. Zhang, Z. Fang, and B. Bensaou. Adaptive power control for single channel ad hoc networks. In *Proceedings of the IEEE ICC Conference*, volume 5, pages 3156 – 3160, May 2005.

.1 Proof of Proposition 1

Intuitively, if the pricing term $\alpha_i p_i$ is too high, then terminals will prefer not to transmit. We consider the best response of player (terminal) i of the original game together with the SNR constraint of link i, given the powers of all other players ($\mathbf{p}_{-i}$). Accordingly, the SNR constraint can be expressed as $p_i y_i \geq \mathrm{SNR}_{th}$, or $p_i \geq \frac{\mathrm{SNR}_{th}}{y_i}$. Combining this equation with the constraint C_1, we get the following inequality:

$$y_i \geq \frac{\mathrm{SNR}_{th}}{P_{max}}. \qquad (.1)$$

Taking $\frac{\partial u_i}{\partial p_i} = 0$, we get $p_i = \frac{1}{\alpha_i} - \frac{1}{y_i}$. By considering the SNR constraint, the best response of player i, p_i^*, becomes:

$$p_i^* = \begin{cases} 0, & \text{if } p_i \leq 0 \text{ or } (0 < p_i < \frac{1}{y_i}\mathrm{SNR}_{th} \text{ and } u_i(\frac{\mathrm{SNR}_{th}}{y_i}) \leq 0) \\ \frac{\mathrm{SNR}_{th}}{y_i}, & \text{if } 0 < p_i < \frac{1}{y_i}\mathrm{SNR}_{th} \text{ and } u_i(\frac{\mathrm{SNR}_{th}}{y_i}) > 0 \\ \frac{1}{\alpha_i} - \frac{1}{y_i}, & \text{if } \frac{1}{y_i}\mathrm{SNR}_{th} \leq p_i \leq P_{max} \\ P_{max}, & \text{if } p_i \geq P_{max} \end{cases} \tag{.2}$$

The case when $0 < p_i < \frac{1}{y_i}\mathrm{SNR}_{th}$ needs to be carefully considered. Link i should decide whether to stay silent ($p_i = 0$) or set $p_i = \frac{1}{y_i}\mathrm{SNR}_{th}$ by comparing the utility functions $u_i(p_i = 0)$ and $u_i(p_i = \frac{\mathrm{SNR}_{th}}{y_i})$. It is obvious that $u_i(p_i = 0) = 0$ and $u_i(p_i = \frac{\mathrm{SNR}_{th}}{y_i}) = \ln(1+\mathrm{SNR}_{th}) - \alpha_i p_i$. If $u_i(\frac{\mathrm{SNR}_{th}}{y_i}) > 0$, the terminal will use the power $\frac{\mathrm{SNR}_{th}}{y_i}$ to transmit; otherwise, there is no feasible solution and the sender of link i should stay silent and wait until the next contention period.

Since p_i is essentially a function of α_i, we can express the best response of link i as a function of α_i, as follows:

$$p_i^* = \begin{cases} 0, & \text{if } \alpha_i > \frac{\ln(1+\mathrm{SNR}_{th})}{\mathrm{SNR}_{th}} y_i \\ \frac{\mathrm{SNR}_{th}}{y_i}, & \text{if } \frac{y_i}{1+\mathrm{SNR}_{th}} < \alpha_i < \frac{\ln(1+\mathrm{SNR}_{th})}{\mathrm{SNR}_{th}} y_i \\ \frac{1}{\alpha_i} - \frac{1}{y_i}, & \text{if } \frac{1}{P_{max}+\frac{1}{y_i}} < \alpha_i < \frac{y_i}{1+\mathrm{SNR}_{th}} \\ P_{max}, & \text{if } \alpha_i < \frac{1}{P_{max}+\frac{1}{y_i}} \end{cases} \tag{.3}$$

The upper bound on α_i is therefore $\frac{\ln(1+\mathrm{SNR}_{th})}{\mathrm{SNR}_{th}} y_i$.

.2 *Proof of Proposition 2*

From Fig. 3, we can see that when α_i is less than $\frac{y_i}{1+y_i P_{max}}$, node i will always try to transmit at power P_{max}. Since $y_i \geq \frac{\mathrm{SNR}_{th}}{P_{max}}$, $\frac{y_i}{1+y_i P_{max}} \geq \frac{\mathrm{SNR}_{th}}{1+\mathrm{SNR}_{th}} \frac{1}{P_{max}}$. Thus, α_i needs to be lower bounded by $\frac{\mathrm{SNR}_{th}}{1+\mathrm{SNR}_{th}} \frac{1}{P_{max}}$ regardless of y_i. Otherwise, node i will always use P_{max} to transmit.

.3 Proof of Proposition 3

For the lower bound, $\frac{\text{SNR}_{th}}{1+\text{SNR}_{th}}\frac{1}{P_{max}} < \frac{1}{P_{max}}$. For the upper bound, since $y_i \geq \frac{\text{SNR}_{th}}{P_{max}}$, it must be that $\frac{\ln(1+\text{SNR}_{th})}{\text{SNR}_{th}}y_i \geq \frac{\ln(1+\text{SNR}_{th})}{P_{max}}$. Since $\text{SNR}_{th} \geq 3$ dB, $\ln(1+\text{SNR}_{th}) > 1$. Therefore, $\frac{\ln(1+\text{SNR}_{th})}{P_{max}} > \frac{1}{P_{max}}$.