

Energy Optimization under Informed Mobility

Chiping Tang, *Member, IEEE*, and Philip K. McKinley, *Member, IEEE*

Abstract—Energy optimization is important in wireless ad hoc networks, where node battery power is usually limited. Research results show that such a network can exploit controlled node mobility to reduce communication-related energy consumption. However, node movement itself usually consumes energy. In this paper we study the energy optimization problem that accounts for energy costs associated with both communication and physical node movement. We refer to this model as *informed mobility*. We first review the theoretical foundations on how to reduce total communication energy consumption, as well as increase system lifetime, by combining node movement and transmission power adaptation. Next, we describe and analyze the informed mobility optimization problem. Based on this analysis, we introduce localized algorithms and protocols for informed mobility. We propose iMobif, a flow-based informed mobility framework that collects network information for mobility decision making. We demonstrate how to use iMobif to minimize total communication energy consumption as well as to maximize system lifetime. We compare the performance of iMobif to that of systems with no mobility or only cost-unaware mobility. Simulation results show iMobif is effective in reducing energy consumption relative to such systems.

Index Terms—Mobile ad hoc network, energy optimization, informed mobility, localized algorithm, routing protocol, node position, system lifetime.

1 INTRODUCTION

IN wireless ad hoc networks, wireless nodes communicate with each other by sending data flows either directly or through intermediate relays. The network topology, and specifically, the paths of flows, significantly affect communication energy efficiency at individual nodes. Excessive or disproportionate energy consumption among nodes can lead to premature failure of the network. To address this problem, several energy optimization approaches including clustering [1] and topology control [2], [3], [4], have been proposed. In these approaches, wireless nodes use techniques such as data aggregation and transmission power adjustment to reduce energy consumption.

If some or all of the nodes in the network are mobile (for example, as in a fleet of microrobots [5]), then it is possible to complement the above energy management strategies with approaches that account for, and even exploit, node mobility. In *mobility-aware* approaches, the system changes its behavior in response to, or by predicting, node movements. This strategy has been used for topology control [6], energy optimization [7], network throughput improvement [8], and system security improvement [9]. In *mobility-aided* approaches, the subject of this paper, nodes actually change their locations to optimize certain performance metrics. For example, a mobile sensor node with sufficient energy can move about a sensor field to collect data from other sensor nodes, reducing energy consumption at those nodes. This

strategy has been proposed as a means to recover a disconnected topology [10], increase sensor surveillance coverage [11], [12], as well as reduce total energy consumption [13], [14], [15]. Depending on initial network topology and the power model, such *controlled* mobility (as opposed to random mobility) can reduce energy consumption by up to 50 percent [13].

Despite the fact that node movement itself typically consumes energy, however, many prior studies of controlled mobility have not accounted for this factor. An exception is the work of Goldenberg et al. [13], which specifically addresses this trade-off. Those authors show that it is possible to numerically compare the benefit of mobility with its cost; a node is allowed to move only when the benefit exceeds the cost. Simulation results indicate the benefit outweighs the cost when the number of data bits in the flow surpasses a certain threshold. In this paper, we extend that work by designing algorithms and protocols for the collection and distribution of the benefit/cost information, thus enabling *local* decision-making on controlled mobility.

In this *informed* mobility optimization problem, the cost and benefit of controlled mobility are weighed in mobility decisions. Factors affecting this tradeoff include the mobility pattern, the network topology, and the data transmission pattern. For such a strategy to be practical, both the cost and benefit calculations should be conducted in a distributed fashion. In addition, such calculations should not include any system parameters that are not measurable by individual nodes. To address this need, we propose *iMobif*, a flow-based framework to facilitate decision-making related to mobility.

In iMobif, flow sources determine the current mobility strategy and status (enabled or disabled). They inform every node on the flow path of the strategy (and its on-off status) using data packets. After receiving a data packet, an intermediate node calculates its new target location according to the current mobility strategy and, if the strategy is

• C. Tang is with Microsoft Corporation, One Microsoft Way, Redmond, WA 98052. E-mail: chipingt@microsoft.com.

• P.K. McKinley is with the Department of Computer Science, Michigan State University, 3115 Engineering Building, East Lansing, MI 48824-1027. E-mail: mckinley@cse.msu.edu.

Manuscript received 22 Feb. 2005; revised 8 June 2005; accepted 31 Aug. 2005; published online 26 June 2006.

Recommended for acceptance by I. Stojmenovic, S. Olariu, and D. Simplot-Ryl.

For information on obtaining reprints of this article, please send e-mail to: tpd@computer.org, and reference IEEECS Log Number TPDI-0175-0205.

enabled, starts moving toward that location. At the same time, the node calculates the cost and benefit of the mobility strategy using local information, aggregates the combined cost-benefit value with the corresponding value in the packet header, and forwards the packet to the next node in the flow path. When the packet eventually arrives at the flow destination, that node can evaluate the overall performance of the mobility strategy based on the aggregate information. If the node determines that a status change is needed, it sends a notification packet back to the source, which updates the mobility strategy status and uses the next data packet to inform all flow nodes.

In addition to changing the mobility strategy status of the existing nodes in a flow, iMobif also supports changing the composition of the flow path. Specifically, iMobif can add new nodes to a flow path so as to further reduce total energy consumption. To achieve this capability, a flow node periodically broadcasts "hello" messages containing the residual energy and location information regarding itself and its flow neighbors. Neighboring nodes that are not currently on the flow path can use this information to determine whether or not they should join the flow path, reporting the cost-benefit comparison results to the flow node. The flow node then chooses the result that maximizes the cost-benefit gain and instructs the chosen node to move toward the flow. The chosen node periodically checks the "hello" messages to determine if the movement remains worthwhile. It stops moving if the expected benefit is not sufficient to counterbalance the cost. Otherwise, the node will eventually join the flow.

To our knowledge, this is the first work to address an adaptive mobility strategy based on online cost-benefit information in wireless ad hoc networks. The approach is general in that iMobif can be tuned for different energy optimization goals by changing the mobility strategy and the corresponding cost-benefit aggregate function. In this paper, we demonstrate the effectiveness of iMobif by implementing two different functions, one to minimize total energy consumption and the other to maximize system lifetime (we define *system lifetime* as the time interval between system initialization and the first time at which a node in the system fails to communicate with others due to energy depletion). While the mobility strategy for the first function is adopted directly from [13], the strategy for the second function is novel. We reported preliminary results on this work in [16].

The remainder of the paper is organized as follows: In Section 2, we review the theoretical foundations on how to reduce total communication energy consumption as well as increase system lifetime by using a combination of node movement and transmission power adaptation. In Section 3, we describe the informed mobility optimization problem and propose solutions that use global information. In Section 4, we present the localized algorithms that make informed mobility decisions at individual nodes. In Section 5, we describe the operation of the iMobif framework. We present performance evaluation results in Section 6, and discuss related work in Section 7. In Section 8, we make concluding remarks and discuss potential research directions.

2 PRELIMINARIES

In this section, we introduce the energy models used in this study, and describe the positioning of nodes so as to minimize communication energy consumption and maximize system lifetime.

2.1 Energy Models

It is well-known that the power needed for successful wireless data transmission is determined by the distance between communication entities and the noise level of the communication channel. In this network-layer study, we adopt a transmission power model similar to the one used in [13]. Let $P_T(d)$ be the power needed for data transmission across distance d , then

$$P_T(d) = a + bd^\alpha,$$

where a , b , and α are constants dependent on the characteristics of the communication channel. The value of α is usually greater than or equal to 2. The energy consumption for transmitting l data bits across distance d is

$$E_T(d, l) = l \cdot P_T(d).$$

In addition to transmission energy consumption, we also consider the energy consumption for node mobility, or *mobility cost*. Of course, mobility cost is dependent on the actual trace of node movement. For simplicity, we adopt a distance proportional cost model similar to the one used in [13]. In this model, the mobility cost $E_M(d)$ can be calculated from the distance traversed, d :

$$E_M(d) = kd,$$

where k is a constant dependent on the environment and the mass of the mobile node. The distance proportional model is reasonable for wheeled vehicles (see discussion in [13] for details).

2.2 Optimal Node Positions

In this study, we assume that the source and destination nodes of a flow are stationary, but that other nodes are free to move to new locations. We note that in many situations, topology constraints may limit node movement. Such a constraint may be characteristic of the application itself. For example, in a sensor network it might be required that every physical area within the sensing range be covered by at least one sensor node. On the other hand, the constraint might relate to the performance of low-level communication protocols. For example, a limit may be placed on the physical density of nodes so as to reduce channel contention and mutual interference. Evaluating the performance of the proposed methods under such constraints is deferred to a future study. In this paper, the only constraint imposed on the network topology is that it remain connected. Under this assumption, let us begin by presenting two theorems on optimal node configurations.

Theorem 1 [13]. *To minimize total transmission energy consumption, the optimal positions of all N nodes in a one-to-one flow must lie entirely on the straight line between the source node n_0 and the destination node n_{N-1} . Furthermore, all the nodes should be evenly spaced on the line.*

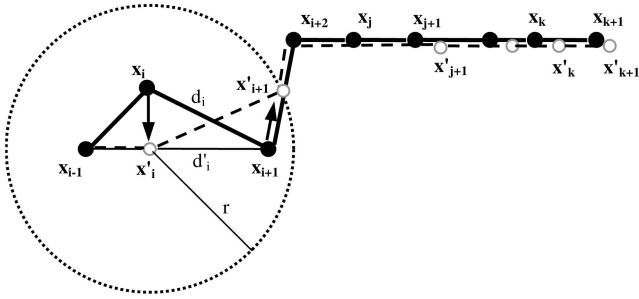


Fig. 1. Increasing system lifetime by moving relevant nodes.

To maximize system lifetime, it is necessary to adjust node positions to balance residual energy. A straightforward strategy is to reduce transmission distance for nodes with low residual energy and increase the transmission distance for nodes with high residual energy. We define *bottleneck nodes* in a network as those with minimum residual energy.

