Controllably Mobile Infrastructure for Low Energy Embedded Networks

Arun A. Somasundara, Aman Kansal, *Student Member*, *IEEE*, David D. Jea, *Student Member*, *IEEE*,
Deborah Estrin, *Fellow*, *IEEE*, and Mani B. Srivastava, *Senior Member*, *IEEE*

Abstract—We discuss the use of mobility to enhance network performance for a certain class of applications in sensor networks. A major performance bottleneck in sensor networks is energy since it is impractical to replace the batteries in embedded sensor nodes post-deployment. A significant portion of the energy expenditure is attributed to communications and, in particular, the nodes close to the sensor network gateways used for data collection typically suffer a large overhead as these nodes must relay data from the remaining network. Even with compression and in-network processing to reduce the amount of communicated data, all the processed data must still traverse these nodes to reach the gateway. We discuss a network infrastructure based on the use of controllably mobile elements to reduce the communication energy consumption at the energy constrained nodes and, thus, increase useful network lifetime. In addition, our approach yields advantages in delay-tolerant networks and sparsely deployed networks. We first show how our approach helps reduce energy consumption at battery constrained nodes. Second, we describe our system prototype which utilizes our proposed approach to improve the energy performance. As part of the prototyping effort, we experienced several interesting design choices and trade-offs that affect system capabilities and performance. We describe many of these design challenges and discuss the algorithms developed for addressing these. In particular, we focus on network protocols and motion control strategies. Our methods are tested using a practical system and do not assume idealistic radio range models or operation in unobstructed environments.

Index Terms—Mobile networking, controlled mobility, wireless sensor networks.

◆

1 INTRODUCTION

Energy is a key concern in wireless sensor network design because, once the nodes are embedded into the environment, it becomes impractical to replace their batteries. One of the major energy expenditures is in communicating the sensor readings, in raw or processed form, from the sensors to a central user location. Usually, these readings are relayed to a base station using ad hoc multihop routes in the sensor network. A problem with this approach, however, is that the nodes closer to the base station relay data from nodes that are farther away. Thus, the nodes closest to the base station consume batteries faster than the remaining network, leading to a nonuniform depletion of energy in the network. Once the nodes with connectivity with the base station exhaust their energy, the network is disconnected and, hence, considered dead for all practical purposes.

A significant advantage in network lifetime can be gained if the energy spent in relaying data can be saved. An alternative for data transfer that does not involve relaying it over multiple hops is to use mobility. A mobile node moving through the network deployment region can collect data from the static sensor nodes over a single hop radio link as and when the mobile node is within radio range of the static nodes. This naturally avoids multihop relaying and reduces the energy overhead at nodes near the base station, enabling the network to last longer. This may increase the latency of data transfer, but is acceptable in several delay-tolerant applications, such as in environmental studies [1], [2], [3]. The field experts in these studies, such as biologists and ecologists, are interested in studying the behavior of the environment and do not need to take any real-time action. As a result, it is not an issue if data does not reach them as soon as it is generated. In other scenarios, certain event information or processed data may be transmitted over multihop wireless routes to achieve low delay for immediate actions, but the bulk raw data could be collected by the mobile node for detailed analysis with a much longer delay.

In certain applications, mobile elements already exist in the deployment environment and a network node can be attached to these mobile elements for data collection [4], [5], [6], [7]. Otherwise, mobile nodes can be added to the system [8], [9], [10] for data collection. In this case, the mobile node is a part of the network infrastructure itself and can be controlled by the network as required. We consider the latter approach in our work.

Another advantage of using a mobile element is that it can handle sparse or disconnected networks. Disconnected networks may be desirable in certain applications due to the following reasons: First, the placement of sensor nodes (dictated by the requirements of field experts) may lead to

---

- *A.A. Somasundara is with the Broadcom Corporation, 3151 Zanker Road, San Jose, CA 95134. E-mail: aruns@broadcom.com.*
- *A. Kansal and D.D. Jea are with the Department of Electrical Engineering, University of California Los Angeles, ATTN: 56-125KK, Engg-IV Building, C/O: Fe Asuncion, 420 Westwood Plaza, MC# 951594, Los Angeles, CA 90095-1594. E-mail: {kansal, dcjea}@ee.ucla.edu.*
- *D. Estrin is with the Department of Computer Science, University of California Los Angeles, 3531H Boelter Hall, Box 951596, Los Angeles, CA 90095-1596. E-mail: destrin@cs.ucla.edu.*
- *M.B. Srivastava is with the Department of Electrical Engineering, University of California Los Angeles, MC# 951594, 6731-H Boelter Hall, Los Angeles, CA 90095-1594. E-mail: mbs@ee.ucla.edu.*

Manuscript received 29 Nov. 2004; revised 2 Mar. 2005; accepted 14 June 2005; published online 15 June 2006.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-0317-1104.

disconnected networks. Second, even to sample the whole area of deployment, the sampled physical phenomenon may have a low spatial variation, i.e., it may be sufficient to place sensors at distances which are larger than the available communication range in the environment. For example, several interesting networked sensing applications occur in marine environments [11], but the radio range is severely constrained underwater and an inordinately high number of nodes would be required to keep the network connected. Such a density of network nodes may interfere with the sensed environment itself and may hence be undesirable. Mobile nodes are already being developed for marine environments [12]. In the above scenarios, instead of deploying additional nodes just to maintain wireless connectivity, a mobile node could be used to collect data from disconnected components. An alternative method to maintain connectivity in sparse networks is to increase the communication range such that network is connected. But, since the radio energy grows super-linearly (typically with an exponent between 2 and 4) with the transmission range while the energy used by a mobile increases only linearly with distance moved (at a given speed), using a mobile node is again advantageous. Also, increasing transmit power alone cannot always increase the range sufficiently in cluttered environments with multipath and fading concerns.

Further, the mobile node can be used for enabling other network functionality not feasible with static nodes alone, such as energy replenishment [13], [14], coverage repair [15], [16], and localization [17].

In this paper, we consider some of the key design issues for a network which uses a controllably mobile node for data transfer. We exploit the control over the motion of the mobile node to improve data gathering performance. The paper is organized as follows: Section 2 summarizes the related work in the use of mobility for communication. We then analyze the energy trade-offs involved in the use of controlled mobility in Section 3 and the throughput trade-offs in Section 4. Section 5 summarizes the prototype system developed for our investigations, including the hardware platforms and the software architecture. It also describes the key algorithm development required for realizing our proposed data collection approach. Our network protocol design is presented in Section 6 and the motion control strategy is presented in Section 7. Sections 8 and 9 present the system evaluation in experiments and simulations. Finally, Section 10 concludes the paper.

2 RELATED WORK

Various types of mobility have been considered in the domain of wireless networks for facilitating data transfer. These can be broadly classified into three categories: random, predictable, and controlled.

Random mobility, assuming all nodes are mobile, has been shown to improve data capacity [18] and similar results were also shown to hold for mobility constrained to a single dimension [19]. An algorithm for routing data among randomly mobile users was suggested in [20], where data is forwarded to nodes which have recently encountered the intended destination node. Random motion of mobile entities was also used for communication in [6], [5], where the mobile entities were zebras in [6] and whales in [5]. In [4], [21], randomly moving humans and animals act as "data mules" and collect data opportunistically from sensor nodes when in range. The average latency performance with mobility as a means to transfer data was calculated in [22]. However, in all cases of random mobility, the worst-case latency of data transfer cannot be bounded. This unbounded latency may lead to excessive data caching at mobile relays and result in buffer overflows. Thus, data in transit may have to be dropped before being delivered to the destination, making it harder to provide transport layer reliability.

