On the Effectiveness of Movement Prediction To Reduce Energy Consumption in Wireless Communication

Srijan Chakraborty Yu Dong David K. Y. Yau John C. S. Lui

***Abstract*—Node movement can be exploited to reduce the energy consumption of wireless network communication. The strategy consists in delaying communication until a mobile node moves close to its *target* peer node, within an application-imposed deadline. We evaluate the performance of various heuristics that, based on the movement history of the mobile node, estimate an optimal time (in the sense of least energy use) of communication subject to the delay constraint. We evaluate the impact of node movement model, length of movement history maintained, allowable delay, single hop versus multiple hop communication, and size of data transfer on the energy consumption. We also present measurement results on an iPAQ pocket PC that quantify energy consumption in executing the prediction algorithms. Our results show that, with relatively simple and hence efficient prediction heuristics, energy savings in communication can significantly outweigh the energy expenses in executing the prediction algorithms. Moreover, it is possible to achieve robust system performance across diverse node movement models.**

Keywords: mobile computing, wireless networking, energy management, movement prediction.

I. Introduction

Limited battery power of mobile devices, e.g., laptops, handhelds, and sensors, is a major concern in wireless mobile computing. A lot of research, representing both sofware and hardware approaches, has been conducted to increase the battery lifetime of these devices. Network communication, in particular, can be quite energy-expensive and should be targeted for possible saving. In this paper, we aim to reduce the energy cost of communication by applications that are delay tolerant. Specifically, our techniques will be most useful if significant nodal movement is possible within a tolerable delay. For example, consider a system that tracks and monitors the body conditions of Tour de France cycling competition winner Lance Armstrong during one of the Tour stages. One can use a mobile device to monitor his heartbeat and blood sugar level and periodically transmit the data back to his coaches and supporting medical personnel. Note that the envisioned mobile monitoring system will have to be extremely light in weight and hence have very limited battery. Energy efficiency of the system is crucial. (On the other hand, the proposed solution will be of limited value to real-time applications with low delay constraints; e.g., a person walking while broadcasting real-time video from her portable camcorder.)

Our strategy to conserve energy in wireless communication is based on the observation that the reduction in physical distance between two communicating parties in a wireless network often results in reduced energy use. In the case of single hop communication, this is obvious if we use transmission power control. Since transmission power is roughly proportional to the square of the distance between two communicating nodes, reduced distance implies reduced transmission power requirement and, consequently, less energy consumption.

However, a more general scenario is one where transmission is multi-hop, especially in the case of ad hoc networks. In this case, decrease in physical distance between two nodes does not necessarily imply reduction in network distance (i.e., number of hops between the source and the destination). It also depends on the network state and node density of the network. However, if the network is not very sparse, it is then likely that reduction in physical distance between two nodes will result in reduced network distance also. Moreover, we can expect the length of the individual hops to be smaller. Both of the factors together can save communication cost in terms of energy use.

This brings us to the problem of predicting when two nodes will move closer to each other. We first consider the situation where a set of mobile nodes in an ad hoc network is communicating with a fixed node, henceforth mentioned as the *target*. Later, we relax this assumption of a fixed target and consider the case where all the nodes are mobile. Then, if we predict that a mobile node will move closer to the target, we can postpone communication until the future time subject to an application imposed deadline. For the movement prediction, we will first consider four basic heuristics based on simple statistics about past movement. Then, we will more formally approach our problem drawing analogy with the secretary problem, and develop the prediction heuristics accordingly. Our heuristics make use of the location history of a mobile node. Location information is used in many routing protocols [11], [19] to improve performance. Su *et al.* [17] use mobility prediction, based on location information, to improve the performance of their ad hoc routing protocol.

Research was supported in part by the National Science Foundation under grant numbers CCR-9875742 and CNS-0305496. An extended abstract of this paper was presented at ACM SIGMETRICS 2003, San Diego, CA, and appears in the conference proceedings [5]. S. Chakraborty is with Microsoft Corporation, Redmond, WA; work done while he was a graduate student at Purdue (email: `srijanc@microsoft.com`).

Y. Dong and D. K. Y. Yau are with the Department of Computer Science, Purdue University, West Lafayette, IN (emails: {`dong,yau`}`@cs.purdue.edu`).

J. C. S. Lui is with the Department of Computer Science and Engineering, The Chinese University of Hong Kong (email: `cslui@cse.cuhk.edu.hk`).

Such location information can be obtained using the Global Positioning System (GPS). We also investigate the issue of buffer requirements due to postponement of messages at the source nodes.

The amount of energy saving depends on various factors, e.g., the heuristics used, mobility model of the mobile nodes, length of movement history maintained, maximum allowable delay, data transmission duration, single hop or multi-hop transmission. We perform simulations to explore various combinations of these parameters and their performance in terms of the amount of energy saved. Our results show that considerable energy saving is possible with relatively simple and hence efficient heuristics.

A. Organization of the paper

The balance of the paper is organized as follows. The background of our work, including the system and mobility models and four basic movement prediction heuristics, is presented in Section II. Section III explains how we can use solutions to the well known secretary problem to solve our problem of network communication. Section IV introduces heuristics informed by the secretary problem for determining when to postpone a communication and when to communicate. Extensive simulation results and measurement results on an iPAQ pocket PC are presented in Section V to quantify the performance of our heuristics. Related work in discussed in Section VI. Section VII concludes.

II. Background

This section presents the background of our work. The system model and node mobility model are described. Four basic heuristics that exploit simple statistics about a node's movement history are also introduced.

A. System Model and Assumptions

Our system model is the following. We consider the case where mobile nodes in an ad hoc network are communicating with a stationary host, which we call the *target*[1]. The entire network is divided into virtual grids. We assume that each node knows its position using GPS and consequently can associate itself with a grid. We assume slotted time and that a node remembers its movement *history* as the grid IDs visited in the previous n time slots. We also assume that all nodes know the position y of the target.

B. Terminology

There are mainly three parameters used in our prediction strategy: the history length n, maximum allowable delay k for which a communication can be postponed, and a probability threshold p_{th}, when our decision is probability-based. Other than these parameters, we define the following terms. Let N be the total number of grids in the network. $S_h = \{x_1, x_2, ..., x_n\}$ is the sequence of n previous grid positions visited by the mobile node h, where x_i is the ith grid ID and x_n is the most recent grid ID visited in the sequence. A window $W(i, i+l-1) = \{x_i, x_{i+1}, ..., x_{i+l-1}\}$ $(1 \leq i \leq n-l+1)$ of size l for node h is a sequence of l grid IDs occurring consecutively in S_h. A window is thus a subsequence of S_h. We also define $d(i, j)$ as the Euclidean distance between two grid positions i and j. Lastly, for a statement Q, we define the truth function $T(Q)$ as:

$$T(Q) = \begin{cases} 1, & \text{if } Q \text{ is true.} \\ 0, & \text{if } Q \text{ is false.} \end{cases}$$

C. Mobility Model

What heuristic we should use may depend on the mobility model of the mobile nodes. A mobility model is a probabilistic process that defines the movement pattern of a mobile node, whereas a movement history is a *trajectory* or a *sample path* of the movement model. The formal definitions of both are given in [3].

