Upper Bounding Service Capacity in Multihop Wireless SSMA-Based Ad Hoc Networks

Shirong Du[1], John N. Daigle[1], and Bahram Alidaee[2]

[1] Center for Wireless Communications, Electrical Engineering Department, University of Mississippi, MS 38677, USA
{sdu,wcdaigle}@olemiss.edu
[2] MIS/POM Department, School of Business Administration, University of Mississippi, MS 38677, USA
Balidaee@bus.olemiss.edu

Abstract. Upper bounds on the service carrying capacity of a multihop, wireless, SSMA-based ad hoc network are considered herein. The network has a single radio band for transmission and reception. Each node can transmit to, or receive from, multiple nodes simultaneously. We formulate the scheduling of transmissions and control of transmit powers as a joint, mixed-integer, nonlinear optimization problem that yields maximum return at minimum power subject to SINR constraints. We present an efficient tabu search-based heuristic algorithm to solve the optimization problem and rigorously assess the quality of the results. Through analysis and simulation, we establish upper bounds on the VoIP call carrying capacity of the network as function of various parameters. We discuss the pros and cons of using SSMA as a spectrum sharing technique in wireless ad hoc networks.

Keywords: wireless ad hoc networks, TDMA/SSMA, transmission scheduling and power control, joint optimization problem, optimality, tabu search algorithm, VoIP call carrying capacity, upper bounds.

1 Introduction

The objective of this paper is to establish upper bounds on the VoIP call carrying capacity of the spread spectrum multiple access (SSMA)-based wireless ad hoc network as a function of network topology, network connectivity, routing strategy, spreading gain, transmit power, and overhead. We investigate the impact of various factors on the upper bounds and per-call energy consumption through analysis and simulations.

The spreading codes are Gold codes and they are assigned on a link basis; that is, each link connecting two nodes is assigned a spreading code. Both simultaneous transmission scheduling and transmit power control are accomplished through the solution of a joint optimization problem at each scheduling slot. We focus our analysis on VoIP because it has universally accepted models and quality objectives.

J. Zheng et al. (Eds.): ADHOCNETS 2009, LNICST 28, pp. 94–109, 2010.

A mobile ad hoc network (MANET) is a collection of mobile nodes that organize themselves into a communication network and manage themselves according to a collection of distributed algorithms without the benefit of a pre-existing infrastructure. Since maximum transmit power may be limited, the communication path from a source to its destination is multi-hop, and, in that case, information is forwarded hop-by-hop. Thus, a node is actually a router and a processing station at the same time [1].

The IEEE 802.11 distributed coordination function (DCF) has been proposed as an operation protocol for a MANET. The DCF employs carrier sense multiple access with collision avoidance (CSMA/CA) and a random exponential backoff procedure following a busy medium condition [2]. However, due to its contention and backoff natures, DCF has low bandwidth efficiency and throughput, and also has no mechanisms to guarantee packet delivery within a delay bound [3]. In addition, since the time to transmit the payload of a voice packet is only a small portion of the total time to transmit the packet, the capacity to accommodate voice traffic in DCF-based networks is severely limited.

In [4], the suitability of using the IEEE 802.11 DCF in multi-hop wireless ad hoc networks is investigated via simulation of a MANET with an 8-node string topology. The problems found include serious TCP instability and unfairness, and it was concluded that the DCF is not suitable for a multi-hop network because it does not provide an effective technique for managing the network's wireless resources and cannot provide assured QoS [3],[4].

It is commonly agreed that future infrastructures for multi-hop MANETs should feature efficient management of the wireless medium. Along these lines, direct sequence code division multiple access (DS-CDMA) has been proposed as a means of achieving efficiency [5], [6], [7], [8]. The performance of a controlled access CDMA (CA-CDMA) has been evaluated and compared with that of the IEEE 802.11 scheme [6], [7]. It is shown that CA-CDMA can achieve up to a 280% increase over the throughput of the IEEE 802.11 scheme, but the increase results from an increase in the number of simultaneous transmissions due entirely to power control. It is shown that CA-CDMA requires less than 50% of the energy required by the IEEE 802.11 scheme. In addition, the throughput enhancement increases with node density, because CA-CDMA bounds the transmission power.

In general, at any given time a given node can either be transmitting, receiving or inactive. The decision of which nodes will be in each of the categories at each transmission slot is called scheduling. The problem of transmission scheduling and power control for ad hoc networks has been studied extensively in literature. In [9], a joint link scheduling and power control problem for sensor networks is formulated as an NP-hard optimization problem that provides tunable tradeoffs between throughput, energy, and latency. Exponential and polynomial complexity greedy-based heuristic algorithms are also presented. Reference [10] presents a distributed power control scheme for time division multiple access (TDMA)/CDMA-based ad hoc networks supporting multicast traffic and introduces a distributed joint scheduling and power control algorithm that eliminates strong interferers and enables power control. On the other hand, joint scheduling and power control is not provided. Reference [11] analyzes a TDMA-based wireless ad hoc network and provides a centralized joint power control, scheduling, and routing algorithm. This reference compares scheduling with and without joint power control, and concludes that with joint power control, the network achieves significantly larger throughput and less delay. Joint scheduling and power control is often formulated as a sequential problem in the literature. For example, [12] considers scheduling and power control jointly, but formulates the problems in two separate phases.

