Cooperative Multicast for Maximum Network Lifetime

Ivana Maric *Member, IEEE* and Roy D. Yates *Member, IEEE*

***Abstract*— We consider cooperative data multicast in a wireless network with the objective to maximize the network lifetime. We present the *Maximum Lifetime Accumulative Broadcast (MLAB)* algorithm that specifies the nodes' order of transmission and transmit power levels. We prove that the solution found by MLAB is optimal but not necessarily unique. The power levels found by the algorithm ensure that the lifetimes of the active relays are the same, causing them to fail simultaneously. For the same battery levels at all the nodes, the optimum transmit powers become the same.**

The simplicity of the solution is made possible by allowing the nodes that are out of the transmission range of a transmitter to collect the energy of unreliably received overheard signals. As a message is forwarded through the network, nodes will have multiple opportunities to reliably receive the message by collecting energy during each retransmission. We refer to this cooperative strategy as *accumulative multicast*. Cooperative multicast not only increases the multicast energy-efficiency by allowing for more energy radiated in the network to be collected, but also facilitates load balancing by relaxing the constraint that a relay has to transmit with power sufficient to reach its most disadvantaged child. When the message is to be delivered to all network nodes this cooperative strategy becomes *accumulative broadcast* [1]. Simulation results demonstrate that cooperative broadcast significantly increased network lifetime compared to conventional broadcast. We also present the distributed MLAB algorithm for accumulative broadcast that determines the transmit power levels locally at the nodes.

***Index Terms*— Cooperative multicast, cooperative broadcast, maximum network lifetime, optimum transmit powers, distributed algorithm.**

I. Introduction

We consider the problem of energy-efficient multicasting in a wireless network. In the multicast problem, a message from a *source* node is to be delivered efficiently to a set of *destination* nodes. When the set of destination nodes includes all the network nodes (except the source), the multicast problem reduces to the broadcast problem. When there is only one destination node, multicast reduces to unicast and the problem becomes that of routing to one destination node. Prior work on this subject has been focused on the minimum-energy broadcast problem with the objective of minimizing the total transmitted power in the network. This problem was shown in [2]–[4] to be NP-complete. Several heuristics for constructing energy-efficient broadcast trees have been proposed; see [2], [3], [5]–[8] and references therein.

However, broadcasting data through an energy-efficient tree drains the batteries at the nodes unevenly causing higher drain relays to fail first. A performance objective that addresses this issue is *network lifetime* which is defined to be the time duration until the first node battery is fully drained [9]. Finding a broadcast tree that maximizes network lifetime was considered in [10]–[13]. The problem of maximizing the network lifetime during a multicast was addressed in [14]. In [15], it was shown that the use of directional antennas can improve both the energy-efficiency and the lifetime of the network as compared to the omnidirectional case. Because the energies of the nodes in a tree are drained unevenly, the optimal tree changes in time and therefore the authors [11], [13], [14] distinguished between the *static* and *dynamic* maximum lifetime problem. In a static problem, a single tree is used throughout the broadcast session whereas the dynamic problem allows a sequence of trees to be used. Since the latter approach balances the traffic more evenly over time, it generally performs better. For the static problem, an algorithm was proposed that finds the optimum tree [11]. For the special case of identical initial battery energy at the nodes, the optimum tree was shown to be the minimum spanning tree. In a dynamic problem, a series of trees were used that were periodically updated [11] or used with assigned duty cycles [13].

Wireless formulations of the above broadcast problems assume that a node can benefit from a transmission only if the received power is above a threshold required for reliable communication. This is a pessimistic assumption. A node for which the received power is below the required threshold, but above the receiver noise floor, can collect energy from the unreliable reception of the transmitted information.

Moreover, it was observed in the relay channel [16] that utilizing unreliable overheard information is essential to achieving capacity. This idea is particularly suited for the multicast problem, where a node has multiple opportunities to receive a message as the message is forwarded through the network. We borrow this idea and re-examine the multicast problem under the assumption that nodes accumulate the energy of unreliable receptions. We refer to this particular cooperative strategy as *accumulative multicast* and in the special case of broadcast, as *accumulative broadcast* [1]. The minimum energy accumulative broadcast problem was formulated and addressed in [1], [17], [18]. The problem was shown to be NP-complete. An energy-efficient heuristic was proposed that demonstrated the improvement of accumulative broadcast over

Manuscript received October 15, 2003; revised August 3, 2004. This work was supported by New Jersey Commission on Science and Technology and NSF grant NSF ANI 0338805.

The authors are with Wireless Network Information Laboratory (WINLAB), Department of Electrical and Computer Engineering, Rutgers University, Piscataway, NJ 08854 USA (e-mail: ivanam@winlab.rutgers.edu; ryates@winlab.rutgers.edu).

the conventional broadcast. Under a different physical model, this problem was independently considered in [19] and again shown to be NP-complete. Furthermore, the same idea, for a packet level system model with the additional constraint of a power threshold for signal acquisition, was recently proposed under the name *Hitch-hiking* [20].

In this paper, we address the problem of maximizing the network lifetime by employing the accumulative multicast. As in the conventional broadcast problem, we impose a *reliable forwarding* constraint that a node can forward a message only after reliably decoding that message.

We show that the maximum lifetime multicast problem has a simple optimal solution and propose the *Maximum Lifetime Accumulative Broadcast (MLAB)* algorithm that finds it. The solution specifies the order of transmissions and transmit power levels at the nodes. The power levels given by the solution ensure that the lifetimes of relay nodes are the same and thus, their batteries die simultaneously. As shown later, this is due to the accumulative multicast that naturally allows for load balancing. Assigning the powers such that all the nodes fail at the same time has its equivalent in the problem of maximizing network lifetime during routing. In that problem, the network lifetime is maximized when the data is sent over multiple routes all with the same minimum lifetime [21]. Moreover, the simplicity of the solution allows us to formulate a distributed MLAB algorithm for the accumulative broadcast that uses local information at the nodes and is thus better suited for networks with large number of nodes.

The paper is organized as follows. In the next section, we give the network model and in Section III, we formulate the problem. In Section IV we present the MLAB algorithm that finds the optimal solution and in Section V we show the benefit of accumulative broadcast to the network lifetime compared to the conventional broadcast. In Section VI we present the distributed MLAB algorithm. Proofs of all theorems are given in the Appendix.

II. System model

We consider a wireless network of N nodes such that from each transmitting node k to each receiving node m, there exists an AWGN channel of bandwidth W characterized by a frequency non-selective link gain h_{mk}. We further assume large enough bandwidth resources to enable each transmission to occur in an orthogonal channel, thus causing no interference to other transmissions. Each node has both transmitter and receiver capable of operating over all channels.

A receiver node j is said to be in the transmission range of transmitter i if the received power at j is above a threshold that ensures the capacity of the channel from i to j is above the code rate of node i. We assume that each node can use different power levels, which will determine its transmission range. The nodes beyond the transmission range will receive an unreliable copy of a transmitted signal. Those nodes can exploit the fact that a message is sent through multiple hops on its way to other nodes. Repeated transmissions act as a repetition code for all nodes beyond the transmission range.