If we assume a mobility model that follows the uniform random distribution, then the uncertainty of future locations is maximum. Consider the simple strategy where a mobile node always postpones any network communication if its current grid ID is not equal to y. Then in the next $k-1$ time units, the node communicates only if it is in the same grid ID as the target. If it does not move to the same grid ID as the target within the deadline, then it communicates at the kth time unit without waiting any further. The following theorem establishes the validity of the simple strategy:

Theorem 1: *The expected energy saving is always positive in the case of the uniform random mobility model using the above mentioned strategy.*

Proof: When the mobility model is uniform random, the probability that a mobile node will be in a grid ID, say i $(1 \leq i \leq N)$, at some time unit t is 1/N. Therefore, the probability that a mobile node will not be in the grid ID y in all the next k time units is $(1 - 1/N)^k$. Let C_i be the cost of network communication if the mobile node is in grid ID i, and C'_y be the expected cost of network communication if the mobile node is not in grid ID y, where y is the target position. Then,

$$C'_y = \sum_{i=1, i \neq y}^{N} \frac{1}{N} C_i.$$

We postpone any network communication only when the current grid ID is not equal to y. So when the current grid is y, applying our heuristic does not change the communication cost. The expected cost of communication, when the current location is not y, and when we do not consider postponing any communication, is

$$E_o = C'_y$$

On the other hand, the expected cost of communication when we apply our heuristic is

$$E_m = [1 - (\frac{N-1}{N})^k] C_y + (\frac{N-1}{N})^k C'_y$$

[1]Later in the paper, we will relax the fixed target assumption.

Hence, the expected energy saving or gain is

$$\begin{aligned} E_g &= E_o - E_m \\ &= [1 - (\frac{N-1}{N})^k](C'_y - C_y) \end{aligned}$$

Since C_y is the minimum among all C_i's, for $i = 1, 2, ..., N$, the second factor in the last equality is always positive. The first factor is also positive. Therefore, the expected gain in energy is always positive.

QED

However, the uniform random mobility model is not a realistic mobility model, since it does not preclude jumps, sharp stops and turns. In reality, mobile nodes show a more regular and smoother movement pattern. Many mobility models have been proposed which try to mimic realistic user movement patterns [9], [8], [4], [18], [20], [21]. As the mobility pattern of a mobile node becomes more regular, a good predictive algorithm should be able to predict the node's future movement with a higher level of confidence. For most of our experiments, we use the random waypoint mobility model, which is widely used in many ad hoc networking experiments in the literature. The model is originally proposed in [21] and later modified in [29] to ensure that the speed of nodal movement will converge to a reasonable non-zero steady-state value. In the model used for our experiments, we have incorporated these modifications [29] (please see Section V for the details). We have also devised an extension to the random waypoint mobility model, which we call the *regular waypoint* mobility model, that introduces a controlled degree of regularity in the movement pattern. In both of these mobility models, a mobile node starts at some random point in the network, chooses its next destination, moves steadily towards that destination, pauses for some time, again chooses a next destination, and so on. The difference between the regular and the random waypoint mobility models is in the way the next destination is chosen. While in the random waypoint model the next destination is chosen randomly, in the case of the regular waypoint model, we use the following Choose_next_destination algorithm.

First, for each mobile node we define two locations called *home* and *work* and a parameter called *regularity*, denoted by r ($0 \leq r \leq 100$). Each mobile node starts from its home grid, and depending on the value of r, it oscillates between the home and work grid positions. The regularity is the percentage of time the mobile node sticks to its movement pattern of home-work-home while choosing the next destination. When $r = 0$, it means that the movement model is the standard random waypoint model. On the other hand, if $r = 100$, the mobile node always alternates between its home and work grids without any diversions. Intermediate values of r give us various degrees of diversions. Initially, when a mobile node starts from home, it chooses the next destination as work with a probability equal to $r/100$. Once it reaches its work location, it chooses home as its next destination with a probability equal to $r/100$ and any other location with probability $1 - r/100$. We also define a maximum time limit, called $periodicity$, by which a node must choose either home or work as its next destination, if r is non-zero. The values of periodicity and r together decide how regular a movement pattern is. If we set the value of periodicity to be more than or equal to the history length, then the regularity of the movement pattern depends only on the value of r.

Algorithm `Choose_next_destination` shows how we choose the next destination for the regular waypoint model. In the algorithm, `previous_time` is the last time when either `home` or `work` grid was chosen as the `next_destination`. The flag `on_the_way_to` is either `home` or `work`, depending on whether the last visited grid among `home` and `work` is `work` or `home`, respectively.

```
Algorithm Choose_next_destination
if(current_time - previous_time > periodicity) {
    if(on_the_way_to == work) {
        next_destination = work;
        on_the_way_to = home;
    }
    else {
        next_destination = home;
        on_the_way_to = work;
    }

    previous_time = current_time;
}
else {
    random_number = generateRandomNumber
                mod 100;
    if(random_number < regularity) {
        if(on_the_way_to == work) {
            next_destination = work;
            on_the_way_to = home;
        }
        else {
            next_destination = home;
            on_the_way_to = work;
        }

        previous_time = current_time;
    }
    else
        next_destination = Random_destination
        _other_than_home_or_work;
}
```

D. Power Saving Strategy

Consider the situation where a mobile node wants to communicate with the target. We have the movement history of the node. Based on this history, we predict the future positions of the mobile node and decide whether waiting for at most k time slots will bring the node any closer to the target. If according to our predicted information, there is a fairly good chance that the mobile node will come closer to the target within k time units, the mobile node will postpone the communication. Then the node will wait for some appropriate time within the next k time slots, when it moves closer to the target, to carry out the communication. The node has to pick such a time slot for communication based on some heuristic. If, by chance, the node does not come any closer to the target in $k-1$ time units, the node will carry out the communication at the kth time unit without waiting any further. In that case, if the mobile node is further from the target than it was, it will likely end up using more energy. This is the misprediction penalty.

E. Choosing Parameters

The choice of various parameters affects the performance of our heuristics. The value of n should be chosen such that it incorporates the movement pattern of the mobile nodes. If too small, it will not have enough information to predict effectively. If too large, computational cost as well as time will adversely affect the performance. The value of k is application specific and should be chosen according to application deadline constraints. When our decision is based on some probability p, which quantifies the likelihood that the node will move closer to the target, we need to set a threshold value p_{th} such that we postpone the communication if $p \geq p_{th}$, or else we communicate immediately. The threshold value will mainly depend on the heuristic we use to calculate p.

F. Heuristics

This section gives the essentials of four basic heuristics based on simple statistics about a node's movement history.

1) **Binary distance (BD) heuristic**: We calculate the probability p that a mobile node will be in grid ID y (recall that y is the target's grid location) within the next k time units as follows:

$$\begin{aligned} p &= P(W(n+1, n+k) \text{ contains } y) \\ &= (\text{Number of windows in } S_h \text{ of size } k \\ &\quad \text{containing } y)/(n-k+1) \\ &= (\frac{1}{n-k+1}) \sum_{i=1}^{n-k+1} T(W(i, i+k-1) \\ &\quad \text{contains } y), \end{aligned}$$

where $(n-k+1)$ is the total number of windows in S_h of size k. If $p \geq p_{th}$, we decide to postpone communication with the target.

2) **Binary Markov distance (BMD) heuristic**: This heuristic uses an order-m Markov model for calculating the probability that a mobile node will be in grid ID y within the next k time units. The probability that $x_{n+1} = y$ is calculated as:

$$P(x_{n+1} = y) = \sum_{i_1}^{N} \sum_{i_2}^{N} \dots \sum_{i_m}^{N} P(x_{n+1} = y |$$

$$x_{n+1-m} = i_1, x_{n+2-m} = i_2, \dots, x_n = i_m)$$

Using this, we calculate the probability p that a mobile node will be in grid ID y within the next k time units. We wait if $p \geq p_{th}$, or else we communicate immediately.

3) **Markov distance (MD) heuristic**: The MD heuristic is a variation of the previous heuristic based on the Markov model. Let R be the set of all possible routes that can be taken by the mobile node in the next k time units, and let R_1, where $R_1 \subseteq R$, contain those routes in R that have at least one location closer to the target than the current distance. Then we calculate the probability that a mobile node will move closer to the target as:

$$p = \frac{\sum_{\rho \in R_1} \text{Probability of taking the route } \rho}{\sum_{\rho \in R} \text{Probability of taking the route } \rho}$$

If this probability is greater than or equal to p_{th}, communication is postponed. The same calculation can be repeated at each of the next $k-1$ time units, with the deadline decreased by one after each decision point. At any point in time, if $p < p_{th}$, the mobile node communicates with the target. If communication is postponed for all the $k-1$ time units, it is performed at the kth time unit.

