Joint scheduling and power control supporting multicasting in wireless ad hoc networks

Kang Wang a,*, Carla F. Chiasserini b, John G. Proakis c, Ramesh R. Rao c

a Qualcomm, Inc., 5775 Morehouse Dr., San Diego, CA 92121, USA
b CERCOM—Dipartimento di Elettronica, Politecnico di Torino, Italy
c ECE Department, University of California, San Diego, La Jolla, CA 92093, USA

Received 24 July 2004; received in revised form 25 May 2005; accepted 3 June 2005
Available online 5 July 2005

Abstract

This paper addresses the problem of power control in a multihop wireless network supporting multicast traffic. We face the problem of forwarding packet traffic to multicast group members while meeting constraints on the signal-to-interference-plus-noise ratio (SINR) at the intended receivers. First, we present a distributed algorithm which, given the set of multicast senders and their corresponding receivers, provides an optimal solution when it exists, which minimizes the total transmit power. When no optimal solution can be found for the given set of multicast senders and receivers, we introduce a distributed, joint scheduling and power control algorithm which eliminates the weak connections and tries to maximize the number of successful multicast transmissions. The algorithm allows the other senders to solve the power control problem and minimize the total transmit power. We show that our distributed algorithm converges to the optimal solution when it exists, and performs close to centralized, heuristic algorithms that have been proposed to address the joint scheduling and power control problem.

Keywords: Wireless ad hoc networks; Scheduling; Power control; Multicasting

1. Introduction

Current and future wireless networks need to support a wide variety of services with high data rate and reliability requirements. The limited energy available at the communication nodes, coupled with the often hostile channel conditions and the limited radio resources, make the problem of providing reliable high data rate services challenging.

Here we focus on multicast services in ad hoc wireless networks. Multicasting enables data delivery to multiple recipients in a more efficient...
manner than traditional unicasting and broadcast-
ing. A packet is duplicated only when the delivery path toward the traffic destinations diverges at a node, thus helping to reduce unnecessary transmissions. For this reason, multicasting is a high desirable feature in ad hoc networks, where bandwidth and nodes’ energy are precious resources. However, some critical issues need to be addressed to provide multicasting in an ad hoc network. A comprehensive discussion on multicast protocols and related issues in an ad hoc environment can be found in [1]. Efficient solutions to the formation of multicast trees in ad hoc networks are proposed in [2–7], while MAC protocols for ad hoc multicasting are presented in [8–10].

In this paper, we address the problem of power control. The power control issue has been well studied in cellular systems [11–13] and ad hoc networks supporting unicast traffic [14,15], to provide the desired link quality, alleviate interference to others, and minimize power consumption. Since we focus on multicasting, the objective of this work is to determine the optimal values of transmit power to be used at the multicast senders, so that the requirements on the signal-to-interference-plus-noise-ratio (SINR) at all intended receivers are fulfilled and the total transmission power is minimized. We formulate the problem as a linear programming problem and propose a distributed, iterative algorithm that converges to the optimal solution. Often, however, the power control problem does not have any solutions, due to the presence of some strong interferers. In this case, as proposed in [15] in the context of ad hoc unicasting, a scheduling\(^1\) algorithm has to be executed to eliminate strong interferers and allow the remaining nodes to solve the power control problem. To the best of our knowledge, our work is the first solution dealing with the problem of distributed, joint scheduling and power control in ad hoc networks that support multicasting.

We introduce the system model and the formulation of the power control problem in Section 2. In Section 3 we present a distributed, iterative power control algorithm, and we prove that this scheme converges to the optimum solution, when it exists. In Section 4 we consider the case where, given the set of multicast senders and receivers, the optimization problem does not have feasible solutions. We present our distributed, joint scheduling and power control scheme, the so called Distributed Joint Scheduling and Power Control (DJS) scheme, and prove some interesting properties of this algorithm. In Section 5, we show by simulation that DJS performs close to some centralized, heuristic algorithms that have been proposed for joint scheduling and power control in wireless networks. Furthermore, we present some simulation results showing that the values of transmit power obtained through the DJS scheme converge to the optimum, when it exists. Finally, Section 6 reviews some related work and Section 7 concludes the paper.

2. System model and assumptions

We consider an ad hoc network composed of stationary nodes, each of them equipped with an omnidirectional antenna. Nodes access the channel by using a TDMA/CDMA scheme with a fixed time slot duration, which accounts for the packet transmission time and a guard time interval. Links between any pair of nodes are assumed to be bi-directional.

We focus on the case of multicast traffic connections. We assume that for each traffic connection the multicast tree has already been constructed and there is no conflict in the transmission setup, i.e., each receiver is associated with only one sender at a time. We are not concerned with traffic routing from the multicast source to the destinations. Rather, we focus on next neighbor transmissions, i.e., sending packet traffic to the specified neighbors while meeting constraints on the SINR at the intended receivers [15]. As mentioned above, we first study the problem of power control and try to find an optimal solution (Section 3); then, we consider the case where an optimal solution does not exist and address the problem of traffic connection scheduling, i.e., how to determine which connections should not be allowed so that

\(^1\) Note that here “scheduling” means allowing/dis-allowing traffic connections so as to ensure that admitted connections enjoy a sufficiently high SINR.
the admitted connections can enjoy a sufficiently high SINR (Section 4).

Let us consider a set of senders denoted by \( S \), and a set of receivers, denoted by \( R \). \( s \) and \( r \) indicate the number of senders and receivers, respectively. Since we deal with multicasting, we have that: \( s \leq r \), i.e., each sender sends data packets to at least one receiver. We define \( P^t_k \) as the transmit power of the generic node \( k \). Every sender causes interference to any receivers, and the amount of interference depends on the power and the propagation attenuation of the transmitted signal. We assume that the signal attenuation over the radio channel is either constant or slowly changing. Short feedback packets using different CDMA codes are encoded with a strong error correction code so that they are always correctly received by the destination nodes.