We have seen voluminous research of VoIP call carrying capacity in the context of IEEE 802.11 WLANs, especially those in which packet exchange takes place through access points [13], [14]. Studies of VoIP call carrying capacity in the context of IEEE 802.11 ad hoc networks have also been presented in some literature [15], [16]. But for a TDMA/SSMA-based ad hoc network with optimized transmission scheduling and power control, its VoIP call carrying capacity is not well understood, to the best of our knowledge. In this paper, we establish upper bounds on the VoIP call carrying capacity as function of network topology, network connectivity, routing strategy, spreading gain, transmit power, and overhead. Our upper bounds are obtained through optimal scheduling and power control. We consider a TDMA/SSMA-based network where multiple transmissions from a node and multiple receptions at a node are allowed. To the best of our knowledge, the joint optimization problem for such a network has not previously been well formulated. Here we present a formulation of the joint problem with detailed scheduling constraints and power constraints. In our formulation, scheduling and power control are not separated, and transmissions can only be scheduled if there exist corresponding feasible transmit powers. We also present a heuristic algorithm based on the tabu search method efficiently to obtain a suboptimal solution. The resulting upper bounds can then be used to judge the relative quality of operational algorithms.

The rest of the paper is organized as follows. In Section 2, we describe our network model and present our joint transmission scheduling and power control problem formulation. In Section 3, we present our solution methodology, and we investigate the performance of our algorithm with respect to its ability to find an optimal solution as well as its execution time. In Section 4, we introduce our simulations of VoIP call carrying capacity of our TDMA/SSMA-based ad hoc network with optimized transmission scheduling and power control. In Section 5, we draw conclusions.

2 Joint Transmission Scheduling and Power Control Problem Formulation

The wireless ad hoc network considered in this paper is a graph, $G = (\mathcal{V}, \mathcal{E})$, consisting of a collection of communication nodes, or vertices, $\mathcal{V}$, and a collection of links, or edges, $\mathcal{E}$. The collection of edges, $\mathcal{E}$, is determined according to the neighbor discovery mechanism and the network connectivity requirement. We denote the edges between node u and node v by $e_{\{u,v\}}$, where $\{u, v\}$ denotes the set of elements u and v. The propagation path between each pair of vertices, u and v, has a positive cost $\ell_{\{u,v\}}$, called the *propagation loss* or *path loss* between nodes u and v. The propagation loss is assumed to be symmetrical and is calculated as the ratio of, $P_{T_u}(P_{T_v})$, the power transmitted from node u (v), to, $P_{R_v}(P_{R_u})$, the power received at node v (u), or

$$\ell_{\{u,v\}} = \frac{P_{T_u}}{P_{R_v}} = \frac{P_{T_v}}{P_{R_u}}. \tag{1}$$

In general, $\ell_{\{u,v\}}$ is determined by a suitable propagation model, and its magnitude is a factor in deciding whether or not there is an edge between u and v.

The network has a single radio channel with the bandwidth of 22 MHz. A set of *spreading codes*, $\mathcal{C}$, is available to be used in the network based on TDMA/SSMA. Spreading codes are assigned to $\mathcal{E}^*$ on link basis, and the single radio channel is shared by using SSMA among $\mathcal{E}$. Codes can be reused on a spatial basis. At slot k, a subset of $\mathcal{E}$, denoted by $\mathcal{E}^*$, is selected for VoIP packet transmission based on our formulated transmission scheduling and power control optimization problem. A link $e_{\{i,j\}}$ with the spreading code $c_{\{i,j\}}$ in $\mathcal{E}^*$, is half-duplex; that is, if node u is in transmitting mode, the node v is in the receiving mode and vice versa. Although a node cannot transmit and receive simultaneously, it is capable of transmitting to multiple receivers or receiving from multiple transmitters at a given time provided the SINR criteria is met.

Our joint optimization problem is then to find $\mathcal{E}^*$, or a transmission set $\mathcal{T}(k)$ with the corresponding transmit power set $\mathcal{P}_T(k)$, to maximize the scheduling return. Let the ordered pair (i, j) denote the transmission from node i to node j. We define $X_{ij} = 1$ if (i, j) is scheduled at transmission interval k and 0 otherwise. Then $\mathcal{T}(k)$ and $\mathcal{P}_T(k)$ can be written as follows:

$$\mathcal{T}(k) = \{(i,j) | i \in \mathcal{V}, j \in \mathcal{N}(i), X_{ij} = 1\}, \tag{2}$$

$$\mathcal{P}_T(k) = \{P_{T_{ij}} | i \in \mathcal{V}, j \in \mathcal{N}(i), X_{ij} = 1\}, \tag{3}$$

where $\mathcal{N}(i)$ denotes node i's neighbors and $P_{T_{ij}}$ denotes the required transmit power for transmission (i, j). Define $S_{\min}$ as the minimum allowable received SINR. Then $P_{T_{ij}}$ can be determined as follows:

$$P_{T_{ij}} \geq S_{\min} \ell_{\{i,j\}} \Big(\frac{N_t}{G} + I_{ij} \Big), \tag{4}$$

$$I_{ij} = \frac{1}{G} \sum_{\substack{(e,f) \in \mathcal{T}(k) \\ (e,f) \neq (i,j), \\ c_{\{e,f\}} \neq c_{\{i,j\}}}} \frac{P_{T_{ef}}}{\ell_{\{e,j\}}} + \frac{1}{G_0} \sum_{\substack{(g,h) \in \mathcal{T}(k) \\ (g,h) \neq (i,j), \\ c_{\{g,h\}} = c_{\{i,j\}}}} \frac{P_{T_{gh}}}{\ell_{\{g,j\}}}, \tag{5}$$

where I_{ij} is the total interference at receiver j for transmission (i, j), N_t is the total noise at receiver j, $c_{\{e,f\}}$ is the spreading code used by transmission (e, f), G is the processing gain between two different spreading codes, and G_0 is the effective processing gain of a reused spreading code with respect to an offset version of the same code.

