Energy Conserving Routing in Wireless Ad-hoc Networks

Jae-Hwan Chang and Leandros Tassiulas
Department of Electrical and Computer Engineering & Institute for Systems Research
University of Maryland at College Park
College Park, MD 20742
{jhchang,leandros}@isr.umd.edu

Abstract— **An ad-hoc network of wireless static nodes is considered as it arises in a rapidly deployed, sensor based, monitoring system. Information is generated in certain nodes and needs to reach a set of designated gateway nodes.** Each node may adjust its power within a certain range that determines the set of possible one hop away neighbors. Traffic forwarding through multiple hops is employed when the intended destination is not within immediate reach. The nodes have limited initial amounts of energy that is consumed in different rates depending on the power level and the intended receiver. **We propose algorithms to select the routes and the corresponding power levels such that the time until the batteries of the nodes drain-out is maximized.** The algorithms are local and amenable to distributed implementation. When there is a **single power level,** the problem is reduced to a maximum flow problem with node capacities and the algorithms converge to the optimal solution. When there **are multiple power levels** then the achievable lifetime is close to the optimal (that is computed by linear programming) most of the time. **It turns out that in order to maximize the lifetime, the traffic should be routed such that the energy consumption is balanced among the nodes in proportion to their energy reserves, instead of routing to minimize the absolute consumed power.**

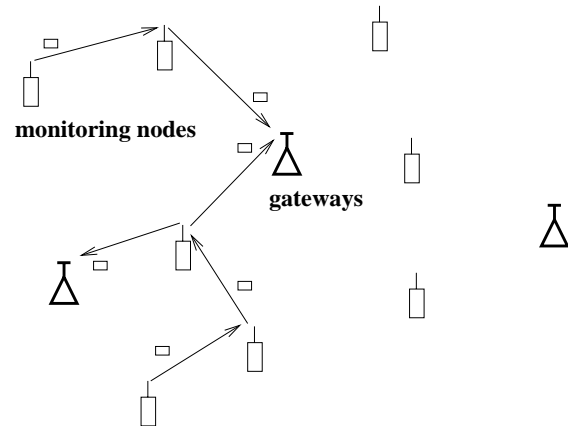*Keywords*— **energy-sensitive routing, wireless ad-hoc networks, sensor networks**

Fig. 1. A multi-hop wireless ad-hoc network is depicted where the nodes are randomly distributed and the information generated at the monitoring nodes are to be delivered to the gateway nodes.

I. INTRODUCTION

CONSIDER a group of wireless static nodes randomly distributed in a region as in Fig.1, where each node has a limited battery energy supply used mainly for the transmission of data. Assume that at each node some type of information is generated as it monitors the data such as sound or vibration in its vicinity using the sensor, and the information needs to be delivered to a set of gateway nodes. These wireless nodes are assumed to have the capability of packet forwarding, i.e., relaying an incoming packet to one of its neighboring nodes, and the transmitted energy level can be adjusted to a level appropriate for the receiver to be able to receive the data correctly if the receiver is within the transmission range. Upon or before a new arrival of information either generated at the node itself or forwarded from the other nodes, routing decision has to be made so that the node knows which of its neighboring nodes to forward its data to. Note that the routing decision and the transmission energy level selection are intrinsically connected in this power-controlled ad-hoc network since the power level will be adjusted depending on the location of the next hop node.

An example scenario for this type of wireless ad-hoc network may include a wireless sensor network where the sensors gather acoustic, magnetic, or seismic information and send the information to its gateway node which has more processing power for further processing of the information or has larger transmission range for the delivery of the information to a possibly larger network for retrieval by a remote user.

Most of the previous works on routing in wireless ad-hoc networks deal with the problem of finding and maintaining correct routes to the destination during mobility and changing topology [1], [6], [11]. In [1], [6], the authors presented a simply implementable algorithm which guarantees strong connectivity and assumes limited node range. Shortest path algorithm is used in this strongly connected backbone network. However, the route may not be the minimum energy solution due to possible omission of the optimal links at the time of the backbone connection network calculation. In [11], the authors developed a dynamic routing algorithm for establishing and maintaining connection-oriented sessions which uses the idea of predictive re-routing to cope with the unpredictable topology changes. Some other routing algorithms in mobile wireless networks can be found in [15], [12], [9], [14], which, as the majority of routing protocols in mobile ad-hoc networks do, use shortest-path routing where the number of hops is the path length.

The problem of minimum energy routing has been addressed before in [1], [6], [16], [10], [8], [18], [17], and [7]. The approach in those works was to minimize the total consumed en-

ergy to reach the destination, which minimizes the energy consumed per unit flow or packet. If all the traffic is routed though through the minimum energy path to the destination the nodes in that path will be drain-out of batteries quickly while other nodes, which perhaps will be more power hungry if traffic is forwarded through them, will remain intact. Instead of trying to minimize the consumed energy, the performance objective of maximizing the lifetime of the system[3], which is equivalent to maximizing the time to network partition[18] has been considered. In [18], the problem of maximizing the time to network partition was reported as NP-complete. In [3] we identified the maximum lifetime problem as a linear programming problem. Therefore, it is solvable in polynomial time. The work in [3] considered the single destination version of the problem, while here we extend the problem to the multicommodity case, where each commodity has a its own set of destinations.

In our study the topology of the network is static and the routing accounts to finding the traffic splits that balance optimally the energy consumption. Hence the results are applicable to networks which are either static, like the sensor networks we mentioned earlier, or whose topology changes slowly enough such that there is enough time for optimally balancing the traffic in the periods between successive topology changes.

This paper is organized as follows: In section II, the problem is formulated. In section III, we propose a class of flow augmentation algorithms that use the shortest cost path. In section IV, we extend the flow redirection algorithm to cover the multicommodity case. In section V, random graphs are generated in order to evaluate the performances of these algorithms. Finally in section VI, some concluding remarks are made.

II. Routing for the Maximum System Lifetime

The wireless ad-hoc network in consideration is modeled as a directed graph $G(N, A)$ where N is the set of all nodes and A is the set of all directed links (i, j) where $i, j \in N$. Let S_i be the set of all nodes that can be reached by node i with a certain power level in its dynamic range. We assume that link (i, j) exists if and only if $j \in S_i$. Let each node i have the initial battery energy E_i, and let $Q_i^{(c)}$ be the rate at which information is generated at node i belonging to commodity $c \in C$, where C is the set of all commodities. Assume that the transmission energy required for node i to transmit an information unit to its neighboring node j is e_{ij}, and the rate at which information of commodity c is transmitted from node i to node j is called the flow $q_{ij}^{(c)}$. Further, let Q_i and q_{ij} be the aggregate flows of all commodities, i.e.,

$$Q_i = \sum_{c \in C} Q_i^{(c)}, \qquad (1)$$

and

$$q_{ij} = \sum_{c \in C} q_{ij}^{(c)}. \qquad (2)$$

We are given, for each commodity c, a set of origin nodes $O^{(c)}$ where the information is generated, i.e.,

$$O^{(c)} = \{ i \mid Q_i^{(c)} > 0, i \in N \}, \qquad (3)$$

and a set of destination nodes $D^{(c)}$ among which any node can be reached in order for the information transfer of commodity c be considered done.