We assume that interference caused by simultaneous transmissions is treated as noise. Let \( i \) denote the node sending a packet to receiver node \( j \). Node \( j \) receives a transmission from \( i \) successfully if the corresponding SINR at node \( j \) is equal to or greater than a given threshold \( \gamma_j \), i.e.,

\[
\text{SINR}_j = \frac{q_{ij}P^t_i}{\sum_{k \neq i}q_{kj}P^t_k + N_0} \geq \gamma_j,
\]

where \( q_{kj} \) is the propagation attenuation of the signal from sender \( k \) to receiver \( j \). \( N_0 > 0 \) is the noise power, and \( L \) is the system processing gain. We can re-write (1) as

\[
P^t_i \geq I_j(\tilde{P}) \triangleq \frac{\gamma_j \left( N_0 + \frac{1}{L} \sum_{k \neq i}q_{kj}P^t_k \right)}{q_{ij}}, \tag{2}
\]

where \( \tilde{P} = [P^t_{S(1)}, P^t_{S(2)}, \ldots, P^t_{S(r)}]^T \) and \( S(j) \) is the associated sender of receiver \( j \). If we denote the set of all associated receivers of sender \( i \) by \( R(i) \), (2) must be satisfied for all \( j \in R(i) \). Further, our goal is to have the expression in (2) to be satisfied for all nodes in \( R \). Thus, by defining \( \tilde{P} = [P^t_1, P^t_2, \ldots, P^t_s]^T \), and \( \tilde{I}(\tilde{P}) = [I_1(\tilde{P}), \ldots, I_r(\tilde{P})]^T \), the problem addressed in this work can be formulated as

\[
\textbf{P:} \quad \text{minimize} \sum_{k=1}^{s} P^t_k \quad \text{subject to} \quad \tilde{P} \geq \tilde{I}(\tilde{P}). \tag{4}
\]

In the following, we will denote with \( \tilde{P}^* \) the optimal solution to problem \( \textbf{P} \).

### 3. Distributed power control algorithm

Problem \( \textbf{P} \), as expressed in (4), is a linear programming problem which can be easily solved in a centralized fashion [16]. However, due to the distributed nature of wireless ad hoc networks, a distributed algorithm is highly desirable.

In [11], Yates proposes a framework for uplink power control in a cellular radio environment where a sender has only one associated receiver. There, a simple iterative power control algorithm is needed, namely,

\[
P(n + 1) = \tilde{I}(\tilde{P}(n)). \tag{5}
\]

Under (5), each sender updates its transmit power such that the target SINR will be met exactly, assuming that the interference from other senders is constant. Yates proves that as \( n \to \infty \), \( \tilde{P}(n) \) converges to \( \tilde{P} = \tilde{I}(\tilde{P}) \), if such a solution exists. Notice that for each iteration a sender needs feedback only from its associated receivers, since we have,

\[
I_j(\tilde{P}(n)) = \frac{\gamma_j P^t_i(n)}{\text{SINR}_j(n)}. \tag{6}
\]

More specifically, at the \( n \)th iteration, sender \( i \) will need to receive from its associated receivers the value of SINR corresponding to its last transmission; while, \( P^t_i(n) \) is known to the sender, as well as \( \gamma_j \), which is the target SINR value.

Now we show that the multicast power control problem in an ad hoc network environment is indeed similar to the unicast power control problem presented in [11].

**Lemma 1.** If (4) has feasible solutions, then there exists a unique \( \tilde{P}^* \) for which (4) is satisfied and \( \sum_{i=1}^{s} P^t_i \) is minimized. Further, for each sender \( i \), there exists at least one associated receiver \( j \in R(i) \) for which (2) is achieved with equality.

**Proof.** See Appendix [17].
Lemma 1 suggests that we could achieve optimum power control by maintaining only the SINR requirement of the weakest associated receiver for each sender, thus converting the multicast power control problem to the unicast one.

Let us define:

\[ I^\max_i(\bar{P}) = \max_{j \in R(i)} I_j(\bar{P}) \] and

\[ \bar{I}(\bar{P}) = \left[ I^\max_1(\bar{P}), \ldots, I^\max_s(\bar{P}) \right]^T. \]

Using the method in [11], we can show that (5) will also converge to the optimum solution \( \bar{P}^* \) which satisfies \( \bar{P}^* = \bar{I}(\bar{P}^*) \) and minimizes \( \sum_{i=1}^{s} P_i^l \), provided that (4) has feasible solutions. We prove this through Lemmas 1 and 2, and Theorem 1, which are presented in the following. Furthermore, in Lemma 4 we show that, if conditions on convergence are satisfied, then convergence to the optimum is approximately exponentially fast.

**Lemma 2.** Starting from \( \bar{P}(0) = 0 \), \( \bar{P}(n) \) is a monotonically increasing sequence bounded by \( \bar{P}^* \). Starting from a feasible solution \( \bar{P}(0) \), \( \bar{P}(n) \) is a monotonically decreasing sequence bounded by \( \bar{P}^* \).

**Proof.** Recall that \( N_0 > 0 \). Starting from \( \bar{P}(0) = 0 \), observe that \( \bar{I}(\bar{P}_1) > \bar{I}(\bar{P}_2) \) if \( \bar{P}_1 \geq \bar{P}_2 \) with at least one inequality. Note that \( \bar{P}(1) > 0 = \bar{P}(0) \), therefore \( \bar{P}(2) = \bar{I}(\bar{P}(1)) > \bar{I}(\bar{P}(0)) = \bar{P}(1) \). Similarly \( \bar{P}(3) > \bar{P}(2) \), and \( \bar{P}(n) \) is a monotonically increasing sequence. According to Lemma 1, a unique \( \bar{P}^* \) exists. Suppose \( \bar{P}(n) < \bar{P}^* \), it follows that:

\[ \bar{P}^* = \bar{I}(\bar{P}^*) > \bar{I}(\bar{P}(n)) = \bar{P}(n+1). \]

Thus \( \bar{P}(n) \) is upper bounded by \( \bar{P}^* \).