Predictable mobility was used in [7]. A network access point was mounted on a public transportation bus moving with a periodic schedule. The sensor nodes learn the times at which they have connectivity with the bus and wake up accordingly to transfer their data. However, it was assumed in [7] that the mobile node comes within direct radio range of all the static nodes, which may not be true in practice, because the trajectory of the bus may not be designed for data collection purposes. We design a network protocol which supports nodes at multiple hops from the mobile node and helps find the minimum energy route for such nodes to deliver their data to the mobile node.

Controlled mobility was considered in our previous work [8], where a robot acts as a mobile base station. The speed of the mobile node was controlled to help improve network performance. We extend that work in three ways: First, we analytically explore the energy trade-off when the energy of moving the mobile node is also considered. We also analytically show that our proposed approach does not lead to any significant differences in network data throughput compared to the use of wireless multihop relaying in a static network. Second, we design an enhanced communication protocol, discussing the trade-offs of the alternative approaches in the design space. Third, we consider the design of the motion control strategy in greater depth. The number of motion states used is analyzed and, also, the effect of speed on buffer requirements is explored. Controlled mobility was also used in [9], where a mobile node is used to route messages between nodes in sparse networks. However, all nodes are assumed to have short range mobility and can modify their locations to come within direct range of the mobile node, which has long range mobility and is used for transferring data. The communication protocol uses two radios, which may not be available in many scenarios. The system has been evaluated in simulations; we evaluate our system both in implementation and simulations.

In [23], mobile nodes in a disconnected ad hoc network modify their trajectories to come within communication range and [24] considered moving the intermediate nodes along a route so that the distances between nodes are minimized and lower energy is used to transmit over a shorter range. This system also assumes that all nodes are mobile, which may be expensive or infeasible in many deployments where node locations depend on sensing or application requirements. A mobile base station was also used in [10] to increase network lifetime. Their technique leads to a more uniform distribution of energy consumption by repeatedly relocating the base station, which changes the bottleneck nodes which are closest to the base station and results in load-sharing the burden of relaying. However, the total energy consumed is not reduced. Our system not only reduces the energy consumed in bottleneck nodes, but also conserves total energy by utilizing shorter routes. Further, while they do not consider the energy spent in moving the base station, our approach is advantageous even when the motion energy is accounted for. The above systems are studied only analytically or in simulation. We implement a working prototype which involves addressing several practical issues and detailed network protocol design.

A scheduling problem for the mobile node with buffer constraints on static nodes and variable sampling rates at each static node is studied in [25]. We address the issues of practical communication protocol design and motion control.

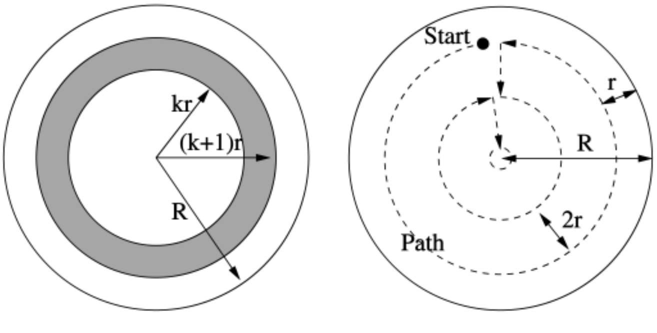
3 QUANTIFYING THE ENERGY ADVANTAGE

The main motivation for using a mobile element is that it increases the lifetime of the network by reducing the relaying of data. To quantify potential savings, we consider the energy requirements with and without a mobile node. Here, we also consider the energy used by the mobile for locomotion.

Suppose the network is deployed over a circular disk of area A and radius R. A set of n sensor nodes is uniformly and randomly deployed over this area. The base station, where all sensor nodes must send their data, is located at the center of the circular area. Since we are comparing only the communication methods, we do not consider the sensing and other processing energy costs. The radio is active only when communicating and is in sleep mode otherwise to minimize energy consumption. Suppose each node must send b bits of data to the egress point, in an interval of time T. To compare the minimum energy cost, we assume that T is large enough to accommodate the most energy efficient transmission schedule at the available bandwidth. Also, assume that an ideal Medium Access Control (MAC) is available and no energy is wasted in collisions.

3.1 Static Network

First, consider the static network where all data is relayed over a multihop topology. To calculate the minimum number of hops a packet must travel, we proceed as follows: Let the transmission range of the nodes be r. The nodes within distance r from the center are within one hop of the sink. The nodes beyond r but within $2r$ are at least two hops from the sink. While some nodes within this distance range may need more than two hops, we calculate the energy, assuming only two hops to get the minimum consumption for the static network. Similarly, nodes which are at a distance, s, from the base station, such that $kr < s \leq (k+1)r$ for some integer k, have to relay their data over at least $k+1$ hops. If the innermost annulus is denoted the zeroth annulus ($k = 0$), then the data from nodes in the kth annulus is transferred over $k+1$ hops, leading to k receptions and $k+1$ transmissions. (The energy for reception at the base station is not included.)



(a) (b)

Fig. 1. Energy consumption calculation. (a) Dividing the static network into concentric annuli to count the minimum number of relay hops required for each node and (b) path for the mobile node to collect data from the entire network. (a) Static network. (b) Network with a mobile node.

The number of nodes at each distance can be calculated as follows: Divide the circular deployment region into concentric annuli, as shown in Fig. 1a.

The number of nodes in a region of area a is approximately equal to $(a/A)n$ at large n since the deployment is uniformly random. The area $a(k)$ of an annulus lying between two circles of radii $(k+1)r$ and kr is

$$a(k) = \pi[\{(k+1)r\}^2 - k^2r^2]. \tag{1}$$

The number of such annuli is $\lceil R/r \rceil$ in the circular area of radius R and, hence, $k \in \{0, \ldots, \lceil R/r \rceil - 1\}$. If R is not a multiple of r, the outermost annulus will have an area smaller than that given by the formula above. It will be $A - \pi x^2 r^2$, where A is the total area of the deployment region and $x = \lceil R/r \rceil - 1$.

Now, the minimum number of transmissions for a single message originating at each of the nodes in the $k\text{th}$ annulus is:

$$n\frac{a(k)}{A}(k+1) \tag{2}$$

and the number of receptions can be similarly evaluated.The communication costs include transmitting a packet and relaying (receiving and transmitting) packets for other nodes.