We wish to transmit packets whose value is maximal. Denote by $\mathcal{Q}_{ij}(k)$ the collection of packets backlogged at node i and whose next hop is to node j just prior to transmission slot k. Define $B(q)$ as the return that would be realized if packet q is transmitted during slot k, and

$$B_{ij,\max}(k) = \max_{q}\{B(q) | q \in \mathcal{Q}_{ij}(k)\}. \tag{6}$$

Definition of a broad variety of return functions is possible. As an example, an appropriate return value for a packet might be based on the remaining time per hop. For example, in the case of VoIP, a packet needs to arrive at its destination within 150 ms after it is generated. Thus, if the number of remaining hops, h, and the packet generation time, t_0, of a packet are known, then the remaining time per hop would be $[150 - (t_k - t_0)]/h$, where t_k denotes the time of the k-th transmission slot. In this case, the best choices for transmission scheduling are not necessarily the packets at the head of the queue at a station but the packets with the least remaining times per hop.

The solution to the following nonlinear programming problem, Problem P1, can be solved to find the set $\mathcal{T}(k)$ that maximizes the return function Z_J over all feasible transmission schedules and the set of transmission powers $\mathcal{P}_T(k)$ that minimizes total power consumption Z_P for the $\mathcal{T}(k)$.

Problem P1

$$\text{Minimize } Z_P = \sum_{(i,j)\in\mathcal{T}(k)} P_{T_{ij}} \text{ for maximal } Z_J = \sum_{i\in\mathcal{V}} \sum_{j\in\mathcal{N}(i)} X_{ij} B_{ij,\max}(k), \tag{7}$$

subject to

$$X_{ij} + X_{di} \leq 1 \quad \forall \quad i \in \mathcal{V}, j \in \mathcal{N}(i), d \in \mathcal{N}(i), \tag{8}$$

$$\sum_{j\in\mathcal{N}(i)} X_{ij} \leq N_{T_{\max}} \quad \forall \quad i \in \mathcal{V}, \tag{9}$$

$$\sum_{i\in\mathcal{N}(j)} X_{ij} \leq N_{R_{\max}} \quad \forall \quad j \in \mathcal{V}, \tag{10}$$

$$P_{T_{ij}} \geq 0 \quad i \in \mathcal{V}, j \in \mathcal{N}(i), \tag{11}$$

$$P_{T_{ij}} - X_{ij} P_{T_{\max}} \leq 0 \quad \forall \quad i \in \mathcal{V}, j \in \mathcal{N}(i), \tag{12}$$

$$\sum_{j\in\mathcal{N}(i)} P_{T_{ij}} \leq P_{T_{\max}} \quad \forall \quad i \in \mathcal{V}, \tag{13}$$

$$\frac{P_{T_{ij}}}{S_{\min}\ell_{\{i,j\}}} - X_{ij}\left(\frac{N_t}{G} + I_{ij}\right) \geq 0 \quad \forall \quad i \in \mathcal{V}, j \in \mathcal{N}(i). \tag{14}$$

Constraint (8) guarantees that if station i is transmitting, then station i cannot be the target of any other transmission because of half-duplex. In (9) and (10), $N_{T_{\max}}$ and $N_{R_{\max}}$ are the maximum number of transmissions and receptions allowed at a station. In (12), $P_{T_{\max}}$ is the maximum transmit power of a station; (12) restricts individual transmission and (13) restricts the total power. In (11) and (12), if $X_{ij} = 1$, then $P_{T_{ij}} > 0$ and $P_{T_{ij}} \leq P_{T_{\max}}$; but if $X_{ij} = 0$, then $P_{T_{ij}} = 0$. Constraint (14) guarantees that the SINR threshold, $S_{\min}$, is met by any feasible $\mathcal{T}(k)$ and $\mathcal{P}_T(k)$ at each receiver. Problem P1 not only has a nontraditional objective function but it is also nonlinear in that it involves products of variables. In Problem P1, variable X_{ij} is integral–in fact, binary– and $P_{T_{ij}}$ is continuous for power control. Thus, Problem P1 is a mixed-integer nonlinear programming (MINLP) problem.

We emphasize here that problem P1 is not a weighted maximum matching problem. Matching problems have been used in variety of settings in distributed scheduling, as for example in [17] and [18]. Given a graph G = (V,E), maximal matching finds a subset of edges, M, of maximum cardinality such that each vertex of G is adjacent to at most one edge in M. A matching is a one-to-one assignments between nodes. However in problem P1, a chosen node can send information to many nodes, i.e., not a one-to-one match. Furthermore, a node receiving information from another node cannot send out information to any other nodes, and a node sending information to another node can not receive information from any other nodes.

3 Tabu Search-Based Heuristic Algorithm

Herein we present a tabu search-based algorithm to find a suboptimal solution for Problem P1. We also evaluate its optimality performance and run time complexity. Tabu search is a high-level heuristic procedure for solving optimization problems, designed to avoid the trap of local optimality [19]. The core of the tabu search is embedded in its short term memory process constituting a form of aggressive exploration that seeks to make the best *move* possible subject to certain restrictions. The restrictions are designed to prevent the reversal, or sometimes repetition, of some moves, by rendering certain moves forbidden (tabu). A critical step in tabu search is to choose the best *admissible* candidates. This involves the creation and update of a candidate list and the evaluation of each move in the list.