Starting from a feasible \( \bar{P}(0) \), \( \bar{P}(1) = \bar{I}(\bar{P}(0)) \leq \bar{P}(0) \) with at least one inequality, therefore \( \bar{P}(2) = \bar{I}(\bar{P}(1)) < \bar{I}(\bar{P}(0)) = \bar{P}(1) \) and \( \bar{P}(n) \) is a monotonically decreasing sequence. Suppose \( \bar{P}(n) > \bar{P}^* \), it follows that:

\[ \bar{P}^* = \bar{I}(\bar{P}^*) < \bar{I}(\bar{P}(n)) = \bar{P}(n+1). \]

Thus \( \bar{P}(n) \) is lower bounded by \( \bar{P}^* \). \( \square \)

**Lemma 3.** If (4) has feasible solutions, then starting from \( \bar{P}(0) = 0 \), or any feasible solution other than \( \bar{P}^* \), \( \bar{P}(n) \) will converge to \( \bar{P}^* \) as \( n \to \infty \) by the iteration described in (5).

**Proof.** According to Lemma 1, a unique \( \bar{P}^* \) exists. According to Lemma 2, \( \bar{P}(n) \) is a monotonically increasing sequence upper bounded by \( \bar{P}^* \) if \( \bar{P}(0) = 0 \). The uniqueness of \( \bar{P}^* \) implies that \( \bar{P}(n) \) must converge to \( \bar{P}^* \). The same holds when \( \bar{P}(0) \) is a feasible solution other than \( \bar{P}^* \). \( \square \)

**Theorem 1.** If (4) has feasible solutions, then starting from any \( \bar{P}(0) \), \( \bar{P}(n) \) will converge to \( \bar{P}^* \) as \( n \to \infty \) by the iteration described in (5).

**Proof.** According to Lemma 1, a unique \( \bar{P}^* \) exists. Observe that the interference function has the scalability property, i.e., for all \( \alpha > 1 \), \( \bar{I}(\alpha \bar{P}) > \bar{I}(\bar{P}) \).

For any \( \bar{P}(0) \), we can find \( \alpha > 1 \) such that \( \bar{P}_n(0) \triangleq \bar{P}^* \geq \bar{P}(0) \geq 0 \triangleq \bar{P}_1(0) \). It follows that:

\[ \bar{I}(\bar{P}_n(n)) \geq \bar{I}(\bar{P}(n)) \geq \bar{I}(\bar{P}_1(n)). \]

We have shown by Lemma 3 that \( \bar{I}(\bar{P}_1(n)) \) converges to \( \bar{P}^* \). By scalability of the interference function \( \alpha \bar{P}^* = \alpha \bar{I}(\bar{P}) > \bar{I}(\alpha \bar{P}^*) \), so \( \alpha \bar{P}^* \) is feasible and \( \bar{I}(\bar{P}_n(n)) \) also converges to \( \bar{P}^* \). Hence \( \bar{I}(\bar{P}(n)) \) converges to the unique, optimum solution whenever a feasible solution exists. \( \square \)

**Lemma 4.** If (4) has feasible solutions, \( \bar{P}(n) \) converges to \( \bar{P}^* \) approximately exponentially fast.

**Proof.** Suppose at iteration step \( n \), receiver \( r(i) \) has the maximum interference function among the associated receiver set \( R(i) \) of sender \( i \), i.e., \( r(i) \triangleq \arg \max_{j \in R(i)} \bar{I}_j(\bar{P}) \). Note that \( r(i) \) may change with iteration step. The iteration in (5) can be re-written as

\[ \bar{P}(n + 1) = \bar{I}(\bar{P}(n)) = \bar{b}_n + \bar{C}_n \bar{P}(n), \]

where \( \bar{b}_n = N_0 \left[ \frac{\gamma_{r(i)}(1)}{\varphi_{r(i)}}, \frac{\gamma_{r(2)}(1)}{\varphi_{r(2)}}, \ldots, \frac{\gamma_{r(s)}(1)}{\varphi_{r(s)}} \right]^T \) and \( \bar{C}_n \) is an \( s \times s \) matrix defined as

\[ \bar{C}_n = \left[ \begin{array}{cccc}
\bar{C}_{11} & \bar{C}_{12} & \cdots & \bar{C}_{1s} \\
\bar{C}_{21} & \bar{C}_{22} & \cdots & \bar{C}_{2s} \\
\vdots & \ddots & \ddots & \vdots \\
\bar{C}_{s1} & \bar{C}_{s2} & \cdots & \bar{C}_{ss}
\end{array} \right]. \]
When a feasible solution exists, according to Theorem 1, (7) converges to \( \bar{P}^* \). Therefore \( \bar{P}_n \) and \( C_n \) have to stay constant after a finite iteration step \( n^* \). Let us denote \( \bar{r}_n = \lim_{n \to +\infty} \bar{P}_n = \bar{b}_n \) and \( C_n^* = \lim_{n \to +\infty} C_n = C_n^* \). By (7) we have
\[
\bar{P}(n) = \left(I + C_n^* + \cdots + C_n^*\bar{r}_n^{-1}\right)\bar{b}_n + C_n^*\bar{r}_n^{-1} \bar{P}_n^*.
\] (8)

When a feasible solution exists, the moduli of the eigenvalues of \( C_n^* \) are strictly less than 1, hence \( I + C_n^* + C_n^{2*} + \cdots + C_n^{n-1} = (I - C_n^*)^{-1} \) and \( \lim_{n \to +\infty} C_n^*\bar{r}_n = 0 \) [13]. Noting \((I - C_n^*)\bar{P}^* = \bar{b}_n\), we have
\[
\lim_{n \to +\infty} \bar{P}(n) = (I - C_n^*)^{-1}\bar{b}_n = \bar{P}^*.
\] (9)