The lifetime of node i under a given flow $\mathbf{q} = \{q_{ij}\}$ is given by

$$T_i(\mathbf{q}) = \frac{E_i}{\sum_{j \in S_i} e_{ij} \sum_{c \in C} q_{ij}^{(c)}}. \qquad (4)$$

Now, let us define the *system lifetime* under flow $\mathbf{q}$ as the length of time until the first battery drain-out among all nodes in N, which is the same as the minimum lifetime over all nodes, i.e.,

$$
\begin{aligned}
T_{sys}(\mathbf{q}) &= \min_{i \in N} T_i(\mathbf{q}) \\
&= \min_{i \in N} \frac{E_i}{\sum_{j \in S_i} e_{ij} \sum_{c \in C} q_{ij}^{(c)}}.
\end{aligned} \qquad (5)
$$

Our goal is to find the flow that maximizes the system lifetime under the flow conservation condition. The problem can be written as follows:

$$\text{Maximize} \quad T_{sys}(\mathbf{q}) = \min_{i \in N} \frac{E_i}{\sum_{j \in S_i} e_{ij} \sum_{c \in C} q_{ij}^{(c)}}$$

$$\text{s.t.} \quad q_{ij}^{(c)} \geq 0, \qquad \forall i \in N, \forall j \in S_i, \forall c \in C,$$

$$\sum_{j:\, i \in S_j} q_{ji}^{(c)} + Q_i^{(c)} = \sum_{k \in S_i} q_{ik}^{(c)}, \quad \forall i \in N - D^{(c)}, \forall c \in C. \qquad (6)$$

Fig.2 illustrates the flow conservation condition for commodity c at node i, and it should be noted that the condition applies to each commodity separately.

In the following we show that the problem is a linear programming problem[13]. The problem of maximizing the system lifetime, given the information generation rates $Q_i^{(c)}$ at the set of origin nodes $O^{(c)}$ and the set of destination nodes $D^{(c)}$ for each commodity c, is equivalent to the following linear programming problem:

$$\text{Maximize} \quad T \qquad (7)$$

$$\text{s.t.} \quad \hat{q}_{ij}^{(c)} \geq 0, \qquad \forall i \in N, \forall j \in S_i, \forall c \in C, \quad (8)$$

$$\sum_{j \in S_i} e_{ij} \sum_{c \in C} \hat{q}_{ij}^{(c)} \leq E_i, \qquad \forall i \in N, \qquad (9)$$

$$\sum_{j:\, i \in S_j} \hat{q}_{ji}^{(c)} + T Q_i^{(c)} = \sum_{k \in S_i} \hat{q}_{ik}^{(c)}, \quad \forall i \in N{-}D^{(c)}, \forall c \in C, (10)$$

where $\hat{q}_{ij}^{(c)} = T q_{ij}^{(c)}$ is the amount of information of commodity c transmitted from node i to node j until time T.
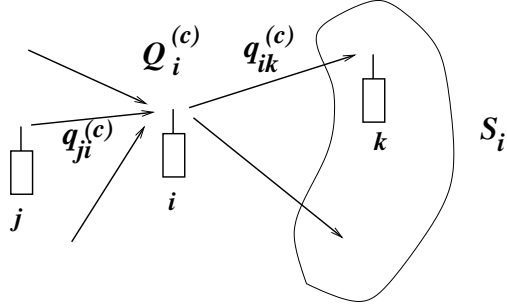
Fig. 2. The conservation of flow condition at node i for each commodity c requires that the sum of information generation rate and the total incoming flow must equal the total outgoing flow.

The linear program given above can be viewed as a variation of the conventional maximum flow problem with node capacities[5]. If the transmitted power level at each node is fixed regardless of its next hop node, i.e., if there is no power control,

$$e_{ij} = e_i, \ \ \forall j \in S_i, \tag{11}$$

and the problem is equivalent to the maximum flow problem with node capacities given by

$$\sum_{j \in S_i} \sum_{c \in C} \hat{q}_{ij}^{(c)} \le E_i / e_i, \ \ \forall i \in N. \tag{12}$$

When the capacity of a a node is a fixed quantity as in (12) then the problem can be converted to a link capacity version by replacing the node with two nodes and a link having the same capacity[4], and the max-flow-min-cut theorem[5] can be used. However, in our problem, unlike the above, the amount of resource (or energy in this case) which a unit flow consumes depends on the energy expenditure to the next hop node. Therefore, it is not trivial to find the min-cut nodes, and even if they were found the traffic split at the nodes must also be identified.

III. Flow Augmentation Algorithms

In this section, we propose a class of flow augmentation (FA) algorithms which use the shortest cost path.

The general description of the algorithm is given in the following. At each iteration, each origin node $o \in O^{(c)}$ of commodity c calculates the shortest cost path to its destination nodes in $D^{(c)}$. Then the flow is augmented by an amount of $\lambda Q_i^{(c)}$ on the shortest cost path, where λ is the augmentation step size. After the flow augmentation, the shortest cost paths are recalculated and the procedures are repeated until any node $i \in N$ runs out of its initial total energy E_i. As a result of the algorithm, we obtain the flow which will be used at each node to properly split incoming traffic.

Our objective is to find the best link cost function which will lead to the maximization of the system lifetime. There are three parameters to consider in calculating the link cost c_{ij} for link (i, j). One is the energy expenditure for unit flow transmission over the link, e_{ij}, the second is the initial energy E_i, and the third is the residual energy at the transmitting node i which is denoted by $\underline{E}_i$. A good candidate for the flow augmenting path should consume less energy and should avoid nodes with small residual energy since we would like to maximize the minimum lifetime of all nodes. In [18], each of these were separately considered, which falls short of optimizing the system lifetime. Obviously, both of these can't be optimized at the same time, which means there is a tradeoff between the two. In the beginning when all the nodes have plenty of energy, the minimum total consumed energy path is better off, whereas towards the end avoiding the small residual energy node becomes more important. Therefore, the link cost function should be such that when the nodes have plenty of residual energy, the energy expenditure term is emphasized, while if the residual energy of a node becomes small the residual energy term should be more emphasized.

With the above in mind, the link cost c_{ij} is proposed to be

$$c_{ij} = e_{ij}^{x_1} \underline{E}_i^{-x_2} E_i^{x_3}, \tag{13}$$

where x_1, x_2, and x_3 are nonnegative weighting factors for each item. Note that if $\{x_1, x_2, x_3\} = \{0, 0, 0\}$ then the shortest cost path is the minimum hop path, and if it is $\{1, 0, 0\}$ then the shortest cost path is the minimum transmitted energy path. If $x_2 = x_3$ then normalized residual energy is used, while if $x_3 = 0$ then the absolute residual energy is used. Let's refer to the algorithm as FA(x_1, x_2, x_3) in the rest of the paper indicating the parameters, and the meanings of the parameters are summarized in Table I for reference.