4) **Average distance (AD) heuristic**: The AD heuristic is based on the weighted average distance between a mobile node and the target over all windows of size k in the mobile node's movement history. We calculate the average as follows:

$$average = (\frac{1}{\sum_{l=1}^{n-k+1} w_l}) \sum_{j=1}^{n-k+1} w_j \frac{1}{k} \sum_{i=j}^{j+k-1} d(x_i, y),$$

where w_j is the weight associated with window $W(j, j+k-1)$ $(1 \leq j \leq n-k+1)$. If the current distance between the mobile node and the target is greater than $average$, then the mobile node decides to postpone the communication, or else it communicates immediately. If it decides to wait, then in the next $k-1$ time units, whenever the current distance becomes not greater than $average$, communication is performed. If the node has postponed communication up till the kth time unit, then it performs communication at the kth time unit.

III. Applying the Secretary Problem

We use the heuristics to decide whether to postpone a communication until a future point in time or not. Once a node decides to postpone the communication, the next problem is to decide when within the next k time slots to communicate. There is one obvious procedure: the mobile node simply applies the same decision criterion for the next $k-1$ time units decreasing the deadline by one time unit at each decision step. If it decides to postpone communication in all these $k-1$ time slots, it finally communicates at the kth time slot.

Alternatively, the node can take the following approach. The problem of deciding when in the next k time units to communicate is analogous to the well known secretary problem. The secretary problem is the most common name for the sequential evaluation and selection problem in which one must make an irrevocable choice from a number of applicants whose values are revealed only sequentially. The simplest version of the secretary problem has the following characteristics [22]:

1) There is only one position available for a secretary.
2) The number of candidates N for the post is known.
3) The applicants are interviewed sequentially in random order, each order being equally likely.

4) The decision-maker can rank all the applicants from best to worst without ties.
5) The decision to reject or accept an applicant is solely on the relative ranking of those applicants interviewed so far.
6) An applicant once rejected cannot be later recalled.
7) The decision maker is satisfied with nothing but the very best.

In our case, at each time slot the node has the choice of either communicating with the target or postponing the communication. As in the secretary problem, we are presented with a sequence of choices one by one and once rejected a choice cannot be later recalled. The secretary problem is a well studied problem. It has many variations. One category of variations is based on the degree of knowledge about the population distribution from which candidates are drawn. In this category there can be three possible types:

1) No information.
2) Partial information, and
3) Full information.

In the no information case, the assumption is that the functional form of the distribution is completely unknown so that our decision is based solely on the relative ranks of choices. In the full information case, perfect information about the distribution is known. So the scores of choices as well as the relative ranks are also known. In the partial information case, the scores are drawn from a partially known distribution whose parameters are unknown *a priori* but can be estimated as we proceed with the evaluation.

Many optimum and heuristic solutions for different variations of the problem have been proposed [7]. Our problem is similar to the partial information secretary problem. We do not have the perfect information about the population distribution, which depends on the mobility model of the mobile nodes. However, neither are we completely ignorant – we have some information in the form of location history of a mobile node. The rest of this section discusses some solutions of the secretary problem and how we can apply those solutions in our case with suitable modifications.

A. Best-choice(r) Algorithm

One of the solutions for the secretary problem is given by the Best-choice(r) algorithm. It says: reject the first $r - 1$ candidates. Then accept the next candidate whose relative rank is 1 among the candidates seen till now. It can be shown that Best-choice($\frac{1}{e}$) is the best possible algorithm. This Best-choice($\frac{1}{e}$) algorithm accepts the best candidate with probability $\frac{1}{e} \approx 0.368$. This particular case is well known as the 37% rule [7]. However, we cannot apply this solution to our problem directly. Our problem of network communication differs from the secretary problem in several aspects. Unlike the secretary problem, we have some information about the population distribution (the distribution of distance, from a mobile node to the target, as a variable) in the form of a history of previous values (samples). We can also have ties, i.e., two distances can be the same. In our case, we can consider the number of candidates N as the number of time slots for which we have history information plus the number of time slots we can delay before communicating with the target. Since in most of the cases, the amount of delay we can incur is much less than 37% of the amount of history we have, we can adapt this 37% rule in our case as follows: We communicate at the first chance when the distance between the communicating node and the target is less than or equal to the least seen so far. Section IV presents this more formally as our Least Distance (LD) heuristic.

B. Single Threshold Selection Strategy

According to this strategy we select the first candidate whose value exceeds a prespecified threshold value v. This solution is applicable to the full information problem [23], where perfect information is known about the population distribution, from which the candidates are drawn. In our case, we do not have perfect knowledge about the population distribution. However, we can estimate some statistical parameter, e.g., average or median, from the history information we have. Then based on the estimated parameter we can set a threshold value such that we communicate as soon as we cross that threshold. Section IV proposes a single threshold heuristic based on average distance between a mobile node and the target.

C. One-bounce Rule

The one-bounce rule states that we should keep checking values as long as they go up. As soon as they go down we stop postponing any more and take the current value [24]. A direct analogy of this rule, that we apply to our problem, is to postpone as long as the distance between the mobile host and the target is decreasing, and communicate as soon as the distance starts increasing. However, this strategy ignores the history other than the last sample. Instead of using this strategy as a separate heuristic, we use this idea in conjunction with other heuristics as described in Section IV.

IV. Heuristics Informed by the Secretary Problem

Informed by the secretary problem, we are able to refine our basic heuristics to increase their effectiveness. Moreover, a new heuristic based on the 37% rule solution of the secretary problem can be devised. We present these further solutions in this section. In addition, we show how the prediction heuristics can be easily modified to effectively handle the more general case of a *moving* target.

A. Directional and recursive average distance (DAD and RAD) heuristics

The basic AD heuristic has been presented in Section II. It has very little computational overhead compared with the ones based on the Markov model. It also performs well across various movement models. There can be a problem with the basic AD heuristic. Once we decide to postpone a communication, we compare the *average distance* each time with the current distance between the communicating mobile node and the target. As long as the current distance is greater than the average, we keep postponing the communication for $k-1$ time slots. Finally, we communicate at the kth time slot, if we have postponed at all the $k-1$ time slots. Now, consider the situation when a mobile node is moving away from the target for all the k time slots. In this case, once we decide to postpone the communication at the beginning, we will keep postponing for the next $k-1$ time slots, since the distance between the mobile node and the target is increasing. In this scenario, the AD heuristic always chooses the worst case and communicates at the kth time slot.

To solve this problem we introduce a new heuristic, Directional Average Distance (DAD) heuristic, based on the AD heuristic that includes the direction of movement of a mobile node in the decision process. This heuristic is motivated by the one-bounce rule mentioned in Section III. Before comparing the current distance with the calculated average, we first check whether the node is moving away from the target. We postpone communication as long as the node is moving towards the target and the current distance is greater than the average. At any point of time, if we find that the node is moving away from the target, we communicate immediately irrespective of whether the current distance is less than the average or not. In this modification, we assume that when a node is going away from the target, it will not reverse its direction within the next k or less time slots. However, this is not guaranteed. The performance of DAD heuristic thus depends on the maximum allowable delay k.

Another modification of the basic AD heuristic is possible. Instead of calculating the average only once, we calculate the average at each of the $k-1$ time slots and we compare that new average with the current distance. We call this variation the *recursive* AD (RAD) heuristic.