From (8) and (9) we can see that \( \bar{P}(n) \) converges to \( \bar{P}^* \) exponentially fast after iteration step \( n^* \). \( \square \)

In a real scenario, every sender can use a limited transmit power. Let us denote with \( P_i^{\max} \) the maximum allowed transmit power of sender \( i \). \( P_i^{\max} \) can be initialized to a pre-determined value, e.g., based on the node’s energy capability, and it can be adapted to account for the remaining battery energy of the node. Let us now say that (4) is feasible when there exists solutions to (4) such that \( P_i \leq P_i^{\max} \), for any \( i \). An important property of the iteration in (5) is that, when (4) is feasible, the initial power levels can be properly selected so that the transmit powers at every iteration will not exceed their maximum allowed value. We prove this in Lemma 5.

**Lemma 5.** (a) If (4) is feasible, then starting from any \( \bar{P}(0) \leq \bar{P}^* \), \( \bar{P}(n) \leq \bar{P}^{\max} \) for every \( n \), where
\[
\bar{P}^\text{max} = [P_1^{\max}, P_2^{\max}, \ldots, P_s^{\max}]^T; \quad (b) \quad \text{If } \bar{P}^\text{max} < \bar{P}(0) \leq \bar{P}^\text{max} \text{ and } \bar{P}(0) \text{ is a feasible solution, then } \bar{P}(n) \leq \bar{P}^\text{max} \text{ for every } n.
\]

**Proof.** (a) If \( \bar{P}(0) < \bar{P}^* \), then \( \bar{P}(1) = I(\bar{P}(0)) < I(\bar{P}^*) = \bar{P}^* \). Therefore \( \bar{P}(n) \) is upper bounded by \( \bar{P}^* \). The fact that (4) is feasible implies that: \( \bar{P}^* \leq \bar{P}^{\max} \), hence \( \bar{P}(n) < \bar{P}^* \leq \bar{P}^{\max} \) for every \( n \).

(b) is obvious since if \( \bar{P}(0) \) is feasible solution, Lemma 2 shows that \( \bar{P}(n) \) is a monotonically decreasing sequence bounded by \( \bar{P}^* \). Thus, for any \( n \), \( \bar{P}(n) < \bar{P}(0) \leq \bar{P}^{\max} \). \( \square \)

Lemma 5 shows that we do not need to use a very small initial power vector; indeed, the transmit power levels at the various iterations will never exceed their maximum limits, as long as the initial power is smaller than the optimal power vector.

Also, Lemma 5 indicates that a sender does not need global information to realize that (4) is not feasible: it can draw this conclusion as soon as it finds out that its transmit power will exceed the power limit. Whenever problem \( \mathbf{P} \) does not have any feasible solutions, a scheduling algorithm is needed, that prevents some senders from transmitting, thus enabling the other senders to perform power control [15].

We address the joint scheduling and power control problem in the next section.

### 4. Distributed joint scheduling and power control algorithm

Consider the case where (4) is not feasible and a scheduling scheme is required to remove some
transmissions and allow the admitted senders to execute power control [15]. We highlight that the optimum joint scheduling and power control that maximizes the number of admitted connections, has exponential complexity and, hence, it cannot be of any practical use. A centralized, heuristic scheduling algorithm have been proposed in [18]. Here, we present a Distributed Joint Scheduling and Power Control (DJSPC) algorithm, which enables the candidate senders to independently determine which connection is admissible. A connection refers to the transmission between a sender and a receiver, hence in a multicast environment a sender may have multiple connections. If none of a sender’s connections are admissible, the sender will not transmit.

Our joint scheduling and power control scheme works in a simple way. At the beginning, every connection is assumed to be admissible, and the transmit power levels are set to reasonable initial values such that \( P(0) < P^\text{max} \), if (4) is feasible. How to choose the initial values in practice is discussed in Section 5. At the \( n + 1 \)th iteration, sender \( i \)'s algorithm is as follows.

1. If all of sender \( i \)'s connections are marked non-admissible, set \( P'_i(n + 1) = 0 \).
2. Otherwise, do the following:
   (a) Calculate \( I_j(\tilde{P}(n)) \) for every \( j \in R_n(i) \), where \( R_n(i) \) is the set of associated receivers of \( i \) whose connections are marked admissible.
   (b) Mark \( j \)'s connection non-admissible if \( I_j(\tilde{P}(n)) > P^\text{max}_j \), and updates \( R_{n+1}(i) \) by removing non-admissible connections from \( R_n(i) \).
   (c) Set \( P'_i(n + 1) = 0 \) if \( R_{n+1}(i) = \emptyset \), otherwise:
      \[
      P'_i(n + 1) = I_j^\text{max}(\tilde{P}(n)) = \max_{j \in R_{n+1}(i)} I_j(\tilde{P}(n)).
      \]

At the \( n + 1 \)th iteration, receiver \( j \) estimates SINR, and sends it to its associated sender. Notice that, in order to compute \( I_j(\tilde{P}(n)) \) with \( j \in R_n(i) \), a sender needs feedback only from its associated receivers (see (6)). If, after the \( n + 1 \)th iteration, a sender’s transmit power is zero, it will not transmit. Otherwise, it transmits to the associated receivers (whose connections are admissible) using the obtained value of transmit power.

In the practice, a sender should notify the corresponding receiver whose connection has been marked non-admissible. This can be done by piggybacking the information in the next iteration. This surely has two benefits: (i) the receiver is informed that it should not expect any packets, and (ii) the receiver no longer wastes power and radio resources by sending back SINR estimates to the sender.

In the Lemma below, we prove that the proposed joint scheduling and power control algorithm has the following desirable properties.