Figure 1 shows a flow chart of our tabu search-based heuristic algorithm. In the initialization step, we initialize the scheduling solution set S_0, the current best scheduling solution set S^*, the transmit power set P^* that corresponds to S^*, the tabu list T_L that records the schedules that have already been checked, and the tabu tenure list T_N that records the tabu tenure value for each transmission. Initially, all X_{ij} in S_0 are set to zero. In each iteration, a transmission *adding* or *dropping* procedure takes place depending on whether an additional transmission that satisfies the three scheduling constraints (8), (9), and (10) in Problem P1 is available or not. If feasible, a new transmission is added into S_0, and then we obtain a tentative scheduling solution set S_1. Operationally, a new variable, $X_{ij,\text{in}}$, is brought into the current solution set S_0 based on the selection criterion. If S_1 is already in the tabu list T_L after the adding, we just discard the addition,

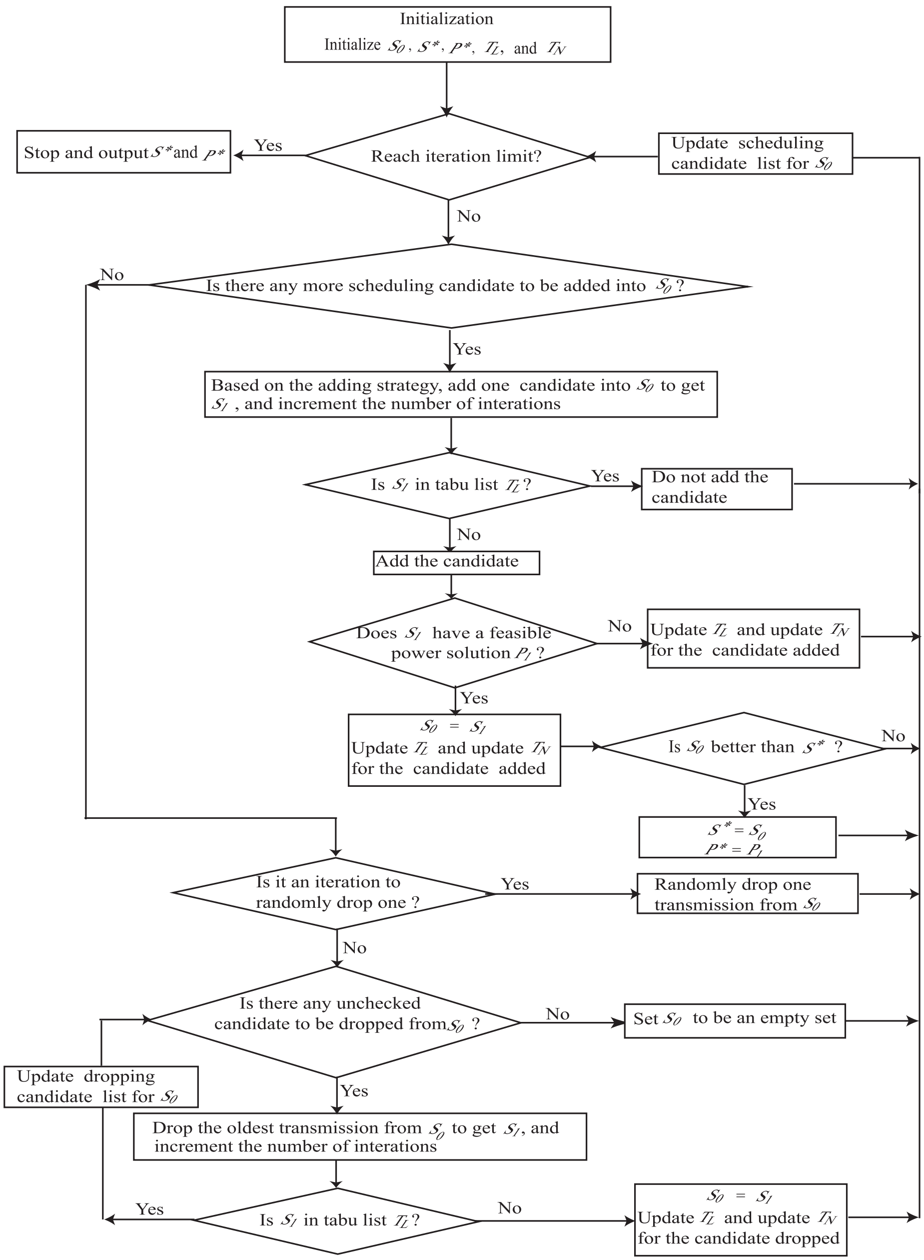

Fig. 1. Flow chart for the tabu search-based algorithm

update the candidate list, and go on to the next iteration; otherwise, we check whether a feasible power solution P_1 exists for the current S_1. If it does not exist, we also discard the addition. But if it does exist, we then step forward to use S_1 as the current solution S_0 and go to the next iteration. In this way the adding procedure schedules more and more transmissions iteration by iteration until no more transmissions can be added.

When we cannot add any more transmissions into S_0, a transmission dropping procedure is triggered. In the dropping procedure, the oldest transmission in S_0 is deleted from S_0. More specifically, one X_{ij} , say $X_{ij,\text{out}}$, which has stayed in S_0 for the longest time, is chosen from among the solution variables to be removed from the current solution set. In this way, we step back a little in the search, and then start another adding procedure to see whether we can get a better solution. By using adding and dropping procedures for a certain number of iterations, we can record and output the best scheduling solution S^* and its corresponding power solution P^* that has been found so far. Thus, instead of solving Problem P1 directly to obtain an optimal scheduling solution $\mathcal{T}(k)$ and its power solution $\mathcal{P}_T(k)$, we use our algorithm to greedily schedule as many transmissions with high returns as possible.