B. Least Distance (LD) heuristic

As we will see in Section V, the LD heuristic is the *simplest and most effective* heuristic presented in this paper. The heuristic is based on the 37% rule as described in Section III-A. According to the 37% rule, the first 37% of the candidates are just evaluated, but not accepted. Then we take the candidate whose relative rank is the first among the candidates seen so far. In our case, we assume that, we have already seen the first 37% or more of the candidates as the location history. We first find the least distance d_{min} between a mobile node and the target in the history of that node:

$$d_{min} = Min_{\forall x \in S_h} d(x, y)$$

Then, in each of the next k time slots we check if the current distance d is less than or equal to d_{min}. At any time slot, if we find $d \leq d_{min}$, we communicate immediately, else we communicate at the kth time unit.

C. Heuristics in the case of Moving Target

Our assumption that there is a fixed target in the network with which all the other mobile nodes communicate may not present a realistic scenario for an ad hoc network. We now relax this assumption and discuss how the heuristics can be easily modified to effectively handle the case when both the communicating peers are mobile. We assume that the sender knows the synchronized location history of the destination with respect to itself. This is possible in the case of a routing protocol that periodically exchanges mobile node locations as routing information [1].

The required modifications to the heuristics to accommodate for a moving target are very simple: we just replace y, the location of the fixed target, with the location history of the moving target. Assume that a mobile node s with location history $S_s = x_1, x_2, \ldots, x_n$ wants to communicate with another mobile node r with location history $S_r = y_1, y_2, \ldots, y_n$. The MD heuristic requires no change other than that R and R_1 are now defined with respect to S_r instead of y.

The equation to calculate average distance for the AD heuristic is changed to the following:

$$average = \left(\frac{1}{\sum_{l=1}^{n-k+1} w_l}\right) \sum_{j=1}^{n-k+1} w_j \frac{1}{k} \sum_{i=j}^{j+k-1} d(x_i, y_i),$$

In case of the LD heuristic, the equation to calculate the minimum distance changes to:

$$d_{min} = Min_{i=1}^{n} d(x_i, y_i)$$

V. Experiments

To evaluate the performance of our heuristics we perform simulations and some measurement experiments on an iPAQ pocket PC. We begin by describing our simulation setup and results. We then present measurement results for energy consumption due to CPU processing on the iPAQ. Finally, we will conclude this section with a performance comparison of the proposed heuristics and their variations.

A. Mobile network simulations

This section presents the simulation setup and experimental results, which are classified according to the heuristic used. Along with the percentage energy saving, we evaluate the percentage decrease in average distance at the time of communication. The BD and BMD heuristics did not show significant energy saving in our preliminary experiments. Because of limited space, we do not further evaluate them in this paper; interested readers are referred to our technical report for the details. Here, we report simulation results for the MD, AD, and LD heuristics. We also discuss the impact of movement regularity, communication duration, and minimum speed of nodal movement on the energy saving, and how postponing communication will affect the communication delay and buffer requirement at the source node.

1) Simulation setup: To evaluate the performance of our heuristics, we use the ns-2 simulator [25] with the ad-hoc routing extensions contributed by CMU [21]. For all our ns-2 experiments, we first generate a node movement file using the CMU node-movement generation tool *setdest*. Then we use the node movement file to generate two traffic pattern files –

one using the standard protocol and the other after applying our heuristics. We use DSR [26] as our routing protocol and the standard *two-ray-ground* model as the propagation model. In experimenting with a mobile target, a communicating node can directly obtain location information about its target by accessing the target's node movement file. We have not currently implemented the actual exchange of location information between nodes through an enchanced routing protocol as suggested in Section IV-C.

All the experiments use the random waypoint mobility model modified according to the specifications in [29]. As discussed in [29], the original random waypoint model may produce unreliable simulation results because it fails to maintain a meaningful steady-state average speed of nodal movement. Instead, the average speed decreases over time, and becomes zero as the simulation time goes to infinity. To solve the problem, lower and upper bounds on the nodal speed, denoted as $speed_{min}$ and $speed_{max}$ respectively, are introduced. With the modifications, the average nodal speed at steady state can be shown to be about $\frac{speed_{max}-speed_{min}}{2}$.

In our experiments, we set the maximum nodal speed $speed_{max}$ to be 10 m/s and the minimum nodal speed $speed_{min}$ to be 1 m/s for both the random and regular waypoint models. Nodes move in a 150 m by 150 m area at speeds uniformly distributed between these bounds and with zero pause time. We have 10 nodes moving in the network and one node (the target) always fixed at the center.

Simulation traffic was generated by all the mobile nodes acting as continuous bit-rate (CBR) sources. The packet size was 1500 bytes and inter-packet interval was 0.005 seconds. We give each mobile node enough initial energy to run each simulation for 20,000 seconds. During these 20,000 seconds, each mobile node communicates with the target 1000 times, each time for a duration of 1 second. For all our heuristics, the delay k is measured in periods of 10 seconds; i.e., a delay of 10 means a 100 second delay. The transmission and reception energy values are chosen to reflect the values of a standard WaveLAN network card.

Each experiment is repeated in five independent runs. A reported data point is the average over the five runs. An error bar marks one standard deviation above and one below the average.

2) Experimental results for MD heuristic: Figure 1 presents the experimental results for the MD heuristic for the random and regular waypoint mobility models, for different p_{th} values. For these experiments, $m = 3$, and the value of k is 5. The regularity in case of the regular waypoint mobility model is 80%.

The percentage energy saving for the MD heuristic with the random waypoint model is up to 37.2%. In the case of the regular waypoint mobility model, the MD heuristic may achieve higher percentage energy saving in most cases. This is expected because the MD heuristic is designed to learn patterns in movement history. From these experimental results, we observe that the amount of energy saving increases as the value of p_{th} decreases. This is because a high probability threshold value may lose many opportunities for saving energy,

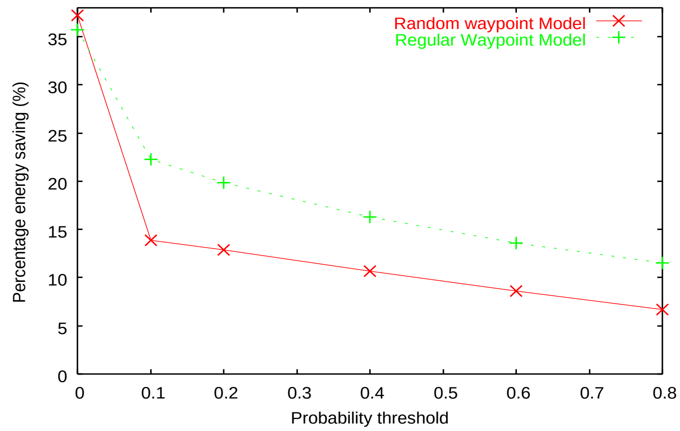

Fig. 1. Percentage energy saving as a function of p_{th} for MD heuristic.

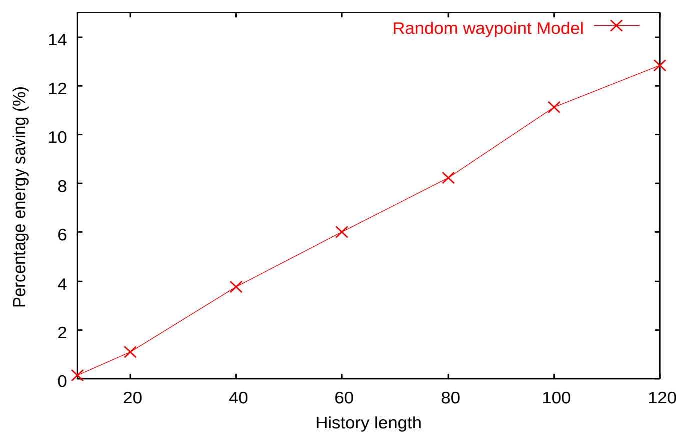

Fig. 2. Percentage energy saving as a function of history length for MD heuristic.

since we are being too conservative in trying to postpone communication.