**Lemma 6.** Starting from any \( \tilde{P}(0) < P^\text{max} \), the DJSPC scheme has the following properties: (a) it will converge; (b) if (4) is feasible, all connections will be marked admissible and the result will converge to \( \bar{P} \).

**Proof.** (a) is obvious when (4) is feasible. Similar to (8), we have
\[
\tilde{P}(n) = \bar{b}_{n-1} + C_{n-1}\bar{b}_{n-2} + \cdots + C_{n-1}C_{n-2} \cdots C_1\bar{b}_0
+ C_{n-1}C_{n-2} \cdots C_0\bar{P}(0).
\] Notice that every entry in the matrices and vectors are positive. If (4) is not feasible, \( \bar{b}_{n-1} + C_{n-1}\bar{b}_{n-2} + \cdots + C_{n-1}C_{n-2} \cdots C_1\bar{b}_0 \) does not converge. Let us show this by contradiction. If it converges, starting from \( \tilde{P}(0) = 0 \), we would have
\[
\tilde{P}(n) = \bar{b}_n + C_n\bar{P}(n)
\] as \( n \to +\infty \). This means as \( n \to +\infty \), \((1 - C_n)\bar{P}(n) = \bar{b}_n \) and therefore (4) is feasible, which is contradictory to the assumption.

Now we have shown that \( \bar{b}_{n-1} + C_{n-1}\bar{b}_{n-2} + \cdots + C_{n-1}C_{n-2} \cdots C_1\bar{b}_0 \) does not converge when (4) is not feasible. Noting that the last term in (10) is always a positive vector and that \( \bar{b}_{n-1} + C_{n-1}\bar{b}_{n-2} + \cdots + C_{n-1}C_{n-2} \cdots C_1\bar{b}_0 \) is monotonically increasing, some elements in \( \tilde{P}(n) \) will exceed those in \( P^\text{max} \) as \( n \) increases. When this happens at least one connection will be marked non-admissible. This process continues until the set of admissible connections converges.

(b) is obvious since if (4) is feasible, no connection will be marked non-admissible by virtue of Lemma 5. \( \Box \)
4.1. Discussion on the DJSPC performance

We now turn our attention to the performance of the DJSPC scheme. It would be interesting to compare DJSPC to the optimum algorithm which maximizes the number of admitted connections. Unfortunately the optimum joint scheduling and power control has exponential complexity since it has to search for a combination of connections that is feasible and also includes the most number of connections.

In [18], Zander studies a heuristic, centralized scheduling algorithm where at each iteration the connection with the minimum normalized SINR (current SINR normalized by target SINR) is removed, whenever a sender’s next power exceeds its limit. After removal of the “weakest” link, the iteration starts again from the beginning. We first ask ourself whether this removal policy based on SINR is a good algorithm. The following Lemma seems to support it.

**Lemma 7.** With a constant set of senders and receivers, the minimum normalized SINR among all receivers is monotonically increasing under the proposed power control algorithm.

**Proof.** We have,

\[ \tilde{P}(n + 1) = I_j(\tilde{P}(n)) = \frac{\gamma_j P_j(n)}{\text{SINR}_j(n)}. \]

The normalized SINR is given by

\[ \text{SINR}_j(n) = \frac{\frac{P_j(n)}{\gamma_j}}{\frac{P_j(n)}{n + 1}}. \]

Suppose the minimum normalized SINR of active connections at step \( n \) is \( \beta_n = \min_j \text{SINR}_j(n) < 1 \), i.e. \( \beta_n \tilde{P}(n + 1) \leq \tilde{P}(n) \) where equality is achieved for at least one element of the vectors. We have

\[ \beta_n \tilde{P}(n + 2) = \beta_n \tilde{I}(\tilde{P}(n + 1)) < \tilde{I}(\beta_n \tilde{P}(n + 1)) \leq \tilde{I}(\tilde{P}(n)) = \tilde{P}(n + 1). \]

The first inequality is due to the scalability property of the interference function. Since \( \beta_{n+1} \tilde{P}(n + 2) \leq \tilde{P}(n + 1) \) with equality achieved for at least one element, we can see that \( \beta_{n+1} > \beta_n \), i.e., the minimum normalized SINR is monotonically increasing with iterations when the set of senders and receivers is constant, even if there is no solution to (4). □

Lemma 7 implies that the infeasibility of (4) is due to bounded minimum normalized SINR. However, we found that the removal algorithm based on normalized SINR does not perform well, rather it performs even worse than our distributed algorithm. This is because that normalized SINR does not capture the “weakness” of a connection very well. Although the minimum normalized SINR of all connections is monotonically increasing, the individual SINR’s may decrease with iterations. Consider, for instance, the case of a weak connection, \( i \), with a low normalized SINR of 0.01 at iteration \( n \). At iteration \( n + 1 \), sender \( i \) boosts its transmit power by 100 times. Consider also that at iteration \( n \) all other connections have normalized SINR close to 1, and their transmit powers remain almost the same at \( n + 1 \). So, at \( n + 1 \) connection \( i \)'s normalized SINR is almost 1, but some other connections may have a lower normalized SINR due to the greatly increased interference from \( i \).

Conversely, we observe that the interference function \( I(\tilde{P}) \) captures the weakness of a connection quite well. The interference function is basically the normalized interference from other transmissions and, when no connection is removed, it is monotonically increasing under the DJSPC algorithm. Therefore a good centralized removal policy is as follows: remove the connection with the maximum interference function among all connections whenever a sender’s power exceeds its limit.

In the next section, we present a simulation-based study of the DJSPC performance, and compare the results that we obtain with our scheme with the performance of the centralized heuristic algorithm removing the connection with the minimum normalized SINR, as well as the one removing the connection with the maximum interference function.