Theorem 2. *To maximize system lifetime, the optimal positions of all N nodes in a one-to-one flow must lie entirely on the straight line between the source node n_0 and the destination node n_{N-1} . Let \mathbf{x}_i and e_i be the location and residual energy, respectively, of node n_i (the i th node in the flow). Then, for any $0 \leq i < N-1$, $0 \leq j < N-1$, $\frac{P_T(d_i)}{P_T(d_j)} = \frac{e_i}{e_j}$, where $d_i = |\mathbf{x}_i - \mathbf{x}_{i+1}|$ is the distance between nodes n_i and n_{i+1} .*

Proof. Assume node n_i is not on the line between n_{i-1} and n_{i+1} in an optimal configuration. We show system lifetime can be increased by moving n_i to some point on the line. By moving n_i from location \mathbf{x}_i to \mathbf{x}'_i on the line, as shown in Fig. 1, the distance from n_i to n_{i+1} can be reduced without increasing the distance between any other node pairs. Let $d'_i = |\mathbf{x}'_i - \mathbf{x}_{i+1}| < d_i$ be the distance between \mathbf{x}'_i and \mathbf{x}_{i+1} , and M be the number of bottleneck nodes in the network. We show how the distance reduction ($d_i - d'_i$) can be propagated to all M bottleneck nodes as follows:

1. If n_i is a bottleneck node, let $r = d_i - \frac{d_i - d'_i}{M}$.
2. Otherwise, let $r = d_i$.
3. Draw a circle around \mathbf{x}'_i with radius r ; node n_{i+1} falls within this circle since $r \geq d'_i$.
4. If node n_{i+2} also falls within this circle, move n_{i+1} to \mathbf{x}_{i+2} .
5. Otherwise, let \mathbf{x}'_{i+1} be the point where the line $\mathbf{x}_{i+1}\mathbf{x}_{i+2}$ intersects the circle; move n_{i+1} to \mathbf{x}'_{i+1} .

As shown in Fig. 1, the above approach can always reduce the distance between n_{i+1} and n_{i+2} while maintaining the distance between n_i and n_{i+1} (if n_i is not a bottleneck node) or reducing the distance between n_i and n_{i+1} (if n_i is a bottleneck node). Similarly, we can change the location of all n_j , $j \neq i$, in this way so that the distance reduction $d_i - d'_i$ is propagated and distributed to all bottleneck nodes, and the distance between any bottleneck node and its downstream node is reduced by at least $\frac{d_i - d'_i}{M}$. Consequently, all bottleneck nodes can reduce their energy consumption and the system lifetime can be increased. This is contradictory to the assumption that the network is already in an optimal configuration.

Therefore, all nodes in an optimal configuration should be located on the straight line between n_0 and n_{N-1} .

Next, we show that for any $0 \leq i < N-1$, $0 \leq j < N-1$, $\frac{P_T(d_i)}{P_T(d_j)} = \frac{e_i}{e_j}$ when all nodes are located on a straight line. Assume the energy consumption of all activities other than communication is negligible. Let $t_i = \frac{e_i}{P_T(d_i)}$ be the lifetime of node n_i , and t be the system lifetime. We need to prove that in an optimal configuration $t_i = t$ for all $0 \leq i < N$. It is obvious $t_i \geq t$. We show $t_i = t$ for all $0 \leq i < N$ by contradiction, specifically, that if $t_j > t$ for some node n_j , then the system lifetime can be increased and the current configuration is not optimal.

Let n_k be the first bottleneck node for $k > j$ with $t_k = t$. Assume there are M such bottleneck nodes. As shown in Fig. 1, we can move node n_{j+1} to location $\mathbf{x}'_{j+1} = \mathbf{x}_{j+1} + \delta_{j+1}$ so that the lifetime $t < t'_j = \frac{e_j}{P_T(d'_j)} < t_j$. We then shift all nodes from n_{j+2} to n_k to the end of n_{N-1} by distance δ_{j+1} so that the distance between them is not changed. Now, move node n_{k+1} to location $\mathbf{x}'_{k+1} = \mathbf{x}'_k + d_k - \frac{\delta_{j+1}}{M}$ so that $d'_k = d_k - \frac{\delta_{j+1}}{M}$. Then, the lifetime of n_k is increased to $t'_k = \frac{e_k}{P_T(d'_k)} > \frac{e_k}{P_T(d_k)} = t_k = t$. Similarly, we can increase the lifetime of all M bottleneck nodes and, thus, increase the system lifetime. This is contradictory to the assumption that the current configuration is optimal. Therefore, in an optimal configuration $\frac{e_i}{P_T(d_i)} = t$ for all nodes n_i and, thus, $\frac{P_T(d_i)}{P_T(d_j)} = \frac{e_i}{e_j}$ for any $0 \leq i < N-1$, $0 \leq j < N-1$. \square

As shown in the above theorems, energy consumption can be optimized by moving nodes to their optimal positions and setting the transmission power of each node to barely reach the next node in the flow. Obviously, the optimal position of node n_i for total energy consumption minimization is

$$\mathbf{x}_i = \mathbf{x}_0 + (\mathbf{x}_{N-1} - \mathbf{x}_0) \cdot \frac{i}{N-1}.$$

Calculating the optimal positions needed to maximize system lifetime is more complicated. From Theorem 2 and the transmission power model, we know, for each $0 < i < N$

$$\frac{a + b(d_{i-1})^\alpha}{a + b(d_i)^\alpha} = \frac{e_{i-1}}{e_i}, \quad \text{and} \quad \sum_{i=0}^{N-2} d_i = D = |\mathbf{x}_0 - \mathbf{x}_{N-1}|.$$

The closed-form solutions for d_{i-1} and d_i are very complicated or even unattainable for $\alpha > 2$. To make the calculations practical, we can use a simple approximation:

$$\frac{(d_{i-1})^\alpha}{(d_i)^\alpha} = \frac{e_{i-1}}{e_i}.$$

Under this simplification, the optimal position of n_i for system lifetime maximization is

$$\mathbf{x}_i = \mathbf{x}_0 + (\mathbf{x}_{N-1} - \mathbf{x}_0) \cdot \frac{\sum_{j=0}^{i-1} (e_j)^\frac{1}{\alpha}}{\sum_{j=0}^{N-2} (e_j)^\frac{1}{\alpha}} \quad \text{for } 0 < i < N-1.$$

Next, we show how to take both transmission energy consumption and mobility cost into consideration to achieve overall energy optimization.

3 THE INFORMED MOBILITY OPTIMIZATION PROBLEM

In this section, we introduce the informed mobility optimization problem. First, we present the definition of the problem. Then, we show how to solve the problem for flows with fixed composition. Last, we discuss the solution for flows with no composition constraint.

3.1 Problem Definition

In the informed mobility optimization problem, the position and initial residual energy of each node is given. The goal is to find the optimal node positions, under network topology constraints, that maximize the energy utilization $G(\mathbf{E}, \mathbf{X})$, which is defined as a function on the current node energy set \mathbf{E} and position set \mathbf{X} .

Definition 1. *The informed mobility optimization problem.*

Let \mathbf{N}_A be the node set of a mobile ad hoc network A . Given the initial energy set \mathbf{E} and position set \mathbf{X} , which contains the initial node energy pair $\langle n_i, e_i \rangle$ and node position pair $\langle n_i, \mathbf{x}_i \rangle$ for each node $n_i \in \mathbf{N}_A$, find the optimal position set \mathbf{X}' ($\langle n_i, \mathbf{x}'_i \rangle$ for each node n_i) such that the network topology constraint $C(\mathbf{X}')$ is satisfied and the energy utilization function $G(\mathbf{E}', \mathbf{X}')$ is maximized. According to the cost model (Section 2.1), this goal is equivalent to maximizing

$$G(\mathbf{E} - E_M(d(\mathbf{X}, \mathbf{X}')), \mathbf{X}') = G(\{e_i - k|\mathbf{x}_i - \mathbf{x}'_i|\}, \{\mathbf{x}'_i\}),$$

while satisfying $C(\{\mathbf{x}'_i\})$, where $|\mathbf{x}_i - \mathbf{x}'_i|$ is the distance between positions \mathbf{x}_i and \mathbf{x}'_i .

The solution to the informed mobility optimization problem depends on the energy utilization function G and the topology constraint C . In this work we study two flow-based subproblems under the topology constraint (as mentioned above) of preventing network disconnection. Specifically, we assume there exist multiple flows \mathbf{F}_i in network A , each of which transmits λ_i data bits from a source node to a destination node. Let node $n_{(i,j)}$ be the j th node in flow $\mathbf{F}_i \in \mathbf{F}_A$, where \mathbf{F}_A is the flow set of network A . A network node A may be involved in multiple flows. It expends a certain amount of energy to transmit the data bits of each flow (except for the flow in which the node is the destination) to the next node in the flow. According to the transmission power model described in Section 2.1, the minimum transmission energy consumption for each node n_i is

$$E_T^i(\mathbf{X}) = \sum_{j, n_i = n_{(j,k_j)} \in \mathbf{F}_j} E_T(|\mathbf{x}_{(j,k_j)} - \mathbf{x}_{(j,k_j+1)}|, \lambda_j),$$

and the minimum unit transmission energy consumption for each node n_i is

$$E_t^i(\mathbf{X}) = \sum_{j, n_i = n_{(j,k_j)} \in \mathbf{F}_j} E_T(|\mathbf{x}_{(j,k_j)} - \mathbf{x}_{(j,k_j+1)}|, 1).$$

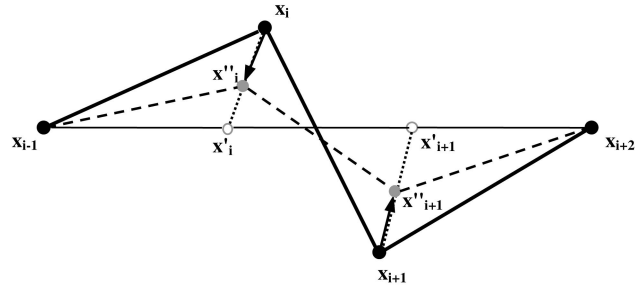


Fig. 2. Nodes moving to their optimal positions.