In the case of the Markov model, once the order of the model is fixed, incorporating more history information does not require extra space. Thus, we can increase the history length to get better performance since, when the history length is larger, a node can learn its movement pattern better, which results in more energy saving. Figure 2 shows how energy saving increases with the increase of the history length in the case of the random waypoint mobility model. The value of k for this set of experiments is 5.

3) Experimental results for AD heuristic: Figure 3 shows the percentage energy saving with the AD heuristic for different values of k for both single hop and multi-hop transmissions. As we increase the value of k, a mobile node can save more energy since it now has more opportunities (i.e., a longer time) to move closer to the destination. From Figure 3, we can see that the percentage energy saving increases rapidly as we increase k initially. However, after some point it tends to stabilize. Figure 4 presents the percentage decrease in average distance for the same set of experiments.

As we can see from Figure 3, the percentage saving in energy is much more in the case of single hop communication.

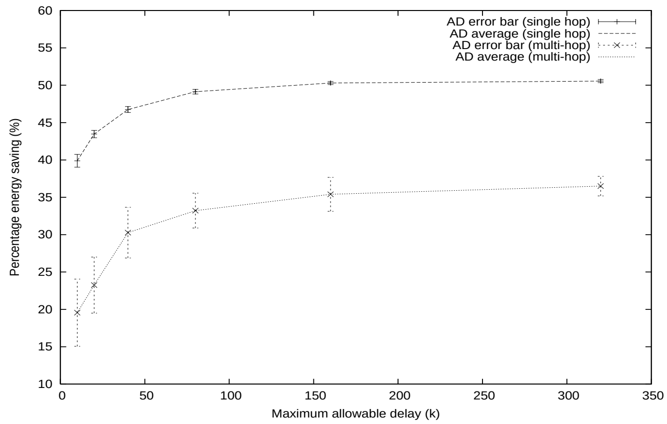

Fig. 3. Percentage energy saving as a function of k for AD heuristic.

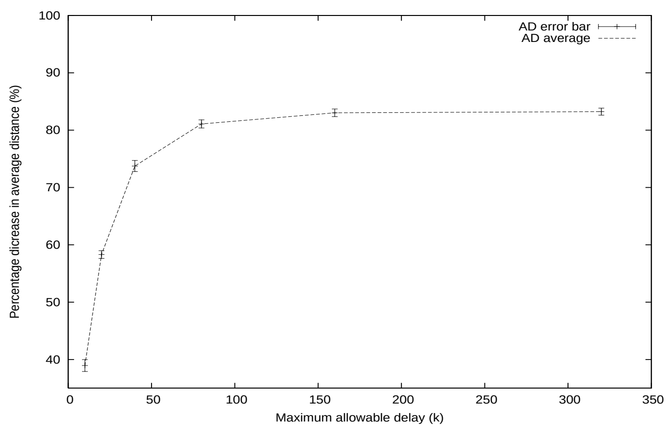

Fig. 4. Percentage decrease in average distance as a function of k for AD heuristic.

This is expected, because in the case of single hop communication, a smaller distance directly translates into reduced energy use due to transmission power control. However, the amount of saving in the case of multi-hop communication depends on factors such as the network state and node density. Multi-hop transmission is more realistic in the case of an ad-hoc network. For experiments with single-hop communication, we always use transmission power control; i.e., we calculate the required transmission power level for a node such that the target is just able to receive from this node. Here, notice that enough transmission power does not necessarily guarantee successful reception of the packet due to other factors, e.g., channel errors, may result in packet loss. For the multi-hop transmission experiments, however, we do not use transmission power control since we do not assume a mobile node always knows its distance from the neighbors. In the single hop case, however, the target is fixed and its position is available to all nodes. If transmission power control were included in the multi-hop case, we would expect the energy saving to be significantly larger. Hence, the reported energy savings in the experiments for multi-hop communication are conservative.

Experimental results for the basic as well as the variations of the AD heuristic are shown in Figure 5. For a smaller value

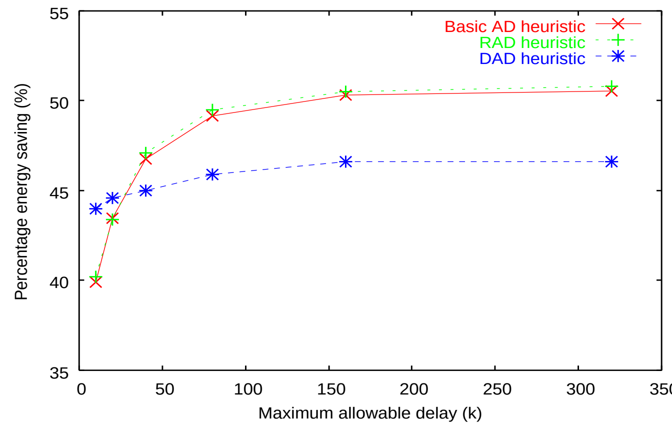

Fig. 5. Percentage energy saving as a function of k for AD, RAD and DAD heuristics.

of k (up to 35 periods), the DAD heuristic performs better than the basic AD and RAD heuristics, while the basic and RAD heuristics show similar performance. However, the DAD heuristic tends to perform less well than the basic one as the maximum allowable delay increases. This is because, as k increases, it becomes more likely that, even if a mobile node is moving away from the target at some point of time, it will revert back towards the target at some future point of time. In other words, in introducing the direction information, we assume that once a mobile node starts moving away from the target, it is not going to turn back before the deadline k. The probability of this assumption being true decreases as the value of k increases.

4) Experimental results for LD heuristic: This heuristic works very well and gives considerable energy savings. First, we present simulation results for this heuristic with the random waypoint mobility model as a function of the history length in Table I. The results show that, up to a certain point, increase in the history length improves performance. After that the amount of energy saving becomes stable. Also, the decrease in average distance is more significant than the energy saving, because our heuristics are designed to decrease this average distance at the time of communication.

Figure 6 shows how the LD heuristic performs with varying delay k both for single hop and multi-hop transmissions. As expected, with increased delay, the amount of energy saving increases. Figure 6 shows the same trend as the one with the AD heuristic. The rate of increase in percentage energy saving is high initially as we increase the value of k, but this rate gradually stabilizes. Figure 7 presents the percentage decrease in average distance for the same set of experiments. In Figure 7, we may find that when the value of k increases up to 40, the percentage decrease in average distance may reach 99%, which means that LD heuristic is always able to predict a closest position to the target when k is large enough.

5) Impact of Regularity: In the regular waypoint model, the regularity of the nodal movement can be controlled by the regularity parameter r. We study the impact of r on the heuristics in this section. Figure 8 and Figure 9 show the percentage energy saving and the percentage decrease in

History Length n	Percentage decrease in average distance	Percentage energy saving	Absolute energy saving (Joule)
10	77.65	52.9	4702
20	92.10	56	4970
40	97.99	57.7	5127
60	98.64	58	5155
80	98.73	58	5156
100	98.73	58	5156
120	98.73	58	5155

TABLE I
ENERGY SAVING USING THE LD HEURISTIC AS A FUNCTION OF HISTORY LENGTH.

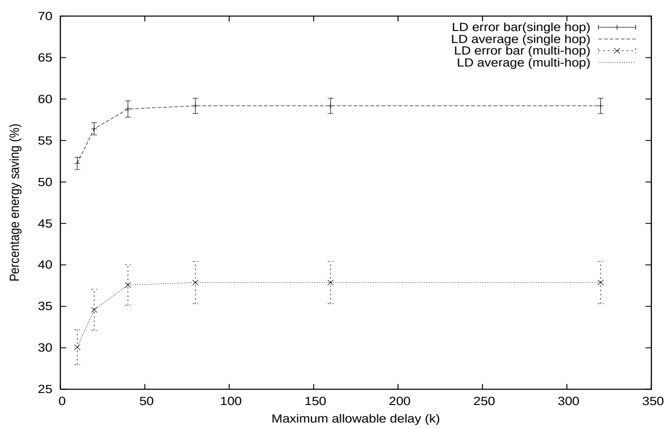

Fig. 6. Percentage energy saving as a function of k for LD heuristic.