After a certain message has been transmitted from a source, labeled node 1, sequence of retransmissions at appropriate power levels will ensure that eventually every destination node has reliably decoded the message. Henceforth, we focus on the multicast of a single message and say that a node is *reliable* once it has reliably decoded that message. Under the reliable forwarding constraint, a node is permitted to retransmit (forward) only after reliably decoding the message. During the multicast, the message is repeatedly transmitted until the set of destination nodes $\mathcal{D}$ becomes reliable.

The constraint of reliable forwarding imposes an ordering on the network nodes. In particular, a node m will decode a message from the transmissions of a specific set of transmitting nodes that became reliable prior to node m. Starting with node 1, the source, as the first reliable node, a solution to the cooperative multicast problem will be characterized by a *reliability schedule*, which specifies the order in which the nodes become reliable. Since the multicast stops after the message has been delivered to D destination nodes, a reliability schedule will not necessarily contain all the network nodes. In general, a multicast reliability schedule is an ordered subsequence of the list of nodes of length M, $D < M \leq N$, that starts with node 1, and contains all destination nodes and a subset a network nodes that relay the message. In the broadcast case, a reliability schedule $[n_1, n_2, n_2, \ldots, n_N]$ is simply a permutation of $[1, 2, \ldots, N]$ that always starts with the source node $n_1 = 1$.

For a given reliability schedule, we refer to the ith node in the schedule as simply node i. After each node $k \in \{1, \ldots, m-1\}$ transmits with average power p_k, the rate in bits per second that can be achieved at node m is [22]

$$r_m = W \log_2 \left(1 + \frac{\sum_{k=1}^{m-1} h_{mk} p_k}{N_0 W}\right) \qquad \text{bits/s,} \tag{1}$$

where N_0 is the one-sided power spectral density of the additive white Gaussian receiver noise.

Let the required data rate $\overline{r}$ be given by

$$\overline{r} = W \log_2 \left(1 + \frac{\overline{P}}{N_0 W}\right) \qquad \text{bits/s.} \tag{2}$$

From (1) and (2), achieving $r_m = \overline{r}$ implies that the total received power at node m has to be above the threshold $\overline{P}$, that is,

$$\sum_{k=1}^{m-1} h_{mk} p_k \geq \overline{P}. \tag{3}$$

After the data has been successfully delivered to the destination nodes, all those nodes are reliable and the feasibility constraint (3) is satisfied at every destination node m. When communicating at rate $\overline{r}$, the required signal energy per bit is $E_b = \overline{P}/\overline{r}$ Joules/bit. This energy can be collected at a node m during one transmission interval $[0, T]$ from a transmission of a single node k with power $p_k = \overline{P}/h_{mk}$, as commonly assumed in wireless broadcasting problems [2], [3], [5], [11]–[14]. However, using the accumulative strategy, the required energy E_b is collected from $m-1$ prior transmissions.

III. Problem Formulation

A *lifetime of a node* i transmitting with power p_i is given by $T_i(p_i) = e_i/p_i$ where e_i is initial battery energy at node i.

The *network lifetime* is the time until the first node failure,

$$T_{net}(\mathbf{p}) = \min_i T_i(p_i) \tag{4}$$

where $\mathbf{p}$ is a vector of transmitted node powers. The problem is to maximize the network lifetime under the constraints that all destination nodes become reliable. For the multicast problem, broadcasting until the subset of destination nodes becomes reliable will solve the problem.

In the conventional multicast problem, the multicast tree uniquely determines the transmission levels; a relay that is the parent of a group of siblings in the multicast tree transmits with the power needed to reliably reach the most disadvantaged sibling in the group. Hence, the arcs in the multicast tree uniquely determine the power levels for each transmission.

In the accumulative multicast, however, there is no clear parent-child relationship between nodes because nodes collect energy from the transmissions of many nodes. Furthermore, the optimum solution may require that a relay transmits with a power level different from the level precisely needed to reach a group of nodes reliably; the nodes may collect the rest of the needed energy from the future transmissions of other nodes. In fact, the optimum solution often favors such situations because all nodes beyond the range of a certain transmission are collecting energy while they are unreliable; the more such nodes, the more efficiently the transmitted energy is being used.

The differences from the conventional multicast problem dictate a new approach. The optimum solution must specify the reliability schedule as well as the transmit power level at each node. Given a schedule, we can formulate a linear program (LP) that will find the optimum solution for that schedule. Such a solution will identify those nodes that should transmit and their transmission power levels. A schedule is an ordered subsequence of M nodes from a network of N nodes,

$$\mathbf{x} = [x_1, \ldots, x_M], \tag{5}$$

with $x_1 = 1$. We say that the *length* of the subsequence $\mathbf{x}$ in (5) is $\|\mathbf{x}\| = M$. Let

$$\{\mathbf{x}\} = \{x_1, \ldots, x_{\|\mathbf{x}\|}\} \tag{6}$$

denote the set of nodes in a schedule $\mathbf{x}$ and let Π_N denote the set of all variable-length ordered subsequences of $\{1, \ldots, N\}$. It follows that the family of all possible schedules is

$$\mathcal{X}_N(\mathcal{D}) = \{\mathbf{x} \in \Pi_N | \mathcal{D} \in \{\mathbf{x}\}, x_1 = 1\} \tag{7}$$

Given a schedule $\mathbf{x}$, we define a gain matrix $\mathbf{G}(\mathbf{x})$ to have i, jth element

$$[\mathbf{G}(\mathbf{x})]_{ij} = \begin{cases} h_{x_i x_j} & i > j, \\ 0 & \text{otherwise}, \end{cases} \tag{8}$$

for $1 \le i, j \le M$. When a node j does not participate in the retransmission of the message, reliable reception by that node is unnecessary and that node can be omitted from the problem formulation. Thus, channel gains corresponding to any node j that is not in schedule $\mathbf{x}$ are not included in $\mathbf{G}(\mathbf{x})$. We can define the problem of maximizing the network lifetime for schedule $\mathbf{x}$ in terms of the vector $\mathbf{p}$ of transmitted powers as

$$\min \ \max_{\mathbf{p}} \frac{p_i}{e_i} \tag{9}$$

$$\text{subject to } \mathbf{G}(\mathbf{x})\mathbf{p} \ge \mathbf{1}\overline{P}, \tag{9a}$$

$$\mathbf{p} \ge \mathbf{0}. \tag{9b}$$

The inequality (9a) contains $M - 1$ constraints as in (3), requiring that the accumulated received power at all nodes in schedule $\mathbf{x}$ (except the source) is above the threshold $\overline{P}$. It should be apparent that power p_i in $\mathbf{p}$ corresponds to the transmit power of node x_i in the schedule $\mathbf{x}$. Alternatively, we can define the problem in terms of *normalized* node powers $\overline{p}_i = p_i e_1 / e_i$ that account for different battery capacities at the nodes; the lifetime at every node i in terms of the normalized power is as if all the batteries were the same: $T_i = e_i/p_i = e_1/\overline{p}_i$. In terms of normalized node powers, Problem (9) can be defined as

$$\min \ \max_{\overline{\mathbf{p}}} \overline{p}_i \tag{10}$$

$$\text{subject to } \overline{\mathbf{G}}(\mathbf{x})\overline{\mathbf{p}} \ge \mathbf{1}\overline{P},$$

$$\overline{\mathbf{p}} \ge \mathbf{0}$$

where each column $\overline{\mathbf{g}}_i$ of the normalized gain matrix $\overline{\mathbf{G}}(\mathbf{x})$ is obtained from the corresponding column $\mathbf{g}_i$ of matrix $\mathbf{G}(\mathbf{x})$ as $\overline{\mathbf{g}}_i = \mathbf{g}_i e_i / e_1$.

