Energy Efficient Communications in Ad Hoc Networks Using Directional Antennas[1]

Akis Spyropoulos and C.S. Raghavendra

Department of Electrical Engineering-Systems
University of Southern California
Los Angeles, CA 90089-2562

Abstract— **Directional antennas can be useful in significantly increasing node and network lifetime in wireless ad hoc networks. In order to utilize directional antennas, an algorithm is needed that will enable nodes to point their antennas to the right place at the right time. In this paper we present an energy-efficient routing and scheduling algorithm that coordinates transmissions in ad hoc networks where each node has a single directional antenna. Using the topology consisting of all the possible links in the network, we first find shortest cost paths to be energy efficient. Then, we calculate the amount of traffic that has to go over each link and find the maximum amount of time each link can be *up*, using end-to-end traffic information to achieve that routing. Finally, we schedule nodes' transmissions, trying to minimize the total time it takes for all possible transmitter-receiver pairs to communicate with each other. We formulate this link problem as solving a series of maximal-weight matching in a graph. Furthermore, we propose a method that can enable our scheduling algorithm to work in a distributed and adaptive fashion. We demonstrate that our algorithm achieves all the possible transmitter/receiver gains possible from using directional antennas. In addition, we illustrate through simulation that our routing scheme achieves up to another 45% improvement in energy cost for routing.**

Index terms—**directional antenna, energy, routing, scheduling, matching.**

I. INTRODUCTION

Wireless ad-hoc networks are multi-hop networks where all nodes cooperatively maintain network connectivity. The ability to be set up fast and operate without the need of any wired infrastructure (e.g. base stations, routers, etc.) makes them a promising candidate for military, disaster relief, and law enforcement applications. Furthermore, the growing interest in sensor network applications has created a need for protocols and algorithms for large-scale self-organizing ad-hoc networks, consisting of hundreds or thousands of nodes.

One important characteristic of such networks is that nodes are energy-constrained. Nodes are battery-operated and frequent recharging or replacement of batteries may be undesirable or even impossible. This makes energy-efficiency an important metric, against which any new protocol/algorithm should be compared. Many different *power-aware* algorithms and protocols have been proposed to conserve the node's energy [9], [11], [12], [13], [14], [15], [16], [19]. Most of these algorithms try to improve the energy efficiency of a certain protocol layer, like the network layer [9], [12], media access control layer [11], link layer [16], [20] or transport layer [15]. Among the different techniques that have been suggested to save power, some of the most commonly used and successful ones are "shutting down nodes" [11], [13], [14], [15], "energy-aware routing" [9], [12] and "scheduling" [20].

The common factor of all the aforementioned protocols and algorithms is the assumption that nodes are equipped with omni-directional antennas. That is, all nodes have a 360^o degree coverage angle and do not need to point/aim at each other, in order to communicate. The advantage of this approach is its simplicity. However, a lot of energy is wasted this way, since the power is broadcasted towards all directions and therefore attenuates rapidly with distance. It may be therefore advantageous to use directional antennas instead.

The use of directional antennas in the context of ad-hoc networks has not been widely explored. Some recent papers [1], [2] suggest the use of multiple directional antennas per node (or multiple beam antennas), in order to increase the throughput of 802.11 media access control protocol [4]. In [17] the author explores the use of *beamforming* antennas in order to improve both throughput and delay in ad-hoc networks. Another paper [3] has suggested the use of multiple directional antennas to reduce the routing overhead of on-demand routing protocols for ad-hoc networks like DSR[5] and AODV[6]. No previous work, however, has identified the energy efficiency of the use of directional antennas or suggested energy efficient communication protocols that can be used with directional antennas. Finally, there are technology related issues that make the deployment of sophisticated multi-beam directional antennas on ad-hoc network nodes seem unrealistic for the time being [7]. We claim, therefore, that the right assumption to make is that of single-beam directional antennas.

In this paper, we propose the use of directional antennas for communications in ad-hoc networks. We argue that the potential energy savings that come from the use of directional antennas can be significant. In order to take advantage of these savings, an algorithm is needed though, that will synchronize potential senders and receivers. Therefore, we propose a four-step algorithm that coordinates node

---

[1] This research is supported by the DARPA Contract F33615-C-00-1633 in the Power Aware Computing and Communications Program

communications efficiently. In addition to enabling communication among nodes using directional antennas, our algorithm tries to further optimize for total energy consumption and network lifetime.

The approach we take is to do energy efficient routing first, in order to find minimum energy paths and then schedule nodes' transmissions, accordingly. The schedule produced is based on end-to-end traffic information and the routing decisions made earlier. Our goal during the scheduling step is to minimize the amount of time it takes to enable all transmitter-receiver pairs to communicate. Finally, we suggest an efficient way to make this algorithm distributed and adaptive/dynamic. The energy savings achieved consist of two major components. The first component, which is the result of using directional antennas instead of omni-directional ones, is proportional to the antenna gain. The second component, which is the result of using energy-efficient routing ranges from 10% to 45%.

The remainder of this paper is organized as follows. In the next section we discuss some implementation issues about directional antennas, we formulate the problem and outline our algorithm. In section *III* we give an in depth description of the algorithm, its individual steps and all techniques used. Section *IV* presents the results of our simulation where we demonstrate the energy efficiency of our algorithm. Finally, section *V* concludes the paper and outlines our future research.