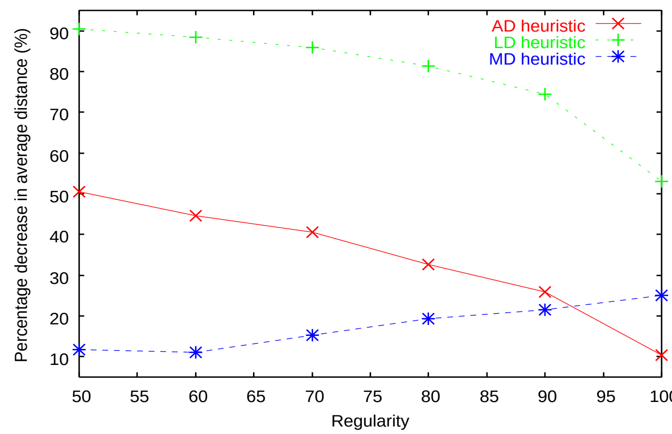

Fig. 8. Percentage decrease in average distance as a function of regularity for MD, AD and LD heuristics.

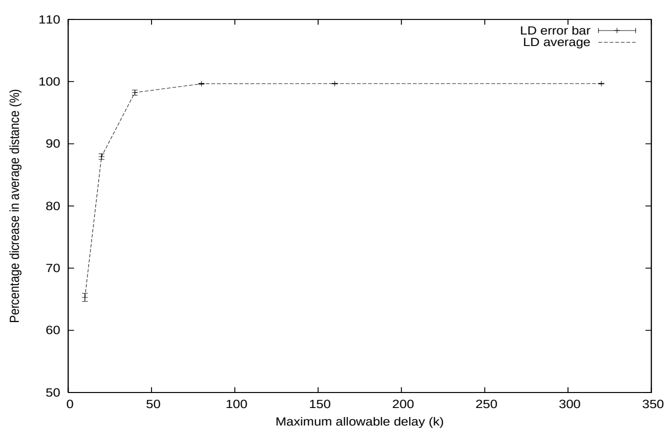

Fig. 7. Percentage decrease in average distance as a function of k for LD heuristic.

average distance, respectively, as we vary the regularity in the case of the regular waypoint mobility model. In Figure 8, when the regularity increases, the percentage decrease in average distance decreases for AD and LD. The first reason is that the AD and LD heuristics are not designed to learn patterns in the movement history of a mobile node for use in the prediction. The second reason is that the more regular the node movement is, the more likely a node will have less chance to move close to the target, if the route between its home and work positions are far from the target. Similarly, the percentage energy savings for AD and LD decrease slightly when the regularity increases, as shown in Figure 9.

For the MD heuristic, as the regularity increases, the percentage decrease in average distance also increases. This is because the Markov model in MD is designed learn the patterns in the movement history. Hence, with increasing regularity in the regular waypoint model, the MD heuristic performs better. However, the percentage energy saving for the MD heuristic begins to decrease when the regularity becomes larger than 80, although the average distance continues to decrease. This is because the heuristic is explicitly designed to reduce the average distance at the time of communication, while other factors may affect the actual energy consumption – e.g., the duration of communication, which is discussed below.

6) Impact of communication duration: The duration of communication affects the performance of our heuristics. The heuristics are designed to find a good time to start the communication, when the distance between the mobile node and the target is less. However, if the duration of the communication is very long, the mobile node may move away from the target while the communication is still going on. In this scenario, our heuristic may not choose the best time to communicate over the entire duration, and the energy saving may be smaller as a result. Table II shows the effect of communication duration on the amount of energy saving.

7) Impact of minimum speed in random waypoint model: We show how different minimum speeds of nodal movement in

Transmission duration (sec)	Percentage energy saving	
	Single-hop	Multi-hop
0.12	11.14	22.40
0.25	28.50	22.86
0.5	37.13	23.49
1	39.89	24.70
2	35.88	24.81

TABLE II
PERCENTAGE ENERGY SAVING USING MODIFIED AD HEURISTIC AS A FUNCTION OF TRANSMISSION DURATION.

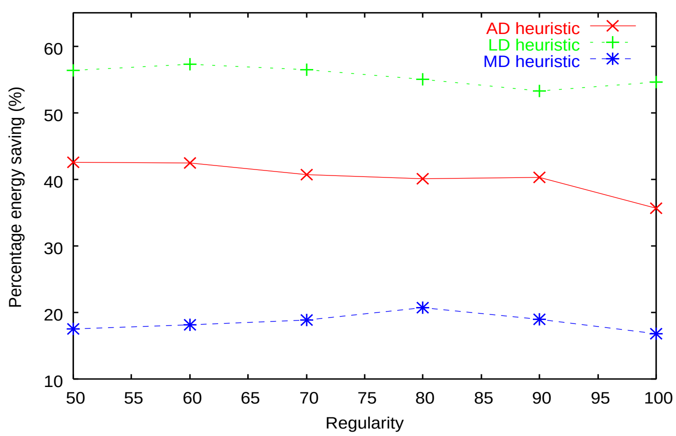

Fig. 9. Percentage energy saving as a function of regularity for MD, AD and LD heuristics.

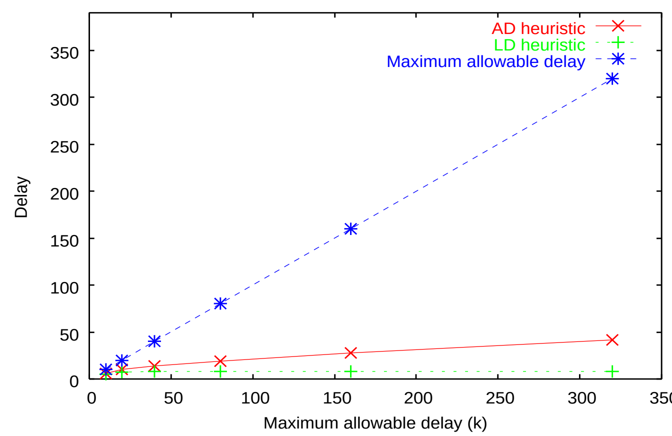

Fig. 11. Maximum allowable delay as compared to the actual observed delay for AD and LD heuristics.

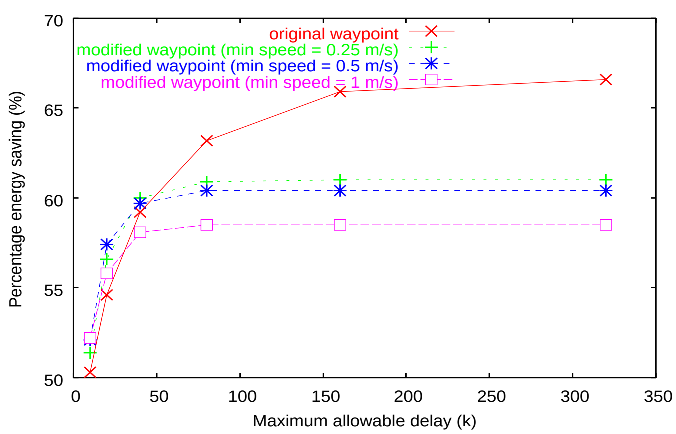

Fig. 10. Percentage energy saving as a function of k for LD heuristics.

the random waypoint model may impact the percentage energy saving for the LD heuristic. The results are shown in Figure 10. From the figure, notice that increasing the minimum speed can somewhat reduce the performance of the LD heuristic. This is because at higher speeds, even if LD can predict a good time to start a communication, it becomes more likely for the node to move away from the target during the communication.