The first subproblem seeks to minimize total energy consumption. We define the energy utilization function G_1 as

$$G_1(\mathbf{E}, \mathbf{X}) = \sum_{\langle n_i, e_i \rangle \in \mathbf{E}} (e_i - E_T^i(\mathbf{X})),$$

in which e_i is introduced to reflect the energy consumption of node movement.

The second subproblem seeks to maximize system lifetime, defined as the number of flow bits transmitted before a node in the flow fails to communicate with others due to energy depletion. We define the energy utilization function G_2 as

$$G_2(\mathbf{E}, \mathbf{X}) = \min_{\langle n_i, e_i \rangle \in \mathbf{E}} \frac{e_i}{E_t^i(\mathbf{X})}.$$

The formulae above show the evaluation of the energy utilization functions depends on the flow compositions. Next, we introduce the methods for calculating optimal node positions.

3.2 Solutions for Flows with Fixed Composition

Section 2.2 showed that the optimal positions of nodes in a flow must lie entirely on the straight line between the flow source and the destination. This observation, however, does not account for the mobility cost. Indeed, it is possible for a node to spend so much energy to move to its “optimal” position, that the energy utilization functions produce lower values than if the node had not moved.

We use an example to illustrate our solutions. In Fig. 2, the nodes (e.g., n_i) move from their initial positions (e.g., \mathbf{x}_i) to their “optimal” positions (e.g., \mathbf{x}'_i) to minimize total energy consumption. At the initial positions, the mobility cost is zero, but the transmission energy consumption is high. As the nodes start moving to their “optimal” positions according to the theorems in Section 2.2, the mobility cost increases while the transmission energy consumption decreases. As shown in the figure, at intermediate positions (e.g., \mathbf{x}'_i), the decrease of total transmission distance $\sum_i |\mathbf{x}_i - \mathbf{x}_{i+1}| - \sum_i |\mathbf{x}'_i - \mathbf{x}'_{i+1}|$ is monotonic in the distance traveled $\sum_i |\mathbf{x}_i - \mathbf{x}'_i|$. The reduction of transmission energy consumption, however, depends on node positions. We propose a simple algorithm shown in Fig. 3: If the mobility cost $E_M(d(\mathbf{X}', \mathbf{X}))$ is not covered by the energy savings $\sum_i (E_T^i(\mathbf{X}) - E_T^i(\mathbf{X}'))$ at the “optimal” positions \mathbf{X}' , then the original positions \mathbf{X} are the true optimal positions; otherwise, positions \mathbf{X}' are the true optimal positions. A similar algorithm can be used to maximize system lifetime.

```

1 ) function GetInformedMobilityPositions(Flow  $F$ ) : PositionSet
2 ) begin
3 )    $X = \text{GetPositions}(F)$ ;  $X' = \text{GetOptimalPositions}(F)$ ;  $E = \text{GetEnergySet}(F)$ 
4 )   CalculateResidualEnergyAfterMovement( $X, X', E, E'$ )
5 )   /* comparing energy utilization with and without mobility */
6 )   if  $G_1(E, X) > G_1(E', X')$  then return  $X$ 
7 )   else return  $X'$ 
8 ) end
9 )
10) procedure CalculateResidualEnergyAfterMovement
    (PositionSet  $X$ , PositionSet  $X'$ , EnergySet  $E$ , output EnergySet  $E'$ )
11) begin
12)   for each  $x_i \in X, x'_i \in X', e_i \in E, e'_i \in E'$  do
13)      $e'_i = e_i - E_M(d(x'_i, x_i))$ 
14) end

```

Fig. 3. Calculating optimal node positions for fixed composition flows.

3.3 Solutions for Flows with Dynamic Composition

By changing the composition of a flow (e.g., adding new nodes to a flow), the value of the energy utilization functions can be further improved. Again we use an example of minimizing total energy consumption to illustrate the solution. Assume flow F includes m nodes, and the distance between the source and the destination is D . Inserting node n' into the flow between nodes n_i and n_{i+1} , and moving nodes to their “optimal” positions according to Theorem 1, the total transmission energy consumption becomes $m \cdot E_T(\frac{D}{m}, \lambda)$. Considering the transmission energy consumption before the insertion, the energy savings is $(m-1) \cdot E_T(\frac{D}{m-1}, \lambda) - m \cdot E_T(\frac{D}{m}, \lambda)$. If the mobility cost is lower than the energy savings, then the total energy consumption is reduced. The algorithm is shown in Fig. 4. This approach can also be used to maximize system lifetime by moving the nodes to their “optimal” positions according to Theorem 2 after the insertion.

3.4 Multiple Flow Considerations

So far, we have discussed the optimal positions for nodes in a single flow. In real scenarios, multiple flows may coexist in one network, and a given node may be included in multiple flows. Moreover, its optimal position with respect to one flow is likely to be different than with respect to another. A coordination mechanism is needed to combine the results for multiple flows.

We propose a solution based on the results from [13]. We imagine there exists a virtual force for each flow that pushes a node from its current position to its “optimal” position determined by that flow. The strength of the force is

proportional to the number of data bits in the flow, and dependent on the distance between the node and its neighboring nodes in the flow. The forces may neutralize each other and generate a new force, which pushes the node to its final position. The final position of node n_i can be calculated as

$$x'_i = \frac{\sum_{j, n_j = n(j, k_j) \in F_j} (|x'_{(j, k_j)} - x_{(j, k_j+1)}|)^{\alpha-2} + |x'_{(j, k_j)} - x_{(j, k_j-1)}|^{\alpha-2} \lambda_j x'_{(j, k_j)}}{\sum_{j, n_j = n(j, k_j) \in F_j} (|x'_{(j, k_j)} - x_{(j, k_j+1)}|)^{\alpha-2} + |x'_{(j, k_j)} - x_{(j, k_j-1)}|^{\alpha-2} \lambda_j},$$

where $x'_{(j, k_j)}$ is the optimal position of node n_i calculated by flow F_j .

After the final target position is obtained, the mobility cost is compared with the aggregate mobility benefit, and node mobility is enabled only when the latter is greater. The algorithm is given in Fig. 5.

4 LOCALIZED ALGORITHMS

The algorithms presented in Section 3 use global information, such as the positions of the source and destination nodes, to make mobility decisions. In real scenarios, it is unlikely that such information is always available at each node. In this section, we propose localized versions of the algorithms. We replace the position and residual energy of all flow nodes with the position and residual energy of the current node and its neighboring nodes in the flow.

4.1 Solutions for Flows with Fixed Composition

First, we propose a localized algorithm for flows with fixed composition based on the algorithm in [13]. The basic idea

```

1 ) procedure JoinFlowUnderInformedMobility(Flow  $F$ , Node  $n$ )
2 ) begin
3 )    $X = \text{GetPositions}(F)$ ;  $X' = \text{GetOptimalPositions}(F)$ ;  $E = \text{GetEnergySet}(F)$ 
4 )   CalculateResidualEnergyAfterMovement( $X, X', E, E'$ )
5 )   /* attempting to insert  $n$  into the flow */
6 )    $F_0 = F$ ;  $n.\text{join}(F_0)$ 
7 )    $X_0 = \text{GetPositions}(F_0)$ ;  $X'_0 = \text{GetOptimalPositions}(F_0)$ ;  $E_0 = \text{GetEnergySet}(F_0)$ 
8 )   /* calculating residual energy assuming  $n$  has joined the flow */
9 )   CalculateResidualEnergyAfterMovement( $X_0, X'_0, E_0, E'_0$ )
10)   /* insert  $n$  into the flow if the energy utilization increases after the insertion */
11)   if  $G_1(E'_0, X'_0) > G_1(E', X')$  and  $G_1(E'_0, X'_0) > G_1(E, X)$  then  $n.\text{join}(F)$ 
12) end

```

Fig. 4. Node insertion algorithm.

```

1) function GetInformedMobilityPositionsInMultipleFlows(FlowSet  $F$ ) : PositionSet
2) begin
3)    $X = \text{GetPositions}(F)$ ;  $E = \text{GetEnergySet}(F)$ 
4)   /* obtaining the optimal positions for each flow */
5)   for each flow  $f_j \in F$  do
6)      $X'_j = \text{GetOptimalPositions}(f_j)$ 
7)     /* calculating the optimal position for each node */
8)     for each node  $n_i \in F$  do begin
9)        $u = 0$ ;  $w = 0$ 
10)      for each flow  $f_j \in F$  do begin
11)        if  $n_i = n_{(j,k)} \in f_j$  then begin
12)           $x'_{(j,k)} = X'_j(k)$ ; /* obtaining the optimal position for a single flow */
13)          /* combining the optimal position of the current flow with other flows */
14)           $u = u + ((d(x'_{(j,k)}, x_{(j,k+1)}))^{\alpha-2} + (d(x'_{(j,k)}, x_{(j,k-1)}))^{\alpha-2}) \cdot \lambda_j \cdot x'_{(j,k)}$ 
15)           $w = w + ((d(x'_{(j,k)}, x_{(j,k+1)}))^{\alpha-2} + (d(x'_{(j,k)}, x_{(j,k-1)}))^{\alpha-2}) \cdot \lambda_j$ 
16)        end
17)      end
18)       $x'_i = u/w$ ; /* calculating the weighted optimal position of this node in  $X'$  */
19)    end
20)    /* calculating residual energy after node movement */
21)    CalculateResidualEnergyAfterMovement( $X, E, X', E'$ )
22)    /* comparing energy utilization with and without mobility */
23)    if  $G_1(E, X) > G_1(E', X')$  then return  $X$ 
24)    else return  $X'$ 
25) end

```

Fig. 5. Calculating optimal node positions in multiple flows.

is to have each node constantly adjust its position according to the positions of its neighbors in the flow. In Fig. 8, for example, a node chooses as its next target position the center of the straight line between its two neighbors. It is shown in [13] that the final positions of the nodes will eventually converge to their optimal positions on the straight line. We extend that algorithm by taking mobility

cost into consideration: A node periodically calculates the mobility benefit (transmission energy consumption reduction or node lifetime increase) and the mobility cost, based on its current position and the next target position. The algorithm is shown in Fig. 6.