We now calculate the energy cost corresponding to these activities. If the power consumption of the radio in transmit mode is P_{tx} at a data-rate of W bps, then the energy cost of transmitting $b\text{-bits}$ becomes:

$$E_t(b) = \frac{P_{tx}}{W}b. \tag{3}$$

Similarly, if the power consumption in receive mode is P_{rx}, the reception cost for $b\text{-bits}$ becomes:

$$E_r(b) = \frac{P_{rx}}{W}b. \tag{4}$$

Also, since the radio must enter the active mode from a sleep mode before transmitting each packet, the wake-up energy cost of the radio must be added to it. In sensor networks, this wake-up cost is an important consideration as the nodes are repeatedly duty-cycling to low power modes. In the static case, if a naive forwarding strategy is used or the bandwidth is attempting to be maximally utilized by conducting multiple communications in parallel, then a node may have to wake up once to receive each data packet it has to relay and also once for each data packet it has to transmit. However, to compute the minimum energy required by a multihop relaying scheme, we consider an energy-optimal schedule where a node wakes up once, receives all the packets it has to relay, and immediately transmits them.[1] Thus, each node wakes up only once in the duration T, which is the minimum number of times it must wake up even to send just its own data. This could be achieved using a schedule where nodes in the outermost annulus transmit first, with an ideal MAC (such as a perfectly synchronized TDMA scheme) to avoid collisions. The nodes in an inner annuli, after receiving the data, immediately transmit it to the corresponding nodes along the paths in the next inner annuli. Let the sum of energy costs for waking up from sleep and entering sleep after activity be E_w.

Assume that the energy consumed at the base station is not a concern. Counting the number of transmissions and receptions as discussed above, the minimum total energy consumed in the embedded network, E_{static}, in each period T to send b bits from every node to sink is thus given by:

$$E_{static} = \sum_{k=0}^{\lceil \frac{R}{r} \rceil - 1} n \frac{a(k)}{A}(k+1)E_t(b) + \sum_{k=1}^{\lceil \frac{R}{r} \rceil - 1} n \frac{a(k)}{A} k E_r(b) + \sum_{k=0}^{\lceil \frac{R}{r} \rceil - 1} n \frac{a(k)}{A} E_w, \qquad (5)$$

where the first term corresponds to the cost of transmissions from all annuli using (2), the second term to the receptions, and the third term is the wakeup cost.

3.2 Network with a Mobile Base Station

Assume the same network deployment as above. However, instead of relaying the data over multiple hops, now the data is collected by a mobile node which acts as a base station. Let the path of the base station be such that it comes within r of each node and moves at a speed such that b bits can be transmitted by every node to it in the time the mobile node is within range. As before, T is large enough to allow this. A possible path for the mobile node to cover the entire circular deployment is shown in Fig. 1b, where the mobile moves along concentric circles separated by $2r$, with the first circle in the path at distance r from the outer circle of the area, and the last circle at a distance of r units (or less if R is not a multiple of $2r$) from the center. This scheme covers the entire region. The length of this path, L, is:

$$L = 2\pi \sum_{i=0}^{\lceil \frac{R}{2r} \rceil - 1} [R - (2i+1)r] + 2r\left(\left\lceil \frac{R}{2r} \right\rceil - 1\right), \qquad (6)$$

where the first term corresponds to the length along the concentric circles and the second term is the distance moved to transit between the circles. Suppose the energy consumption in moving a unit distance is E_{mov} for the chosen speed of movement. E_{mov} includes both the locomotion and processing energies used by the traction platform. The total motion energy will then be LE_{mov}. In the static multihop relaying network, data can be transmitted as soon as collected, even though that may not be how a practical network operates due to energy overheads in radio wake-up costs. However, with the mobile node, data at the static nodes may have to be stored in a flash memory until the mobile comes by. We consider the energy spent in writing to flash memory in this case, thus making the comparison harsher on the network with a mobile node. Suppose the energy spent in writing to flash is E_{fl} per bit.

The energy consumed in collecting b bits, in this case, E_{mobile}, is the motion cost for the base station, communication cost for b bits from n embedded nodes to the base station, the wake-up cost, and the energy consumed in writing to flash:

$$E_{mobile} = E_{mov}L + nE_t(b) + nE_w + nbE_{fl.} \qquad (7)$$

Let us compare the expressions derived in (5) and (7). We substitute the various parameters from Mica2 motes datasheet [26], which is also used in our prototype system. The data rate[2] used is $W = 19.2kbps$. The receive power is $P_{rx} = 30mW$. The transmission range obtained in practice for a given power level is not a fixed quantity, but varies with a number of factors, like the environment, sensitivity of receiver, etc. Our system is targeted at natural habitat monitoring. A radio range of $r = 5m$ is reported in [27] for natural habitat environments at 80 percent packet delivery ratio for a power level of $-3dBm$ corresponding to a transmit mode power consumption $P_{tx} = 23.7mW$ [28]. The robotic platform used for the mobile node in our prototype gives an $E_{mov} = 60J/m$ [29]. The flash energy is measured to be $E_{fl} = 36.56nJ/bit$ and we take $b = 500kB$, which is the available flash size on this platform. The term corresponding to the wake-up energy is same in both equations and can hence be neglected. Consider a network deployment region with $R = 100m$. With these values, E_{static} (5) and E_{mobile} (7) are plotted with varying node density in Fig. 2. The unit of node density is the number of nodes within the radio coverage area of a node, i.e., number of nodes in area πr^2.

The graph shows that the energy used in the network with a mobile node is lower after a particular density. Thus, for the same total battery energy provided to the system, the network with a mobile node consumes energy at a slower rate and, hence, could last longer (when the number of nodes and, hence, density is more than a threshold).

The problem of how many neighbors each node should be connected to so that the overall network becomes connected has been studied earlier. For instance, Kleinrock and Silvester [30] set this magic number at 6. In [31], each node was allowed to adjust its transmission range, and a node density of 6 and 8 was derived for different transmission strategies. Recently, Xue and Kumar [32] showed that no such magic number is possible, and the number of neighbors required is $\Theta(\log n)$, where n is the number of nodes. Thus, the number of neighbors per node required to maintain connectivity increases with the number

---

[1] This schedule is optimal for energy minimization only and may not yield the maximum possible data-rate.

[2] The radio hardware allows $38.4kbps$, but due to processing speed limitations, only $19.2kbps$ is used in TinyOS.