II. ANALYSIS

A. Directional Antennas

The power savings of a directional antenna over an omni-directional depend on how narrow the primary beam/lobe is and also how suppressed the secondary lobes are compared to the primary one [7]. We'll use the simplifying assumption that the power radiated in secondary lobes is negligible and that all power is radiated through the (single) primary lobe. Furthermore, we assume that the antenna efficiency is 100%, so that all power fed into the antenna by the power amplifier is effectively converted into radiated power. In this simple abstract model the power savings are captured by the *antenna gain*, which is given by

$$\frac{4\pi}{\theta * \varphi} \tag{1}$$

where θ and φ are elevation and azimuth angles in radians, respectively. If both the transmitter and receiver use directional antennas to communicate, then the total savings will be equal to *Gain(Tx) *Gain(Rx)*, where both transmitter and receiver gains (reciprocity theorem) are given by (1). This is not true in general, especially in the case of simple directional antennas (e.g. 3-4 element arrays). The secondary lobes can reduce the actual gain of the antenna. In addition, interference from secondary lobes and an antenna efficiency of less than 100% (real case) result in a reduction of the radiated power in the direction of the primary lobe (i.e. reduced EIRP – Effective Isotropically Radiated Power), which implies reduced power savings, as well. However, in this paper we're not interested in the exact power gain stemming from the use of a specific directional antenna, nor are we interested in comparing different types of directional antennas in terms of potential power/energy savings. Our understanding is that the use of directional antennas allows nodes to communicate using less power than omni-directional ones, even if we elaborate our model including power losses in secondary lobes, antenna efficiency, etc. Furthermore, continuing advances in directional and smart antenna technology will keep making this potential power savings even higher. The focus of this work instead is how to exploit this high potential of directional antennas for energy savings and convert this potential into actual energy savings.

Although the above gain equation implies extremely high gains for narrow antenna beams, there are limitations on the actual gain that one can achieve for a mobile/wireless node antenna. The size of the terminal is the major restricting factor. The antenna size needs to be equivalent to the wavelength used, in order for it to radiate power efficiently. Furthermore, if more than one antenna elements (e.g. dipoles, patch antennas, etc.) are used to create diversity effects or to increase gain, those elements must be placed apart at distances of the same order of magnitude with the wavelength λ (e.g. λ, $\lambda/4$, etc.). Hence, depending on the size of the terminal (i.e. sensor, PDA, laptop, vehicle, etc.), one cannot easily use more than 3-4 elements for the frequency band currently used for ad-hoc networks (i.e. 2.4GHz). The gain for a 4-element phased-array is around 6-10dBi (depending on the type of the array), which gives a total of 12-20dBi for the transmitter-receiver pair[2].

We use a first order radio model which is similar to the one discussed in [23]. Here the radios are assumed to have power control and can expend the minimum required energy to reach intended recipients. The energy to transmit and receive a bit of information is given by:

$$E_{Tx} = E_{elec} + E_{amp} * d^{a}$$
$$E_{Rx} = E_{elec} \tag{2}$$

E_{elec} (Joules/bit) is the energy consumed in the electronics part of the transmitter (receiver) in order to transmit (receive) a bit of information. E_{amp} (Joules/bit/m$^{\alpha}$) is the energy consumed in the power amplifier part of the transmitter per bit. We can see that the transmission energy depends on the distance between the transmitter and receiver (*a* is the attenuation factor of the environment and can be between 2 and 4 for outdoor applications). This is exactly were the directional *antenna gain* comes into play. E_{amp} is proportional to the transmission power needed to reach the

---

[2] Antenna gain is usually measured in dBi, which is the amount of dB improvement over an (ideal) isotropic antenna that radiates its power uniformly in every direction (i.e. for all θ and φ) and considered as having gain equal to 1.

intended recipient, which in turn is inversely proportional to *Gain(Tx) *Gain(Rx)* [7]. The reduction in overall transmission energy E_{Tx} for a given reduction in power amplifier energy E_{amp}, depends on the specific radio parameters E_{elec} and E_{amp}, distance d and attenuation factor a. However, if nodes are not close by and/or environmental conditions are harsh (high a) the power amplifier energy consumption part is expected to dominate. Finally, continuing advances in low-power electronics are expected to bring E_{elec} further down, unlike E_{amp} which is lower bounded by the power needed to reach a specific node.

In addition to the high achievable gains, there are some more benefits in using directional antennas. The concentration of the radiated power into a narrow beam can significantly reduce the probability of detection, which can be crucial for certain applications (e.g. military applications). Furthermore, the resulting interference is much lower than that for the case of omni-directional antennas. Therefore, network capacity is increased. All these benefits do not come for free, however. Unlike in the case of omni-directional antennas, the transmitter and receiver antenna now have to aim at each other first, before they can start communicating. Consequently, there is a need for a protocol that synchronizes potential transmitters and receivers to make sure that every node has a chance to send a packet to every other node in the network.

Finally, we note that high-gain aperture and horn antennas (commonly used in satellite and terrestrial microwave communications) are not appropriate for ad-hoc networks. They need to be mechanically steered in order to point to a specific direction. This mechanical rotation (and any other kind of mechanical movement, as a matter of fact) consumes large amount of energy and if the antenna has to be redirected frequently (which is the case here) it can monopolize the energy consumption of the wireless terminal. For this reason, we choose to use electronically steerable directional antennas. The energy it takes to electronically steer the beam of directional antenna (e.g. array), is much less than that of mechanically steerable antennas, though not totally negligible. Furthermore, there is a delay overhead (i.e. *slack*) in redirecting the antenna beam, which is implementation dependent. Therefore, we argue that it is better to keep the antenna pointing to the same direction as long as possible, instead of changing direction continuously (e.g. for each packet to be transmitted). This is actually our main motivation to make routing and scheduling decisions on a larger time scale basis, instead of a per-packet basis. We'll use known solutions to synchronize and steer directional antennas [22].