To make mobility decisions regarding system-wide metrics such as the total energy consumption or system

```

1) Algorithm LocalizedAlgorithmForFixedFlows on node  $n_i$ :
2)
3) on receive flow message  $m$ :
4)    $f = \text{thisNode.getFlow}(m.\text{flowid})$ 
5)   if  $m.\text{destination} = \text{thisNode}$  then
6)     UpdateMobilityStatus( $m$ )
7)     /* updating mobility status and sending notification to flow source if necessary */
8)   else begin
9)      $x' = \text{GetNextPosition}(f)$ ; /* calculating next position */
10)    GetMobilityPerformance( $f, m, x, x'$ )
11)    /* obtaining expected energy performance with different mobility strategies */
12)    AggregateMobilityPerformance( $m$ )
13)    /* updating expected aggregate information in the message */
14)     $f.\text{mobility\_enabled} = m.\text{mobility\_enabled}$ ; /* updating local mobility status */
15)    if  $m.\text{mobility\_enabled}$  then move towards  $x'$ 
16)    else stop moving
17)    forward  $m$  to  $f.\text{next}$ 
18)  end
19) procedure UpdateMobilityStatus(Message  $m$ )
20) begin
21)    $f = \text{thisNode.getFlow}(m.\text{flowid})$ 
22)   if  $m.\text{aggregate\_benefit} < m.\text{aggregate\_cost}$  and  $f.\text{mobility\_enabled}$  then
23)     send mobility disable notification to  $m.\text{source}$ 
24)   else if  $m.\text{aggregate\_benefit} > m.\text{aggregate\_cost}$  and not  $f.\text{mobility\_enabled}$  then
25)     send mobility enable notification to  $m.\text{source}$ 
26) end

```

Fig. 6. The localized algorithm for fixed flows.

```

1) function GetNextPosition (Flow  $f$ ) : Position
2) begin
3)   return  $(f.prev.x + f.next.x)/2$ 
4) end
5)
6) procedure GetMobilityPerformance (Flow  $f$ , inout Message  $m$ , Position  $x$ , Position  $x'$ )
7) begin
8)    $m.benefit = E_T(|x - f.next.x|, \lambda_f) - E_T(|x' - f.next.x|, \lambda_f)$ 
9)    $m.cost = E_M(|x - x'|)$ 
10) end
11)
12) procedure AggregateMobilityPerformance (inout Message  $m$ )
13) begin
14)    $m.aggregate\_benefit+ = m.benefit$ 
15)    $m.aggregate\_cost+ = m.cost$ 
16) end

```

Fig. 7. Minimizing total energy consumption using the generic algorithm for fixed flows.

lifetime, it is necessary to aggregate the mobility benefit and mobility cost at individual nodes. To do so, the algorithm exploits the characteristics of flow-based communication: Messages sent from the source to the destination pass through all intermediate nodes. During message propagation, the mobility benefit and cost at each node can be aggregated and placed in a message header. The destination node thus obtains the aggregated mobility benefit and cost of the flow. If the benefit is greater than the cost, then the nodes should move to their next target positions; otherwise they should maintain their current positions. The destination records the mobility status (to move or not to move) of the flow. In the case of a status change, it sends a notification message back to the source. The source places the current mobility status in message headers so that the flow nodes can obtain the status and enforce mobility accordingly.

This algorithm is “generic” in the sense that it can be applied to various informed mobility problems. Next, we will show how to use the generic algorithm to minimize total energy consumption or maximize system lifetime. We note the proposed algorithm is an approximation of the global algorithm described in Section 3. However, the simulation results in Section 6 show the energy performance improvement is significant even though the localized approximation does not always produce an optimal solution.

4.1.1 Minimizing Total Energy Consumption

To minimize total energy consumption, it is necessary to calculate next node positions, estimate the expected transmission energy consumption at the current and next positions, estimate the expected energy consumption for moving to next positions, and obtain the total flow energy consumption at the destination. Specifically, each node should implement the functions *GetNextPosition()*, *GetMobilityPerformance()*, and *AggregateMobilityPerformance()* in the generic algorithm. The function implementations are shown in Fig. 7.

As shown in Fig. 8, a node chooses as its next position the center of the straight line between the previous and next nodes in the flow path. As described in Section 5, this information is locally available. The total energy consumption of the flow can be collected using summation

operation. We set the benefit to be the difference between transmission energy consumption at the current and next positions, and the cost to be the energy consumption for moving to next positions.

4.1.2 Maximizing System Lifetime

To maximize system lifetime, it is necessary to calculate next node positions, estimate expected node lifetime at the current and next positions, and obtain the minimum in-flow node lifetime at the destination. The next node position can be calculated locally using an approximation as described in Section 2.2. The approximation is based on Theorem 2 as well as the transmission power model. The basic idea is to move each node n_i from x_i to a new location x'_i on the line between x_{i-1} and x_{i+1} . The distance $d'_{i-1} = |x_{i-1} - x'_i|$ as well as the distance $d'_i = |x'_i - x_{i+1}|$ is calculated using the location and residual energy information of the previous and the current node. As described in Section 5, all such information is locally available. The calculation is based on the following equations:

$$d'_{i-1} + d'_i = D = |x_{i-1} - x_{i+1}|, \frac{a + b(d'_{i-1})^\alpha}{a + b(d'_i)^\alpha} = \frac{e_{i-1}}{e_i}.$$

The first equation ensures that the new location is on the line between the previous and the next nodes, and the second equation is derived from Theorem 2 directly. As described in Section 2.2, closed-form solutions for d'_{i-1} and d'_i are very complicated or even unattainable for $\alpha > 2$. Hence, we use a simple approximation:

$$\frac{(d'_{i-1})^\alpha}{(d'_i)^\alpha} = \frac{e_{i-1}}{e_i}, \text{ so } x'_i = x_{i-1} + (x_{i+1} - x_{i-1}) \cdot \frac{(e_{i-1})^{\frac{1}{\alpha}}}{(e_{i-1})^{\frac{1}{\alpha}} + (e_i)^{\frac{1}{\alpha}}}.$$

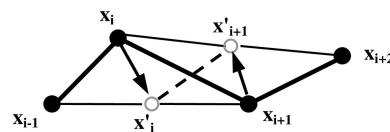


Fig. 8. Calculating the new node location that minimizes total energy consumption.

```

1) function GetNextPosition (Flow  $f$ ) : Position
2) begin
3)   return  $f.prev.x + (f.next.x - f.prev.x) \cdot \frac{(f.prev.e)^{\frac{1}{\alpha}}}{(f.prev.e)^{\frac{1}{\alpha}} + (e)^{\frac{1}{\alpha}}}$ 
4) end
5)
6) procedure GetMobilityPerformance (Flow  $f$ , inout Message  $m$ , Position  $x$ , Position  $x'$ )
7) begin
8)    $m.benefit = \frac{e - E_M(|x - x'|)}{E_T(|x' - f.next.x|, 1)}$ 
9)    $m.cost = \frac{e}{E_T(|x - f.next.x|, 1)}$ 
10) end
11)
12) procedure AggregateMobilityPerformance (inout Message  $m$ )
13) begin
14)    $m.aggregate.benefit = \min(m.aggregate.benefit, m.benefit)$ 
15)    $m.aggregate.cost = \min(m.aggregate.cost, m.cost)$ 
16) end

```

Fig. 9. Maximizing system lifetime using the generic algorithm for fixed flows.

The accuracy of this approximation depends on the relative weight between a and $b(d_i')^\alpha$. In Section 6, we show $b(d_i')^\alpha$ may be much larger than a in real environments, where this approximation is effective in increasing system lifetime.

The minimum lifetime among nodes on the flow path can be collected using the above aggregation procedure, replacing the summation operation with the minimum operation. To make the mobility benefit and cost comparable, we set the benefit to be the system lifetime with mobility, and set the cost to be the system lifetime without mobility. The corresponding function implementations are shown in Fig. 9.

4.2 Solutions for Flows with Dynamic Composition

Next, we describe the localization of the algorithms that change the composition of the flow by inserting new nodes. As described in Section 5, a flow node n_i periodically broadcasts the position and residual energy information of itself as well as its neighbors (n_{i-1}, n_{i+1}) in the flow. A nearby nonflow node n_j overhearing this message calculates the mobility benefit and the cost of joining this flow, with the expected position of n_j calculated using the function *GetNextPosition()* introduced in Fig. 7 or Fig. 9. If the expected gain is greater than the cost, then n_j sends the benefit and cost value to n_i , which may have received similar information from other nonflow nodes. Node n_i selects the node n_k that maximizes the mobility gain, and tentatively attaches that node to the flow. Node n_k periodically checks the mobility benefit and cost of joining the flow. If the mobility benefit is greater than the cost, n_k continues to move toward its expected position; otherwise it stops. When n_k becomes close enough to n_{i-1} and n_i so that it is beneficial for n_k to join the flow, n_k formally inserts itself into the flow between n_{i-1} and n_i . A generic algorithm is shown in Fig. 10. To use the algorithm, an application implements the function *GetMobilityInsertionPerformance()* to determine insertion benefit and cost, as well as the function *CompareMobilityStrategies()* to select better joining nodes. The function implementations for minimizing total energy consumption and maximizing system lifetime are shown in Figs. 11 and 12, respectively.

5 THE iMOBIF FRAMEWORK

We are now ready to describe iMobif, a practical framework for energy optimization under informed mobility. iMobif collects necessary node positions and residual energy information, and passes the information to the algorithms described in Section 4, which determine the proper mobility strategy.

5.1 Overview

In designing the iMobif framework, we make the following assumptions:

1. nodes are energy constrained,
2. nodes are mobile,
3. node movement consumes node energy,
4. nodes can select a transmission power level close to a specified value and use this power level for transmission,
5. nodes can measure residual node energy,
6. nodes can detect their locations,
7. nodes can move to location specified by software applications and protocols,
8. nodes can measure (or estimate from historical data) the energy needed to move to a target location, and
9. nodes can determine the minimum transmission power needed to communicate with nodes within a specific distance.