For any schedule $\mathbf{x}$, we can formulate Problem (10) as a linear program in terms of transmit power levels $\overline{\mathbf{p}}$,

$$\hat{p}^*(\mathbf{x}) = \min_{\overline{\mathbf{p}}} \ \hat{p} \tag{11}$$

$$\text{subject to } \overline{\mathbf{G}}(\mathbf{x})\overline{\mathbf{p}} \ge \mathbf{1}\overline{P}, \tag{11a}$$

$$\overline{\mathbf{p}} \le \mathbf{1}\hat{p} \tag{11b}$$

$$\overline{\mathbf{p}} \ge \mathbf{0}. \tag{11c}$$

If $\hat{p} = \hat{p}^*(\mathbf{x})$, then there exists a power vector $\overline{\mathbf{p}}$ such that (11a) and (11b) are satisfied. It follows that for any $p > \hat{p}$, $\overline{\mathbf{p}} \le \mathbf{1}p$. Thus, for any power $\hat{p} \ge \hat{p}^*(\mathbf{x})$, we say that power $\hat{p}$ is *feasible* for schedule $\mathbf{x}$. Over all possible schedules, the optimum power is

$$p^* = \min_{\mathbf{x} \in \mathcal{X}_N(\mathcal{D})} \hat{p}^*(\mathbf{x}). \tag{12}$$

Equation (12) is a formal statement of the problem from which finding the best schedule corresponding to p^* is not apparent. We will see that the power p^*, may, in fact, be the solution to (11) for a set of schedules, $\mathcal{X}^*$. In the rest of the paper, we will consider only normalized powers and we therefore drop the overline notation; $\mathbf{H}$ will denote the ordinary gain matrix, $\mathbf{G}(\mathbf{x})$ will denote the gain matrix permuted for schedule $\mathbf{x}$, and the power vector will be simply $\mathbf{p}$, with p_i representing either the power of node i or node x_i, as appropriate for the context.

Rather than identifying $\mathcal{X}^*$, we employ a simple procedure that for any power p, determines a collection of schedules for which power p is feasible. In particular, to distribute a message, we let each node retransmit with power p *as soon as possible*, namely as soon as it becomes reliable. We refer to such a distribution as the *ASAP*(p) *distribution*. During the

ASAP(p) distribution, the message will be resent in a sequence of retransmission stages from sets of nodes $Z_1(p), Z_2(p), \ldots$ with power p where in each stage i, a set Z_i that became reliable during stage $i-1$, transmits and makes Z_{i+1} reliable.

Let $S_i(p)$ and $U_i(p)$ denote the reliable nodes and unreliable nodes at the start of stage i. $U_{D,i}(p) \subset U_i(p)$ is the set of unreliable destination nodes at the start of stage i. Then, $Z_1(p) = 1$ and $S_i(p) = Z_1(p) \cup \ldots \cup Z_i(p)$. The set $Z_{i+1}(p)$ is given by

$$Z_{i+1}(p) = \{z \in U_i(p) : p \sum_{k \in S_i(p)} h_{zk} \geq \overline{P}\}. \tag{13}$$

Note that if power p is too small, the ASAP(p) distribution can *stall* at stage i with $S_{i+1}(p) = S_i(p)$ and $U_{D,i}(p) \neq \emptyset$, the empty set. In this case, ASAP(p) fails to distribute the message to all destination nodes. When $U_{D,i}(p) = \emptyset$ at a stage i, the ASAP(p) distribution *terminates successfully*. We will say that ASAP(p) distribution is a *feasible multicast* if it terminates successfully.

The partial node ordering, $Z_1(p), Z_2(p), \ldots$, specifies the sequence in which nodes became reliable during the ASAP(p) distribution. In particular, any schedule $\mathbf{x}$ that is consistent with this partial ordering is a feasible schedule for power p. Nodes that become reliable during the same stage of ASAP(p) can be scheduled in an arbitrary order among themselves since these nodes do not contribute to each other's received power. The following theorem verifies that in terms of maximizing the network lifetime it is sufficient to consider only schedules consistent with the ASAP(p) distribution.

Theorem 1: If $\tilde{p}$ is a feasible power for a schedule $\tilde{\mathbf{x}}$, then the ASAP($\tilde{p}$) distribution is a feasible multicast.

In particular, Theorem 1 implies that for optimum power p^*, the ASAP(p^*) distribution is feasible.

We next present the *Maximum Lifetime Accumulative Broadcast (MLAB)* algorithm, that determines the optimum power p^*. Once the power p^* is given, broadcasting with ASAP(p^*) will maximize the network lifetime.

IV. The MLAB algorithm

We label node 1 as the source and 2 as its closest neighbor (more precisely, the node with the highest link gain to the source). The MLAB algorithm finds the optimum power p^* through a series of ASAP(p) distributions, starting with the smallest possible *candidate broadcast power*, $p = \overline{P}/h_{21}$. Whether ASAP(p) stalls or terminates successfully, we define $\tau(p)$ as the terminating stage. When $p = p^*$, the ASAP(p^*) distribution will terminate in $\tau^* = \tau(p^*)$ stages. When the ASAP(p) distribution stalls at stage $\tau(p)$, we determine the minimum power increase δ for which ASAP($p + \delta$) will not stall at stage $\tau(p)$, in the following way. The increase in broadcast power δ_j needed to make a node $j \in U_{\tau(p)}(p)$ reliable must satisfy

$$\overline{P} = (p + \delta_j) \sum_{k \in S_{\tau(p)}(p)} h_{jk}. \tag{14}$$

We choose $\delta = \min_{j \in U_{\tau(p)}(p)} \delta_j$. We then increase p to $p + \delta$ and *restart* the MLAB algorithm. The algorithm stops when an ASAP(p) distribution terminates successfully.

```
Initialize: $p = \overline{P}/h_{21}$
Start: Set $S_1(p) = \{1\}$; $U_1(p) = S^c$
          apply the ASAP($p$) distribution;
If ASAP($p$) stalls at stage $\tau(p)$:
    for all $j \in U_{\tau(p)}(p)$ calculate:
        $\delta_j = \overline{P}/\sum_{k \in S_{\tau(p)}(p)} h_{jk} - p$;
    Set: $\delta = \min_{j \in U_{\tau(p)}(p)} \delta_j$; $p \leftarrow p + \delta$;
    go to Start;
end
```

The cardinality of S is given by $|S|$. S^c denotes the complement.