The path cost is computed by the summation of the link costs on the path, and the algorithm can be implemented with any existing shortest path algorithms including the distributed Bellman-Ford algorithm[2], which will be used in our simulation.

IV. Flow Redirection Algorithm

In this section, we extend the flow redirection (FR) algorithm[3] to the multicommodity case.

This algorithm is based on the following observation. If we have a single origin and a single destination or if we have multiple origins and destinations but without any constraints on the information generation rates, then under the optimal flow, the minimum lifetime of every path from the origin to the destination with positive flow is the same. Note that the latter case can be converted to a single origin and a single destination version by adding a super origin and a super destination connected to the origins and the destinations respectively with zero energy expenditure links.

The above fact can be shown as follows. Assume that the flow is optimal, i.e., minimum lifetime over all nodes is maximized. If we further assume that the minimum lifetimes of the paths with positive flow to the destination are not all identical then there is a set of path(s) with positive flow whose minimum lifetime is the shortest. We can always increase the minimum

$FA(x_1, x_2, x_3)$	Meaning
$FA(0, 0, 0)$	Minimum hop path
$FA(1, 0, 0)$	Minimum transmitted energy path
$FA(\cdot, x, x)$	Normalized residual energy is used
$FA(\cdot, \cdot, 0)$	Absolute residual energy is used

lifetime of this set of path(s), which is also the system lifetime, by redirecting an arbitrarily small amount of flow to the paths whose lifetime is longer than these paths such that the minimum lifetime of the latter path after the redirection is still longer than the system lifetime before the redirection. This contradicts our assumption that the flow is optimal.

In this algorithm we redirect a portion of each commodity flow at every node in a way that the minimum lifetime of every path with positive flow from the node to the destination will increase or at least will stay the same.

In the following, we describe the implementation of FR. Let's use an imaginary super destination node $\tilde{d}^{(c)}$ where $\tilde{d}^{(c)} \in S_d$ and $e_{d\tilde{d}^{(c)}} = 0$ for all $d \in D^{(c)}$. Let the initial flow be such that from $o \in O^{(c)}$ to $\tilde{d}^{(c)}$ the minimum total transmitted energy path is used with a flow value of $Q_o^{(c)}$. Note that any path to the destination can be used as the initial flow. Each node $i \in N - D^{(c)}$ redirects its outgoing flow of commodity c by subtracting $\epsilon_i^{(c)}$ from the flow of a certain path to $\tilde{d}^{(c)}$ and by adding the same amount to the flow of another path to $\tilde{d}^{(c)}$. It is possible that the flow can be re-routed to a different destination node in $D^{(c)}$. The steps to be taken at each node $i \in N - D^{(c)}$ for each commodity c are as follows:

1. (Determine the Two Paths) Determine the two paths to the destination which are to be involved in the redirection.
2. (Calculate $\epsilon_i^{(c)}$) Calculate the amount $\epsilon_i^{(c)}$ of redirection.
3. (Redirect the Flow) Properly increment and decrement the flows of the two paths determined above by an amount of $\epsilon_i^{(c)}$.

The first step of the algorithm at each node i for commodity c, (Determine the Two Paths), is described in more detail. The goal of this step is to identify the ascent direction.

We will need two different path calculations for each commodity. Let's first form a subnetwork $G_F^{(c)}(N, A_F^{(c)})$ of $G(N, A)$ where $A_F^{(c)} \subset A$ consists only of edges with positive flow, i.e.,

$$A_F^{(c)} = \{(i, j) | q_{ij}^{(c)} > 0, (i, j) \in A\}. \tag{14}$$

Let $P_i^{(c)}$ be the set of all paths in $G_F^{(c)}(N, A_F^{(c)})$ from node i to any of the destination nodes in $D^{(c)}$. For a path $p \in P_i^{(c)}$, define its path length $L_p(\mathbf{q})$ under flow $\mathbf{q}$ as a vector whose

elements are the lifetimes of all the nodes in the path before reaching any of its destination nodes $D^{(c)}$. For example, if path $p \in P_i^{(c)}$ starting from node i traverses nodes $j_1, j_2, \cdots, j_m$ before reaching any node in $D^{(c)}$, then $L_p(\mathbf{q}) = [T_i(\mathbf{q}) \quad T_{j_1}(\mathbf{q}) \quad T_{j_2}(\mathbf{q}) \quad \cdots \quad T_{j_m}(\mathbf{q})]$. The length of path p, $L_p(\mathbf{q})$, is said to be shorter (longer) than the length of path p', $L_{p'}(\mathbf{q})$, if the smallest element of $L_p(\mathbf{q})$ is smaller (larger) than that of $L_{p'}(\mathbf{q})$. We compare the smallest element first since it is the minimum lifetime of all nodes in the path. In case they are the same, the next smallest elements are compared, and so on. If there are more than one smallest elements with the same value then each one is compared separately. Using this so-called lexicographical ordering, the shortest length path from any node to the destination is defined. We modify the distance comparison part of the Bellman-Ford algorithm [2] to obtain the shortest length paths distributively. Let's denote the shortest length path in $G_F^{(c)}(N, A_F^{(c)})$ from node i to the destination node $\tilde{d}^{(c)}$ by $sp^{(c)}(i)$. Note that the shortest path, $sp^{(c)}(i)$, passes through the node which has the minimum lifetime of all downstream nodes of node i. The other path calculation is to find the longest length path in $G(N, A)$ using the same path length vector. If two path lengths are the same, choose the one with less number of elements in the path length vector. The longest length path is the path whose minimum lifetime is the longest. Let's denote the longest length path in $G(N, A)$ from node i to the destination node $\tilde{d}^{(c)}$ by $lp^{(c)}(i)$. Note that the longest length path is the path which, in some sense, has the largest capacity since we will need to assign more flows to this path than any other path in order to make the minimum lifetime of the path to equal the minimum lifetime of the other paths.

Let g denote the next hop node of node i from which path the flow will be subtracted, and let t denote the next hop node of node i to which path the flow will be added, where $g \in S_i$, the *giver*, and $t \in S_i$, the *taker*, are to be carefully chosen among the neighbors of node i. Note that notations such as $g_i^{(c)}$ and $t_i^{(c)}$ could be used, but we use g and t instead for simplicity since there is no ambiguity.

Depending on whether or not the lifetime of node i, $T_i(\mathbf{q})$, is the minimum in the shortest length path $sp^{(c)}(i)$, two different measures are taken.