Assumptions 1, 2, 3, and 4 specify the target environment of iMobif: mobile sensor or mobile ad hoc networks where nodes are battery-driven and transmission power is tunable. Assumption 5 holds for most nodes in such environment. Assumptions 6 and 7 require each node to be equipped with GPS or other positioning devices/algorithms [17]. Assumptions 8 and 9 are usually not supported by hardware. iMobif establishes the energy-distance relationship to provide required information.

Each iMobif node maintains the following information:

1. a list of mobility strategies and their corresponding functions,
2. the current location and residual energy of the node,


```

1 ) Algorithm LocalizedAlgorithmForDynamicFlows on node  $n_i$ :
2 )
3 ) on receive hello message  $m$  from  $n_j$ :
4 )   for each flow  $f$  in  $m.flows$  do begin
5 )     if  $thisNode.isFlowMember(f)$  then continue
6 )      $f.next = n_j$ ; /* adjust flow data for position calculation */
7 )      $x' = GetNextPosition(f)$  /* calculating attach position */
8 )      $GetMobilityInsertionPerformance(f, m, x, x')$ ; /* calculating join benefit/cost */
9 )     if  $m.benefit > m.cost$  then
10 )       if  $thisNode.isAttached(f)$  then move towards  $x'$ 
11 )       else send join request to  $f.next$  with  $m.benefit$  and  $m.cost$  values
12 )     else stop moving
13 )     if  $thisNode.isAttached(f)$  then begin
14 )        $GetMobilityInsertionPerformance(f, m, x, x)$ 
15 )       /* check if it is optimal to join an attached flow */
16 )       if  $m.benefit > m.cost$  then begin
17 )         send flow join message to  $f.prev$  and  $f.next$ 
18 )         join flow  $f$  between  $f.prev$  and  $f.next$ 
19 )       break
20 )     end
21 )   end
22 ) end
23 )
24 ) on receive join request message  $m$  from  $n_j$ :
25 )   if  $thisNode.isFlowMember(m.flowid)$  then begin
26 )      $f = thisNode.getFlow(m.flowid)$ 
27 )     /* choosing the node that maximizes mobility gain to join */
28 )     if  $f.attached$  is null then begin
29 )       send attach message to  $n_j$ 
30 )        $f.attached = m$ 
31 )     end
32 )     else if  $CompareMobilityStrategies(m.benefit, m.cost, f.attached.benefit, f.attached.cost)$ 
33 )      $> 0$  then begin
34 )       send detach message to currently attached node  $f.attached.source$ 
35 )       send attach message to  $n_j$ 
36 )        $f.attached = m$ 
37 )     end
38 )   end

```

Fig. 10. The localized algorithm for dynamic flows.

3. a neighbor table with the identity, location, and residual energy of each neighbor,
4. a routing table, and
5. a flow table containing, for each flow traversing the node, its source, number of total data bits, previous node, mobility strategy and status, destination, and next node.

According to Assumptions 5 and 6 above, a node can measure its current location and residual energy. Such information is placed in the corresponding tables. The

routing table is managed by lower level routing protocols. In protocols such as AODV [18], for instance, each node periodically sends “hello” messages to probe and collect information from neighbors. In iMobif, a node in the system embeds in these “hello” messages its location, residual energy, and local flow table information, so that each node can obtain such information regarding its neighbors.

In the iMobif framework, the source of each flow determines current mobility strategy and status (enabled or disabled). The source selects a strategy from a mobility

```

1 ) procedure GetMobilityInsertionPerformance (Flow  $f$ , inout Message  $m$ , Position  $x$ , Position  $x'$ )
2 ) begin
3 )    $m.benefit = E_T(|f.prev.x - f.next.x|, \lambda_f) - E_T(|f.prev.x - x'|, \lambda_f) - E_T(|x' - f.next.x|, \lambda_f)$ 
4 )    $m.cost = E_M(|x - x'|)$ 
5 ) end
6 )
7 ) function CompareMobilityStrategies
   (Number  $benefit0$ , Number  $cost0$ , Number  $benefit1$ , Number  $cost1$ ) : Integer
8 ) begin
9 )   return  $(benefit0 - cost0) - (benefit1 - cost1)$ 
10 ) end

```

Fig. 11. Minimizing total energy consumption using the generic algorithm for dynamic flows.

```

1 ) procedure GetMobilityInsertionPerformance (Flow  $f$ , inout Message  $m$ , Position  $x$ , Position  $x'$ )
2 ) begin
3 )    $m.benefit = \min(\frac{e-E_M(|x-x'|)}{E_T(|x'-f.next.x|,1)}, \frac{f.prev.e}{E_T(|f.prev.x-x'|,1)})$ 
4 )    $m.cost = \frac{f.prev.e}{E_T(|f.prev.x-f.next.x|,1)}$ 
5 ) end
6 )
7 ) function CompareMobilityStrategies
   (Number  $benefit0$ , Number  $cost0$ , Number  $benefit1$ , Number  $cost1$ ) : Integer
8 ) begin
9 )   return  $\max(benefit0, cost0) - \max(benefit1, cost1)$ 
10) end

```

Fig. 12. Maximizing system lifetime using the generic algorithm for dynamic flows.

strategy list and enables or disables it according to feedback from the destination. The initial mobility status is both application and environment-specific. The iMobif framework is general in that it can be tuned to different energy optimization goals by changing the mobility strategy and the corresponding functions for node positioning, cost-benefit calculation, and information aggregation.

5.2 Operations

As described earlier, the flow source disseminates mobility strategy and status to other nodes along the flow path using message header of data packets. In addition, a source disseminates another flow parameter, the expected residual flow length, measured in data bits. The flow length information is used to calculate the benefit of the mobility, since a change in node location affects energy consumption of all subsequent packets. The flow length estimate is provided by the application.

As described above, node mobility in iMobif framework is triggered by data packets in a packet-by-packet fashion. The evaluation of overall performance is also packet-by-packet. This scheme ensures that the mobility is naturally synchronized without additional synchronization overhead. On the other hand, it implies that there may be oscillations if the cost-benefit combination produces disparate results between successive steps, resulting in a large number of notification packets. Although not formally proved, our simulation results show that only a few notification packets are sent for most flows, implying that the cost-benefit combination is consistent between successive steps.

In addition to processing data packets and sending “hello” messages, iMobif also sends back-to-back probing packets to neighbors to determine energy-distance relationship. The probing packets are sent with decreasing transmission power, with the value of transmission power embedded in the packet. A neighbor overhearing these packets replies with an acknowledgement packet indicating the lowest transmission power of the probing packets it has received. Based on such feedback from different neighbors, a node can establish an energy-distance lookup table and derive the energy-distance relationship and the parameter α . This operation can be conducted with low frequency since channel characteristics are usually static.

6 PERFORMANCE EVALUATION

In this section, we present evaluation results of the iMobif framework. First, we describe the simulation setup,

followed by simulation results for total energy consumption minimization and system lifetime maximization, respectively. Results for situations with and without topology constraints are presented, and the performance of global and localized algorithms are compared. Last, we describe results that shed light on the internal processing of the framework, and evaluate the framework under practical energy constraints. We consider only a single one-to-one flow in this section, and leave the evaluation of multiple flows as future work.

6.1 Simulation Setup

We randomly distribute 100 nodes in a 100m \times 100m area. The communication range of each node is set to 20m. The resultant average number of neighbors per node is approximately 12. We randomly select two nodes as the source and destination of the flow. The network uses greedy routing to forward packets from the source to the destination. We randomly set the initial residual energy of each node and the flow length; details are described in subsequent sections. As described in Section 2.1, we use the transmission power model $P_T(d) = a + bd^\alpha$. We adopt parameter values similar to those in [13]. Specifically, we set $a = 10^{-7}$ J/bit, $b = 10^{-10}$ Jm $^{-\alpha}$ /bit, and vary the value of α . We use the mobility cost model $E_M(d, l) = kd$, and vary the value of k . The maximum distance traveled is 1m in each step. The flow rate is set to 1KBps (8Kbps). In the simulations using iMobif, node mobility is initially disabled.

6.2 Minimizing Total Energy Consumption

Figs. 13a and 13b show the location of the nodes in a typical flow before transmission and after the mobility strategy that is used to minimize total energy consumption reaches a steady state. The size of the circle representing each node is proportional to its residual energy. The figures show the iMobif framework is capable of reaching the optimal configuration using the localized approach. For this mobility strategy, the node location is independent of its residual energy.

In Fig. 14, we compare the total energy consumption of six approaches: an approach without node mobility, an approach with only cost-unaware mobility, the two approaches using the iMobif framework, which are both benefit and cost-aware, and the two approaches using global algorithms. The static approaches do not change the composition of the flow, while the dynamic approaches may add new nodes to a flow path. We use the approach without mobility as the baseline, and define *energy consumption ratio* for the other five approaches

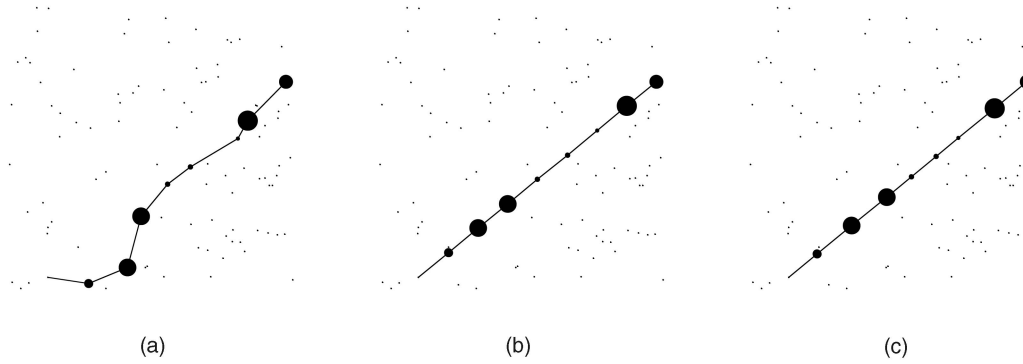


Fig. 13. The effect of controlled mobility on a wireless network. The size of a node is proportional to its residual energy. (a) Original. (b) The effect of the strategy that minimizes total energy consumption. (c) The effect of the strategy that maximizes system lifetime.

as the ratio of the total energy consumed to the energy consumed in the baseline approach. We expect the energy consumption of the five controlled-mobility approaches to be lower than that of the baseline approach. In other words, we expect the energy consumption ratio to be less than 1. In addition, we expect the energy consumption ratio of the approaches using global algorithms to be less than that of iMobif approaches.