B. *Problem Formulation*

Consider an ad hoc network with N wireless nodes, each of which is equipped with a single directional antenna. Each directional antenna is modeled as having a single (primary) radiation lobe, as described in section II.A. The elevation and azimuth angles of the primary lobe are θ and φ radians, respectively. Hence, the antenna gain is given by (1). Antenna beams are electronically steerable and can be pointed to any direction within the 360° azimuth plane. We denote the distance between nodes i and j as d_{ij} and we assume that the signal power attenuates linearly with $(d_{ij})^a$ where a is between 2 (i.e. free space) and 4 (e.g. urban environment). We assume that each node cannot reach every other node in one hop and therefore requires routing through intermediate nodes.

We assume that the amount of traffic generated per time unit by each node i destined for any other node j in the network can be modeled as a stochastic process (e.g. Poisson process). We denote the average arrival rate of traffic at node i destined for node j as f_{ij}. We define the end-to-end average flow matrix $F = \{f_{ij}\}$, as the matrix whose entry at row i and column j is the end-to-end flow f_{ij}. The end-to-end flow elements f_{ij} could be described either in terms of bytes/bits per time unit (i.e. fluid model) or in terms of packets per time unit. Our choice of representation does not affect our analysis and same results apply in either case. Throughout this paper we will assume that traffic flow is expressed in *packets per time unit* and that packets are of fixed size, without any loss of generality. Finally, we assume that the arrival process is stationary and therefore F can be considered constant for the sake of routing and scheduling. Later in the paper we will relax the stationarity assumption and allow the arrival process to be slowly varying in time.

Our goal is to successfully route the generated amount of traffic from each node to every other node using the minimum amount of energy. Assuming that all links can exist simultaneously and that nodes have the capability to route traffic to a specific destination through two or more paths (i.e. traffic balancing), then the centralized version of our problem becomes an *optimal routing* problem [8]. Instead of trying to optimize for average packet delay, we optimize for total energy consumption. In this paper we choose to drop the latter assumption for simplicity. Hence, we assume that all traffic from a specific source to a specific destination follows the same path. Furthermore, after routing has finished, we need to drop the first assumption and take into consideration the connectivity constraints that stem from the use of directional antennas. Finally, during the final phase of our algorithm we try to optimize for end-to-end packet delay for the routing configuration produced.

C. *Algorithm*

In this section we outline our proposed algorithm. It consists of 4 major steps:

1. **Shortest Cost Routing:** In order to find shortest cost paths, we will use the topology generated by considering all the possible links that can exist from each node to its neighbors by pointing the directional antenna into different directions. Clearly, the directional antenna cannot be pointed at multiple neighbors at the same time, but we can consider all the links to identify all possible routing paths. The use of directional antennas reduces interference in general and makes the problems of the hidden terminal and the exposed terminal [24] less

severe. However, there are still some ill situations where different transmitter-receiver pairs can interfere with each other. Consider for example the simple scenario of nodes *{T1, T2, R2, R1}* situated in a straight line with *T1* being the leftmost one and *R1* the rightmost one. Clearly, a transmission from *T1* to *R1* would interfere with a transmission from *T2* to *R2* and we shouldn't consider both *T1-R1* and *T2-R2* links as being able to be up simultaneously for routing purposes. Consequently, situations can arise where two nodes cannot be considered as having a common link, although being within reachable distance. We assume that lower level protocols take care of this kind of problems and make sure that all links considered in the routing step cannot (directly) interfere with each other. We use two different metrics in order to relate link/node cost with energy consumption.

2. **Link flow matrix calculation:** We define the link flow matrix $F' = \{f'_{ij}\}$ as the matrix whose entry at row *i* and column *j* is the traffic flow on the link connecting node *i* to node *j*. If there's no flow on link *i-j* or nodes *i* and *j* are not connected then $f'_{ij} = 0$. In this second step we calculate *F'* from *F,* using the routing information (i.e. routing tables) produced in Step 1.
3. **Topology update:** In this step we drop the assumption that the node antenna can point to different directions at the same time. Therefore, only one link can be up for each node at a time. Using this model and the link flow matrix *F'* calculated in step 2, we examine if the topology configuration used in step 1 can serve the individual link flows calculated in step 2. If the resulting link capacities are higher than the respective offered traffic for all links then we calculate the amount of time each link can be up and proceed to step 4. Otherwise, we use a heuristic to reconfigure the topology into a new one that has better potential to handle the offered load and go back to step 1.
4. **Scheduling:** At this final step, we already have the amount of time each individual link can stay up per time unit (i.e. per round). Our goal is to minimize the duration of the round while serving every individual link for the amount of time that was specified during step 3. This is a version of the general *scheduling problem*. Scheduling problems are usually modeled and solved using graph theoretic techniques. We formulate and solve this scheduling problem using a series of *maximum weighted matchings*.

III. ALGORITHM & PROTOCOLS

A. *Shortest Cost Routing*

The *Shortest Cost Routing* algorithm is a general routing algorithm. Some of its sub-cases are very well-known and widely used in routing algorithms (e.g. shortest path routing and shortest delay routing as in OSPF). There are several algorithms that calculate shortest cost paths to every node from a specific source node. We use *Dijkstra's* algorithm to generate shortest cost paths for each node.