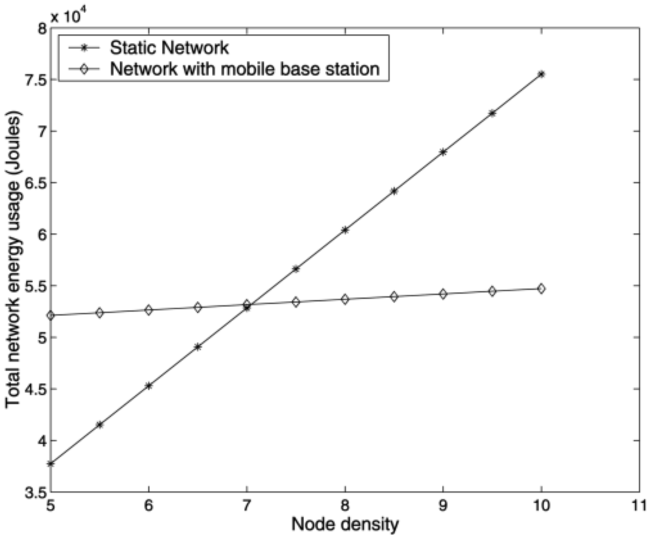


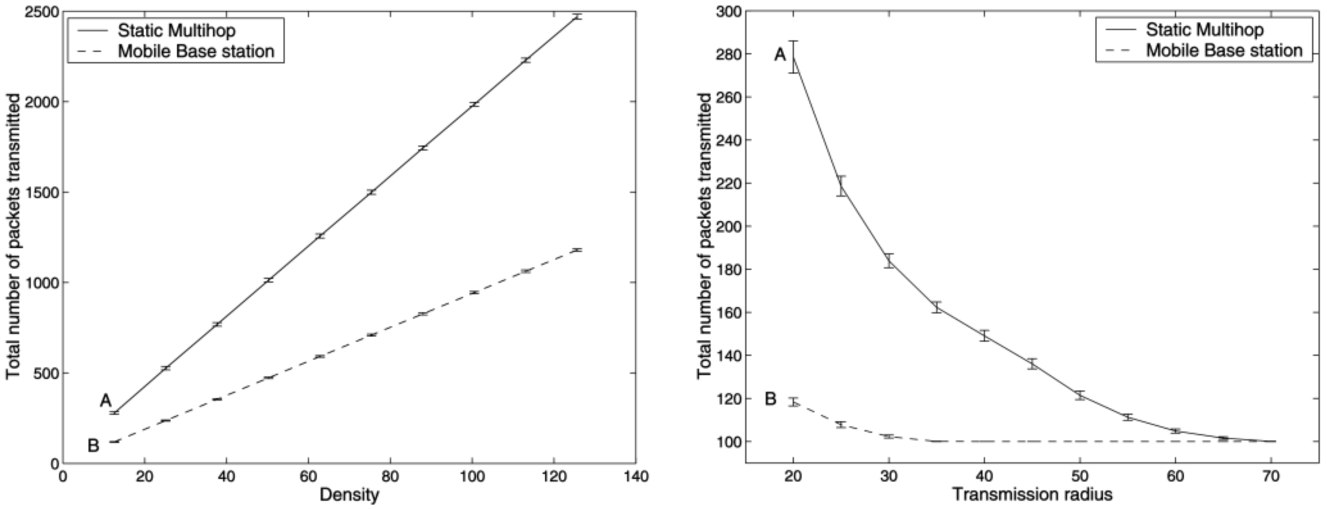
Fig. 2. Comparing the energy consumption for a static network with that of a network using a mobile base station.

of nodes. As seen from the figure, at higher densities, using a mobile node leads to energy efficiency.

In the above calculation, we have added the energy consumed in moving the mobile node. In practice, such a mobile node can be recharged easily, such as at a docking station along its path, and this energy need not be a consideration. Clearly, then, the savings in energy at static nodes due to the reduced relaying overhead will help extend the network lifetime significantly.

In certain cases, such as for the reasons discussed in Section 5, it may not be possible to design a path for the mobile node which allows it to establish direct connection with each static node in the network, as was the case in the above analysis. Below, we simulate a network in a square area of $100m \times 100m$. In the static multihop case, the base station is located at the center, $(50, 50)$. In the mobile case, the mobile base station is constrained to move only along a circle of radius $35m$ centered at $(50, 50)$, which causes some nodes in the square deployment region to be outside its radio connectivity. The data from such nodes is transferred over short multihop routes to nodes which have direct connectivity to the mobile base station. Each static node has one packet to send to the base station (static or mobile). We measure the number of packets transmitted by all the nodes in the network to achieve this goal. (If all nodes are 1-hop from the base station, the total number of packets transmitted = the number of nodes). The simulations are repeated for 100 random topologies and averaged. Fig. 3a shows the total number of packets transmitted with increasing density and a fixed transmission range of 20. We see that as the node density increases, using a mobile base station has greater energy advantage. As before, density is measured as the number of nodes in a transmission circle. Fig. 3b shows the number of packets transmitted as transmission radius increases for a fixed number of nodes (100 nodes). Here, we see that with a smaller transmission radius (and, hence, a correspondingly smaller energy consumption for transmission), the number of packets transmitted by the network drastically reduces when using a mobile base station.

To further visualize the relaying overhead, we plot the average number of packets transmitted by nodes at a particular hop distance from the base station for a 1,000 node network and a transmission radius $r = 20m$ in Fig. 4. Note that when the mobile base station is used, all nodes are within one or two hops from the base station, while the hop distance is larger with static multihop relaying. The figure clearly shows that the energy bottlneck in a static multihop network occurs at the nodes closest to the base station as they suffer the greatest relaying overhead. Once these nodes run out of batteries, the energy



(a)

(b)

Fig. 3. Packet transmissions required to collect one packet from each node, with and without a mobile base station: (a) Number of packets transmitted with increasing density and (b) number of packets transmitted with increasing transmission radius. The points representing the same network scenario (density = 12.57, transmission radius = 20) in the two graphs are labeled: A, B. (a) Changing node density by changing # of nodes; $r = 20$. (b) Changing radio range; number of nodes $= 100$.

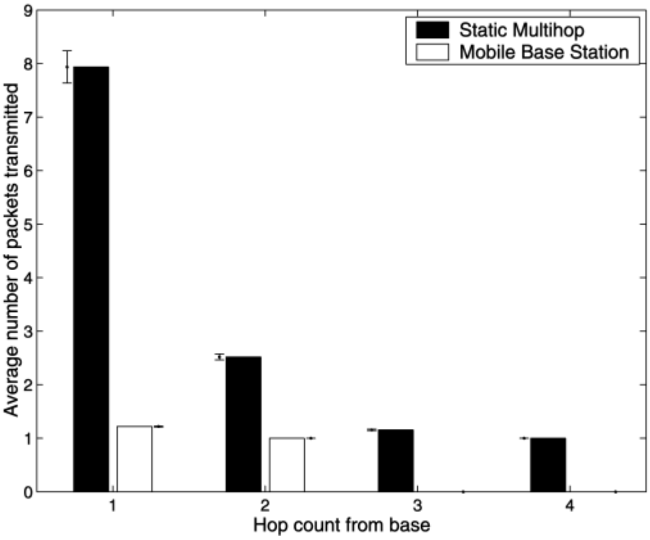


Fig. 4. Relaying overhead for the two data gathering approaches. Nodes are divided into classes with equal hop distance from the base station, and the number of packets transmitted by nodes in each class is averaged.

remaining in rest of the network is of no use since the network is disconnected. In the network with a mobile, all static nodes are within one or two hops and the average energy used at all nodes is more uniform than in the static multihop network.

4 COMPARING DATA RATES

We now analyze the potential data rate for a system using a mobile node for data transfer and compare it with that of static multihop relaying. The static network deployment and the path of the mobile node are as in the previous section, with one minor change for the path of the mobile node to achieve symmetry. When the mobile node completes the innermost circle, it moves back straight to the starting point *Start* of Fig. 1b (during which there is no data collection). This makes the Round Trip Time (RTT) of the mobile as seen by all the nodes the same, whereas, if after completing the innermost circle, the mobile backtracked its path, again traveling in concentric circles, but this time moving outward, the RTT seen by each node would be different, and would lie in $[0, 2L]$. Equation (6) is modified as below, where the new term corresponds to the linear distance moved to reach the *Start* after completing the innermost circle:

$$L = 2\pi \sum_{i=0}^{\lceil \frac{R}{2r} \rceil - 1} [R - (2i+1)r] + 2r\left(\left\lceil \frac{R}{2r} \right\rceil - 1\right) + \underbrace{2r\left(\left\lceil \frac{R}{2r} \right\rceil - 1\right)}. \tag{8}$$

Suppose the radio data rate is W for a point to point link. Suppose that, at any location along its path, the mobile is within the radio range of k nodes at any time and the maximum channel data rate available to each node is W/k. The number of nodes within range, k, can be approximated as $(\pi r^2/\pi R^2)n = (r^2/R^2)n$ for a uniformly random deployment at large n. A static node can be within transmission range for a maximum distance of $2r$. When the shortest distance between the node and the mobile node's path is almost r, they will be within range for a very small portion of the path $\delta > 0$. Taking the average length of the intersection of a randomly placed line segment with a circle, a static node will remain in range for a distance of $(\pi/2)r$. If s is the speed of the mobile node, the average time spent within range is $\pi r/(2s)$. Thus, the average number of bits collected from a static node is $(W/k) * (\pi r/2s)$.