If $T_i(\mathbf{q}) \leq \min[L_{sp^{(c)}(i)}(\mathbf{q})]$ then the lifetime of node i is the minimum over all nodes in the subnetwork consisting of node i and all its downstream nodes, and hence we would like to increase the lifetime of node i. This can be achieved if we redirect a flow at node i to the direction where the required transmission energy per information unit is smaller. In other words, if we choose a flow passing node g we redirect the flow to node t where $e_{it} < e_{ig}$. This can be done in many different ways. One such choice will be to redirect the flow whose energy expenditure to the next hop is the maximum to the direction of minimum energy expenditure to the next hop, i.e.,

$$g = \operatorname*{argmax}_{j \,:\, j \in S_i, \, q_{ij}^{(c)} > 0} e_{ij}, \tag{15}$$

and

$$t = \underset{j \,:\, j \in S_i}{\operatorname{argmin}} \ e_{ij}. \tag{16}$$

Another possibility for the taker is to choose the longest length path whose path length is the longest among all the next hop nodes that has smaller energy expenditure than e_{ig}, i.e.,

$$t = \underset{j \,:\, j \in S_i, \, e_{ij} < e_{ig}}{\operatorname{argMAX}} \ L_{lp^{(c)}(j)}(\mathbf{q}), \tag{17}$$

where MAX denotes the maximum in the lexicographical ordering. The giver doesn't have to be the node with the maximum energy expenditure. In fact we can choose any node with non-minimum energy expenditure. In our algorithm, all these possibilities are used alternately.

On the other hand, if $T_i(\mathbf{q}) > \min[L_{sp^{(c)}(i)}(\mathbf{q})]$ then we would like to increase the lifetime of the minimum lifetime node in the path $sp^{(c)}(i)$ by redirecting some of the flow to another path since the lifetime of that node is the minimum lifetime over all nodes in the subnetwork consisting of node i and its downstream nodes. The giver g is the next hop node of node i in the shortest length path $sp^{(c)}(i)$, and the taker can be either the node whose longest length path is the longest, i.e.,

$$t = \underset{j \in S_i}{\operatorname{argMAX}} \ L_{lp^{(c)}(j)}(\mathbf{q}), \tag{18}$$

or the node with the minimum energy expenditure whose longest length path is longer than the shortest length path of node i, i.e.,

$$t = \underset{j \,:\, j \in S_i, \min[L_{lp^{(c)}(j)}(\mathbf{q})] > \min[L_{sp^{(c)}(i)}(\mathbf{q})]}{\operatorname{argmin}} \ e_{ij}. \tag{19}$$

In fact, it suffices to find any node whose path length of the longest length path is longer than that of the shortest length path of node i. In our algorithm, all these possibilities are used alternately.

Given the two nodes g and t, the flow of the path composed of (i, g) and $sp^{(c)}(g)$ will be re-routed to the path composed of (i, t) and $lp^{(c)}(t)$.

The second step of the algorithm at each node i for commodity c, (Calculate $\epsilon_i^{(c)}$), is described in more detail. The aim of this step is to determine the amount of redirection that guarantees monotonic non-decrease of the system lifetime. The constraints that $\epsilon_i^{(c)}$ should meet are as follows. First, it should be less than or equal to the flow in the path of giver node, i.e.,

$$\epsilon_i \leq q_{ig}^{(c)}, \tag{20}$$

and

$$\epsilon_i \leq q_{jk}^{(c)}, \tag{21}$$

for each link (j, k) in the path $sp^{(c)}(g)$. Furthermore, none of the lifetimes should become shorter than the currently minimum lifetime of the subnetwork consisting of node i and all its downstream nodes since this will lead us to the opposite direction to that of our objective. If $T_i(\mathbf{q}) \leq \min[L_{sp^{(c)}(i)}(\mathbf{q})]$ then none

of the lifetimes in the path $lp^{(c)}(t)$ should become shorter than $T_i(\mathbf{q})$, i.e.,

$$\frac{1}{T_j(\mathbf{q})} + \frac{e_{jk}\epsilon_i^{(c)}}{E_j} \leq \frac{1}{T_i(\mathbf{q})}, \tag{22}$$

for each link (j, k) in the path $lp^{(c)}(t)$. On the other hand if $T_i(\mathbf{q}) > \min[L_{sp^{(c)}(i)}(\mathbf{q})]$ then we need to consider two things. First, none of the lifetimes in the path $lp^{(c)}(t)$ should become shorter than the minimum lifetime of the path $sp^{(c)}(i)$, i.e.,

$$\frac{1}{T_j(\mathbf{q})} + \frac{e_{jk}\epsilon_i^{(c)}}{E_j} \leq \frac{1}{\min[L_{sp^{(c)}(i)}(\mathbf{q})]}, \tag{23}$$

for each link (j, k) in the path $lp^{(c)}(t)$. Second, if $e_{it} > e_{ig}$ then the lifetime of node i may decrease due to the redirection, but it should not become shorter than the minimum lifetime of the path $sp^{(c)}(i)$, i.e.,

$$\frac{1}{T_i(\mathbf{q})} + \frac{(e_{it} - e_{ig})\epsilon_i^{(c)}}{E_i} \leq \frac{1}{\min[L_{sp^{(c)}(i)}(\mathbf{q})]}. \tag{24}$$

Finally, the value of $\epsilon_i^{(c)}$ should be chosen among the values that meet all the constraints stated above. We could either use the maximum $\epsilon_i^{(c)}$ that meets all the constraints or just a fraction of it. To avoid possible oscillations and for faster convergence, we choose the half of the maximum $\epsilon_i^{(c)}$ that meets all the constraints except (20) and (21).

The third step of the algorithm at each node i for commodity c, (Redirect the Flow), is described in more detail.

Subtracting $\epsilon_i^{(c)}$ from the path $sp^{(c)}(g)$ is simple. We made sure that $\epsilon_i^{(c)}$ is less than or equal to what is available in each edge in the path in (20) and (21).

Adding $\epsilon_i^{(c)}$ to the path $lp^{(c)}(t)$ is also simple, but there is a possibility that one or more loops of positive flow value can be formed. These loops should be removed in order to avoid unnecessary energy consumption and to ensure that the path indeed leads to the destination. After adding $\epsilon_i^{(c)}$ to the path $lp^{(c)}(t)$, the formation of one or possibly more loops is checked and the loop(s) are removed link by link along the path. For instance, if link (j, k) in the path $lp^{(c)}(t)$ is checked by calculating the shortest hop distance from node k to node j in the subnetwork $G_F^{(c)}(N, A_F^{(c)})$ of $G(N, A)$. If the distance is finite then at least one loop exists. Remove the loop flow and then repeat the procedure until all loops involving the link (j, k) is removed and then proceed to the next link.