We note that in each iteration in our algorithm, each X_{ij} in S_0 or S_1 is known. Thus, when we seek a power solution for S_1 in our algorithm, constraint (14) becomes linear because in the constraint the products of variables disappear, and then the entire problem becomes linear. Specifically, in the adding procedure of each iteration, Problem P1 is always simplified and converted into the following ordinary linear programming Problem P2:

Problem P2

$$\text{Minimize} \quad Z_P = \sum_{(i,j)\in\mathcal{T}(k)} P_{T_{ij}}, \tag{15}$$

subject to

$$0 \le P_{T_{ij}} \le P_{T_{\max}} \quad \forall \quad (i,j) \in \mathcal{T}(k), \tag{16}$$

$$\sum_{\substack{j\in\mathcal{N}(i)\\(i,j)\in\mathcal{T}(k)}} P_{T_{ij}} \le P_{T_{\max}}, \tag{17}$$

$$\frac{P_{T_{ij}}}{S_{\min}\ell_{\{i,j\}}} - I_{ij} \ge \frac{N_t}{G} \quad \forall \quad (i,j) \in \mathcal{T}(k). \tag{18}$$

Here we design two different adding strategies to select a transmission candidates to be added into the transmission set. Our first adding strategy is to choose the transmission with the highest return while the second adding strategy is based on both the return and the interference of a transmission. Specifically, we calculate the weight for each candidate, select the one with the highest weight, and add it into the transmission set. The weight is calculated as follows:

$$W_{ij} = B_{ij,\max}(k)/\ell_{\{i,j\}}. \tag{19}$$

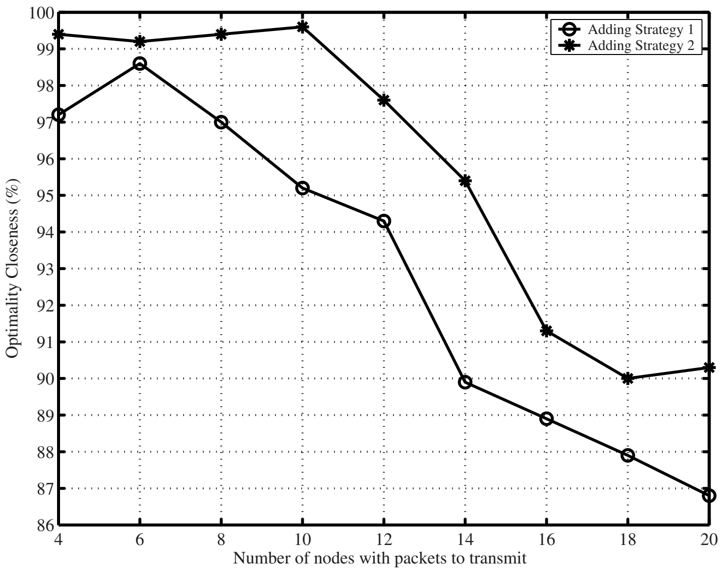

Fig. 2. Optimality of the algorithm with respect to the number of active nodes

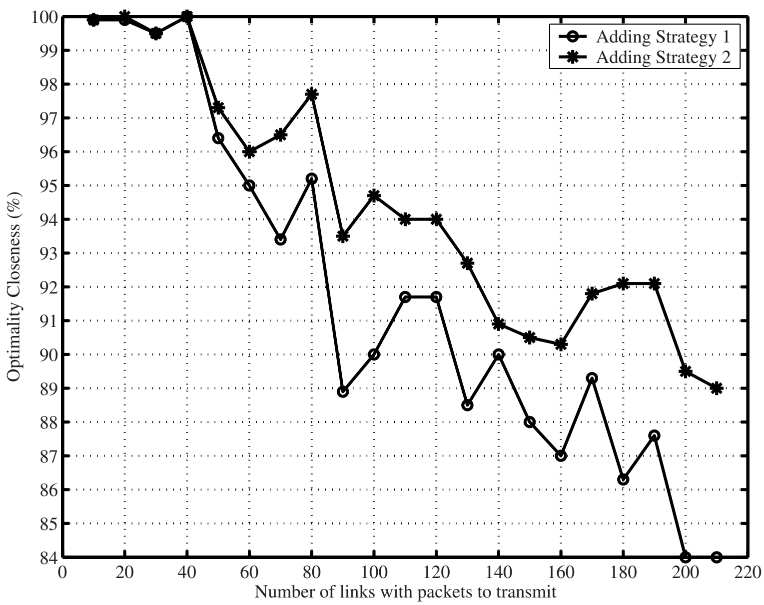

Fig. 3. Optimality of the algorithm with respect to the number of active links

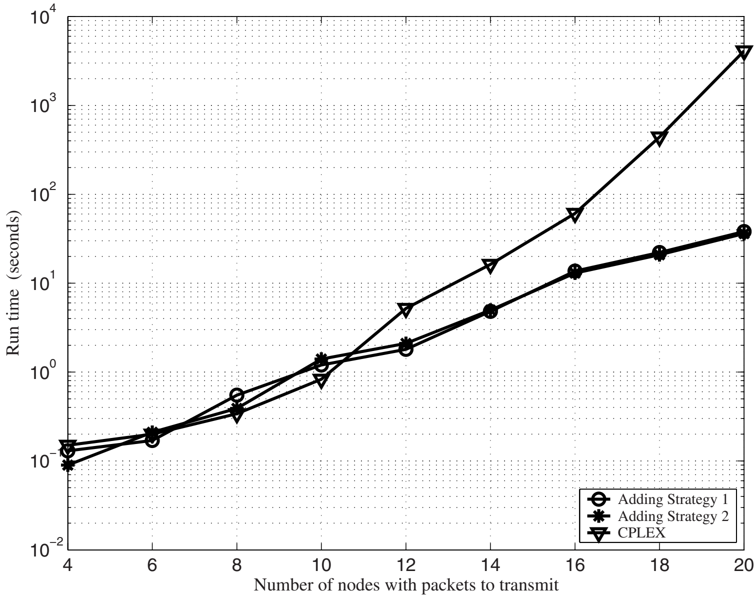

Fig. 4. Run time of the algorithm and CPLEX with respect to the number of active nodes

In order to evaluate the quality and run times of our tabu-based heuristic, we used ILOG CPLEX, a commercial package capable of solving our problem to optimality. We selected a large sample of scheduling problems at random during our simulations and solved the set of problems to optimality using CPLEX. Figures 2 and 3 show performance as a function of the number of active nodes and the number of active links in the networks, respectively. Here we define *active* to mean a node or a link has a packet available to transmit; otherwise, it is inactive. From Figure 2 and Figure 3, we find that the results of our algorithm with the second adding strategy are often within 5% of the optimal values when the number of active nodes is less than 14 or the number of active links is less than 80. When there are 20 active nodes, our algorithm with the second strategy achieves 90% of the optimal return. We can also find that the second adding strategy achieves a 2% to 3% improvement over the first adding strategy when the number of active nodes is more than 8 or the number of active links is more than 80. Thus, in the scheduling of a transmission, we should consider both its return and its interference as well.