In the following tests, the flow length is exponentially distributed with mean 100KB and 10MB, respectively. The mobility cost parameter k is set to 0.1, 0.5, and 1 J/m, respectively. Considering sensor nodes are usually small and light, such energy consumption level for node mobility is feasible. The transmission power parameter α is set to 2 and 3, respectively. We randomly generate 100 flows and collect the statistics.

First, we present the results for mobility without topology constraints. Fig. 14a shows that the energy consumption of the cost-unaware mobility approach is much higher than the baseline approach for short flows. As shown in Fig. 14b, this is due to the fact that the mobility cost is much higher than the transmission cost for short flows and, thus, the mobility benefit is not enough to cover the cost. For longer flows, the cost-unaware mobility approach can reduce energy consumption, as shown in Fig. 14e. However, in most cases, the total energy consumption of the approach with cost-unaware mobility is higher than the approach without mobility. This result shows it is important to take mobility cost into consideration for practical applications. On the other hand, Fig. 14 shows the approach using iMobif framework can achieve lower energy consumption than the approach without mobility for almost all flow instances. Even for flow instances that are long enough to cover the mobility cost, the performance of the iMobif approach is still comparable to that of the cost-unaware approach. This result indicates that the adverse impact of incorrect initial mobility status is limited. Fig. 14 also shows the dynamic-composition approaches that insert new nodes into flows can further reduce energy consumption significantly.

The figures show the performance of iMobif approaches is comparable to that of the approaches using global algorithms in most cases except in Fig. 14a, where the global dynamic-composition approach exhibits much better performance than its iMobif counterpart since the former has inserted much more new nodes into the flow. The latter has missed these candidates since node movement traces specified by the

algorithm are suboptimal in this case. As shown, a global approach has the advantage of reducing node movement distance as well as locating more beneficial nodes for insertions, as long as its computed positions are really “optimal.” The global algorithms described in Section 3 assume monotonic decrease of total energy consumption as nodes move to their “optimal” positions. Therefore, they only compare node energy consumption at the initial positions and the final positions, and make mobility decision based on the comparison results. Our simulation results show, however, the energy decreasing is not necessarily monotonic. An “optimal” global algorithm may need to explore much more combinations of node positions to find the solution, otherwise it may produce no better results than the localized approaches. Moreover, a global approach requires the information on the number of nodes in the flow, which may be dynamic. In addition, the approach assumes all intermediate nodes can move to their optimal positions, which may not be the case due to a topology constraint or lack of mobility at some nodes. Instead, a localized algorithm does not require information on the number of in-flow nodes, and can “elegantly” handle all the cases where in-flow nodes are fixed or flow composition has been changed. Therefore, we claim iMobif is an appropriate solution to the informed mobility problems.

Next, we present the results for mobility with the constraint that a node should not move outside of the transmission range of any of its current neighbors. Fig. 15 shows, under this mobility constraint, the controlled mobility approaches are slightly less effective for energy consumption reduction than in the case of no mobility constraint. Therefore, the space for improvement over the cost-unaware approach is smaller. However, iMobif framework still achieves lower energy consumption in most cases.

6.3 Maximizing System Lifetime

Fig. 13c shows the steady-state location of the nodes for the mobility strategy that maximizes system lifetime. Note that Fig. 13c is actually different from Fig. 13b although they appear similar. Clearly, the distance between a node and the next node in the path is dependent on the node residual energy, indicating that the iMobif framework is effective in placing nodes at their optimal positions using the distributed algorithm.

In Fig. 16, we compare the system lifetime of the three approaches with and without the topology constraint.

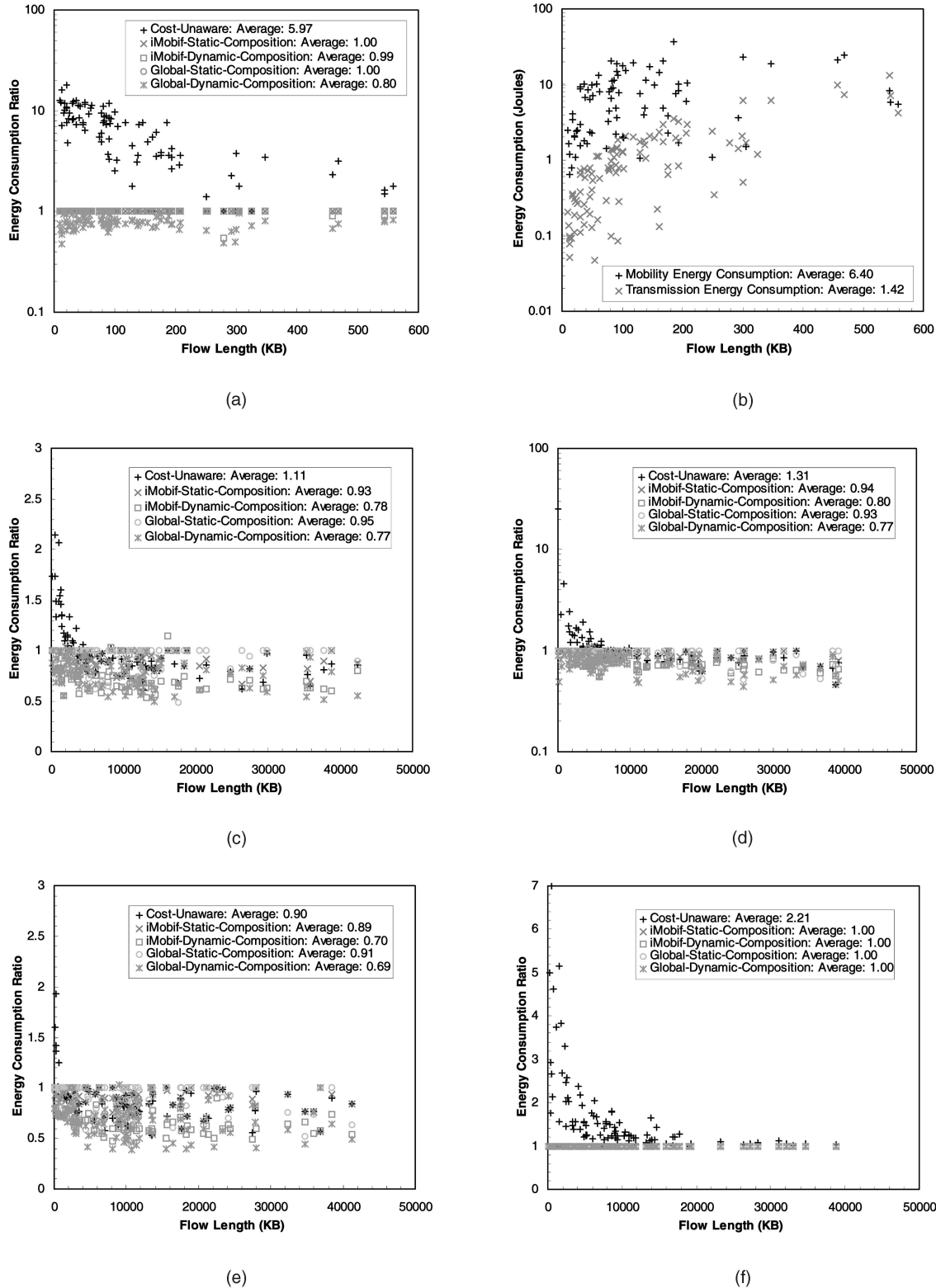


Fig. 14. Performance comparison of the approaches for energy consumption reduction without topology constraint. (a) $k = 0.5, \alpha = 3$, mean flow length 100KB. (b) Comparing mobility and transmission cost. (c) $k = 0.5, \alpha = 3$, mean flow length 10MB. (d) $k = 1, \alpha = 3$, mean flow length 10MB. (e) $k = 0.1, \alpha = 3$, mean flow length 10MB. (f) $k = 0.5, \alpha = 2$, mean flow length 10MB.

Again, we select the approach without mobility as the baseline approach, and define *system lifetime ratio* for the other three mobility approaches as the ratio of the system lifetime to that of the baseline approach. The flow length is exponentially distributed with mean 10MB, and the

mobility cost parameter k is set to 0.5. The transmission power parameter α is set to 3. The node residual energy is randomly chosen between 5 and 10 Joules (we intentionally set low residual energy to produce instances with short system lifetime). As Fig. 16 shows, the system lifetime of the

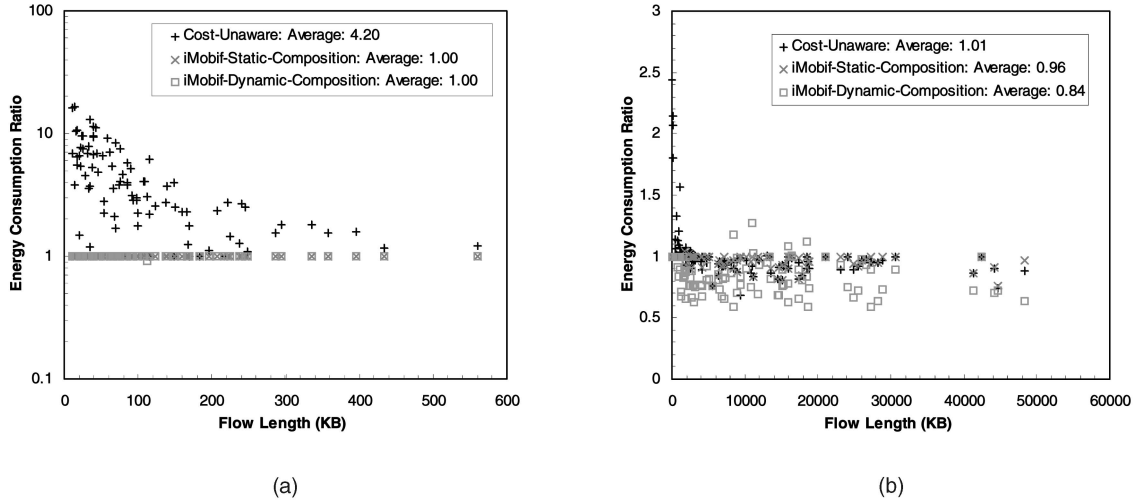


Fig. 15. Performance comparison of the approaches for energy consumption reduction with topology constraint. (a) $k = 0.5$, $\alpha = 3$, mean flow length 100KB. (b) $k = 0.5$, $\alpha = 3$, mean flow length 10MB.

approach with cost-unaware mobility is usually shorter than the approach without mobility, since it is likely that the bottleneck nodes spend too much energy in moving to new locations. On the other hand, Fig. 16 shows the approaches using the iMobif framework can achieve longer system lifetime than the approach without mobility for most flow instances. Although the average improvement is moderate, iMobif can increase system lifetime up to a factor of 2 for some flow instances with static flow composition, and up to a factor of 4 with dynamic flow composition. Even under the topology constraint, iMobif can extend system lifetime by factors of 1.6 and 3 for static and dynamic flow compositions, respectively. These results imply that the approximation algorithm is effective in reducing energy consumption at bottleneck nodes, thereby increasing system lifetime.