8) Actual delay and buffer requirement: When using our power saving strategy, due to postponed sending of messages, queues may build up at each of the mobile nodes acting as a source. During our simulations we measure the actual delay for each of the transmissions. This actual delay along with the message arrival rate at a source node gives us the buffer requirement for that node. Figure 11 shows the actual delay observed as compared to the maximum allowable delay k for both the LD and AD heuristics. The figure clearly shows that the actual delay increases very slowly as compared to k and almost flattens after certain time. At the same time, increase in the value of k up to certain point gives us increased energy saving as shown in Figures 3 and 6. Thus higher energy saving can be achieved by increasing the value of k (i.e., the *allowable* delay) without increasing the *actual* delay significantly. Since the actual delay observed, i.e., the waiting time for the messages in the queue, is bounded, the total number of messages in a queue, i.e., the queue size, is also bounded for an arbitrary but bounded message arrival rate.

B. Measurement results on iPAQ

The prediction algorithms run on the mobile nodes. The required CPU processing incurs an energy cost. Our prediction strategy will effectively reduce energy consumption of the mobile device only if the computational cost, in terms of energy, is less than the energy saving in network communication. Table III shows the energy consumption due to CPU processing of our heuristics on a Compaq's iPAQ 3650 running Linux. A digital multimeter is used to measure the electrical current drawn by the iPAQ. To get an accurate measurement of the current actually consumed by the handheld device, we disconnect the batteries from the iPAQ and use an external DC power supply. A Windows terminal is connected to the multimeter. We run a program in the Windows terminal written

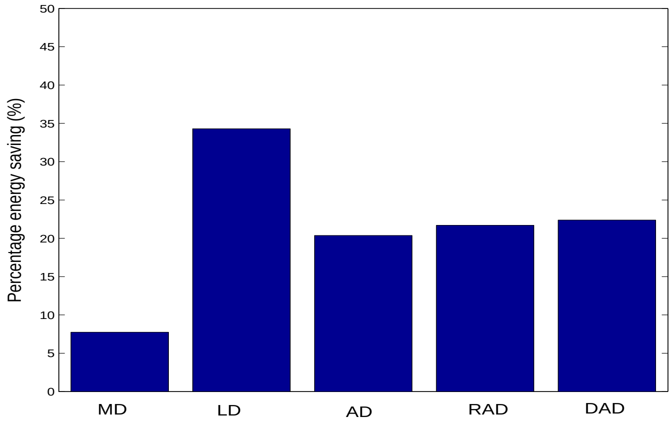

Fig. 12. Performance comparison among the heuristics.

using Agilent SICL (Standard Instrument Control Library) to store the DC current readings taken by the multimeter in a file while we run a prediction algorithm on the iPAQ. These current values, together with the voltage information and the time taken to execute a prediction algorithm, are used to calculate the energy consumption in executing the prediction algorithm.

We run each prediction algorithm 1000 times (same as the number of communications between a mobile node and the target during our simulations) and then multiply that with 10 to reflect the total energy usage by 10 nodes (recall that we used 10 nodes in our simulations) due to CPU processing. So these values can be directly compared with the energy savings shown in all the previous experiments.

The experimental results clearly show that the energy consumption in executing different prediction algorithms (of the order $10J$) are negligible compared to the energy saving in network communication (of the order 10^3J).

C. Summary comparison of the heuristics

Let us present a summary of the results for the MD, AD, DAD, RAD, and LD heuristics, and compare their effectiveness. The MD heuristic, though sophisticated and complex, does not achieve effective energy saving. In comparison, both the LD and AD heuristics are quite simple and highly CPU efficient. However, they can achieve power savings of more than 60% compared with immediate communication. Moreover, the LD and AD heuristics are robust in the sense that it shows significant power savings for both the random and regular waypoint mobility models. The bar graph in Figure 12 shows the relative performance of the heuristics presented in Section IV, for the random waypoint mobility model, multi-hop communication, $k = 10$, and $n = 80$. Other parameters for the different heuristics are chosen to best represent their performance. For the MD heuristic, $p_{th} = 0.10$ and $m = 3$.

Lastly, we summarize the properties of all our heuristics in Table IV.

VI. Related Work

The problem of limited battery life in mobile handheld devices has drawn the attention of the research community for many years now. Many hardware as well as software approaches have been proposed to cope with the problem. In software, the design criteria for operating system as well as various networking protocols should include energy efficiency along with other traditional performance metrics. Network communication is one of the major energy consumer in case of handheld devices and a lot of research has gone into optimizing networking protocols for energy efficiency. On-demand routing [16], [10], [15] has been shown to be more energy efficient compared to traditional pro-active table driven routing. This is because a lot of overhead is incurred in exchanging routing information periodically in case of a pro-active network, whereas in on-demand routing, routing information is exchanged only when necessary. Much previous work on mobility prediction [2][3][6][12][13][14] has focused on resource reservation and quick handoff management between base stations to provide QoS support for cell-network-based mobile wireless users (e.g., with bandwidth reservation and guarantees). The authors in [11], [19], [1] propose the use of location information to reduce energy consumption in routing. Su *et al* [17] uses location information to estimate the expiration time of the link between two adjacent mobile nodes. Based on this prediction, they reconstruct routes before they expire. Grossglauser and Tse [28] consider a mechanism where packet delivery is delayed (although routed through an Their objective, however, is to improve the capacity, instead of reducing the energy use, of ad hoc wireless networks. Das and Bhattacharya [3] propose an information theoretic approach to track mobile users in a Personal Communication Service (PCS) network environment. We also use location information of the mobile nodes in our energy saving strategy, but in a different context.

VII. Conclusion

Wireless networking is rapidly emerging as the future communication technology. Ad hoc networks are becoming increasingly popular because its deployment is very fast and easy. However, the components of an ad hoc network are mostly battery-powered handheld devices. Limited battery life thus is an important issue in ad hoc networking. Our strategy presented in this paper can conserve energy of the mobile nodes, thus increasing the network lifetime. Our strategy predicts a good time for communication to take place, based on the location information of the mobile nodes, and postpones communication until that point. We consider the case of both a fixed target and a moving target. As we have seen in the simulation results, variations in the transmission duration affect the amount of energy saving. It would be instructive to find effective prediction strategies that take into account the transmission duration while predicting a good time for communication. Another issue to explore further is the optimal way to divide the whole network into grids.

References

[1] Basagni S., Chlamtac, I., Syrotiuk, V. R., and Woodward, B. A. A Distance Routing Effect Algorithm for Mobility (DREAM). In Proc. Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking, (MobiCom), pp. 76-84, Dallas, TX, October, 1998.

Prediction Heuristic	Energy consumption due to CPU processing (Joule)
MD heuristic	26.904
Basic AD heuristic	1.209
Recursive AD heuristic	2.512
DAD heuristic	3.527
LD heuristic	0.134

TABLE III
ENERGY CONSUMPTION IN EXECUTING VARIOUS PREDICTION ALGORITHMS.

Heuristic	Computational complexity	Robustness	Effectiveness		Energy consumption by prediction algorithm
			Single hop communication	Multi-hop communication	
MD	Large (O(N^m))	Not much Robust.		Energy saving up to 11%	small(26.90 J)
AD	Small (O(n))	Robust.	Energy saving up to 62%	Energy saving up to 39%	small (1.21 J)
LD	Small (O(n))	Robust.	Energy saving up to 67%.	Energy saving up to 45%	small (0.13 J)

TABLE IV
SUMMARY OF THE PROPERTIES OF ALL OUR HEURISTICS.

[2] V. Bharghavan and M. Jayanth. Profile-based Next-cell Prediction in Indoor Wireless LAN. in Proc. IEEE SICON'97, Apr. 1997

[3] A. Bhattacharya, S. K. Das. LeZi-Update: an information-theoretic approach to track mobile users in PCS networks. Proc. ACM/IEEE MobiCom, August 1999, pp. 1-12.