Figures 4 and 5 compare the run times of our algorithm and the ILOG CPLEX program with respect to the number of active nodes and the number of active links, respectively. CPLEX often consumes less run time than our algorithm when the number of active nodes is less than 10 or the number of active links is less than 100. But as the number of active nodes or active links increases, CPLEX's run time increases much more sharply. When the number of active nodes is 20 or the number of active links is more than 230, our algorithm

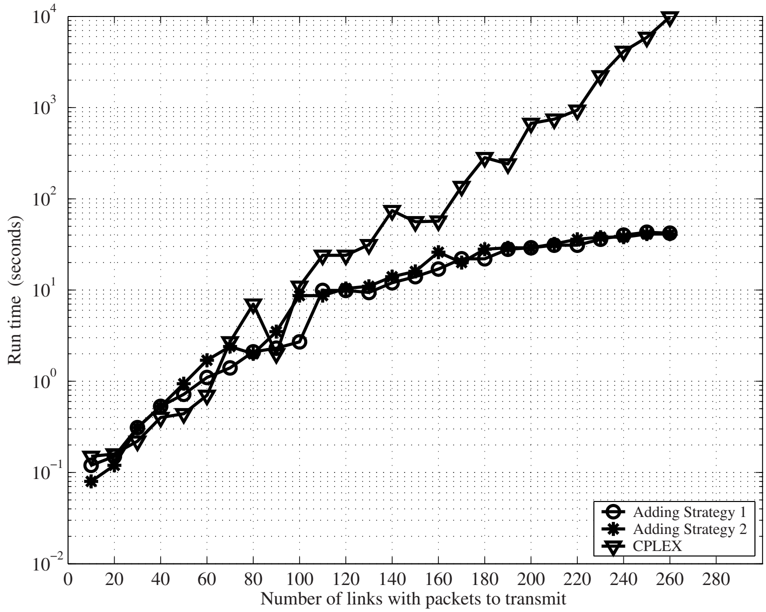

Fig. 5. Run time of the algorithm and CPLEX with respect to the number of active links

consumes less than 2% of CPLEX's run time. Thus, our tabu search-based algorithm can be quite efficient compared to CPLEX's obtaining the exact solution when the network is larger than about 12 active nodes.

4 Simulations and Results

The CCC of a network is defined as the maximum number of VoIP calls that can be carried without violating the VoIP quality constraints. We conducted simulations to investigate the CCC upper bounds as function of network topology, network connectivity, routing strategy, spreading gain, transmit power, and overhead. We scheduled and forwarded VoIP packets for a call based on our Problem P1 and our tabu search-based heuristic algorithm with the second adding strategy. The network configuration was restricted to a random grid having 49 nodes in a 2-dimensional area of 490 meters × 490 meters. The square area was divided into 49 square areas of equal size and then a station was placed at a random position within each small square. In the optimization formulation, $N_{T_{\max}}$ and $N_{R_{\max}}$ were both set to be 4 so that a node was capable of transmitting to as many as 4 neighboring nodes simultaneously or a node was capable of receiving from up to 4 neighboring nodes simultaneously. The propagation loss exponent was 2.0, and propagation loss between each node pair in the network was determined according to a log-normal shadowing model. In the simulations, $S_{\min}$

was 5 dB, and N_t was 2.2e-9 W for the 22 MHz radio band. In addition, the simulation time was more than 30 minutes in each run, the duration of a call was chosen from an exponential distribution with a mean of 3 minutes, a G.711 codec of 64 Kb/s was used, the delay threshold was 150 ms, and the packet drop rate threshold was 1%. In the simulations, a variation of the algorithm presented in [20] and [21] was used to choose the edge set $\mathcal{E}$ to construct networks such that the maximum required transmission power was minimized while providing three-vertex-connectivity (three node-disjoint paths) between every node pair. Specific factors considered were the routing algorithm–Dijkstra's shortest path routing (SP) or Dijkstra's shortest path routing with congestion avoidance (SP with CA), the spreading gain–Barker codes with the code length of 11 (B11) or gold codes with the code length of 63 (G63), and channel access overhead–no overhead or with overhead derived from the IEEE 802.11 standards. When using SP with CA, we tried to find a path from a source to its destination and avoid traversing the nodes with heavy traffic loads. Results gathered from the simulations included the VoIP call carrying capacity (CCC), the average number of simultaneous transmissions (ST), the average hop count (AHC), and the energy consumption per call.

Table 1 shows the results for networks with three-vertex-connectivity, with $P_{T_{\max}}$ set to be 1 W, and under a number of scenarios. In Table 1, it can be seen that the AHC for a call is about equal over all scenarios. It can also be seen that including overhead tends to reduce the number of simultaneous transmissions occurring in the network, but using a congestion avoidance routing algorithm tends to increase the number of simultaneous transmissions, as might be expected. There are more simultaneous transmissions for G63 than B11, but the increase is only about 80% the ratio of 63 to 11. The last column shows that in the absence of synchronization overhead, a spreading gain of 11 yields a CCC about 50% higher than a spreading factor of 63. But, when a reasonable synchronization overhead is included, the CCC for the spreading factor of 11 case drops by about 50% while the CCC for the case of 63 is about the same. One can then deduce that a fast server is always better than a collection of slower servers if the fast server can be kept busy whenever there is traffic to serve. However, if using the fast server requires a setup time, then the collection of slow servers has potential to outperform the fast server. The reason for this is that the slow servers lose the service time in parallel so that the impact of setup time is reduced roughly according to the number of parallel servers.