Our primary concern is the energy-efficiency of the routing paths chosen. Therefore, we need to define appropriate metrics and assign link costs in such a way that it will result in the routing algorithm choosing paths that will be optimal in terms of energy consumption (for the metrics chosen). A discussion on different *energy-aware* metrics and their appropriateness in ad hoc networks can be found in [9]. Each of these metrics captures a slightly different notion of energy-efficiency and is application specific as to which one is the most appropriate. Furthermore, some of these metrics are less straightforward than others to be incorporated into routing algorithms.

In this paper we choose to use only two of the metrics proposed in [9]. These metrics are: *minimize energy consumed per packet* (metric 1 in [9]) and *maximize network lifetime* (metric 4 in [9]). In this paper we will refer to them as metric 1 and metric 2, respectively. We believe they can adequately represent the majority of the cases and can easily be implemented into our routing algorithm. Here is a short description of the two metrics:

1. **Minimize energy consumed per packet:** This is an obvious metric that reflects our intuition about energy conservation. Assume that some packet *j* traverses the path $n_1,\ldots,n_k$ where n_1 is the source and n_k is the destination Let *E(a,b)* denote the energy consumed in transmitting (and receiving) a packet over link *a-b* , where *a* and *b* are neighboring nodes. *E(a,b)* will depend, in this case, on the distance separating node *a* and node *b*. Then the energy consumed for packet *j* is,

   $$e_j = \sum_{i=1}^{k-1} E(n_i, n_{i+1}) \tag{3}$$

   The goal is to minimize e_j, $\forall$ packet *j*
   We implement this metric by assigning each link *a-b* a cost equal (or proportional) to *E(a,b)*. This way, the shortest cost paths produced by the routing algorithm will be the *minimum energy per packet* paths.
2. **Maximize network lifetime:** The goal of this metric is to avoid routing traffic through nodes with depleted energy. Consequently, the time until the first, second,…, final node dies out will be maximized and so will the network lifetime. Each node *i* is assigned a cost/weight w_i which is a function of the remaining energy of the node. The total cost of sending a packet *j* through the path $n_1,\ldots,n_k$ is,

   $$c_j = \sum_{i=1}^{k} w_i \tag{4}$$

   The goal of this metric is to minimize cj, $\forall$ packet j and this way maximize network lifetime. The remaining energy of the node, that is the battery's remaining lifetime, can be directly derived from the output voltage of the battery. In [9] different function costs are suggested based on different battery discharge functions.

B. Flow Matrix Calculation / Topology Update – Modification

Let i denote a source node and j a destination node. The average rate of traffic generated per time unit at node i destined for node j is given by f_{ij}, as mentioned earlier. The *time unit* can be any specific amount of time. It could be chosen so that it simplifies calculations (e.g. 1 second or the time it takes to transmit a packet). Alternatively, it can be the maximum amount of time T_{max} during which flow matrix F does not change significantly and can be therefore considered constant. Let TC_{ij} denote the amount of time flow f_{ij} can be considered constant. Then,

$$T_{\max} = \min_{i,j \in N} \{TC_{ij}\} \qquad (5)$$

Let SP_{kl} denote the set of links over which traffic from node k to node l is routed. Then the link flow matrix elements f'_{ij}, which represent the total number of packets that are routed through link i-j per time unit, are calculated as follows:

$$f'_{ij} = \sum_{k,l=1}^{N} f_{kl} * B_{ij}(k,l), \qquad (6)$$

where $B_{ij}(k,l)$ is a binary function,

$$B_{ij}(k,l) = \begin{cases} 1 & \text{, if link } i-j \in SP_{kl} \\ 0 & \text{, otherwise} \end{cases} \qquad (7)$$

We assumed before that links to different neighbors can be up simultaneously, only in order to take into account all candidate routing paths. However, we now have to drop this assumption since in reality the antenna of the node can only point to one direction at a time. Hence, the *time unit* or *cycle* we defined earlier has to be shared among all possible links for each node. For example, assume node i has two neighbors a, b. Then, f'_{ia} and f'_{ib} are the packets sent per time unit from node i to node a and node b, respectively. Let t_{ia} and t_{ib}, denote the fraction of the time unit link i-a and link i-b should be up, respectively. Then,

$$t_{ia} = \frac{f'_{ia}}{f'_{ia} + f'_{ib}}, \qquad t_{ib} = \frac{f'_{ib}}{f'_{ia} + f'_{ib}} \qquad (8)$$

Let's define,

$$\lambda_i = \sum_{j=1}^{N} f'_{ij} \text{ and } t_{ij} = \frac{f'_{ij}}{\lambda_i}, \ \forall \mathrm{i}, \mathrm{j}. \qquad (9)$$

Link i-j being up means that both the antenna of node i is pointing at node j and the antenna of node j is pointing at node i. Therefore, the maximum up time, say $T_{up}(i,j)$ for link i-j must be equal to the minimum of t_{ij} and t_{ji},

$$T_{up}(i,j) = min\{t_{ij}, t_{ji}\}, \qquad (10)$$

Equation (10) implies that the total up time of a node (i.e. fraction of time a node has more than zero links active) can be less the one. If we assume infinite link capacities at this step (i.e. capacities that are always high enough to handle the offered traffic), then we can safely proceed to the *scheduling* phase. However, if link capacities are restricted, there's a possibility that the fraction of time allocated to one(or more) link(s) is not long enough to serve all the traffic that goes through this(these) link(s). Consequently, the existing topology is not adequate and we need to come up with a different one that can handle the offered amount of traffic better. This is a typical *topological design* problem [8]. The general problem is very broad and may be difficult to formulate and thus solve. The usual approach is to try to solve the problem locally, by either increasing the capacity of the bottleneck link or by adding another link that will carry some of the excess load over a different path. In the case of ad-hoc networks, the capacity of links is usually fixed and dependent on the radio bandwidth. Hence, we can only try adding an additional link. Assuming that nodes are *power controlled* (i.e. can transmit K discrete different levels of power), a new link can be added if a node increases its power level until it discovers a new neighbor. An efficient neighbor discovery protocol in networks with nodes using directional antennas and power control is discussed in [18]. After the link is added we go back to step 1 and restart.