6.4 Notifications and Insertions

As described, iMobif uses a notification message sent from destination to source to change the status of the current mobility strategy. As shown in Fig. 17a, in the simulation

traces the number of notification messages is small, indicating the cost-benefit comparison results are fairly consistent, and there are few oscillations. Again the flow length in this test is exponentially distributed with mean 10MB. The mobility cost parameter k is set to 0.5, and the transmission power parameter α is set to 3.

The results in Sections 6.2 and 6.3 showed that adding new nodes to a flow path can significantly reduce energy consumption, or increase system lifetime, as desired. Next, we study how many insertions are needed to achieve such performance gains. We define *insertion ratio* as the ratio of the number of nodes added relative to the original flow hop count. Fig. 17b plots the CDF of the insertion ratio for 100 simulation runs without the topology constraint, with the parameters $k = 0.1$, $\alpha = 3$, and mean flow length 10MB. The insertion ratios for both aborted and completed insertions are shown. An insertion is aborted when a node is preliminarily attached to a flow, but never finishes joining the flow because the mobility cost becomes higher than the expected benefit. The figure shows many insertions are

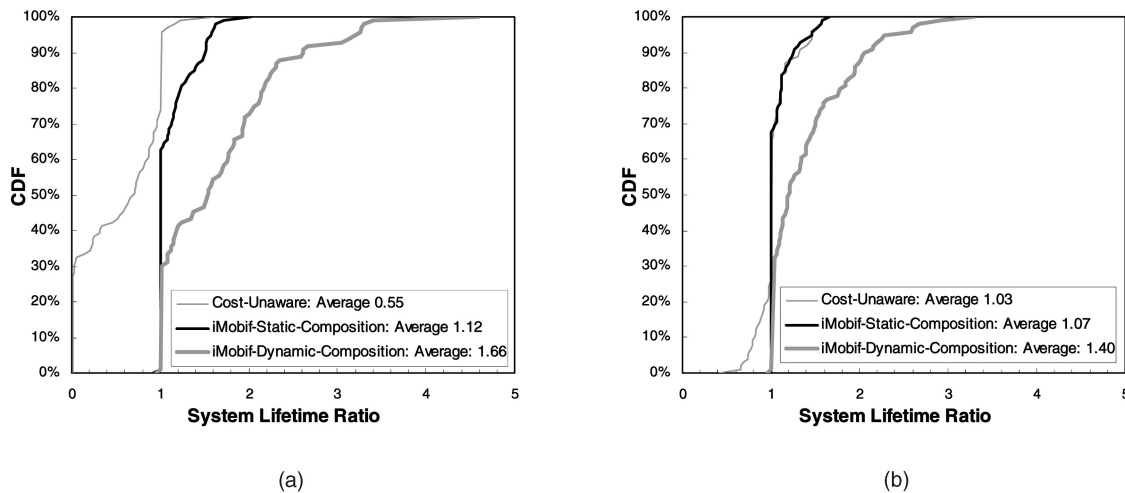


Fig. 16. Performance comparison of the approaches for system lifetime increasing. (a) Without topology constraint. (b) With topology constraint.

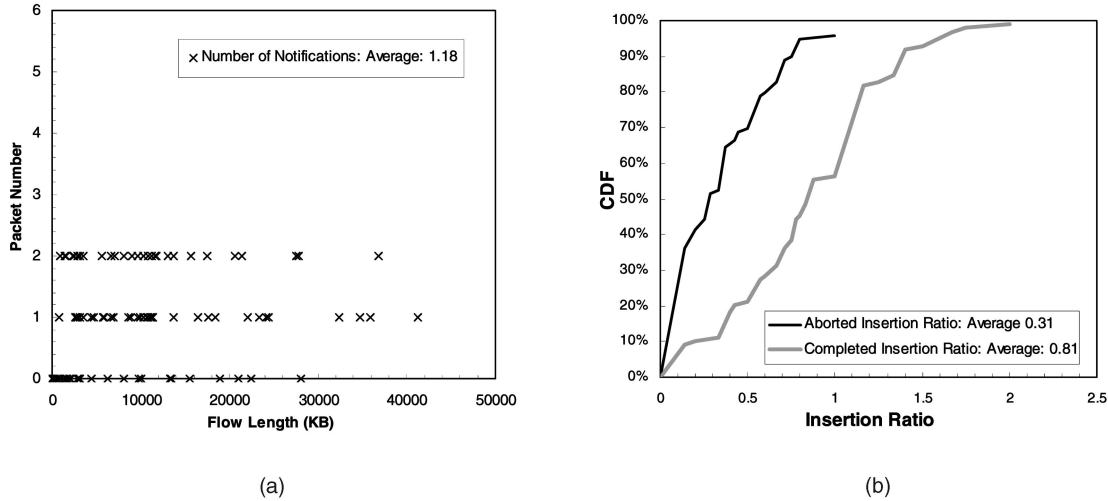


Fig. 17. The number of notifications and insertions. (a) The number of notification messages. (b) The number of inserted nodes.

aborted, preventing further performance loss but causing the previous movement to be wasted. However, the number of completed insertions is much higher, so the overall benefit is still high enough to counterbalance the mobility cost.

6.5 Impact of Practical Issues

In real systems the transmission power level may not be continuously tunable. It is more common that a node provides several levels of transmission power and applications select one of them for transmission. To study the impact of transmission power granularity on the performance of iMobif, we conduct simulation with a granularity value of 1 dBm. The iMobif algorithms compute the transmission power needed based on the propagation distance. The power value is rounded up to the next discrete level, which is used for transmission. Fig. 18a shows iMobif can achieve comparable energy consumption ratio under this realistic setting as the settings with continuously tunable transmission power. This result implies the energy consumption gap between different mobility strategies is usually substantial.

iMobif can reduce transmission energy consumption by adjusting transmission power level. In reality, nodes also consume a fixed amount of energy for receiving messages. To study total energy consumption ratio of iMobif, we set the node receiving power fixed at 10 mW so that the energy consumption for receiving is approximately the same as the energy consumption for transmission in the baseline approach. Fig. 18b shows the iMobif approach with fixed flow composition can still reduce total energy consumption, although with lower improvement. The approach with dynamic flow composition, however, may consume more energy than the baseline approach. This is due to the fact that inserted nodes introduce extra energy consumption for receiving. To solve this problem, we modify the function *GetMobilityInsertionPerformance()* in Fig. 11 to include the expected receiving energy consumption into the cost of node insertion. The modified approach, labeled as “iMobif-Dynamic-Receiving-Aware” in Fig. 18b, exhibits higher performance than the baseline approach. On the other hand, Fig. 18c shows iMobif approaches remain effective in maximizing system lifetime even if receiving energy consumption is considered. Extending the earlier analysis

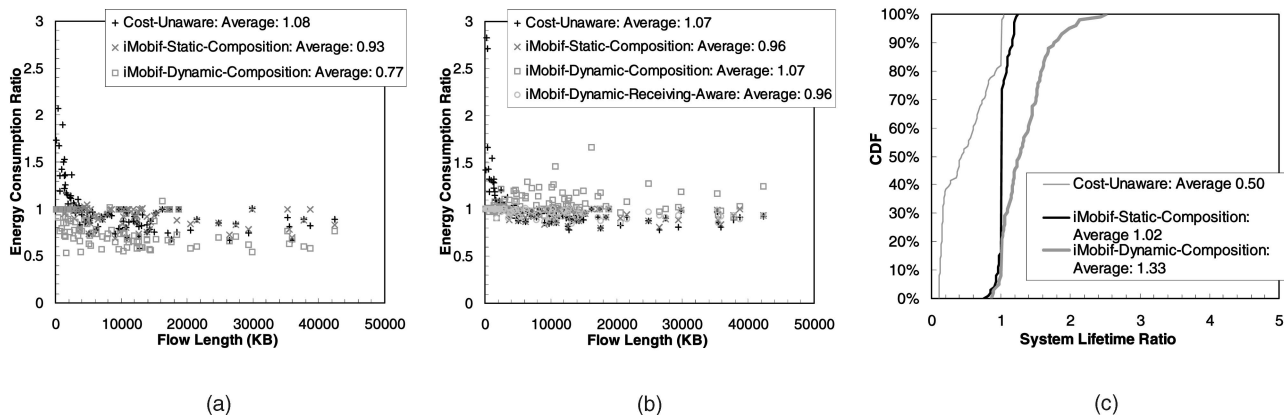


Fig. 18. Performance comparison of the approaches under practical considerations. (a) $k = 0.5$, $\alpha = 3$, mean flow length 10MB, discrete transmission power. (b) $k = 0.5$, $\alpha = 3$, mean flow length 10MB, energy consumption under nontrivial receiving energy. (c) $k = 0.5$, $\alpha = 3$, mean flow length 10MB, system lifetime under nontrivial receiving energy.

to include receiving energy consumption is a topic for future research.