5. Simulation results

We derive the performance of the proposed joint scheduling and power control algorithm for a stationary network whose nodes are randomly
spread over a $100 \times 100$ square region. We focus on a multicast group composed of $N$ nodes, out of which one node is randomly chosen as the multicast source. We consider that the multicast tree is set up by using the MIP scheme [19]. We assume that there are two sets of senders whose transmissions alternate over time. In each odd (even) slot, transmissions are performed by the nodes in the odd (even) layer of the tree, having at least one child. The nodes, which do not transmit in a time slot, act as receivers. An example of a simple multicast tree and of the corresponding transmission scheme are presented in Fig. 1.

We assume that the propagation attenuation between the generic sender $k$ and receiver $j$ is $q_{kj} = 1/d_{kj}^\alpha$, where $d_{kj}$ is the distance between the two nodes, and $\alpha$ is the power decay factor that we take to be equal to 2. The target SINR, $\gamma$, is set to 6 dB for any receiver, $L$ is equal to 4, and $P_{\text{max}}$ is equal to 33 dBm for all senders. Performance metrics are evaluated by Monte Carlo simulations for 2000 independent configurations at each value of $N$.

Fig. 2 presents the average percentage of network instances for which problem $P$, as in (4), is feasible. As shown in the plot, the feasibility probability drops sharply as the number of simultaneous transmissions increases. This clearly indicates the need for a scheduling algorithm.

Next, we consider the case when problem $P$ is not feasible. Fig. 3 shows the average percentage

![A Simple Multicast Tree](image)

Sender | Receiver
--- | ---
A | B, C, D
F | I, J
G | K
H | L

![Fig. 2. Average percentage of not admitted connections vs. the number of nodes in the system ($N$).](image)

Sender | Receiver
--- | ---
B | E, F
D | G, H
K | M, N

![Fig. 3. Average percentage of not admitted connections vs. the number of nodes in the system ($N$).](image)
of multicast connections that are not admitted to transmit according to our scheduling algorithm, as a function of \( N \). We compare the performance of our distributed algorithm with the two centralized, joint scheduling and power control algorithms discussed in the previous section. The best performance is obtained when \( N \) takes values between 8 and 20. When \( N \) is smaller, a very high percentage of connections is not admitted because there are few connections and at least one connection is not admitted. While, when \( N \) is larger than 20, the percentage of not-admitted connections increases with \( N \), due to the higher interference level in the network. However, we observe that DJSPC always outperforms the centralized algorithm that removes the connection with the minimum normalized SINR, and it performs quite close to the centralized algorithm that removes the connection with the maximum interference function.

Fig. 4 shows the average transmit power of the admitted multicast senders after the algorithm convergence is reached. The centralized algorithm that removes the minimum normalized SINR connection has the highest average transmit power, and DJSPC has the lowest average transmit power. This, however, does not mean that DJSPC performs better than the centralized algorithm that removes the connection with maximum interference function; in fact, as shown in Fig. 3, DJSPC removes a larger number of weak connections with respect to the centralized algorithm. Also, we notice that the average transmit power decreases as \( N \) increases. This is because, as larger values of \( N \) are considered, the distance between sender and receiver becomes relatively shorter.

Fig. 5 presents the transmit power level of the admitted senders vs. the number of iterations. The results are obtained through our distributed algorithm, in the case where we have \( N = 15 \), four multicast senders, and the initial transmit power level is set to 0 for all senders. The plot shows that convergence is reached in 5–7 iterations in all cases. Notice that the sharp decrease in sender 1’s power level occurring at the 5th iteration is due to the removal of the connection between sender 1 and one of its receivers. Another connection of sender 1 is removed at the 6th iteration.

Then, we consider the case where a feasible solution to problem \( \mathbf{P} \) exists, so as to compare the performance of our distributed algorithm against the optimal solution. We study a network scenario with \( N = 15 \), out of which four are multicast senders, and we assume that the initial power level is set to 0 for all senders. In Fig. 6, we present the senders’ transmit power as a function of the number of iterations, while in Fig. 7 we show the relative error in the senders’ transmit power that we obtain by comparing the power values given by the DJSPC algorithm and the optimal solution. We can see that for all senders convergence is

![Fig. 4. Average transmit power of admitted multicast senders vs. \( N \).](image-url)

![Fig. 5. Transmit power level of the admitted senders vs. the number of iterations, with \( N = 15 \).](image-url)
achieved quite quickly. More precisely, by looking at Fig. 7, we observe that after 5 iterations, we obtain a relative error smaller than $10^{-2}$ in all cases.

The results shown above are obtained by setting the initial power values to 0, which may be not a practical choice in real world. Senders cannot know whether (4) is feasible nor their optimal power values in advance. However, according to Lemma 5, if (4) is feasible, reasonable large initial transmit power values such that $\bar{P}(0) < \bar{P}$ can be used without decreasing the efficiency of the joint scheduling and power control scheme. On the other hand, if the initial transmit power values are too large a performance degradation occurs. For example, if (4) is feasible, $\bar{P} < \bar{P}(0) < \bar{P}^{\text{max}}$ and $\bar{P}(0)$ is not a feasible power vector, then $\bar{P}(n)$ is not a monotonically decreasing sequence. In this case, $\bar{P}(n)$ could exceed its maximum power limit even though (4) is feasible, and some connections could be unnecessarily removed.

Fig. 8 shows the average number of iterations required to achieve a relative error in transmit power smaller than 10% (upper plot) and the average percentage of multicast connections that are not admitted (lower plot), as functions of the initial power level used by the multicast senders. Results are plotted in both the case where problem P has feasible solutions and the case where no feasible solution exists. We can see that the percentage of connections that are not admitted increases with the initial power value. When P has feasible solutions, all connections are admitted if the initial power value is small. This suggests that the choice of the initial power values is quite critical, and achieving a good tradeoff between scheduling and power control

![Fig. 6. Transmit power level of the multicast senders vs. number of iterations, in the case where a feasible solution exists and $N = 15$.](image1)

![Fig. 7. Relative error in the senders' transmit power, as a function of the number of iterations, in the case where a feasible solution exists and $N = 15$.](image2)

![Fig. 8. Average number of iterations needed for achieving a relative error in transmit power smaller than 10% (upper plot) and the average percentage of multicast connections that are not admitted (lower plot) vs. the initial level of transmit power. The case where a feasible solution exists and no feasible solution exists are shown when $N = 15$.](image3)
efficiency, and the required number of iterations in the DJSPC algorithm is challenging. Finally, we would like to point out that, although we use uniform values in our simulations, the initial power levels and maximum power limits do not need to be the same for all senders.