Fig. 1. MLAB algorithm.

The pseudocode of the algorithm is given in Figure 1. The MLAB algorithm ends after at most $N-1$ restarts. There exists a set of feasible schedules that are consistent with the partial ordering given by the ASAP(p) distribution. The normalized transmit power at all nodes in $S_{\tau(p)}(p)$ is p. Note that the last transmitting set $Z_{\tau(p)}$ could in fact, transmit with power less than p if it is enough for the last set of unreliable destination nodes, $U_{D,\tau(p)}(p)$, to become reliable. Thus, choosing the power level at all nodes to be p is not necessarily a unique solution. While this won't change the network lifetime, the latter solution will reduce the total transmit power in the network. Next we show that the power found by MLAB is in fact the optimum power, that is, $p = p^*$.

Theorem 2: The MLAB algorithm finds the optimum power p^* such that the ASAP(p^*) distribution maximizes the network lifetime.

Finally, we note that the full restarts of the MLAB algorithm are used primarily to simplify the proof of Theorem 2. In fact, when MLAB stalls, it is sufficient for the reliable nodes to offer incremental retransmissions at power Δ^*. This observation will be the basis of distributed algorithm proposed in Section VI.

V. Performance

We now evaluate the benefit of accumulative broadcast to the network lifetime and compare it to the conventional network broadcast that discards overheard data in a network. In particular, networks with randomly positioned nodes in a 10 x 10 square region were generated. The transmitted power was attenuated with distance d as d^α for different values of propagation exponent $\alpha = 2, 3, 4$. The received power threshold was chosen to be $\overline{P} = 1$. Results were based on the performance of 100 randomly chosen networks.

Figure 2 shows the power p for different values of propagation exponent in networks with different node densities. The observed power decrease is due to shorter hops between nodes in denser networks. For equal battery capacities at the nodes, the corresponding network lifetime is shown in Figure 3.

Figures 4 and 5 show the benefit of accumulative broadcast as compared to conventional broadcast in terms of network lifetime. For conventional broadcast, the authors in [11], [12] proposed two algorithms, MSNL and MST, that maximize the

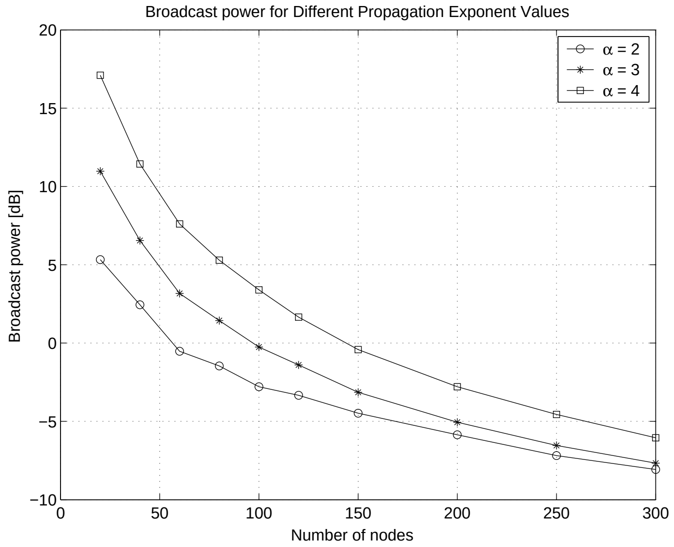

Fig. 2. Broadcast power for different propagation exponent values.

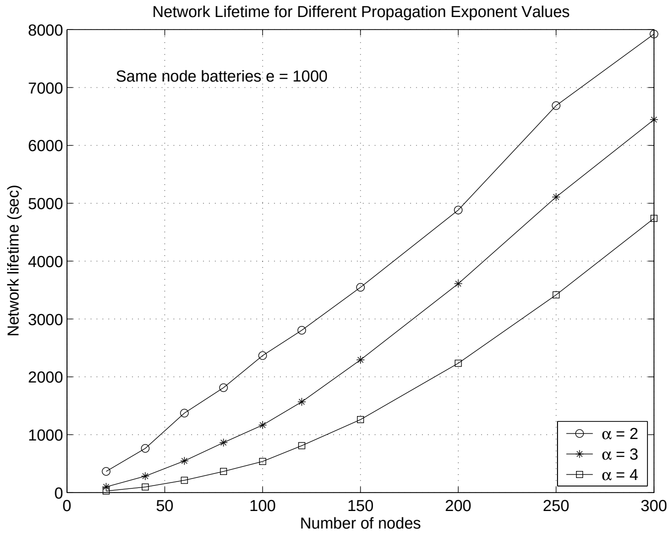

Fig. 3. Network lifetime for different propagation exponent values.

static network lifetime as well WMSTSW, a greedy algorithm that increases the dynamic lifetime. We compare the performance of these algorithms for three different battery energy distribution as given in [11], [12], to the network lifetime found by the MLAB algorithm. We first assume that all the nodes have identical batteries. Then, we consider two different node battery scenarios in which the initial battery energies at the nodes are independent uniform $(0, 1000)$ or uniform $(500, 1000)$ random variables. Several other algorithms to increase the dynamic network lifetime were evaluated in [12] with similar performance to WMSTSW. As expected, we see that solution found by MLAB considerably increases network lifetime. Typically, MLAB increased the network lifetime by a factor of 2 or more. The reason is twofold: first, because the broadcast uses the energy of overheard information enabling for more radiated energy to be captured. And second, because the accumulative broadcast enables MLAB to distribute the load more evenly among the nodes than does the dynamic load balancing in conventional broadcast.

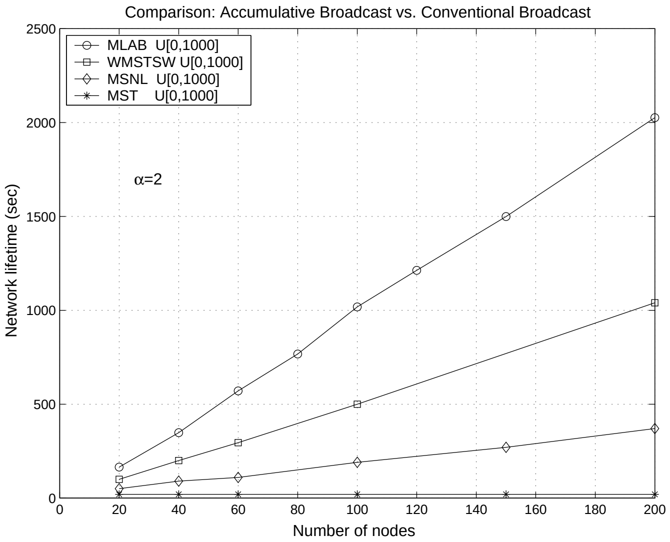

Fig. 4. Network lifetime of accumulative broadcast and conventional broadcast.