7 RELATED WORK

Controlled mobility has been exploited to increase sensor surveillance coverage [11], [12], recover disconnected networks [10], and reduce energy consumption [13] in wireless ad hoc and sensor networks. In [10], the authors proposed an approach that uses controlled mobility to improve communication reliability, specifically, to reconnect a partitioned sensor network. In [11], [12], the authors proposed localized algorithms that increase sensor surveillance coverage by dispatching mobile sensor nodes to new locations not covered by existing sensor nodes. The localized node-insertion algorithm proposed in this paper is similar to the "auction"-based algorithms proposed in [12]. In addition to the above research, other research activities on controlled mobility in sensor networks are reported in [19]. However, most of these approaches do not account for mobility cost. In this paper, we consider energy consumption for both communication and mobility.

In [13], the authors proposed several approaches that adjust the network topology to reduce energy consumption. The application scenarios include single flow, multiple flows, and concat. The approaches are based on the observation that total energy consumption is minimized when all relaying nodes are evenly spaced on a straight line between the source and the destination. In these approaches, each relaying node moves to its ideal position to form the straight line with evenly spaced relays. The cost benefit trade-off is addressed and it is shown that the energy consumption of signal propagation is quadratic (or higher) in the propagation distance, so the energy savings due to changing the locations of relaying nodes can be super-linear in the distance. On the other hand, the mobility cost, or the energy consumption for node movement, is linear in the distance. Therefore, the benefit of node movement will eventually outweigh the mobility cost in term of energy consumption, provided the lifetime of the flow is long enough. The above work, however, does not incorporate the cost-benefit trade-off in the design of mobility strategies. iMobif uses similar distributed algorithms to optimize energy consumption using controlled mobility. However, iMobif also includes online cost-benefit comparisons, which cause the mobility strategy to be enabled or disabled dynamically so as to optimize system performance. In addition, iMobif provides algorithms for the insertion of new nodes into an existing flow, in order to further improve energy efficiency.

8 CONCLUSIONS AND FUTURE WORK

In this paper, we studied the informed mobility optimization problem. We designed localized algorithms to make mobility decisions using both mobility benefit and mobility cost. Based on these algorithms, we proposed iMobif, a framework to coordinate node mobility so as to optimize energy consumption in wireless ad hoc networks. The framework dynamically calculates the cost and benefit of a mobility strategy in a distributed fashion. The results of the

cost-benefit comparison are used to enable or disable the mobility strategy. We demonstrated how to integrate two different mobility strategies into the framework: minimizing total energy consumption and maximizing system lifetime. Simulation results show iMobif is effective in reducing total energy consumption or increase system lifetime, compared to both an approach without mobility and an approach using only cost-unaware mobility.

The performance of iMobif depends on the relative weight between mobility cost and communication energy consumption. iMobif would be most effective in environments where node mobility consumes little power while communication is heavyweight in terms of energy consumption. In a sensor network, one of the target environments of iMobif, node mobility may be lightweight since sensor nodes are usually small and light. Although a flow between two nodes in sensor networks is usually short, the flow resulting from aggregation (of multiple simultaneous flows between a given node pair) or concatenation (of multiple successive flows between this node pair over a long interval) may be large. iMobif can be extended to treat aggregated and/or concatenated flows as a single long flow, and estimate its length based on historical data.

iMobif adjusts node positions to optimize energy consumption. The resultant network topology depends on the composition of, and the spatial and temporal relationship among, flows. An interesting extension of this work is to study the characteristics of the resultant topology and its impact on higher level applications and lower level network protocols. On the other hand, iMobif adjusts node positions using a flow-by-flow approach. A strategy optimal to individual flows is not necessarily optimal to the flows as a whole. Designing globally optimal strategies is challenging. It would also be interesting to study the impact of topology constraint on the efficiency of iMobif. Finally, it would be interesting to study how to extend iMobif to satisfy the requirements of specific applications by using a combination of controlled mobility, transmission power adaptation, flow pattern inference, and mobility collaboration.

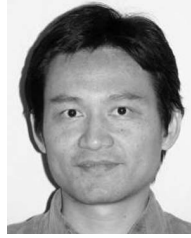
ACKNOWLEDGMENTS

A concise and preliminary version of this paper appeared in the *Proceedings of the Second International Workshop on Wireless Ad Hoc Networking (WWAN 2005)*, Columbus, Ohio, June 2005. This version contains more extensive descriptions of the original algorithms plus a new algorithm that optimizes node selection and node movement simultaneously. This work was conducted while C. Tang was a graduate student at Michigan State University. P.K. McKinley's work was supported in part by US Department of the Navy, Office of Naval Research, under Grant No. N00014-01-1-0744. His work was also supported in part by US National Science Foundation grants CCR-9912407, EIA-0000433, EIA-0130724, and ITR-0313142.

REFERENCES

- [1] S. Bandyopadhyay and E.J. Coyle, "An Energy Efficient Hierarchical Clustering Algorithm for Wireless Sensor Networks," *Proc. IEEE INFOCOM*, 2003.
- [2] A. Cerpa and D. Estrin, "Adaptive Self-Configuring Sensor Networks Topologies," *Proc. INFOCOM*, June 2002.

- [3] Y.-C. Tseng, Y.-N. Chang, and P.-H. Tseng, "Energy-Efficient Topology Control for Wireless Ad Hoc Sensor Networks," *Proc. Int'l Computer Symp. (ICS)*, 2002.
- [4] J. Pan, Y.T. Hou, L. Cai, Y. Shi, and S.X. Shen, "Topology Control for Wireless Sensor Networks," *Proc. Ninth Ann. Int'l Conf. Mobile Computing and Networking (MobiCom)*, Sept. 2003.
- [5] K. Dantu, M. Rahimi, H. Shah, S. Babel, A. Dhariwal, and G. Sukhatme, "Robomote: Enabling Mobility in Sensor Networks," Technical Report CRES-04-006, Dept. Computer Science, Univ. of Southern California, 2004.
- [6] J. Wu and F. Dai, "Mobility-Sensitive Topology Control in Mobile Ad Hoc Networks," *Proc. Int'l Parallel and Distributed Processing Symp. (IPDPS)*, 2004.
- [7] S. Chakraborty, D. Yau, and J. Lui, "On the Effectiveness of Movement Prediction to Reduce Energy Consumption in Wireless Comm. (Extended Abstract)," *Proc. SIGMETRICS*, June 2003.
- [8] M. Grossglauser and D. Tse, "Mobility Increases the Capacity of Ad-Hoc Wireless Networks," *Proc. INFOCOM*, pp. 1360-1369, Mar. 2000.
- [9] S. Capkun, J.-P. Hubaux, and L. Buttyan, "Mobility Helps Security in Ad Hoc Networks," *Proc. Fourth ACM Int'l Symp. Mobile Ad Hoc Networking and Computing (MobiHoc)*, June 2003.
- [10] T. Srinidhi, G. Sridhar, and V. Sridhar, "Topology Management in Ad Hoc Mobile Wireless Networks," *Proc. Real-Time Systems Symp., Work-in-Progress Session*, Dec. 2003.
- [11] G. Wang, G. Cao, and T.L. Porta, "Movement-Assisted Sensor Deployment," *Proc. IEEE INFOCOM*, Mar. 2004.
- [12] G. Wang, G. Cao, and T.L. Porta, "A Bidding Protocol for Deploying Mobile Sensors," *Proc. IEEE Int'l Conf. Network Protocols (ICNP)*, Nov. 2003.
- [13] D. Goldenberg, J. Lin, A.S. Morse, B. Rosen, and Y.R. Yang, "Towards Mobility as a Network Control Primitive," *Proc. MobiHoc*, 2004.
- [14] A. Chakrabarti, A. Sabharwal, and B. Aazhang, "Using Predictable Observer Mobility for Power Efficient Design of Sensor Networks," *Information Processing in Sensor Networks*, Apr. 2003.
- [15] S. Jain, R.C. Shah, G. Borriello, W. Brunette, and S. Roy, "Exploiting Mobility for Energy Efficient Data Collection in Sensor Networks," *Proc. Workshop Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt)*, Mar. 2004.
- [16] C. Tang and P.K. McKinley, "iMobif: An Informed Mobility Framework for Energy Optimization in Wireless Ad Hoc Networks," *Proc. Second Int'l Workshop Wireless Ad Hoc Networking (WWAN 2005)*, June 2005.
- [17] L. Hu and D. Evans, "Localization for Mobile Sensor Networks," *Proc. MobiCom*, Sept. 2004.
- [18] C.E. Perkins and E.M. Royer, "Ad Hoc On-Demand Distance Vector Routing," *Proc. Second IEEE Workshop Mobile Computing Systems and Applications*, pp. 90-100, Feb. 1999.
- [19] C. Tang, "Mobility-Assisted Sensing and Comm. in Sensor Networks," http://www.cse.msu.edu/~tangchip/publication/sensor_mobility.pdf, 2004.



Chiping Tang received the BS degree in computer science from Beijing University in 1992, and the MS and PhD degrees in computer science from Michigan State University in 2002 and 2005, respectively. His research interests include mobile computing and wireless network protocols, Internet computing, distributed algorithms, and collaborative applications. He is a member of the IEEE and the ACM.



Philip K. McKinley received the BS degree in mathematics and computer science from Iowa State University in 1982, the MS degree in computer science from Purdue University in 1983, and the PhD degree in computer science from the University of Illinois at Urbana-Champaign in 1989. Dr. McKinley is currently a professor in the Department of Computer Science at Michigan State University, where he has been with the faculty since 1990. He was a

member of technical staff at Bell Laboratories in Naperville, Illinois from 1982-1990, on leave of absence 1985-1989. He serves as an associate editor for *IEEE Transactions on Parallel and Distributed Systems* and was cochair of the program committee for the 2003 IEEE International Conference on Distributed Computing Systems. His current research interests include adaptive middleware, collaborative applications, mobile computing, and group communication protocols. He is a member of the IEEE and the ACM.

▷ **For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.**