C. Scheduling

We have already converted the initial *connectivity* graph (i.e. graph whose edge weights represent transmission costs) into one where edge weights represent link *up-time* fractions as seen in Fig.1.

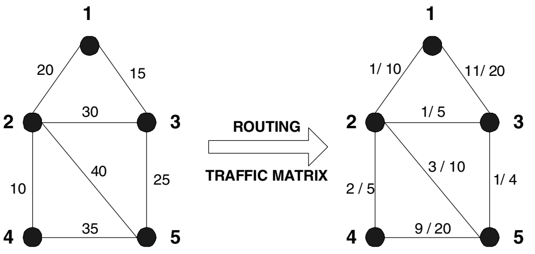

Fig. 1. Conversion of initial connectivity graph into a graph whose edges represent link *up-time* fractions. Edge weights represent transmission costs in the left graph and link "up" times in the right graph.

The final step is to schedule individual links in a way to minimize the total time it takes to "serve" all links. It is possible, and also desirable to have different sender-receiver pairs communicating in parallel, as long as no sender or receiver belongs to more than one pair.

Seeing this problem from a graph theory perspective, we need to choose sets of edges that have no vertices in common. If there weren't any weights on graph edges, then this would be an *edge-coloring* problem [10]. It is known [10], that the minimum number of colors needed is between d_{max} and d_{max+1}, where d_{max} is the maximum node degree. Hence, it would take at least d_{max} to d_{max+1} rounds to schedule all individual links. However, *edge-coloring* is not the optimal thing to do, in this case. Two distinct edge-colorings, both using the minimum number of colors (*edge-chromatic number* of the graph), could have a significant performance difference, as depicted in Fig.2.

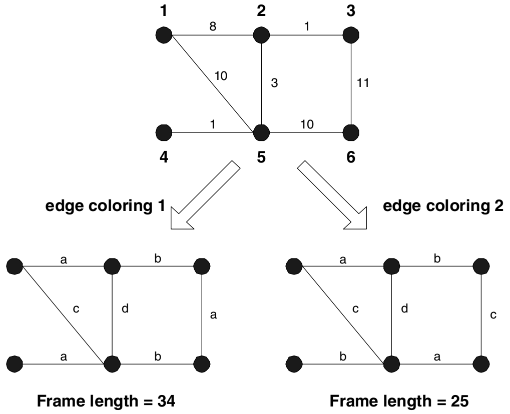

Fig. 2. Two possible edge colorings for an example graph. Edge weights represent number of time units a link has to be up. Frame length is measured in time units as well.

As is evident in Figure 2, we need to schedule links together that have equal or similar weights (i.e. *up-times*). Alternately, we need to choose a *maximum weight matching* [10]. After doing so, we can remove the links that were included in this matching from the graph and try to find another *maximum weight matching* for the *pruned* graph. We repeat this process until there are no edges left in the graph. This series of *maximum weight matchings* is highly efficient in scheduling links of similar weights together, whenever this is possible. There is a lower bound on how well we can do in terms of total frame duration. This lower bound is equal to the maximum of the sum of all edge weights having a vertex in common. Hence, as in the case of edge-coloring we can't do better than d_{max}, in this case we can't do better than,

$$T_{frame}(min) = \max_{i}\{ \sum_{j=1}^{N} w_{ij} \} \quad ,i \in (1,N) \tag{11}$$

For the graph depicted in Figure 2, we can see that this lower bound is 24 and the *bottleneck* node is node 5. The optimal frame length for this scenario is 25, which we do achieve using our *series of maximum weight matchings* scheme.

The duration of each *frame* (i.e. the time it takes to "serve" all links) depends on the total number of matchings necessary, and on the *up-time* of the links included in each matching. If we define this frame time as T_{frame}, the set of links in matching i as $S_m(i)$ and the number of matchings as M then,

$$T_{frame} = \sum_{i=1}^{M} \max_{a \in S_m(i)} \{T_{up}(a)\} \tag{12}$$

D. Initialization / Broadcast / Distributed Version

We have assumed up to this point that our algorithm is centralized and static. Consequently, the routing decisions and the resulting schedule is calculated in some central node based on static traffic information and is then distributed to all nodes in the network. However, our algorithm can be easily converted to a dynamic and distributed one.

We mentioned earlier that the end-to-end traffic flow matrix F is slowly varying in time. Hence, it can be considered constant over a certain time period T_{max}. This way, we know that the final schedule our algorithm produces will be good for at least T_{max}. If, however, we observe the system over a longer time period we'll see that F can change, sometimes significantly. Therefore, the existing schedule will not be optimal any more. Furthermore, it may not even be able to handle the offered amount of traffic. This means that our algorithm has to be rerun and a new schedule has to be produced for every *cycle* of duration T_{max}. Each node could dynamically keep track of the changing statistics (e.g. average arrival rate) of the traffic arrival process. If the traffic pattern is slowly varying then T_{max} will be much higher than the amount of time it takes to produce a new schedule, say T_{init}. Therefore, the overhead of periodically re-calculating the schedule will be insignificant and our algorithm can be adaptive. The time axis will consist of many long *normal operation* and short *schedule update* periods, interleaved as depicted in Fig.3.