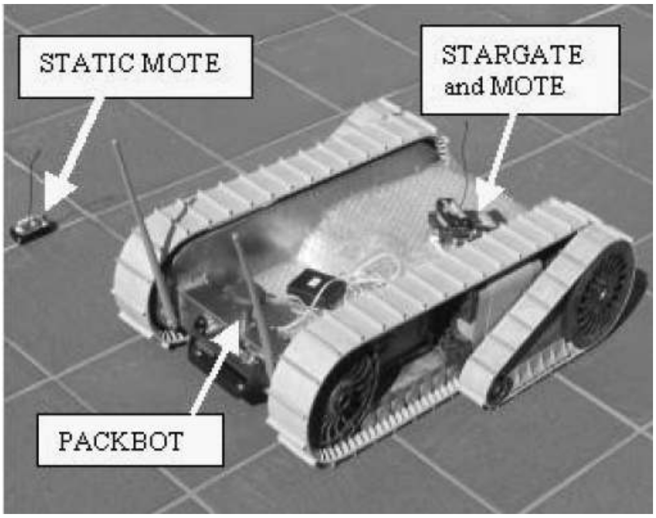


Fig. 5. Various components of the prototype system.

The time taken by the mobile to collect data from all nodes as per the motion scheme described above is L/s. Thus, the per node data-rate, C_m, can be calculated as:

$$\begin{aligned} C_m &= \left(\frac{W}{k} * \frac{\pi r}{2s}\right) \Big/ \left(\frac{L}{s}\right) \\ &= \frac{W}{n}\left(\frac{\pi R^2}{2rL}\right). \end{aligned} \tag{9}$$

This is $O(1/n)$ as R, r, and L are constant for a fixed area. This may be compared to the data-rate derived for static multihop relaying in [33]:

$$C_s = \frac{W}{n}\left(\frac{1}{2 - \pi r^2}\right). \tag{10}$$

That is also $O(1/n)$, differing only in the constants, and, hence, the approach of using a mobile node for saving energy does not lead to any significant compromise on network data-rate.

The delay is potentially higher when using the mobile. The worst-case delay seen by a packet (the time between its generation and collection by the mobile node) is the path traversal time, (L/s). While this is O(1) for fixed area, it can be higher than the time required for routing a packet over a multihop wireless path. Thus, the use of a mobile node for routing is beneficial for delay tolerant applications.

5 PROTOTYPE SYSTEM DESIGN

We now describe our prototype system, shown in Fig. 5, which uses a mobile base station to collect data from multiple static nodes in order to save energy at static nodes. The key hardware components of the system are:

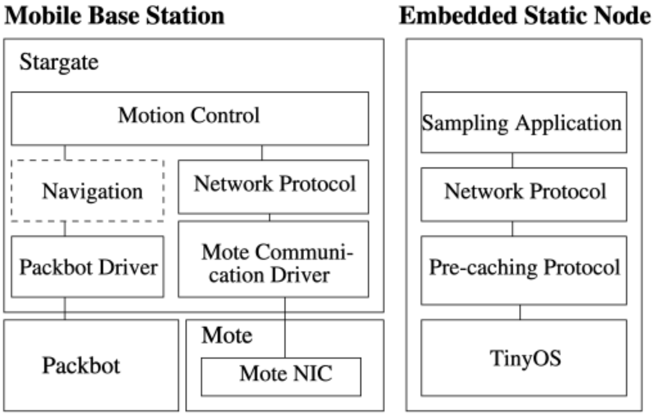


Fig. 6. Software architecture for the prototype system for both the mobile node and the static nodes.

1. Static Sensor Nodes: These are embedded devices deployed for sampling the environment and are energy constrained. In our system, we use Mica2 motes [26].
2. Mobile Node: The mobile node is composed of three elements:
   - Robot: We use a packbot [34] as a traction platform for mobility.
   - Processing Platform: The brain of the mobile node is a Stargate [35], an Xscale processor-based computing device running Linux.
   - Network Interface: A Mica2 mote is attached to the serial port of the Stargate to act as a wireless interface to the static sensor nodes.

5.1 Software Architecture

The software architecture developed for the prototype system is shown in Fig. 6. The functions of the various blocks on the mobile base station and on the static nodes are explained below.

Mobile Base Station. The software for the mobile base station is divided into three parts corresponding to its hardware components:

1. *Packbot Firmware:* The robot firmware is implemented by the manufacturer on the Packbot and a control interface is provided.
2. *Stargate Software:* The software on the Stargate processing platform is developed in Linux. The various blocks are:
   a. *Packbot driver:* This component generates the appropriate Packbot specific commands corresponding to the motion control dictated by the higher layers.
   b. *Navigation:* This component is employed to move the robot autonomously along a preassigned path. The navigation block is not implemented as part of our work as several such solutions based on laser ranging and other methods are available for enabling a robot to follow a preassigned path [36], [37]. In our experiments, we use a straight path and tracking the motion commands to determine location along the path suffices.
   c. *Mote communication driver:* This is essentially a serial port driver and handles the communications with the mote attached to the stargate.
   d. *Network protocol:* This block represents the communication strategy designed for data collection using our mobile base station. This protocol is independent of the algorithm used for controlling the mobility of the mobile node and will work even if the motion of the mobile node was not controlled by the system, as long as the mobile node follows a fixed path, such as in [7]. The exact algorithms developed for this block are discussed in Section 6.
   e. *Motion Control:* This block monitors the data collection performance of the network protocol and controls the motion of the mobile node for enhancing the data collection performance. The exact algorithms designed for this purpose are discussed in Section 7.
3. *Mote Firmware:* The mote shown as part of the mobile base station acts as a network interface to the embedded nodes. It forwards the data and protocol packets received on the wireless channel from the static nodes to the Stargate on the mobile base station over a serial port and vice versa.

Embedded Static Node. The firmware on the static nodes is developed in TinyOS [38]. The static nodes contain two blocks required for the data collection process, labeled the *precaching protocol*, which is responsible for communication among static nodes, and the *network protocol*, which controls the data collection process between the static nodes and the mobile base station. Both of these are discussed in Section 6.

The software design for the system consists of two key challenges—designing the communication protocol for collecting data and designing the motion control algorithms.

5.2 Challenges in Communication Protocol Design

In a realistic system, there are certain constraints on the path of the robot and it may not be able to reach within direct radio range of each static node due to several reasons. First, the traction platform may not be able to move over the complete deployment terrain due to the presence of obstacles. Second, the mobile node may not know the location of all the sensor nodes to determine a path which intersects the radio range of all the deployed nodes. Finally, implementing the autonomous robot navigation on arbitrary terrain is known to be a hard problem. Thus, we use a preexisting path through the deployment environment for the robot. Hence, the robot may not reach within communication range of each static node. The communication protocol must allow those static nodes which are never within direct radio range of the mobile node to transfer their data using a multihop path to the closest location of the mobile node's path. We refer to this step as local multihop routing. These local multihop paths are much shorter than the paths in a static network and the relay overhead is much smaller.