6. Related work

The problem of power control in wireless networks has been widely studied in the context of both cellular and ad hoc networks. The power control algorithms in [18,11,12,20,13] were designed for a cellular environment but they apply to the case of unicast transmissions in ad hoc networks as well. In particular, in [13] a simple distributed algorithm is introduced, which maximizes the SINR at any receivers while minimizing the total transmission power [12]. The problem of optimally controlling the node transmission range in ad hoc networks was addressed in [21,14]. In [22–24] the authors employ power control to adjust the node power level so as to create a desired network topology. In particular, the problem posed in [22] is formulated as a constrained optimization problem. In [25,26] power control is used within the carrier sense multiple access with collision-avoidance (CSMA/CA) MAC scheme to improve spatial channel reuse. However, the methods proposed there apply specifically to CSMA/CA-based systems, and do not guarantee that the allocated transmission power levels are minimum.

With regard to scheduling and admission control in wireless networks, several proposals have appeared in the literature. For instance, in [27–29] scheduling and admission control schemes involving power control issues are proposed for CDMA-based cellular systems supporting multimedia traffic. In the context of ad hoc networks, a scheduling scheme which provides fairness in channel access and maximizes spatial reuse of bandwidth is presented in [30]. Admission control and power control aspects are addressed in [31], where the authors present a distributed scheme which maintains the SIR of active radio links above their required thresholds while new users request admission in the system.

The work closest to ours is presented in [15], where the joint problem of scheduling and power control in an ad hoc network is first addressed. The key idea in [15] is that strong interferers are eliminated via a scheduling phase so that the remaining nodes can solve the power control problem by using the algorithm in [13]. The work in [15] deals with unicast transmissions; furthermore it assumes the existence of a central scheduler which has information on the nodes' geographical position and traffic conditions. Our work differs from [15] in dealing with a multicast traffic scenario and in proposing a distributed algorithm. In addition, unlike [15], we do not have a separate scheduling phase and a power control phase—these two functions are performed within the same algorithm iteration. In [32,33] the problems of routing, scheduling and power control in a multi-hop wireless network are jointly addressed. In [32] the authors use a dynamic programming-based formulation to maximize the system throughput under a transmit power constraint, taking into account the selected route to the destination. Their solution apply to the context of a multihop network overlaid on a cellular structure. In [33] first the authors formulate the joint scheduling and power control problem as a constrained optimization problem whose solution minimizes the total average transmission power in the network. Then, by using shortest path algorithms with properly set link weights, the authors search for a feasible optimum routing. Again, the work in [33] considers a unicast traffic scenario. In [34], a joint scheduling and power control scheme that mitigates inter-cell interference in a cellular environment is proposed.

Relevant to our work are also some papers on multicasting in wireless ad hoc networks. Multicast protocols for ad hoc networks can be found in [2–7,35]. In particular, [6,7] focus on the construction of a power efficient multicast tree and discuss some crucial issues. The work in [6] points to the broadcast nature of wireless communications: when omnidirectional antennas are used, every transmission by a node can be received by all nodes in its transmission range. This implies that the higher the transmission power is, the more nodes can be reached. However, trade-offs exist
among network connectivity, power consumption, as well as interference caused to other simultaneous transmissions. In [7], both physical layer requirements, expressed in terms of SINR, and network layer issues are taken into account in order to derive a multicast tree which minimizes transmission power. Finally, in [36] clustering is used to generate a supernode topology, where each supernode (i.e., cluster-head) acts as forwarding agent for its members. The authors adapt an ad hoc network multicast protocol to be executed on this supernode topology. Multicast data will move from the sender to its cluster-head, then along the supernode topology, and finally from cluster-head to receivers, by using at every step an appropriate power level.

7. Conclusions

In this paper we addressed the problem of power control in ad hoc networks supporting multicast traffic. Specifically, we first proposed a distributed algorithm that, given the set of multicast senders and their corresponding receivers, provides an optimal solution when it exists, i.e., it minimizes the total transmit power. When no optimal solution exists for the given set of multicast communications, we presented a distributed, joint scheduling and power control algorithm which eliminates the weak connections and tries to maximize the number of successful simultaneous transmissions, while minimizing the total transmit power. Our results show that the proposed distributed, joint scheduling and power control algorithm performs close to centralized, heuristic algorithms that have been proposed to address the problem.

Acknowledgements

This work was supported by the Multidisciplinary University Research Initiative (MURI) under the Office of Naval Research Contract N-00014-00-1-0564 and partially by the Italian Ministry for Scientific Research (MIUR) through the PATERN project.

Appendix A

Lemma 1. If (4) has feasible solutions, then there exists a unique $\bar{P}$ for which (4) is satisfied and $\sum_{i=1}^{n} P_{i}^*$ is minimized. Further, for each sender $i$, there exists at least one associated receiver $j \in R(i)$ for which (2) is achieved with equality.