In the following it is shown that FR can have arbitrarily poor performance. For the performance comparison, let's denote the maximum system lifetime obtained using algorithm X by T_{sys}^X, and the optimum system lifetime by T_{sys}^{opt}, and the ratio between these two values is denoted by
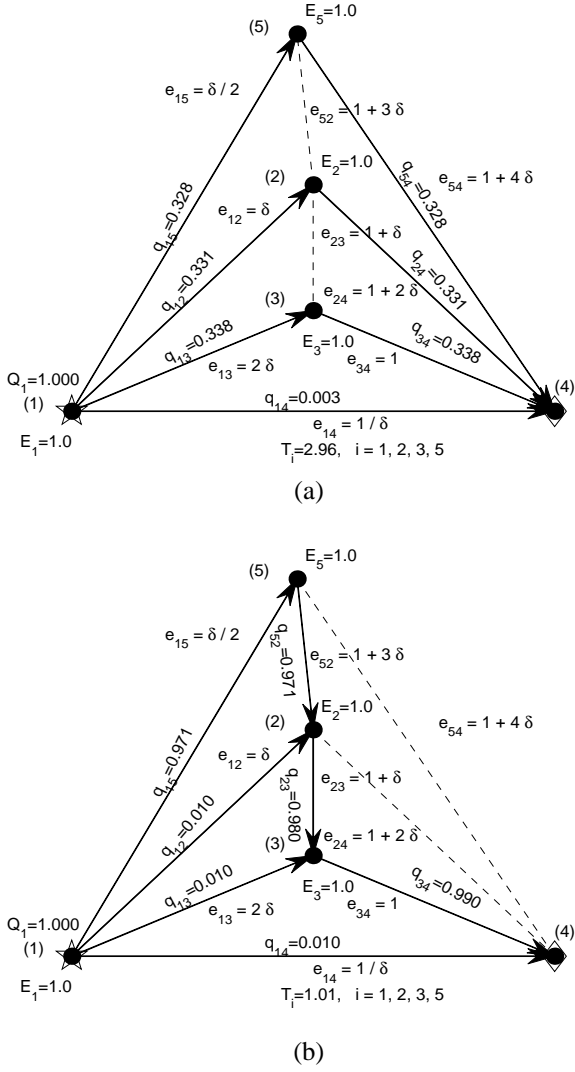
$$R_X = \frac{T_{sys}^X}{T_{sys}^{opt}}, \tag{25}$$

Fig. 3. An example showing local optimum convergence of FR for arbitrary positive constant δ. The flow values and lifetimes correspond to the case when $\delta = 0.01$: (a) global optimum with $T_{sys}^{opt} \approx 2.95$; (b) local optimum with $T_{sys}^{FR} \approx 1.01$.

which will be used throughout the paper as the performance measure.

An example showing the convergence to a local optimum is given in Fig.3, where a single commodity is originated from node 1 and is destined for node 4. δ is an arbitrary constant while the values of flows and lifetimes in the figure are for the case when $\delta = 0.01$. When $\delta = 0.01$, the optimum is $T_{sys}^{opt} \approx 2.95$, but the maximum system lifetime obtained by FR is $T_{sys}^{FR} = 1.01$. We can verify that R_{FR} can be as small as $1/3$ as δ approaches zero.

In the worst case R_{FR} can be shown to be arbitrarily small by expanding the network in a similar fashion. For example, the ratio R_{FR} reaches $1/4$ if we expand the network in Fig.3 by

adding node 6 with $e_{61} = \delta/3$, $e_{65} = 1 + 5\delta$, and $e_{64} = 1 + 6\delta$.

V. PERFORMANCE COMPARISON THROUGH SIMULATION

In this section, random graphs are generated in order to evaluate the performances of the proposed algorithms. The performances are compared with that of minimum transmitted energy (MTE) routing algorithm in order to see how much we gain in terms of the system lifetime compared to the conventional minimum transmitted energy routing algorithm. Comparison is also made with the maximum residual energy path (MREP) routing algorithm proposed in [3], where the path length was a vector whose elements were the link costs given by

$$c_{ij} = (\underline{E}_i - e_{ij}\lambda)^{-1}. \tag{26}$$

The lexicographical ordering was used in comparison of the two length vectors. The idea was to augment the flow on the path whose minimum residual energy after the flow augmentation will be the largest.

It has been shown in [3] that MTE can perform arbitrarily bad by an example. In the following example, it is shown that the minimum hop (MH) routing can perform arbitrarily bad. Fig.4 (a) shows the optimal solution and Fig.4 (b) shows the minimum hop solution. The ratio between the system lifetime obtained by MH and the optimal solution is $R_{MH} = \delta$. As $\delta > 0$ approaches zero, R_{MH} approaches zero. Note that the example scenario is possible since in a wireless environment, path loss is proportional to the square of the distance in free space and in higher orders in urban area, which makes multihop transmission less energy consuming then a single hop counterpart in many cases.

Let there be 20 nodes randomly distributed in a square of size 5 by 5. Assume that the transmission range of each node is limited by 2.5, i.e., $j \in S_i$ if and only if $d_{ij} \leq 2.5$, where d_{ij} is the distance between node i and node j. The energy expenditure per unit information transmission from node i to j is assumed to be given by

$$e_{ij} = \begin{cases} 1.0 \times 10^{-8}, & \text{if } d_{ij} \leq 0.025, \\ \left(\frac{d_{ij}}{2.5}\right)^4, & \text{if } 0.025 < d_{ij} \leq 2.5. \end{cases} \tag{27}$$

Note that there may be cases where no path is available between the origins and the destinations, although it is very rare in our setting. We simply discard these cases to assume the connectivity.

Two different scenarios are simulated: i) single commodity case where information generated at 5 origin nodes need to reach any one of two destination nodes; ii) multicommodity case where each of the 5 origin nodes has its own single designated destination node.

First of all, $FA(x_1, x_2, x_3)$ is simulated to find the best parameters x_1, x_2, and x_3.

Let node i have initial energy of $E_i = 1$ if i is even and $E_i = 2$ if i is odd. In the single commodity case, the origin nodes are given by $O = \{1, 2, 3, 4, 5\}$ and assume the information generation rates are $Q_1 = Q_2 = Q_3 = 2$ and
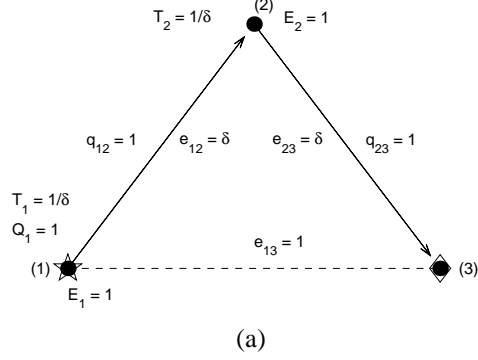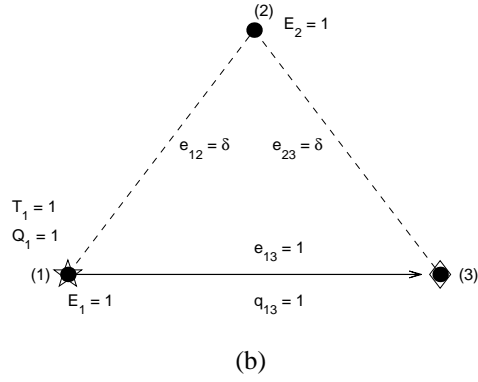
Fig. 4. An example showing that MH can have arbitrarily poor performance where $\delta > 0$ is a positive constant : (a) optimum system lifetime $T_{sysx}^{opt} = 1/\delta$; (b) system lifetime obtained by MH, $T_{sys}^{MH} = 1.00$.
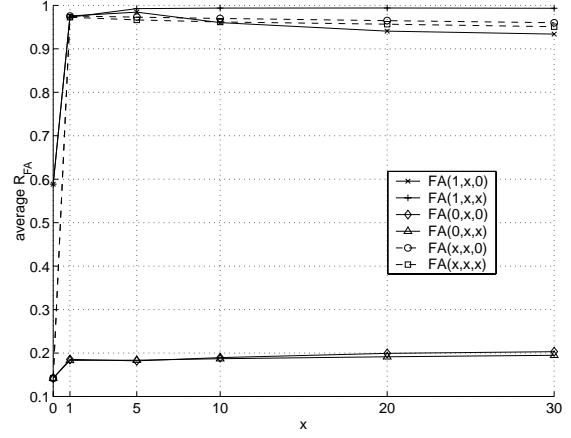