TABLE 1
Description of Design Choices in Motion Control and the Effect of Precaching

Mobility Strategy	Without Pre-caching	With Pre-caching
Mobile stops to service nodes	Unpredictable delays Lower memory requirements More time available for local multihop	Unpredictable delays Higher memory requirements Local multihop occurs before mobile reaches
Mobile moves with fixed trip latency	Predictable delays Lower memory requirements Less time available for local multihop	Predictable delays Higher memory requirements Local multihop occurs before mobile reaches

Another issue in communication protocol design is that the local multihop paths from the static nodes to the mobile node exist only for a short duration, and during this time, all nodes which have active paths to the mobile node will attempt to transfer their data. This can cause potential congestion and MAC collisions. However, partial data routes leading up to the nodes which come within direct connectivity with the mobile node exist even at other times when the last hop to the mobile node is not active. To save data transfer time, static nodes, which do not come within direct range of the mobile node, could transfer their data to nodes close to the mobile node's path, even before the connection with the mobile node is established. We refer to this process as *precaching*, i.e., data from nodes farther off from the mobile node's path is cached using local multihop paths before the mobile node arrives for data collection. We must design a communication protocol which enables the above steps.

5.3 Challenges in Motion Control

Since the mobile node in our system is controlled, we can optimize its motion to enhance the data gathering performance. While the path of the mobile node is fixed, the exact speed profile followed along the path is still flexible. The network connectivity may be different in different network regions due to several reasons and may also vary with time. Practical deployments, such as the experimental sensor networks at James Reserve [2], monitoring ecological and biological parameters, rarely have uniformly deployed sensor nodes. Deployments are determined by field experts and nodes are concentrated at locations near phenomena of interest. Also, the deployment environment is cluttered with several environmental entities. The effects of these factors on the network layer connectivity are:

1. On certain stretches along the path, the mobile node may not have connectivity to any static node.
2. The number of nodes with connectivity may be different in different regions along the path, limiting the data transfer in denser regions due to MAC collisions.
3. The wireless channel may be variable and the presence of environmental obstacles and multipath effects may cause the quality of connectivity to be vastly different in different regions along the path.

To overcome these effects, the mobile node may follow a stop and go strategy so that it stops at several locations along the path to collect data from static nodes. However, this may cause the total time spent for each path traversal to be variable, while a predictable time may be desirable to effectively schedule sleep and power save modes at the static nodes. A fixed low latency path may also be desirable to bound the buffer requirements at static nodes. Suppose the mobile spends a time T_{rc} for recharging after traversing the path once. Suppose the buffer available at static nodes is such that it accommodates data sampled over a period T_{buf}, both locally at the node and from nodes for which this data is precached. Then, the mobile node must complete a path traversal within a duration $T = T_{buf} - T_{rc}$. In such a case, data delivery would have to be maximized within the fixed latency. The speed at which the mobile moves also affects the frequency of its visits and this directly impacts both the achievable data rate and the buffer requirements at the static nodes. A summary of these issues is captured in Table 1. We design a motion control strategy which attempts to achieve a fixed trip latency but maximize data delivery by adapting the speed profile to address network variations.

6 NETWORK LAYER ALGORITHMS

This section describes the network algorithms developed for the system to address the communication protocol challenges mentioned earlier in Section 5.2. The data transfer process is carried out in three phases, as shown in Fig. 7 and discussed in detail below.

6.1 Cluster Formation

The first phase is the formation of clusters within the network. It is a training phase and essential information required for subsequent phases is collected in this phase. This phase lasts only for the duration of a single path traversal by the mobile node and the training data is then used in several subsequent traversals. This training phase may be periodically repeated to adapt to system dynamics (for instance, node additions, deletions). The method used for forming the clusters is described below.

As mentioned earlier, the path followed by the mobile is fixed. In the first path traversal, the mobile node moves along this path, broadcasting BEACON packets periodically. The mobile node is denoted to be at level $L_g = 0$ in the

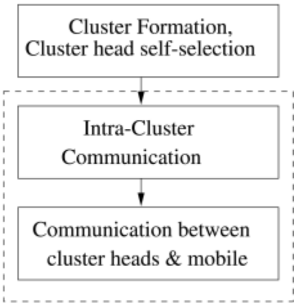


Fig. 7. Three phases in the data collection process. The first phase is used for training only, while the other two continue to operate in parallel.

clusters being created for data collection. The BEACON packet contains three fields:

$$\{SenderIdentity, L_g(sender), Clusterhead\},$$

where $SenderIdentity$ is the identity of the node broadcasting the BEACON packet, $L_g(sender)$ is the value of L_g it has assigned itself, and $Clusterhead$ is the identity of the cluster head for the cluster to which this node belongs. The value of $Clusterhead$ is set to 0 in the BEACON packet broadcast by the mobile node. All the static nodes begin with their level $L_g = \infty$ and follow Algorithm 1.

Algorithm 1: Algorithm for training phase at static nodes (at node with identity = ID.

```
Initialize: $L_g(ID) = \infty$.
while TRUE
  Listen for packets
  if Receive BEACON packet
    if BEACON{$L_g(sender)$} + 1) < $L_g(ID)$
      $L_g(ID)$ := BEACON{$L_g(sender)$} + 1
      if $L_g(ID) == 1$
        $Clusterhead(ID) := ID$
      else
        $Clusterhead(ID)$ := BEACON{$Clusterhead$}
      Set BEACON = {$ID, L_g(ID), Clusterhead(ID)$}
      Broadcast BEACON
```

This algorithm leads to the assignment of a level number L_g at each node which represents its shortest hop count from the mobile node's path and the assignment of a cluster head at each node. Nodes which receive BEACON packets directly from the mobile node will get assigned $L_g = 1$ and these nodes will assign their own ID as the identity of the cluster head within their cluster.

A cluster creation process in progress is shown in Fig. 8. When the BEACON broadcast is started, some nodes which lie close to the mobile node's path but have not yet received the BEACON packet from the mobile node may receive BEACON packets from nodes which have already received the BEACON packet. As a result, they may initially assign themselves a higher value of L_g than the best possible value, but this will get corrected when the mobile node reaches within their connectivity range and such nodes receive the BEACON packet directly from the mobile node. Similarly, all nodes will assign themselves the lowest value of L_g based on the smallest L_g value received in any BEACON packet and, hence, become part of a cluster which leads to the shortest path to the mobile node. For instance, the node with $L_g = 3$ in the figure will receive a BEACON packet from a node with $L_g = 1$, as represented by the dotted line, and, hence, update its level to 2 and become part of cluster C instead of cluster B.

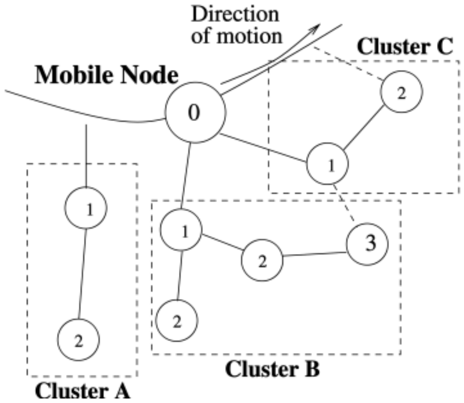


Fig. 8. Cluster formation process in the training phase. The circles represent nodes and the number within the circles represents the value of L_g assigned at the current instant. The dotted lines represent BEACON receptions yet to take place.