Tables 2 and 3 show the performance metrics as a function of maximum transmit power. In the first column of Tables 2 or 3, the maximum transmit power ($P_{T_{\max}}$) varies from 0.010 W to 0.570 W. In these simulations, the algorithm for constructing networks with three-vertex-connectivity is not used. Instead, a node can have a link to any node in its range, and the second column shows the actual minimum vertex-connectivity between any pair of two nodes in the networks for each power setting. The fifth column in Tables 2 or 3 shows the average energy consumption per call. The VoIP call carrying results for Barker codes of length 11 and Gold codes of length 63 are shown in the last column in Table 2 or Table 3.

Table 1. Simulation results for three-vertex-connectivity networks

Scenario	Results		
	AHC	ST	CCC
SP, B11, no overhead	4.3	1.9	11
SP, B11, with overhead	4.3	1.8	8
SP with CA, B11, no overhead	4.3	2.2	12
SP with CA, B11, with overhead	4.3	1.9	8
SP, G63, no overhead	4.3	8.4	8
SP, G63, with overhead	4.2	7.8	7
SP with CA, G63, no overhead	4.3	8.7	8
SP with CA, G63, with overhead	4.3	8.0	8

Table 2. Simulation results for networks with B11, SP, overhead, and various transmit powers

$P_{T_{\max}}$ (W)	Results				
	Min-conn	AHC	ST	Energy/call (J)	CCC
0.059	1	4.30	1.1	1.1	5
0.130	3	2.60	1.2	1.6	10
0.292	8	1.80	1.2	2.2	13
0.519	14	1.30	1.2	2.9	16
0.810	24	1.11	1.2	3.3	18
1.167	34	1.07	1.2	3.8	20
1.588	40	1.01	1.2	4.2	21
2.074	45	1.01	1.2	4.3	21
2.625	48	1.00	1.2	4.5	22
3.241	48	1.00	1.2	4.4	22

Table 3. Simulation results for networks with G63, SP, overhead, and various transmit powers

$P_{T_{\max}}$ (W)	Results				
	Min-conn	AHC	ST	Energy/call (J)	CCC
0.010	1	4.30	2.3	0.8	2
0.025	3	2.60	2.4	1.3	4
0.050	8	1.80	2.4	2.0	6
0.100	14	1.30	2.5	2.4	8
0.140	24	1.11	2.6	4.0	9
0.200	34	1.07	3.3	4.2	13
0.280	40	1.01	4.4	4.4	18
0.350	45	1.01	6.1	4.8	28
0.450	48	1.00	7.9	5.5	35
0.570	48	1.00	10.6	6.0	45

Table 4. Simulation results for networks with G63, SP and overhead, and without power control for each transmit power

$P_{T_{\max}}$ (W)	Results				
	Min-conn	AHC	ST	Energy/call (J)	CCC
0.010	1	4.30	1.1	1.2	0
0.025	3	2.60	1.2	2.2	1
0.050	8	1.80	1.5	3.4	2
0.100	14	1.30	1.6	4.8	4
0.140	24	1.11	1.6	5.1	5
0.200	34	1.07	1.8	9.0	6
0.280	40	1.01	2.5	9.4	10
0.350	45	1.01	3.2	11.0	13
0.450	48	1.00	4.1	15.3	17
0.570	48	1.00	4.8	17.8	20

From Tables 2 and 3, we can see that the CCC and the energy per call increase as the $P_{T_{\max}}$ increases. This can be explained as follows. When $P_{T_{\max}}$ increases, more simultaneous transmissions can take place in a network, and the number of hops that a call needs to get through decreases at the same time. Thus, more VoIP calls can be supported as the transmission range increases. From Tables 2 and 3, we can also see that when the transmission range increases to some certain levels such that the network becomes fully-connected, a very large number of calls can be supported with the cost of high power consumption. In general, there is a trade-off between VoIP call carrying capacity and power consumption. In a MANET, when energy consumption is one of the major concerns, $P_{T_{\max}}$ needs to be carefully chosen. Finally, we compare the capacity of B11 in Table 2 with that of G63 in Table 3. From the CCC columns in Table 2 and Table 3, we find that when $P_{T_{\max}}$ is less than about 0.14 W, the capacity of G63 is about a half of that of B11. But when $P_{T_{\max}}$ is more than about 0.35 W, the capacity of G63 is much more than that of B11. This dramatic increase is the results of the increase in connectivity due to increased power and the ability to schedule more simultaneous transmissions using the G63 code.

Table 4 shows the results for networks with G63, under the log-normal path loss model, and without power control. In Table 4, a node can only transmit at $P_{T_{\max}}$. More specifically, for Table 4, Problem P1 is unchanged except that the constraint (12) is set to be as follows instead:

$$P_{T_{ij}} - X_{ij} P_{T_{\max}} = 0 \;\; \forall \;\; i \in \mathcal{V}, j \in \mathcal{N}(i). \tag{20}$$

When we compare Table 4 with Table 3, we find that the capacity of the networks without power control is generally a half or less than a half of the capacity of the networks with power control. In addition, the networks without power control need significantly more energy to support a call than the networks with power control. Recall that the current wireless ad hoc network with IEEE 802.11 DCF is contention-based and usually does not implement power control. Thus, we conclude that future wireless ad hoc network should have power control if at all possible.