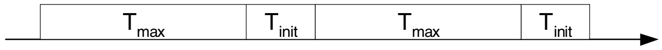

Fig. 3. Time axis consisting of *normal operation* and *schedule update* periods.

In order for the algorithm to be distributed, as well, we need a scheme to communicate the traffic flow information from each node to every other node (i.e. all-to-all communication). This way, all nodes will have the same version of F. If every node runs, subsequently, the same algorithm on the same F, then every node will obviously produce the same correct version of the schedule. We assume that the topology is known in advance (i.e. no mobility), or topology updates are made known using some link-state algorithm and the distributed version of Dijkstra's algorithm to calculate shortest cost paths. The communication of traffic flows from each node to every other node occurs during T_{init}. Each node i has a vector of values to broadcast, which we'll call traffic vector L_i. Each value represents the average amount of traffic node *i* generates for a specific destination. Then,

$$L_i = \{f_{i1}, f_{i2}, f_{i3}, \dots, f_{iN}\}$$

We need a broadcast algorithm that will,

$$\text{send } L_i \text{ to every other node } j \neq i, \ \forall i$$

A good way to perform this all-to-all communication of L_i*'s* is to choose one node in the network, say node *R,* and construct a binomial tree rooted on this node. Furthermore, we define two distinct phases, namely the *gather* phase and the *broadcast* phase. During the *gather* phase, each node *i* sends its L_i to node *R*, following the schedule computed for the binomial tree, towards the root. We assume that all nodes know the binomial tree and root node. It's out of the scope of this paper how this binomial tree is computed and how the root is chosen. We plan however, to explore this issue and define an efficient solution for it, in future work. After *R* has collected all L_i's, it goes the *broadcast* phase. During this phase node *R* broadcasts a packet containing all L_i's (including L_R) utilizing the same binomial tree. A different node is chosen for every *schedule update* cycle, in a round-robin fashion, in order to be the root of the tree. This way, the communication overhead of gathering and broadcasting back every L_i, is equally divided among all nodes. The reason why a binomial tree is the best choice for broadcasting using directional antennas can be seen in Fig. 4.

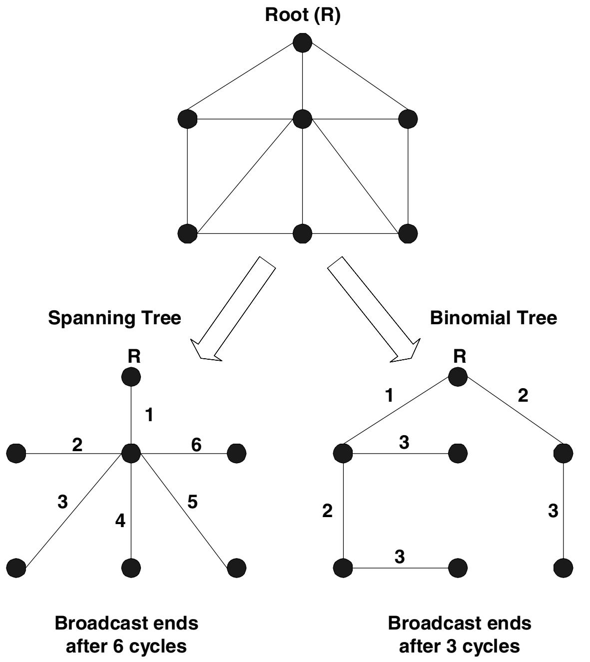

Fig. 4. Comparison of a binomial and a spanning tree used for broadcasting in an example ad hoc network consisting of nodes with directional antennas. Edge numbers indicate during which time cycle a link is up.

IV. SIMULATION RESULTS

For our simulations we generate random topologies consisting of 10-20 nodes. We make sure that each graph produced is connected. Furthermore, we can define the average *degree* of the vertices of the graph as an input parameter. The average vertex degree is related to the *connectivity* of the graph. Thus, if we choose the average vertex degree to be equal to *k*, then the graph will be *k-connected* (for the average case). Higher *k* means that there are more possible paths over which traffic can be routed. Therefore, a good routing algorithm will have a broader range of paths to choose from and is expected to perform better.

We choose to use network lifetime as the metric against which we will compare all different schemes. Specifically, we measure the time until the *1st node* in the network runs out of battery. It is topology dependent how many nodes in the network have to "die" and in what sequence, until the graph is not connected any more and the network is considered non-functional. It would be hard therefore, to compare different algorithms based on when the 2nd, 3rd, etc. die or when the network gets disconnected. Furthermore, as we explained in section *III.A*, the conclusions we would draw from such a comparison would be very similar to the ones drawn by considering only the 1st node that runs out of energy.

In Fig.5, we compare four different configurations:

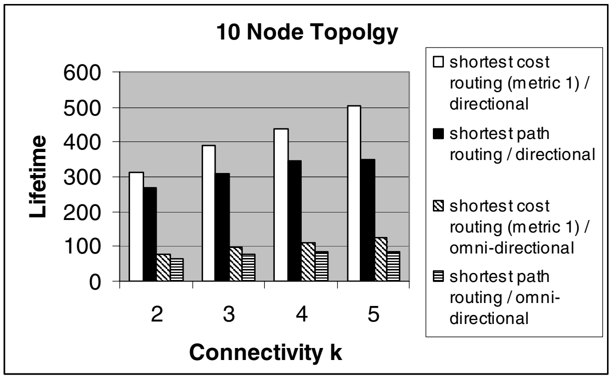

Fig. 5. Performance comparison (in terms of network lifetime) of four different schemes, applied to networks consisting of 10 nodes.