Fig. 5. The average performances of FA(x_1, x_2, x_3).

Fig. 6. The worst case performances of FA(x_1, x_2, x_3).

$Q_4 = Q_5 = 1$. The destination nodes are $D = \{19, 20\}$. In the multicommodity case $O^{(i)} = \{i\}$ and $D^{(i)} = \{i + 15\}$ for $i = 1, 2, 3, 4, 5$ with $Q_i^{(i)} = 2$ for $i = 1, 2, 3$ and $Q_i^{(i)} = 1$ for $i = 4, 5$.

Figs.5 and 6 show the results for single commodity case when $\lambda = 0.01$. Multicommodity case results are not shown here since they were similar to the single commodity case. In all cases, FA$(1, x, x)$ was the best in terms of both average and worst case performance. It should be noted that even with $x = 1$, $R_{FA(1,x,x)}$ was always over 0.8 of the optimal and about 0.98 of the optimal on the average. FA$(0, x, 0)$ and FA$(0, x, x)$ was the worst with average performance of about 0.2, which means that by considering only the residual energy without taking the energy expenditure into account the system lifetime can't be improved much. It's better than the MH solution but considerably worse than all the others which considers the energy expenditure term. The results also suggest that we use the normalized residual energy instead of the absolute residual energy, which can be more clearly seen in Fig.6 by comparing FA$(1, x, x)$ with FA$(1, x, 0)$.

Figs. 7 and 8 plot the average and the worst case performance of the best FA$(1, x, x)$ for various values of λ. We could observe that as λ got smaller, the performance was better. Note that the worst case of FA$(1, 50, 50)$ when $\lambda = 0.001$ was 0.9929. The curves weren't monotonically increasing, but we can see that for smaller λ it is so up to a larger x than the curves of larger λ. This phenomenon can be best explained as follows. While the shortest cost path may indeed be the optimal direction for the flow augmentation, it is only so for a certain amount of flow. As soon as λ gets larger than this amount, monotonicity of the convergence breaks. A somewhat similar behavior, though not exactly identical, can be found in many optimization methods using descent direction [13], where a procedure called line search is done to guarantee monotonic convergence.

Let's compare the performances of the other algorithms. In both single commodity and multicommodity case, let each node i have initial energy $E_i = 1$ and assume that the information generation rate at each origin node $o \in O^{(c)}$ is $Q_o^{(c)} = 1$ for each commodity.

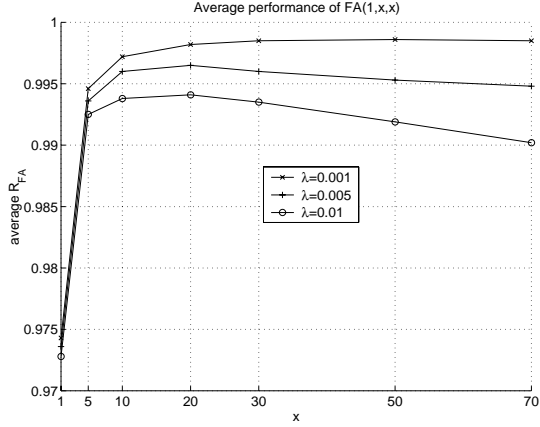The single commodity case results are presented first. Be-

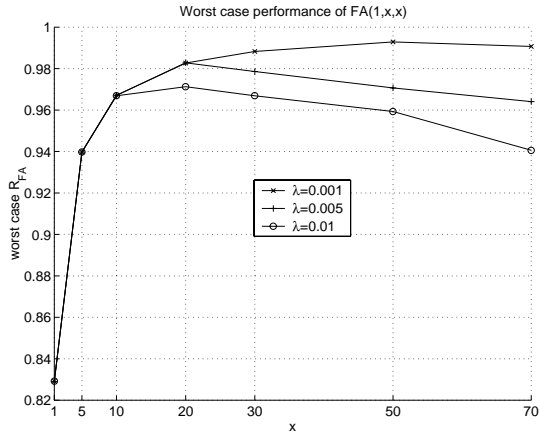Fig. 7. The average performance of $FA(1, x, x)$ for various values of λ.

Fig. 8. The worst case performance of $FA(1, x, x)$ for various values of λ.

fore going into the statistics, let's compare the algorithms by an example graph, where origin nodes are given by $O = \{1, 2, 3, 4, 5\}$ and destination nodes are given by $D = \{19, 20\}$. Figs.9, 10, and 11 show the solutions of MTE, FR, and $FA(1, 50, 50)$ with $\lambda = 0.001$, respectively. The true optimum is $T_{sys}^{opt} = 6.31$. One can observe that the advantage of our algorithms over MTE lies in the fact that the traffic is more spread out. The system lifetime obtained by $FA(1, 50, 50)$ and FR were more than four times as long as that of MTE in this example, and both were close to the optimal.