At the end of the first phase, the static network will be divided into multiple clusters with exactly one cluster head in each cluster. Certain clusters may consist of only a single node which is itself the cluster head.

6.2 Intracluster Communication

The second phase is designed to address the issue of ensuring that partial routes to the mobile node are exploited for transferring data even when the complete route from a static node to the mobile node is not active, so as to save communication time when the mobile node is within the connectivity range. This process is called precaching and is carried out using a protocol based on the directed diffusion protocol described in [39]. The directed diffusion protocol is summarized briefly. The sensor nodes in the network are divided into data sources and one or more data collection devices, called sinks. A sink expresses interest in a particular sensor data satisfying some constraint. This interest is broadcast from each sink periodically, containing the constraint on data and a time to live (TTL) field. The TTL field denotes the number of hops for which the interest will be rebroadcast. The nodes which hear the interest store it in their interest cache and rebroadcast it after decrementing the TTL field by 1. As the interest propagates through the network, it establishes reverse paths toward the sink, along which the data in response to the interest will be returned to the sink. These reverse paths are denoted as gradients and are optimized over the course of their usage to adapt to runtime dynamics and network loads. Whenever a node collects new data, it will check its interest cache, and if this data matches any of the constraints received in an interest, it will send the data along the corresponding reverse path. The packet is thus relayed to the base station. To handle network dynamics and changing topologies, there is an expiration time for the interests at the sources (within which duration it should have heard a new interest message being broadcasted periodically by the sink).

In our system, the cluster heads act as sinks. It may be noted that the cluster heads are also generating data. Each cluster head stores the data generated locally and the data collected using diffusion from the nodes within its cluster. Each node responds only to the interest message originated from the cluster head of its own cluster and, hence, diffusion within each cluster occurs independently of diffusion in other clusters. The end result is that data collected at static nodes continues to rapidly reach the cluster heads along the reverse paths found by the directed diffusion protocol within each cluster and is ready to be transferred to the mobile node when such connectivity is available. In our prototype, we use an all-inclusive interest, which means that all samples collected by a sensor node are to be delivered to the mobile node.

There are other choices for the communication protocol used in networks where the complete path does not exist continuously, such as the Delay Tolerant Networking (DTN) protocol described in [40]. In DTN, packets are grouped together into bundles and the complete bundle is transferred to the next custodian node. The custodian is a node which accepts the responsibility of eventual delivery to the final destination. This approach is useful when several hops along the route are intermittent. Our mechanism is conceptually similar to the DTN approach, with the next custodian for each node being the cluster head. However, while the DTN protocols have been implemented on Internet-class networks, our network protocol is implemented on lower resource embedded nodes.

6.3 Communication between Cluster Heads and the Mobile Node

The next step for a data packet to be delivered is the transfer from a cluster head to the mobile node. This step must occur along an intermittently available link. For this data transfer, a key requirement is to determine when the connectivity between a cluster head and the mobile node is available. Communication should start when the connection is available and stop when the connection no longer exists, so that the cluster head does not continue to transmit data when the mobile node is no longer receiving it.

We do not use information about the location of the static nodes and the mobile node because such location data may not be available in all systems. Further, even when location data is available, the radio connectivity calculated using a simple range-based model may not correspond exactly to the real connectivity observed in the network due to the presence of wireless channel variations and multipath effects.

We solve the above problem by using an acknowledgment-based protocol between cluster heads and the mobile node. The mobile node, in all subsequent path traversals after a training phase, periodically broadcasts a POLL packet, announcing its presence and soliciting data as it proceeds along the path. The POLL is transmitted at fixed intervals, T_{poll}, and the time between these transmissions is used for data communication. This POLL packet is used by the cluster heads to detect when the mobile node is within connectivity range. The cluster head which receives the POLL will start transmitting data packets to the mobile node. The mobile node acknowledges each data packet from a cluster head. This acknowledgment helps the cluster head to realize that the connection is active and the data was reliably delivered. The acknowledged data packet can then be cleared from the cluster head cache. The data packet to be transmitted is chosen from the data cache at the cluster head in the following manner. The cluster head stores the data packets received in a separate queue for each node in its cluster. It also stores its own locally generated data in another queue. Whenever a packet is to be transmitted, one packet is chosen from these queues in a round-robin fashion and, hence, each nonempty queue is served sequentially. Once the cluster head transmits a data packet to the mobile node, it starts two timers:

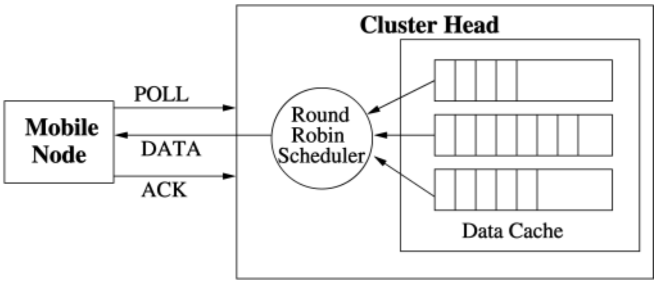


Fig. 9. Communication between the mobile node and the cluster heads. Each packet queue at the clusterhead corresponds to data packets received from some static node within the cluster.

1. *Retransmit Timer:* The unacknowledged data packet is retransmitted when this timer expires. These retransmissions help overcome the effect of packet losses due to channel errors and MAC layer collisions with other cluster heads with simultaneous connectivity. The value of this timer is higher than the round trip delay for receiving an acknowledgment over the CSMA MAC protocol used on the channel.
2. *Connection Lost Timer:* The cluster head stops retransmitting the packet to the mobile node if it does not receive a POLL message broadcast from the mobile node before this timer expires. This helps the cluster head to detect when the mobile node has moved away and connectivity is no longer available. The value of this timer is set equal to $3 \times T_{poll}$. This redundancy of 3 ensures that the connection is not assumed lost if up to two POLL messages are missed due to channel error or MAC collision.

The above process is summarized in Fig. 9. While the figure shows only one cluster head, multiple cluster heads may be simultaneously communicating with the mobile node and a CSMA MAC protocol is used on the channel.

7 ADAPTIVE MOTION CONTROL

We now describe the algorithms used to control the motion of the mobile node. This is a higher layer activity which uses information about the data collection performance from the network protocols. The network protocols described in the previous section continue to operate in parallel with the motion control algorithms described here.

Motion can be controlled in two domains: space and time. Control in space refers to the exact trajectory followed by the mobile node, which, in our system, is fixed due to the reasons described in Section 5.2. Here, we address the control in time, which refers to the exact speed profile of the mobile node along the trajectory. The trajectory is a closed path which the mobile node repeatedly traverses. The key design consideration here is to traverse the path within a specified latency constraint and attempt to maximize the data delivered as discussed in Section 5.3.

Let the latency constraint specified be T. Given T and the length of the path, L, the speed at which the mobile node moves can be naively set to $s = (L/T)$. However, even within the latency constraint, we can adapt the node speed according to the network conditions to improve the data collection performance. For instance, the mobile node can spend more time in regions where the network is dense or the channel is obstructed and less time in regions where there are no or few nodes in connectivity. This observation is the key insight to our speed control algorithm. The process consists of two steps: determining the congested regions and allocating additional time in these regions to the correct proportion.