This way we can identify how much savings come from the use of directional antennas instead of omni-directional ones and how much come from using energy-efficient routing itself. Furthermore, for each configuration, we depict how *connectivity k* affects performance. We assume a directional antenna of modest gain (i.e. not too difficult to implement and incorporate in a wireless node). Specifically, we assume that both the transmitter and receiver antenna gain is equal to 2 (3 dB)[3]. Hence, the total *path gain* is equal to 4 (6 dB).

---

[3] In this case we compare the gain of a directional antenna with that of a half-wavelength dipole antenna. The half-wavelength dipole is commonly used in both cellular phones and ad hoc network terminals and is what we consider

Finally, we note that each result is the average over 100 random topologies.

We can clearly see, in Fig.5, the anticipated 4x improvement that comes from using directional antennas (i.e. configurations 1, 2), instead of omni-directional ones (i.e. configurations 3, 4), for both routing schemes. Furthermore, the improvement part that is the result of using *minimum-energy-per-packet routing* instead of *shortest path routing* is 10%-30%. Finally, it is worth noticing the behavior of the two routing schemes in relation with network connectivity. It is evident that *minimum-energy-per-packet routing* performs better with increasing *k*, unlike *shortest path routing*, which does not as effectively take advantage of the multiplicity of routing paths.

The influence of network size on the previous four schemes is depicted in Fig.6, where we present similar results for networks consisting of 20 nodes. Again, we run all algorithms for 100 random graphs and take the average. The reason network lifetime is reduced for all four cases, has to do with our end-to-end traffic generator. The more the nodes in the network, the higher the total amount of traffic each node generates. Consequently, more traffic is going over the network per time unit and energy is depleted more quickly. However, we are mainly interested on how network size affects the relative performance of the four different configurations. It is evident that the 4x improvement stemming from the use of directional antennas does not change significantly. However, the improvement of *minimum-energy-per-packet routing* over *shortest path routing* is less pronounced (i.e. 6%-25%). Finally, we can observe a similar behavior as for the 10-node topology in relation to network connectivity *k*.

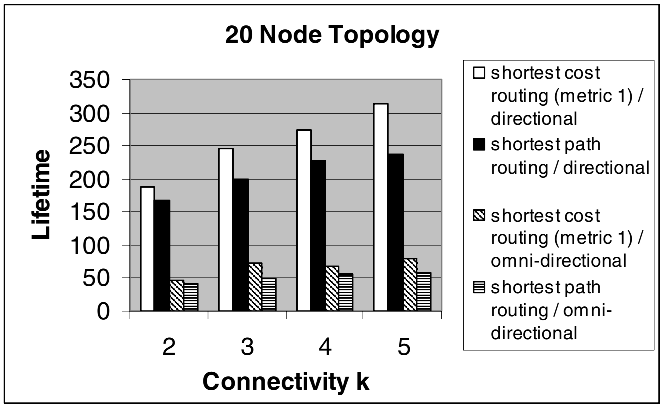

Fig. 6. Performance comparison (in terms of network lifetime) of four different schemes, applied to networks consisting of 20 nodes.

Finally, in Fig.7 we compare our two shortest cost routing algorithms for both the case of directional antennas and that of omni-directional ones. Once more, we observe the 4x improvement coming from the use of directional antennas. Additionally, it is evident that the 2^{nd} metric, which optimizes for network lifetime, does a better job in keeping network nodes alive. The improvement of the 2^{nd} metric over the 1^{st} one is about 7%-20%. Furthermore, we can see that the 2^{nd} metric takes better advantage of the variety of different paths that exist when connectivity is high. Finally, the energy savings of metric 2 over *shortest path routing* are between 15%-45%, for both directional and omni-directional antennas.

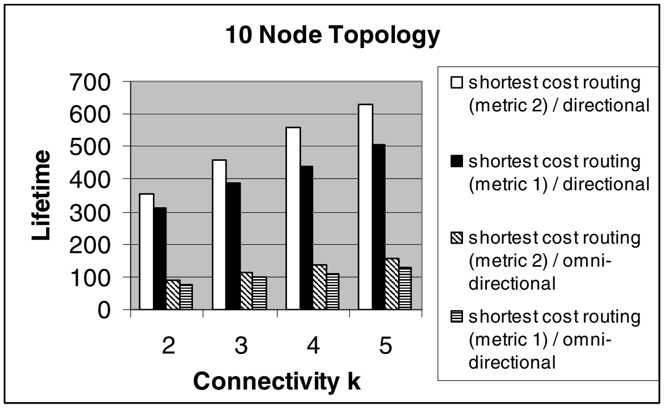

Fig. 7. Performance comparison (in terms of network lifetime) of our two metrics used for shortest cost routing, applied to networks consisting of 10 nodes.

V. CONCLUSIONS AND FUTURE RESEARCH

In this paper, we demonstrated the benefits of using directional antennas in ad hoc networks. We presented an energy-efficient algorithm for routing and scheduling in ad hoc network with nodes using directional antennas. We showed that using our algorithm we can decrease the total energy consumption and thus increase network lifetime by a factor, which is proportional to the antenna gain. In addition, simulation results demonstrate up to another 45% improvement in network lifetime that is achieved by using energy-aware routing, instead of conventional routing schemes (e.g. minimum hop routing).