[4] S. Biaz, G. D. Holland, Y. B. Ko, and N. H. Vaidya. Evaluation of protocols for wireless networks. Parallel and Distributed Processing Techniques and Applications, July 1999.

[5] S. Chakraborty, D. K. Y. Yau, and J. C. S. Lui, On the effectiveness of movement prediction to reduce energy consumption in wireless communication (extended abstract). Proc. ACM SIGMETRICS, San Diego, CA, June 2003.

[6] J. Chan and A. Seneviratne, "A Practical User Mobility Prediction Algorithm for Supporting Adaptive QoS in Wireless Networks," in *Proc. of IEEE International Conference on Networks,* Sep. 1999, Brisbane, Australia

[7] P. R. Freeman. The secretary problem and its extensions: A review. International Statistical review 51, 189-206, 1983.

[8] Z. J. Haas. A New Routing Protocol for Reconfigurable Wireless Networks. In Proc. IEEE 6th International Conference in Universal Personal Communications, 1997, pp 562-566.

[9] X. Hong, M. Gerla, G. Pei, and C. C. Chiang. A Group Mobility Model for Ad Hoc Wireless Networks. In Proc. ACM/IEEE MSWiM'99, Seattle, WA, August 1999, pp. 53-60.

[10] David B. Johnson, David A. Maltz, Josh Broch. DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks. Ad Hoc Networking, edited by Charles E. Perkins, Chapter 5, pp. 139-172, Addison-Wesley, 2001.

[11] Youngbae Ko, N. H. Vaidya. Location-Aided Routing (LAR) Mobile Ad Hoc Networks. In Proc. ACM/IEEE Mobicom, October 1998, Dallas, TX.

[12] D. Levine, I. Akyildiz and M. Naghshineh, "A Resource Estimation and Call Admission Algorithm for Wireless Multimedia Networks Using the Shadow Cluster Concept," *IEEE/ACM Trans. on Networking,* 5(1), Feb. 1997, pp 1-12.

[13] G. Liu and G. Maguire Jr., "A Class of Mobile Motion Prediction Algorithms for Wireless Mobile Computing and Communications," *ACM/Baltzer MONET,* 1(2), 1996, pp. 113-121

[14] (7)T. Liu, P. Bahl and I. Chlamtac, "Mobility Modeling, Location Tracking, and Trajectory Perdition in Wireless ATM Networks," *IEEE JAC,* 16(6), Aug. 1998, pp. 922-936

[15] Vincent D. Park and M. Scott Corson. A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks. In Proc. IEEE INFOCOM '97, Kobe, Japan, April 1997.

[16] Charles E. Perkins and Elizabeth M. Royer. Ad hoc On-Demand Distance Vector Routing. In Proc. 2nd IEEE Workshop on Mobile Computing Systems and Applications, New Orleans, LA, February 1999, pp. 90-100.

[17] W. Su, S. J. Lee, M. Gerla. Mobility Prediction and Routing in Ad Hoc Wireless Networks. International Journal of Network Management, Wiley & Sons, 2000.

[18] D. S. Tan, S. Zhou, J. Ho, J. S. Mehta, H. Tanabe. Design and Evaluation of an Individually Simulated Mobility Model in Wireless Ad Hoc Networks. In Proc. Communication Networks and Distributed Systems Modeling and Simulation, 2002, San Antonio, TX.

[19] Ya Xu, John Heidemann, Deborah Estrin. Geography-informed Energy Conservation for Ad-hoc Routing. In Proc. Seventh Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom), Rome, Italy, July 2001.

[20] http://may.cs.ucla.edu/projects/maisie/models.html

[21] Carnegie Mellon University Monarch Project. CMU Monarch Project Home Page. Available at http://www.monarch.cs.cmu.edu/.

[22] Ferguson, T.S.. Who Solved the Secretary Problem?. Statistical Science, 1989, 4, p282-296.

[23] Y. H. Chun, H. Moskowitz, R. Plante. Single Threshold Selection Strategy in the Group Interview Problem. INFOR: Information Systems and Operational Research. Vol 35, No. 3 (August 1997), pp.157-169.

[24] J.D. Hey.Search for Rules for Search. Journal of Economic Behaviour and Organization (1982), 3, pp.65-81.

[25] L. Breslau *et al*. Advances in network simulation. IEEE Computer, 33(5):59-67, May 2000. Expanded version available as USC TR 99-702b at http://www.isi.edu/ johnh/PAPERS/Bajaj99a.html.

[26] D. Johnson, D. A. Maltz. Dynamic source routing in ad hoc wireless networks. Mobile Computing, T. Imielinski and Hank Korth, Eds. Kluwere Academic Publishers, 1994.

[27] M. P. Quine, J. S. Law. Exact Results for Secretary Problem. Journal of Applied Probability. 33, pp.630-339, 1996.

[28] M. Grossglauser, D. Tse. Mobility Increases the Capacity of Ad-hoc Wireless Networks. IEEE/ACM Transactions on Networking, August 2002, Vol. 10 , Issue 4, pp. 477-486.

[29] J. Yoon, M. Liu and B. Noble, "Random Waypoint Considered Harmful," in *Proc. IEEE INFOCOM,* 2003,

VIII. AUTHORS' BIOGRAPHIES

Srijan Chakraborty received the B.S. degree in Computer Science from the Birla Institute of Technology and Science (BITS), Pilani, India in 2001, and the M.S. degree in Computer Science from Purdue University in 2003. Presently he is working at Microsoft Corporation. His research interests are in mobile networking and distributed systems. At Purdue, his research work focused on the problem of reducing energy consumption of battery-powered devices (e.g., laptops and handhelds) using application-aware run-time adaptation, protocol enhancement, and routing optimization.

Yu Dong received the B.E. degree from the School of Automation and Information Engineering, University of Science and Technology Beijing (USTB), China in 1995, the M.E. degree in Information System Engineering from Osaka University, Japan in 2001, and the M.S. degree in Computer Science from Purdue University in 2003. He is currently working towards his Ph.D degree at Purdue. His research interests are in networking, including wireless ad-hoc and overlay networks, multimedia systems, and VLSI design. Since 1999, he has been a student member of the IEEE.

David K. Y. Yau received the B.Sc. (first class honors) degree from the Chinese University of Hong Kong, and the M.S. and Ph.D. degrees from the University of Texas at Austin, all in computer sciences. From 1989 to 1990, he was with the Systems and Technology group of Citibank, NA. He was the recipient of an IBM graduate fellowship, and is currently an Associate Professor of Computer Sciences at Purdue University, West Lafayette, IN. He received an NSF CAREER award in 1999, for research on network and operating system architectures and algorithms for quality of service provisioning. His other research interests are in network security, value-added services routers, and mobile wireless networking. David is a member of the ACM and IEEE. He serves on the editorial board of the *IEEE/ACM Transactions on Networking*.

John C. S. Lui received his Ph.D. in Computer Science from UCLA. He worked in the IBM T. J. Watson Research Laboratory and in the IBM Almaden Research Laboratory/San Jose Laboratory before taking up an academic position at the Chinese University of Hong Kong. Currently, he is leading a group of research students in the Advanced Networking and System Research Group. His research encompasses both systems and theory. His current research interests are in theoretical/applied topics in data networks, distributed multimedia systems, network security, OS design, mathematical optimization, and performance evaluation. John received the Vice-Chancellor's Exemplary Teaching Award in 2001. He is an associate editor of the Performance Evaluation Journal, a member of the ACM, a senior member of IEEE, and an elected member of the IFIP WG 7.3. John serves as the TPC co-chair of ACM Sigmetrics 2005. His personal interests include films and general reading.