VI. Distributed MLAB Algorithm

We next describe a distributed MLAB algorithm for accumulative broadcast that determines broadcast power locally at each node. Nodes are assumed to have no knowledge of link gains (distances) to other nodes at the beginning of the algorithm. The distributed algorithm will be run at the beginning of a broadcast session during the broadcast of the first message. Let q denote the broadcast power determined by the distributed MLAB. Once the power q is determined, data will be broadcasted through ASAP(q) distribution. In a static network where the same power q is used throughout a long broadcast session, the initial overhead to determine q will be small compared to the amount of broadcast data.

The distributed implementation of MLAB algorithm has to resolve the following:

1) *When should a reliable node decide to increase the broadcast power?*
2) *How much should a reliable node increase the broadcast power?*

When the ASAP(p) distribution stalls, determining the necessary power increase δ, requires global knowledge of network gains and cannot be computed locally at a node. In the distributed MLAB algorithm, the broadcast power will be increased in steps of size Δ, for some small fixed power Δ. Further, during the initial broadcast phase while the algorithm is run to determine q, we let Δ be the transmit power of every transmission. A reliable node intending to transmit with power $n\Delta$ for some $n > 1$ will instead repeatedly transmit for n times, each time with power Δ. A transmission from a node i with power Δ will be overheard by a number of nodes that define a Δ-*neighborhood* $N_i(\Delta)$ of node i. Nodes will belong to $N_\Delta(i)$ if they can detect the presence of a signal sent at

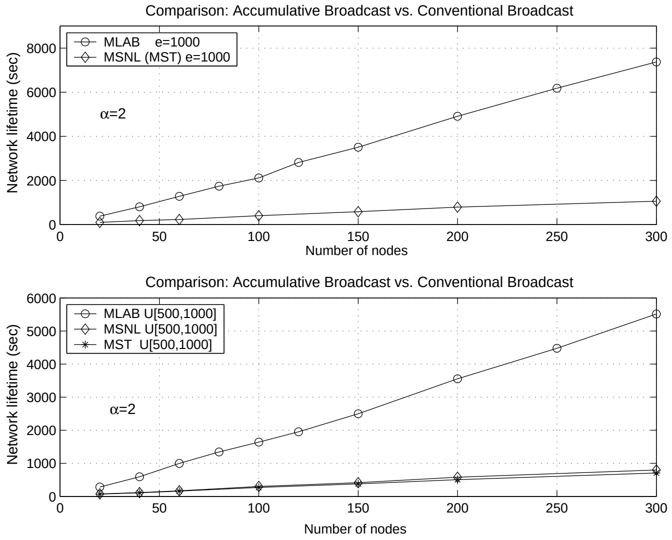

Fig. 5. Network lifetime of accumulative broadcast and conventional broadcast.

node i, although their received power may not be sufficient for reliable decoding.

Overhearing a broadcast from a node k will enable node i to determine the link gain h_{ik} and identify node k as its reliable neighbor. During the algorithm, node i will keep track of its set of reliable neighbors, $R_i \subset N_i(\Delta)$. From the number of repeated transmissions at node k, node i will also be able to determine the current transmit power at node k. Because the transmit power will not necessarily be the same at all nodes all the time, node i will keep track of transmit power $p_i(k) = n_i(k)\Delta$ for every $k \in R_i$, where $n_i(k)$ is a number of repeated broadcasts by node k. In addition, once reliable, node i will keep track of its unreliable neighbors, U_i. An unreliable node j will send NACK$_j$ control messages to identify itself. As node i becomes reliable, it will broadcast with maximum power among its reliable neighbors in R_i,

$$p(R_i) = \Delta \max_{k \in R_i} \{n_i(k)\}.$$

While reliable, whenever it overhears a transmission that increases the power $p(R_i)$, node i will repeat the broadcast to meet it. In that way, the current maximum transmit power in the network $\overline{q}$ will *propagate* until all reliable nodes have transmitted with that power. A reliable node that overhears no transmissions for time $T_o > \tau(\Delta)$ and has unreliable neighbors, will decide to increase its transmit power. At the end of the algorithm, power $\overline{q}$ will determine the broadcast power q. A detailed description of the algorithm is given in pseudocode in Figure 6.

Constraining the power of each transmission to Δ defines Δ-neighborhoods and allows nodes to determine the link gains within their Δ-neighborhoods. Therefore, power Δ defines the network topology and has to be high enough to guarantee network connectivity [23]. In the distributed MLAB algorithm, it is sufficient that under power Δ, the network is connected in the overheard sense. That is, in the underlying graph, a link between two nodes exists if they can overhear each other.

```
At each node $i$ do:
initialize $R_i = \emptyset$, $p_R = 0$;
while ($p_R < \overline{P}$) do:
    when data received with $P$ from $k$:
        collect data; $p_R \leftarrow p_R + P$;
        if $k \notin R_i$:
            $h_{ik} = P/\Delta$, $n_i(k) = 0$, $R_i \leftarrow R_i \cup \{k\}$;
            send NACK$_i$ reliably to $k$;
        end %if
        $n_i(k) = n_i(k) + 1$;
end % while
as ($p_R \geq \overline{P}$) do once:
    decode the message;
    set $n_i(i) = \max_{k \in R_i} \{n_i(k), 1\}$;
    broadcast the decoded message once;
    $U_i = \{j : j$ that responded with NACK$_j\}$;
    broadcast $n_i(i) - 1$ times;
while ($p_R \geq \overline{P}$) do:
    when data received from node $k$:
        update $n_i(k) = n_i(k) + 1$;
        if $n_i(k) > n_i(i)$:
            $n_i(i) \leftarrow n_i(i) + 1$, broadcast;
        if $k \notin R_i$: $R_i \leftarrow R_i \cup \{k\}$, $U_i \leftarrow U_i \setminus \{k\}$;
    if no data received for $T_o$ and $U_i \neq \emptyset$:
        broadcast;
        $n_i(i) \leftarrow n_i(i) + 1$;
    end %if
end % while
```

Received power at a node is denoted p_R.

Fig. 6. Distributed MLAB algorithm.

During MLAB, we assume that the network is connected under power Δ.

This assumption is not essential for the algorithm and can be relaxed by letting MLAB algorithm rely on preexisting network topology. Different distributed algorithms for determining network topology have been proposed (see [24], [25]) and typically employ short HELLO control packets exchanged at the nodes. Given the power P_c and rate r_c of control packets, HELLO packets define one-hop neighborhood $N_i(P_c)$ for node i as all nodes that can reliably receive a HELLO$_i$ packet sent at node i. A version of the distributed MLAB algorithm can then be run on the top of the topology defined by neighborhoods $N_i(P_c)$ instead of $N_i(\Delta)$.