The performances of the algorithms are presented in Table II, and in Fig.12 average and worst cases of the algorithms are compared. Note that $\lambda = 0.001$ was used for MREP and FA. For each algorithm a total of 200 randomly generated graphs were simulated. $R_{FA(1,50,50)}$ was always over 0.99 of the optimal, i.e., even in the worst case. $FA(1, 1, 1)$'s performance was comparable to MREP's. While the average of R_{MTE} was about 0.7310, the average system lifetime of FR, MREP, and $FA(1, x, x)$ for $x \geq 1$ were above 0.95 of the optimal. R_{FR} and R_{MREP} were over 0.9 in about 90 % of the case while that

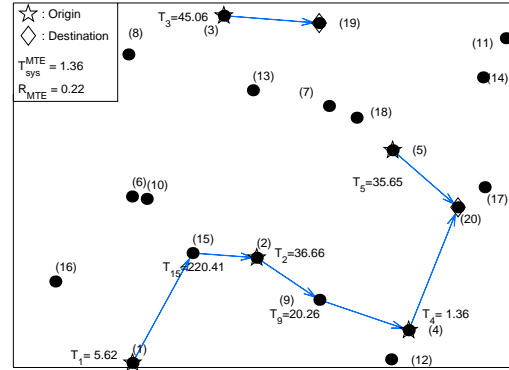| Algorithm X	$avg\ R_X$	$min\ R_X$	$Pr\{R_X > 0.9\}$
MTE	0.7310	0.1837	33%
FR	0.9596	0.6878	88%
MREP	0.9572	0.8110	89%
$FA(1, 1, 1)$	0.9744	0.7347	94%
$FA(1, 50, 50)$	0.9985	0.9911	100%

Fig. 9. An example showing the solution by MTE for single commodity case where nodes 1 through 5 are the origin nodes, and nodes 19 and 20 are the destination nodes where any one of the two nodes need to be reached.

of MTE was so in only 33 % of the case. The worst case of R_{MTE}, R_{FR} and R_{MREP} were 0.1837, 0.6878, and 0.8110, respectively. Although it was shown earlier that both MTE and FR can perform arbitrarily bad in the worst case, simulation results were in favor of FR. The average gain in the system lifetime obtained by the proposed algorithms were between 49 % and 55 % compared with MTE.

In the multicommodity case, commodity $i \in C$ where $C = \{1, 2, 3, 4, 5\}$ is assumed to be generated at node i and its destination node is node $i + 15$ among 20 randomly distributed nodes. Figs.13, 14, and 15 show examples of multicommodity case solutions by MTE, FR, and $FA(1, 50, 50)$ with $\lambda = 0.001$ respectively, where only the aggregate flows are depicted. In this example, the optimal system lifetime is $T_{sys}^{opt} = 7.81$, and the system lifetime obtained by FR and $FA(1, 50, 50)$ were more than one and a half times as long as that of MTE, where both were close to the optimal.

In the multicommodity case, the performances of the algorithms given in Table III and Fig.16 showed similar behavior to the single commodity case. $\lambda = 0.001$ was used for MREP and $FA(1, 50, 50)$. $R_{FA(1,50,50)}$ was always over 0.99 of the optimal, i.e., including the worst case, and again $FA(1, 1, 1)$'s performance was comparable to MREP's. While the average R_{MTE} was 0.6982, those of R_{FR} and R_{MREP} were 0.8862
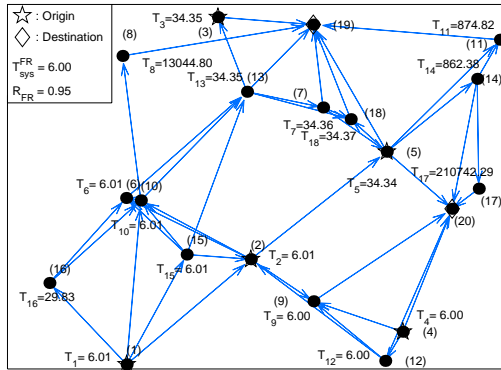
Fig. 10. An example showing the solution by FR for single commodity case where nodes 1 through 5 are the origin nodes, and nodes 19 and 20 are the destination nodes where any one of the two nodes need to be reached.
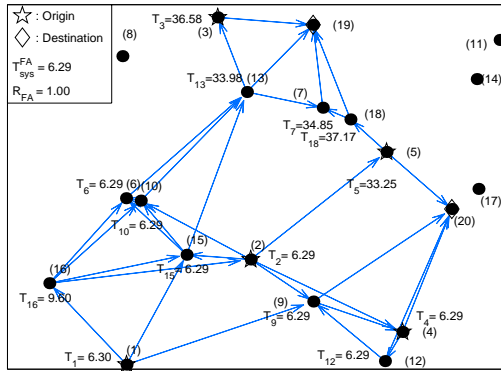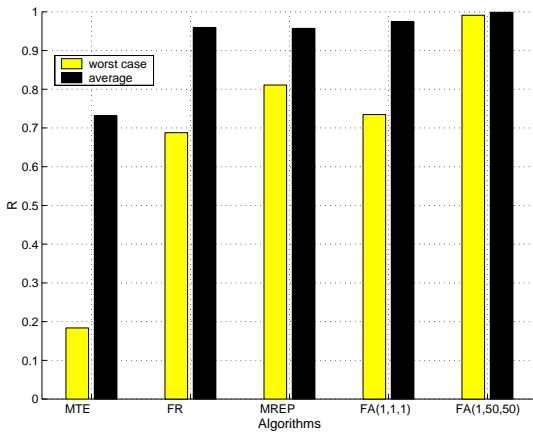

Fig. 11. An example showing the solution by $FA(1, 50, 50)$ when $\lambda = 0.001$ for single commodity case where nodes 1 through 5 are the origin nodes, and nodes 19 and 20 are the destination nodes where any one of the two nodes need to be reached.

Fig. 12. The comparison of average and worst case performances of all three algorithms are made in the single commodity case.

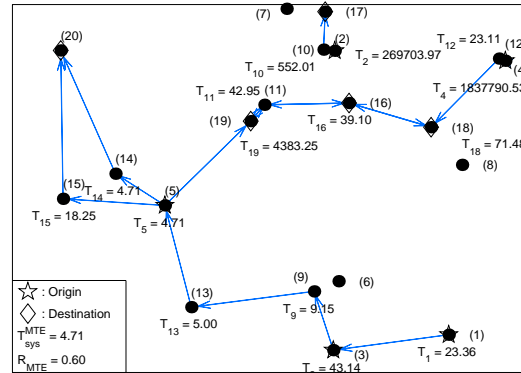| Algorithm X	$avg\ R_X$	$min\ R_X$	$Pr\{R_X > 0.9\}$
MTE	0.6982	0.2201	25%
FR	0.8862	0.4297	54%
MREP	0.9349	0.7298	69%
$FA(1,1,1)$	0.9565	0.7178	86%
$FA(1,50,50)$	0.9974	0.9906	100%

Fig. 13. An example showing the solution by MTE for multicommodity case where nodes 1 through 5 are the origin nodes and nodes 16 through 20 are the corresponding destination nodes, respectively.

and 0.9349, respectively. R_{FR} and R_{MREP} were over 0.9 in 54 % and 69 % of the case respectively, while that of MTE was so in only 25 % of the case. The worst cases of R_{MTE}, R_{FR}, and R_{MREP} were 0.2201, 0.4297, and 0.7298 respectively. While the performances of MTE, FR, and MREP deteriorated compared with single commodity case, the performance of $FA(1, 50, 50)$ was still very close to the optimal. The average gain in the system lifetime obtained by the proposed algorithms were between 40 % and 62 % compared with MTE.