Proof. Let us define $\bar{b} = N_{0}[\gamma_1, \ldots, \gamma_r]^T$ and,

$$Q = \begin{bmatrix} -\frac{1}{L} q_{11} & \cdots & q_{S(1)1} & \cdots & -\frac{1}{L} q_{1s} \\ -\frac{1}{L} q_{12} & q_{S(2)2} & \cdots & \cdots & -\frac{1}{L} q_{2s} \\ \vdots & \vdots & \vdots & & \vdots \\ -\frac{1}{L} q_{1r} & \cdots & q_{S(r)r} & \cdots & -\frac{1}{L} q_{rs} \end{bmatrix}.$$ 

it is straightforward to show that (4) is equivalent to

$$Q\bar{P} \geq \bar{b}. \quad (11)$$

We can show that: If (4) has feasible solutions, then there exists a unique $\bar{P}$ such that

(i) it satisfies: $Q\bar{P} \geq \bar{b}$, and
(ii) it maximizes $\sum f(P_{i})$ for any strictly decreasing function $f$.

By doing so we prove Lemma 1.

If (4) is feasible, it is easy to see that there exists a $\bar{P}$ which satisfies $Q\bar{P} \geq \bar{b}$ and maximizes $\sum f(P_{i})$ for a (not any) strictly decreasing function $f$. With this $\bar{P}$, we can show that for each sender $i$ there is at least one receiver in $R(i)$ whose SINR is exactly equal to its target SINR. That is, by denoting such a receiver by $r(i)$, we have: $(Q\bar{P})_{r(i)} = b_{r(i)}$. We show this by contradiction [17].

Suppose there exists a sender $i$ such that all receivers in $R(i)$ exceed their target SINR. We can reduce $P_{i}^{*}$ by a certain amount and leave the transmit power of the other nodes unchanged, so that at least one receiver in $R(i)$ reaches exactly its target SINR while the other receivers still exceed theirs. Observe that, since the interference from sender $i$ is reduced, the SINR’s at all the other
receivers are still met. Moreover, $f$ being strictly decreasing, the new power vector, $\vec{P}$ increases $\sum f(P'_i)$, which contradicts the assumption that $\vec{P}$ is a maximizer.

Now, let us consider a $s \times s$ square matrix $Q'$ created by taking for every sender $i$, the $r(i)$'th row of $Q$. Also, consider a $s \times 1$ vector $\vec{b}$ created by taking the $r(i)$'th element of $\vec{b}$, for $1 \leq i \leq s$. This is equivalent to considering the unicast transmission problem, for which we have: $Q'\vec{P} = \vec{b}'$. In such a case, $Q'$ is a full rank matrix if there exists a feasible power vector [37]. If so, we can write: $\vec{P} = (Q')^{-1}\vec{b}'$. For any feasible $\vec{P}$, it must satisfy $Q'\vec{P} \geq \vec{b}'$ and therefore $\vec{P} \geq \vec{P}$ [37].

This result shows that the optimal solution $\vec{P}$ is Pareto Optimal for the multicast case too, i.e., $\vec{P} \leq \vec{P}$ for any other $\vec{P}$ such that $Q\vec{P} \geq \vec{b}$.

By using the result proved above, we conclude that $\vec{P}$ satisfies $Q\vec{P} \geq \vec{b}$, is unique and maximizes $\sum f(P'_i)$ for any strictly decreasing function $f$. □

References


Kang Wang received his BSEE in 1996 from the Nanjing University of Posts & Telecommunications in China. He received his MS and Ph.D. from the Ohio State University in 2000 and the Ph.D. from the University of California at San Diego in 2004. Since then he joined Qualcomm at San Diego as a senior systems engineer.

At Qualcomm, he has been involved in 3G high speed EV-DO wireless network design. His current research interests include routing and multicast protocols for wireless ad hoc networks, power efficient ad hoc network design, and the general areas of digital communications.

Carla-Fabiana Chiasserini graduated with a summa cum laude degree in Electrical Engineering from the University of Florence in 1996. She did her graduate work at the Politecnico di Torino, Italy, receiving the Ph.D. degree in 1999. Since then she has been with the Department of Electrical Engineering at Politecnico di Torino, where she is currently an Assistant Professor. Since 1999, she has worked as a visiting researcher at the University of California, San Diego, California.

Her research interests include architectures, protocols and performance analysis of wireless networks for integrated multimedia services.

John G. Proakis received the BSEE from the University of Cincinnati in 1959, the MSEE from MIT in 1961 and the Ph.D. from Harvard University in 1967. He is an Adjunct Professor at the University of California at San Diego and a Professor Emeritus at Northeastern University. He was a faculty member at Northeastern University from 1969 through 1998 and held the following academic positions: Associate Professor of Electrical Engineering, 1969–1976; Professor of Electrical Engineering, 1976–1998; Associate Dean of the College of Engineering and Director of the Graduate School of Engineering, 1982–1984; Interim Dean of the College of Engineering, 1992–1993; Chairman of the Department of Electrical and Computer Engineering, 1984–1997. Prior to joining...
Northeastern University, he worked at GTE Laboratories and the MIT Lincoln Laboratory.


Ramesh R. Rao received his Honours Bachelor's degree in Electrical and Electronics Engineering from the University of Madras in 1980. He did his graduate work at the University of Maryland, College Park, MD, receiving the MS degree in 1982 and the Ph.D. degree in 1984. Since then he has been on the faculty of the department of Electrical and Computer Engineering at the University of California, San Diego, where he is currently Professor and Director of the San Diego Division of the California Institute of Telecommunications and Information Technology. Prior to this appointment, he served as the Director of the UCSD Center for Wireless Communications and was the Vice Chair of Instructional Affairs in the Department of Electrical and Computer Engineering.

His research interests include architectures, protocols and performance analysis of wireless, wireline and photonic networks for integrated multi-media services.

He served as the Editor of the Information Theory Society Newsletter from 1993 to 1995 and is the founding Web Editor of the Information Theory Society web site. He has been a senior member of the IEEE since 1990 and was elected to the IEEE Information Theory Society Board of Governors (1997–1999 and 2000–2002). He is the Editor for Packet Multiple Access of the IEEE Transactions on Communications and is a member of the Editorial Board of the ACM/Baltzer Wireless Network Journal as well as IEEE Network magazine.