Note that decreasing the rate r_c reduces the power P_c necessary for network connectivity by reducing the receiver power threshold needed for reliable communication. Connectivity in overheard sense, required for Δ-neighborhoods, reduces this threshold to its minimum value necessary to acquire a signal or decode a packet header and thus reduces necessary power for connectivity. Therefore, it may be reasonable to assume that under power Δ, network is connected. The next theorem shows that the algorithm is correct and finishes in finite time.

Theorem 3: The distributed MLAB algorithm makes every network node reliable in finite time.

The running time and performance of the algorithm are dependent on value of parameters Δ and T_o. In fact, we have the following theorem.

Theorem 4: For large enough T_o, $T_o > \tau(\Delta)$, power q found by the distributed MLAB is within Δ of the optimum solution; that is, $q \in [p^*, p^* + \Delta)$.

Thus, by choosing smaller Δ, solution found by the distributed MLAB approaches the optimum, at the expense of longer running time due to the larger T_o and smaller step size Δ. When the distributed MLAB does not rely on a preexisting topology, there is a lower bound on Δ to guarantee network connectivity. At the other extreme, for Δ large enough to guarantee full connectivity (every node can overhear every transmission), T_o can be chosen to be 0. The optimal tuning of the algorithm parameters has yet to be determined.

VII. Conclusion

In this paper, we addressed the cooperative multicast network lifetime problem and proposed the *Maximum Lifetime Accumulative Broadcast (MLAB)* algorithm that finds an optimum solution. The constant power levels found by the algorithm ensure that the lifetimes of the active relays are the same, causing them to fail simultaneously. Furthermore, the MLAB algorithm solves both the cooperative broadcast and cooperative unicast problem that are useful for many applications. In sensor networks, for example, the unicast problem arises in any scenario where the data collected at the sensors is gathered by a central station.

The ASAP(p^*) solution found by MLAB is *static* since it stays constant throughout the multicast session. In conventional broadcast, the constraint that a node is made reliable by the transmission of a single relay, causes the relay with the most disadvantaged child to drain its battery fastest. Consequentially, the optimality of a spanning tree that maximizes the network lifetime for a given initial battery levels is temporary and dynamic tree updates [12], [13] are needed for load balancing. In a cooperative multicast using the ASAP(p^*) distribution, all relays will be draining their batteries evenly; however, a set of *leaf nodes* $U_{\tau^*}(p^*)$ will never transmit and will have full batteries even when the relay nodes die. An significant question is whether the undepleted batteries of these leaf nodes can be exploited by a dynamic multicast strategy.

After multiple uses of the ASAP(p^*) distribution, reexamination of the maximum lifetime problem (12), as expressed in terms of normalized powers, will show for each non-relay node $j \in U_{\tau^*}(p^*)$ that the outgoing normalized link gains h_{kj} have increased by the ratio of the full battery energy of node j to the depleted battery of node 1. Although one can show that reconfiguring the multicast distribution to maximize the residual network lifetime results in the very same ASAP(p^*) distribution, it would be mistake to conclude that ASAP(p^*) policy is an optimal dynamic policy. In fact, similar to the conventional broadcast, a dynamic strategy with time varying powers can extend the network lifetime. For example, in the four node network shown in Figure 7, the source node 1 wishes to send messages to the destination node 4. With initial battery powers $e_i = 1$ and required received power $\overline{P} = 1$, the ASAP solution use transmissions by nodes 1, 2 and 3, each with power $p^* = 2/3$. The lifetime of each node is $e_i/p^* = 3/2$. On the other hand, alternating between schedule $\mathbf{x} = [1, 2]$ with power vector $\mathbf{p} = [1/3, 1, 0]$ and schedule $\mathbf{x}' = [1, 3]$ with power vector $\mathbf{p}' = [2/3, 0, 1]$ results in a system which has average transmit power of $1/2$ for each node and resulting network lifetime 2. In this case, dynamic switching between schedules, corresponding to routing packets along multiple routes, yields a network lifetime larger than that of the ASAP distribution, the optimal static policy. A general solution for the optimal dynamic cooperative multicast remains an open problem at this time.

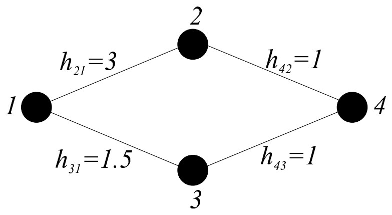

Fig. 7. Four node network example.

In this paper, the cooperative multicast was proposed for the AWGN channel with constant link gains. However, the cooperation between the nodes in the fading channel offers the additional benefit as a form of diversity [26]–[28]. It would also be interesting to consider the implications of time varying channels to the accumulative multicast problems.

VIII. Appendix

Proof: Theorem 1

The proof is by induction on k, where k is the index to a sequence of stages during the ASAP($\tilde{p}$) distribution. We prove by induction that at the start of stage k, nodes $\{x_1, \ldots, x_k\} \subset S_k(\tilde{p})$. In case that the number of stages is $\tilde{\tau} = \tau(\tilde{p}) < M$, we define $S_k(\tilde{p}) = S_{\tilde{\tau}}(\tilde{p})$ for all $\tilde{\tau} < k \leq M$. The idea is that ASAP($\tilde{p}$) makes nodes reliable at least as soon as the schedule $\tilde{\mathbf{x}}$.

Case $k = 1$ is obvious since $S_1(\tilde{p}) = \{1\}$ for any $\tilde{p}$. Next we assume that $\{x_1, \ldots, x_k\} \subset S_k(\tilde{p})$. This implies

$$\tilde{p} \sum_{x_j \in S_k(\tilde{p})} g_{k+1,j} \geq \tilde{p} \sum_{x_j \in \{x_1, \ldots, x_k\}} g_{k+1,j} \geq^{(a)} \overline{P} \qquad (15)$$

where (a) follows from the feasibility of power $\tilde{p}$ for schedule $\tilde{\mathbf{x}}$, because under schedule $\tilde{\mathbf{x}}$, node x_{k+1} is made reliable by transmissions of $\{x_1, \ldots, x_k\}$. We conclude that $x_{k+1} \in S_{k+1}(\tilde{p})$ and since $\{x_1, \ldots, x_k\} \subset S_k(\tilde{p}) \subset S_{k+1}(\tilde{p})$, it follows that $\{x_1, \ldots, x_{k+1}\} \subset S_{k+1}(\tilde{p})$. Thus, $\{x_1, \ldots, x_M\} \subset S_M(\tilde{p})$, implying the ASAP($\tilde{p}$) distribution makes all the nodes in a schedule $\tilde{\mathbf{x}}$, and thus all destination nodes, reliable. □

Proof: Theorem 2

Suppose the last restart of the MLAB algorithm occurs when the power is p_0 and the ASAP(p_0) distribution stalls at stage

$\tau_0 = \tau(p_0)$. This implies

$$p_0 \sum_{k\in S_{\tau_0}(p_0)} h_{jk} < \overline{P}, \qquad j \in U_{\tau_0}(p_0). \tag{16}$$

In this case, we restart MLAB with broadcast power $p_0 + \delta_0$ where $\delta_0 = \min_{j\in U_{\tau_0}(p_0)} \delta_j$ and δ_j satisfies

$$(p_0 + \delta_j) \sum_{k\in S_{\tau_0}(p_0)} h_{jk} = \overline{P}. \tag{17}$$

This implies

$$(p_0 + \delta_0) \sum_{k\in S_{\tau_0}(p_0)} h_{jk} \le \overline{P}, \qquad j \in U_{\tau_0}(p_0). \tag{18}$$

Since this is the last restart of MLAB, the ASAP($p_0 + \delta_0$) distribution is a feasible multicast. It follows that $p^* \le p_0 + \delta_0$ since p^* is the optimal broadcast power. To show that $p^* = p_0 + \delta_0$ requires the following lemma.