VI. CONCLUSION

In power-controlled wireless ad-hoc networks, battery energy at network nodes is a very limited resource that needs to be utilized efficiently. One of the conventional routing objectives was to minimize the total consumed energy in reaching the destination. However, the conventional approach may drain out the batteries of certain paths which may disable further information delivery even though there are many nodes with plenty of energy. Therefore, we formulated the routing problem with the objective of maximizing the system lifetime given the sets of origin and destination nodes and the information generation rates at the origin nodes, and proposed a class of flow augmentation algorithms and a flow redirection algorithm which balance the energy consumption rates among the nodes in proportion to
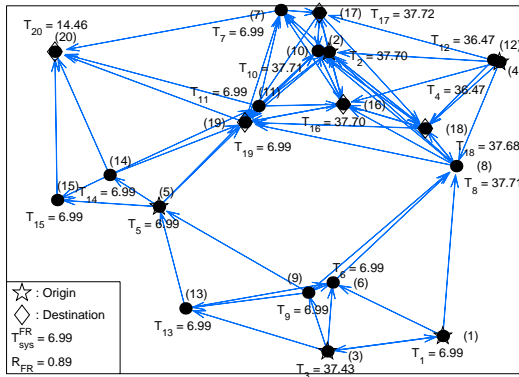
Fig. 14. An example showing the solution by FR for multicommodity case where nodes 1 through 5 are the origin nodes and nodes 16 through 20 are the corresponding destination nodes, respectively.
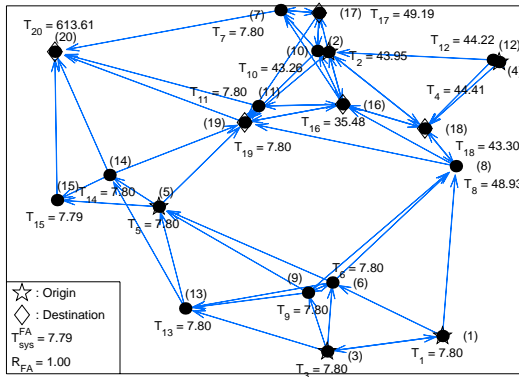

Fig. 15. An example showing the solution by $FA(1, 50, 50)$ when $\lambda = 0.001$ for multicommodity case where nodes 1 through 5 are the origin nodes and nodes 16 through 20 are the corresponding destination nodes, respectively.
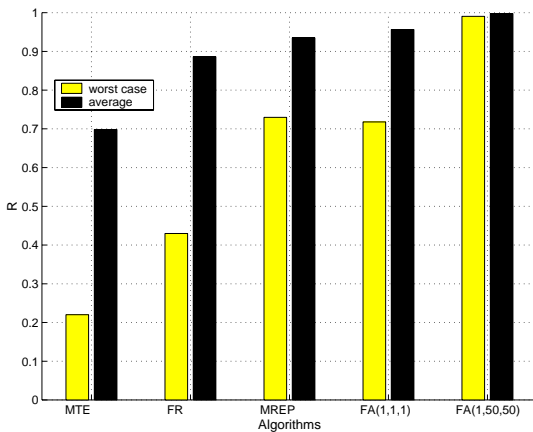

Fig. 16. The comparison of average and worst case performances of all three algorithms are made in the multicommodity case.

their energy reserves. The proposed algorithms are local and amenable to distributed implementation and showed close to the optimal performance most of the time, significantly improving the system lifetime, that is, as much as 60 % on the average over the conventional minimum transmitted energy routing.

REFERENCES

[1] Dennis J. Baker and Anthony Ephremides, "The architectural organization of a mobile radio network via a distributed algorithm," *IEEE Transactions on Communications*, vol. COM-29, no. 11, pp. 56–73, Jan. 1981.
[2] Dimitri Bertsekas and Robert Gallager, *Data Networks*, Prentice-Hall, Inc., 2nd edition, 1987.
[3] J.-H. Chang and L. Tassiulas, "Routing for maximum system lifetime in wireless ad-hoc networks," in *Proceedings of 37-th Annual Allerton Conference on Communication, Control, and Computing*, Monticello, IL, Sept. 1999.
[4] Wai-Kai Chen, *Theory of Nets: Flows in Networks*, Wiley, 1990.
[5] T. Cormen, C. Leiserson, and R. Rivest, *Introduction to Algorithms*, McGraw-Hill and MIT Press, 1990.
[6] Anthony Ephremides, Jeffrey E. Wieselthier, and Dennis J. Baker, "A design concept for reliable mobile radio networks with frequency hopping signaling," *Proceedings of the IEEE*, vol. 75, no. 1, pp. 56–73, Jan. 1987.
[7] M. Ettus, "System capacity, latency, and power consumption in multihop-routed SS-CDMA wireless networks," in *Proceedings of IEEE Radio and Wireless Conference (RAWCON) 98*, Colorado Springs, CO, Aug. 1998, pp. 55–58.
[8] R.G. Gallager, P.A. Humblet, and P.M. Spira, "A distributed algorithm for minimum weight spanning trees," Tech. Rep. LIDS-P-906-A, Lab. Inform. Decision Syst., Massachusetts Inst. of Technol., Cambridge, MA, Oct. 1979.
[9] D. Johnson and D. Maltz, "Dynamic source routing in ad hoc wireless networks," *Mobile Computing*, 1996.
[10] Teresa H. Meng and Volkan Rodoplu, "Distributed network protocols for wireless communication," in *Proceedings of the 1998 IEEE International Symposium on Circuits and Systems, ISCAS'98*, Monterey, CA, June 1998, vol. 4, pp. 600–603.
[11] A. Michail and A. Ephremides, "A distributed routing algorithm for supporting connection-oriented service in wireless networks with time-varying connectivity," in *Proceedings Third IEEE Symposium on Computers and Communications, ISCC'98*, Athens, Greece, June 1998, pp. 587–591.
[12] S. Murthy and J.J. Garcia-Luna-Aceves, "An efficient routing protocol for wireless networks," *ACM Mobile Networks and Applications Journal, Special Issue on Routing in Mobile Communication Networks*, 1996.
[13] Stephen G. Nash and Ariela Sofer, *Linear and Nonlinear Programming*, McGraw-Hill, 1996.
[14] Vincent D. Park and M. Scott Corson, "A highly distributed routing algorithm for mobile wireless networks," in *Proc. IEEE INFOCOM'97*, Kobe, Japan, 1997.
[15] C. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance vector routing (DSDV) for mobile computers," in *ACM SIGCOMM*, Oct. 1994.
[16] Volkan Rodoplu and Teresa H. Meng, "Minimum energy mobile wireless networks," in *Proceedings of the 1998 IEEE International Conference on Communications, ICC'98*, Atlanta, GA, June 1998, vol. 3, pp. 1633–1639.
[17] Timothy Shepard, "Decentralized channel management in scalable multihop spread spectrum packet radio networks," Tech. Rep. MIT/LCS/TR-670, Massachusetts Institute of Technology Laboratory for Computer Science, July 1995.
[18] S. Singh, M. Woo, and C.S. Raghavendra, "Power-aware routing in mobile ad hoc networks," in *Proceedings of Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking*, Dallas, TX, Oct. 1998, pp. 181–190.