5 Conclusions

SSMA-based communications can facilitate multiple simultaneous transmissions. We have formulated the transmission scheduling and power control in an SSMA-based network as a joint optimization problem. We have designed an efficient tabu search-based heuristic algorithm to solve the optimization problem and rigorously assess the quality of the results. Through analysis and simulation, we have established upper bounds on the VoIP call carrying capacity of the network as function of network topology, connectivity, routing strategy, spreading gain, maximum transmit power, and scheduling overhead. Our established upper bounds are based on the assumption that there is a central control mechanism to implement the optimized transmission scheduling and power control in the network. Since a wireless ad hoc network is distributed and such a mechanism does not exist in the SSMA-based network, we can expect that the actual capacity of the network should be less than our results. Thus, based on the difference between our upper bounds and the actual capacity of the network, we can measure how well the network can utilize available resources.

Our results indicate that use of SSMA has potential to increase the traffic handling capability of an ad hoc network provided a suitable scheduling and power control algorithm can be developed. This increased capacity would be especially important in cases where the required capacity has been underestimated. Thus, it appears well worthwhile to attempt to develop light-weight distributed algorithms for SSMA-based systems. This is especially true in light of the fact that the networking community has concluded that IEEE 802.11 does not provide a suitable environment for deploying algorithms that guarantee service quality.

References

1. Royer, E., Toh, C.: A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks. IEEE Personal Communications 6, 46–55 (1999)
2. IEEE Computer Society LAN MAN Standards Committee: IEEE Std 802.11 Wireless Medium Control (MAC) and Physical Layer (PHL) Specifications. IEEE Press, New York (1997)
3. Xu, S., Saadawi, T.: Does IEEE 802.11 MAC Protocol Work Well in Multi-hop Wireless Ad Hoc Networks? IEEE Communications Magazine 39, 130–137 (2001)
4. Xu, S., Saadawi, T.: Revealing the Problems with 802.11 Medium Access Control Protocol in Multi-hop Wireless Ad Hoc Networks. Computer Networks: The International Journal of Computer and Telecommunications Networking 38, 531–548 (2002)
5. Lal, S., Sousa, E.: Distributed Resource Allocation for DS-CDMA Based Multimedia Ad Hoc Wireless LANs. IEEE Journal on Selected Areas in Communications 17, 947–967 (1999)

6. Muqattash, A., Krunz, M.: CDMA-Based MAC Protocol for Wireless Ad Hoc Networks. In: MobiHoc, pp. 153–164. ACM Press, New York (2003)
7. Muqattash, A., Krunz, M., Ryan, V.: Solving the Near-Far Problem in CDMA-Based Ad Hoc Networks. Ad Hoc Networks 1, 435–453 (2003)
8. Meshhadany, T., Ajib, W.: New CDMA-Based MAC Protocol for Ad Hoc Networks. In: IEEE 66th Vehicular Technology Conference, pp. 91–95. IEEE Press, New York (2007)
9. Lu, G., Krishnamachari, B.: Energy Efficient Joint Scheduling and Power Control for Wireless Ad Hoc Networks. In: IEEE Sensor and Ad Hoc Communications and Networks, IEEE SECON, pp. 362–373. IEEE Press, New York (2005)
10. Wang, K., Chiasserini, C., Rao, R., Proakis, J.: A Joint Solution to Scheduling and Power Control for Multicasting in Wireless Ad Hoc Networks. EURASIP Journal on Applied Signal Processing 2005, 144–152 (2005)
11. Li, Y., Ephremides, A.: A Joint Scheduling, Power Control, and Routing Algorithm for Ad Hoc Wireless Networks. Ad Hoc Networks 5, 959–973 (2007)
12. Elbatt, T., Ephremides, A.: Joint Scheduling and Power Control for Wireless Ad Hoc Networks. IEEE Transactions on Wireless Communications 3, 74–85 (2004)
13. Medepalli, K., Gopalakrishnan, P., Famolari, D., Kodama, T.: Voice Capacity of IEEE 802.11b, 802.11a and 802.11g Wireless LANs. In: IEEE Global Telecommunications Conference, pp. 1549–1553. IEEE Press, New York (2004)
14. Cai, L., Xiao, Y., Shen, X., Mark, J.: VoIP over WLAN: Voice Capacity, Admission Control, QoS, and MAC. International Journal of Communication Systems 19, 491–508 (2006)
15. Fasolo, E., Maguolo, F., Zanella, A., Zorzi, M.: VoIP Communications in Wireless Ad Hoc Network with Gateways. In: IEEE Symposium on Computers and Communications, pp. MW69–MW74. IEEE Press, New York (2007)
16. Oliveira, C., Siqueira, M.: Simultaneous VoIP Calls Capacity Over an 802.11 Ad Hoc Network. In: International Conference on Wireless Networks, pp. 470–476. CSREA Press, Irvine (2006)
17. Akkaya, K., Younis, M.: A Survey on Routing Protocols for Wireless Sensor Networks. Ad Hoc Networks 3, 325–349 (2005)
18. Wu, K., Srikant, R.: Regulated Maximal Matching: A Distributed Scheduling Algorithm for Multi-Hop Wireless Networks with Node-Exclusive Spectrum Sharing. In: IEEE Conference on Decision and Control 2005 and 2005 European Control Conference, pp. 5342–5347. IEEE Press, New York (2005)
19. Glover, F.: Tabu Search: A Tutorial. Interfaces 20, 74–94 (1990)
20. Li, N., Hou, J.: Localized Fault-Tolerant Topology Control in Wireless Ad Hoc Networks. IEEE Transactions on Parallel and Distributed Systems 17, 307–320 (2006)
21. Yang, L.: Building k-Connected Neighborhood Graphs for Isometric Data Embedding. IEEE Transactions on Pattern Analysis and Machine Intelligence 28, 722–728 (2006)