We are currently working on other routing problems, including multicasting and broadcasting in ad hoc networks with directional antennas. In future work, we plan to explore variations of scheduling problems that arise in this context. Furthermore, we plan to incorporate our algorithm into ns-2 network simulator [21].

REFERENCES

[1] Y. B. Ko, V. Shankarkumar, and N. H. Vaidya, “Medium access control protocols using directional antennas in ad-hoc networks,” *Proc. IEEE INFOCOM’2000*, March 2000.

[2] A. Nasipuri, S. Ye, J. You, and R. E. Hiromoto, “A MAC protocol for mobile ad hoc networks using directional antennas,” *Proc. IEEE Wireless Communications and Networking Conference (WCNC’2000)*, 2000.

[3] A. Nasipuri, J. Mandava, H. Manchala, and R. E. Hiromoto, “On-demand routing using directional antennas in mobile ad hoc

---

omni-directional, because of its uniform radiation pattern on plane φ. However, the dipole antenna has a gain of 1.5 dBi itself. In this section therefore, when we say that an antenna has a gain of 2 (3 dB), for example, it means that the actual antenna gain is 4.5 dBi.

networks," *Proc. IEEE Computer Communications and Networks,* 2000.

[4] IEEE Local and Metropolitan Area Network Standards Committee, Wireless LAN medium access control (MAC) and physical layer (PHY) specifications, IEEE standard 802.11-1999, 1999.

[5] D. Johnson, D. Maltz, Yih-Chun Hu, and Jorjeta G. Jetcheva, "Dynamic source routing (dsr)," Internet Draft, draft-ietf-manet-dsr-06.txt, November 2001.

[6] C. Perkins, E. Belding-Royer, and S. Das, "Ad Hoc On Demand Distance Vector (AODV) Routing." *IETF Internet draft*, draft-ietf-manet-aodv-09.txt, November 2001 (Work in Progress).

[7] C. A. Balanis, *Antenna Theory: Analysis and Design*, 2nd ed. *New York: Wiley*, 1997.

[8] D. Bertsekas and R. Gallager, *Data Networks*, Englewood Cliffs, *NJ: Prentice Hall*, 2nd ed., 1992.

[9] S. Singh, M. Woo, and C. S. Raghavendra, "Power-Aware Routing in Mobile Ad-Hoc Networks," *Proc. ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM'1998)*, 1998.

[10] N. Christofedes, *Graph Theory: An Algorithmic Approach*, Academic Press Inc., 1975.

[11] S. Singh, and C. S. Raghavendra, "PAMAS: Power Aware Multi-Access protocol with Signalling for Ad Hoc Networks," ACM Computer Communications Review, 1999.

[12] Jae-Hwan Chang, and Leandros Tassiulas, "Energy conserving routing in wireless ad-hoc networks," *Proc IEEE INFOCOM '2000*, March 2000.

[13] Y. Xu, J. Heidemann, and D. Estrin, " Geography Informed Energy Conservation for Ad Hoc Routing," *In Proceedings of the Seventh ACM/IEEE International Conference on Mobile Computing (ACM MOBICOM) and Networking* Rome, Italy, July 16-21 2001.

[14] B. Chen, K. Jamieson, R. Morris, H. Balakrishnan, "Span: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks," *In Proceedings of the Seventh ACM/IEEE International Conference on Mobile Computing (ACM MOBICOM) and Networking* Rome, Italy, July 16-21 2001.

[15] R. Kravits, and P. Krishnan, "Application-Driven Power Management for Mobile Communication," *ACM/Baltzer Wireless Networks*, 1999.

[16] M. Zorzi, and R.R. Rao, "Energy constrained error control for wireless channels", *IEEE Personal Communications*, vol. 4,pp. 2733, Dec. 1997.

[17] R. Ramanathan, "On the Performance of Ad Hoc Networks with Beamforming Antennas," *In the Proceedings of ACM Symposium on Mobile Ad hoc Networking and Computing (MobiHoc'2001)*, 2001.

[18] T. Y. Liu, "Link Acquisition Protocols for a Mobile Communication Network with Directional/Adaptive Antennas," Ph.D Thesis, *Communication Science Institute (CSI) / Univeristy of Southern California,* May 1998.

[19] K. Nakano, S. Olariu, and A. Zomaya, "Energy-efficient deterministic routing protocols in radio networks," *Proc IEEE International Conference on Parallel Processing,* 2000.

[20] B. Prabhakar, E. U. Biyikoglu, and A. E. Gamal, "Energy-efficient transmission over a wireless link via lazy packet scheduling," *Proc IEEE INFOCOM'2000*, March 2000.

[21] http://www.isi.edu/nsnam/

[22] J. C. Liberti, Jr., and T. S. Rappaport, *Smart Antennas for Wireless Communication: IS-95 and Third Generation CDMA Applications,* Prentice Hall, 1999.

[23] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan. "Energy-Efficient Communication Protocols for Wireless Microsensor Networks," *In Proc. of Hawaiian Int'l Conf. on Systems Science*, January 2000.

[24] F.A. Tobagi, and L. Kleinrock, "Packet Switching in Radio Channels: Part II - The Hidden Terminal Problem in CSMA and Busy-Tone Solution," *IEEE Trans. on Communications COM-23*, pp. 1417-1433, December 1975.