Lemma 1: For any power $p' < p_0 + \delta_0$, the ASAP(p') distribution stalls at stage $\tau' = \tau(p')$ with $S_{\tau'}(p') \subset S_{\tau_0}(p_0)$.

Lemma 1 implies that if $p^* < p_0 + \delta_0$, then the ASAP(p^*) distribution will stall, which is a contradiction of Theorem 1. Thus, at the final restart of the MLAB algorithm, the power is $p_0 + \delta_0 = p^*$.

Proof: Lemma 1

Let $\mathcal{F} = S_{\tau'}(p') \setminus S_{\tau_0}(p_0)$. First, we show by contradiction that $\mathcal{F}$ is an empty set. Suppose $\mathcal{F}$ is nonempty. Let $\tau_{\mathcal{F}}$ denote the first stage in which a node $j' \in \mathcal{F}$ was made reliable by the ASAP(p') distribution. Thus,

$$\overline{P} \le p' \sum_{k\in S_{\tau_{\mathcal{F}}}(p')} h_{j'k}. \tag{19}$$

Moreover, $S_{\tau_{\mathcal{F}}}(p') \subset S_{\tau_0}(p_0)$ since up to stage $\tau_{\mathcal{F}}$, all nodes that were made reliable by ASAP(p') belong to $S_{\tau_0}(p_0)$. Hence,

$$\begin{aligned} \overline{P} &\le p' \sum_{k\in S_{\tau_0}(p_0)} h_{j'k} & (20)\\ &<^{(a)} (p_0 + \delta_0) \sum_{k\in S_{\tau_0}(p_0)} h_{j'k} & (21)\\ &\le^{(b)} \overline{P} & (22) \end{aligned}$$

since (a) follows from $p' < p_0 + \delta_0$ and (b) follows from Equation (18). Thus we have the contradiction $\overline{P} < \overline{P}$ and we conclude that $\mathcal{F}$ is empty, $S_{\tau'}(p') \subset S_{\tau_0}(p_0)$, and $U_{\tau_0}(p_0) \subset U_{\tau'}(p')$. Second, we observe that ASAP(p') stalls at stage τ' since for all $j \in U_{\tau'}(p')$,

$$\begin{aligned} p' \sum_{k\in S_{\tau'}(p')} h_{jk} &\le p' \sum_{k\in S_{\tau_0}(p_0)} h_{jk} & (23)\\ &< (p_0 + \delta_0) \sum_{k\in S_{\tau_0}(p_0)} h_{jk} \le \overline{P}. & (24) \end{aligned}$$

□

Proof: Theorem 3

In a network that is connected under power Δ, there is a path from the source node to every other node in the network. Consider a path from node 1 to some node K. We relabel the nodes such that the path is given by $[1, 2, \ldots K]$. For any reliable node $1 \le k \le K - 1$ such that $k + 1$ is unreliable, it holds that $k + 1 \in U_k$. By distributed MLAB, node k will increase its transmit power whenever it overhears no transmissions for T_o, until $k + 1$ is reliable. Thus, eventually all the nodes on the path will be reliable. This holds for any path for any node K.

We next find an upper bound on $\mathcal{T}_i$, time it takes for a node i to make all of its neighbors reliable. An upper bound on the number of transmissions needed at a node i to make node $j \in N_i(\Delta)$ reliable, neglecting the energy node j may have collected from transmission from other nodes, is $\lceil \overline{P}/h_{ji}\Delta \rceil$. In the worst case, node i will wait for T_o between any two consecutive broadcasts. Thus,

$$\begin{aligned} \mathcal{T}_i &\le \max_{j\in N_i(\Delta)} \left\{ T_o \left\lceil \frac{\overline{P}}{\Delta h_{ji}} \right\rceil \right\} & (25)\\ &= T_o \left\lceil \frac{\overline{P}}{\Delta \min_{j\in N_i(\Delta)}\{h_{ji}\}} \right\rceil . & (26) \end{aligned}$$

Since $j \in N_i(\Delta)$, it follows that $h_{ji} \ne 0$ and therefore $\mathcal{T}_i$ is finite for every node i. Since there is only a finite number of nodes, all nodes will be made reliable in finite time. □

Proof: Theorem 4

To prove Theorem 4, we next upper bound the time $\mathcal{T}(k)$ it takes for maximum transmit power $\overline{q} = k\Delta$, $k > 0$ to propagate through the network.

Lemma 2: Let T be the duration of a single transmission and let $\tau(k) = N(k + 1)T$. Then, $\mathcal{T}(k) < \tau(k)$.

Proof: Lemma 2

The time it takes for one node to transmit with $\overline{q}$ is upper bounded by kT, the case when the node never previously transmitted. Since the node may have to wait for NACKs for additional time T, the total time at a node is upper bounded by $(k + 1)T$, Since the propagation cannot take more than N hops, the total time is upper bounded by $N(k + 1)T$. □

To prove Theorem 4, we first observe that power q is lower bounded by p^*: before the power p^* is reached, there are always nodes that are unreliable and the distributed MLAB does not stop at the reliable nodes. The power p^* is reached for $\overline{q} = \lceil p^*/\Delta \rceil \Delta < p^* + \Delta$ and no further increase in power is neccessary. By Lemma 2, $\overline{q}$ will propagate in less than $\tau(\Delta) = N(\lceil p^*/\Delta \rceil + 1)T$ time. If $T_o \ge \tau(\Delta)$, no node will increase $\overline{q}$ before all reliable nodes transmitted with $\overline{q}$. However, at that point all network nodes will be reliable and distributed MLAB will stop at all nodes with $q = \overline{q}$. □

References

[1] I. Maric and R. Yates, *Efficient Multihop Broadcast for Wideband Systems*, Book chapter: multiantenna channels: capacity, coding and signal processing, DIMACS workshop on signal processing for wireless transmission ed., G. J. Foschini and S. Verdu, Eds. American Mathematical Society, Oct. 2002, vol. 62.

[2] A. Ahluwalia, E. Modiano, and L. Shu, "On the complexity and distributed construction of energy-efficient broadcast trees in static ad hoc wireless networks," in *Proc. of Conf. on Information Science and Systems*, Mar. 2002, pp. 807–813.

[3] M. Cagalj, J. Hubaux, and C. Enz, "Energy-efficient broadcast in all-wireless networks," *ACM/Kluwer Mobile Networks and Applications (MONET); to appear*, 2003.

[4] W. Liang, "Constructing minimum-energy broadcast trees in wireless ad hoc networks," in *Proc. of International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'02)*, June 2002, pp. 112–122.

[5] J. Wieselthier, G. Nguyen, and A. Ephremides, "On the construction of energy-efficient broadcast and multicast trees in wireless networks," in *Proc. of INFOCOM'00*, Mar. 2000, pp. 585–594.

[6] ——, "Energy-efficient broadcast and multicast trees in wireless networks," *Mobile Networks and Applications (MONET)*, vol. 7, pp. 481–492, Dec. 2002.

[7] F. Li and I. Nikolaidis, "On minimum-energy broadcasting in all-wireless networks," in *Proc. of Local Computer Networks (LCN 2001)*, Nov. 2001.

[8] J. Cartigny, D. Simplot, and I. Stojmenovic, "Localized minimum-energy broadcasting in ad hoc networks," in *Proc. of INFOCOM'03*, Mar. 2003.

[9] J. H. Chang and L. Tassiulas, "Routing for maximum system lifetime in wireless ad-hoc networks," in *Proc. of 37-th Annual Allerton Conference on Communication, Control and Computing*, Sept. 1999.

[10] J. Wieselthier, G. Nguyen, and A. Ephremides, "Resource management in energy-limited, bandwidth-limited, transceiver-limited wireless networks for session-based multicasting," *Computer Networks*, vol. 39, pp. 113–131, June 2002.

[11] I. Kang and R. Poovendran, "Maximizing static network lifetime of wireless broadcast adhoc networks," in *Proc. of ICC'03*, May 2003.

[12] ——, "Maximizing network lifetime of broadcasting over wireless stationary adhoc networks," *submitted.*

[13] R. J. M. II, A. K. Das, and M. El-Sharkawi, "Maximizing lifetime in an energy constrained wireless sensor array using team optimization of cooperating systems," in *Proc. of the International Joint Conf. on Neural Networks, IEEE World Congress on Computational Intelligence*, May 2002.

[14] P. Floreen, P. Kaski, J. Kohonen, and P. Orponen, "Multicast time maximization in energy constrained wireless networks," in *Proc. of ACM/IEEE Mobicom 2003 Workshop on Foundations of Mobile Computing*, Sept. 2003.

[15] J. Wieselthier, G. Nguyen, and A. Ephremides, "Energy-aware wireless networking with directional antennas: The case of session-based broadcasting and multicasting," *IEEE Transactions on Mobile Computing*, vol. 1, no. 3, pp. 176–191, July 2002.

[16] T. Cover and A. E. Gamal, "Capacity theorems for the relay channel," *IEEE Trans. on Information Theory*, vol. 25, no. 5, pp. 572–584, Sept. 1979.

[17] I. Maric and R. Yates, "Efficient multihop broadcast for wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 22, no. 6, pp. 1080 – 1088, Aug. 2004, special issue on fundamental performance limits of wireless sensor networks.

[18] ——, "Efficient multihop broadcast for wideband systems," in *Proc.of the Allerton Conference on Communications, Control and Computing, Monticello, IL*, Oct. 2002.

[19] Y. Hong and A. Scaglione, "Energy-efficient broadcasting with cooperative transmission in wireless sensory ad hoc networks," in *Proc. of Allerton Conference*, Oct. 2003.

[20] M. Agarwal, J. Cho, L. Gao, and J. Wu, "Energy efficient broadcast in wireless ad hoc networks with hitch-hiking," in *Proc. of INFOCOM'04*, June 2004.

[21] J. H. Chang and L. Tassiulas, "Energy conserving routing in wireless ad-hoc networks," in *Proc. of INFOCOM'00*, Mar. 2000.

[22] E. Telatar, "Capacity of multi-antenna gaussian channels," in *Europ. Trans. Telecommunications*, vol. 10, Nov. 1999, pp. 585–595.

[23] P. Gupta and P. R. Kumar, "Critical power for asymptotic connectivity in wireless networks," *Stochastic Analysis, Control, Optimization and Applications: A Volume in Honor of W.H. Fleming*, 1998.

[24] N. Li and C. Hou, "BLMST: a scalable, power efficient broadcast algorithm for wireless sensor networks," in *IEEE INFOCOM 2004, submitted.*

[25] S. Narayanaswamy, V. Kawadia, R. S. Sreenivas, and P. R. Kumar, "Power control in ad-hoc networks: Theory, architecture, algorithm and implementation of the COMPOW protocol," in *European Wireless 2002*, Feb. 2002, pp. 156–162.

[26] J. N. Laneman, D. N. C. Tse, and G. W. Wornell, "Cooperative diversity in wireless networks: efficient protocols and outage behavior," *IEEE Trans. on Information Theory, to appear*.

[27] A. Sendonaris, E. Erkip, and B. Aazhang, "User Cooperation Diversity–Part I: System Description," *IEEE Transactions on Communications*, vol. 51, no. 11, pp. 1927 – 1938, November 2003.

[28] A. Catovic, S. Tekinay, and T. Otsu, "Reducing transmit power and extending network lifetime via user cooperation in the next generation wireless multihop networks," *Journal on Communications and Networks*, vol. 4, no. 4, pp. 351–362, Dec. 2002.

PLACE
PHOTO
HERE

Ivana Maric received the B.S. in electrical engineering from University of Novi Sad, Yugoslavia in 1995 and the M.S. in electrical engineering from Rutgers University, Piscataway in 2000. She is a currently working toward the Ph.D. degree in Wireless Information Network Laboratory (WINLAB) at the same university. She was a summer intern in AT&T Research Labs, Red Bank, NJ in 1998. Her research interest include network information theory, wireless ad hoc networks and more specifically, wireless relay channels and cooperative broadcast.

PLACE
PHOTO
HERE

Roy D. Yates received the B.S.E. degree in 1983 from Princeton University, and the S.M. and Ph.D. degrees in 1986 and 1990 from M.I.T., all in Electrical Engineering. Since 1990, he has been with the Wireless Information Networks Laboratory (WINLAB) and the E&CE department at Rutgers University. Presently, he is an Associate Director of WINLAB and a Professor in the ECE Dept. He is a co-author (with David Goodman) of the text *Probability and Stochastic Processes: A Friendly Introduction for Electrical and Computer Engineers* published by John Wiley and Sons. Roy Yates is a co-recipient of the 2003 IEEE Marconi Paper Prize Award in Wireless Communications. His research interests include power control, interference suppression and spectrum regulation.