7.1 Selecting the Congested Regions

To ensure practical applicability, we design an algorithm that is not based on the idealistic radio disc model. This model does not hold in practice, as is observed in several experimental studies such as [41]. We also do not use geographic information about the location of the static nodes as location may not be available or the error in the location estimate may well be of the same order of magnitude as the radio range of the static devices.

The congested regions are not represented using geographic location, but using node identities. The mobile node records each distinct node identity it receives data from. Note that the identity of the node which collected the data is preserved in the data packet even when it is precached at the cluster head. A fixed number, k, of the total nodes is selected dynamically for each path traversal to be in the congested set. There are three reasons for a node to belong to this set. The node (or its cluster head, if it is not on the path of the mobile):

1. was in the connectivity range with the mobile node for a very small duration,
2. is located in a high density region, where the bandwidth is shared among several nodes and the data rate is further reduced due to MAC layer collisions, or
3. suffers from a poor wireless channel.

The consequence of all the above effects is similar. In addition, if the size of the cluster this node belongs to is big (the cluster head does round-robin), congestion results. The congestion set is identified by counting the number of data packets received from each node at the end of one path traversal. The node identities are sorted by the number of packets received from each and the k nodes with the lowest packet count are selected to be in the congested set. The congested set discovered in each path traversal is used to control the motion in the next traversal.

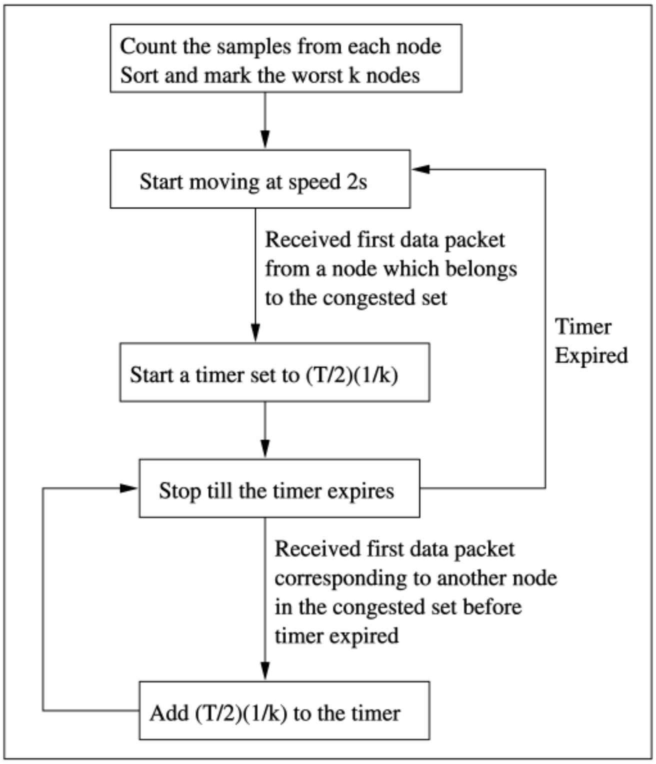


Fig. 10. Adaptive motion control algorithm used on the mobile node.

7.2 Differential Time Allocation

Instead of moving at the naive speed s required to satisfy the latency constraint, if the mobile node moves at a faster speed $2s$ in some regions, it will have extra time available for congested regions. At speed $2s$, the mobile can complete a path traversal in time $T/2$. This provides an extra time $T/2$ for servicing congested regions. This time is allocated among the k congested nodes as follows: When the mobile node receives a data packet corresponding to one of the nodes in the congested set, it stops and initiates a timer for time T_{stop} given as:

$$T_{stop} = \frac{1}{k}(T/2). \tag{11}$$

While stopped, if the node receives a data packet corresponding to another node in the congestion set, it adds an additional duration of T_{stop} to the running value of the timer. Thus, in regions where multiple congested nodes are present, the mobile node ends up spending a longer period of time. The precise algorithm running on the mobile node in each path traversal (after the initialization phase of cluster formation) is shown in the flowchart in Fig. 10.

It may be recalled from the previous section that the first path traversal is used for cluster formation, wherein no data is collected. As a result, for the second traversal, there is no history to begin with and the congested set is not known. The congested set can be chosen randomly for the second traversal. Data collected in the second traversal is then used to find the congested set for the third traversal and so on for all subsequent traversals.

To visualize the effect of this motion control algorithm, we plot the time spent at different regions along the path of the mobile node for one traversal (after the training phase

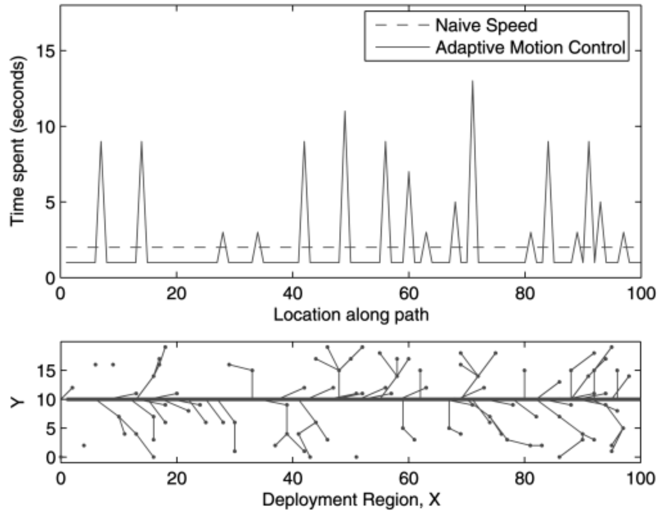


Fig. 11. Visualizing the adaptive speed control strategy for a sample run on a random topology. The X-axis in both subplots shows the distance along the horizontal dimension of the deployment region and is at the same scale for exact correspondence. Some nodes stay disconnected as the topology is randomly generated.

and after a congestion set has been acquired for at least one traversal) in Fig. 11. The lower subplot shows the nodes in the random network deployment in an area of $100m \times 20m$, around the path of the mobile node. The connections among the nodes show the clusters formed. The horizontal line in the middle of the lower subplot represents the fixed path of the mobile node passing through the network deployment and the mobile moves from left to right along this path. Lines are also shown from nodes which are cluster heads to the first point along the mobile node's path where connectivity is expected. The radio range is $r = 5$, while several nodes are farther than r from the mobile node's path and, hence, have to use cluster heads for forwarding their data. The upper subplot shows the time spent by the mobile node along the path in different regions. The distance is $100m$, and the latency T is $200s$. This leads to a naive speed of $50cm/s$. At naive speed, the time spent is equal in all regions throughout the path traversal. This is shown by the dashed line. With adaptive speed, the time spent is increased in certain regions. The X-axis is the same for both the subplots. It may be observed that the regions where the mobile node spends more time correspond to stretches along the path where several cluster heads or cluster heads of larger clusters are connected to the mobile and, hence, more data is required to be transmitted.

Note: The algorithm, as presented, assumes that all the nodes are sampling at the same rate and, thus, have an equal amount of data to send. When this is not the case, such as when each node samples at a different sampling rate or when the interest is limited to only a subset of the data, the total number of packets a node intends to send is also included in each data packet. Using that value and the number of samples actually received from a node, the mobile node can calculate the percentage of samples it collected in a path traversal from each node. This percentage can then be used to form the ordering to select the k nodes.

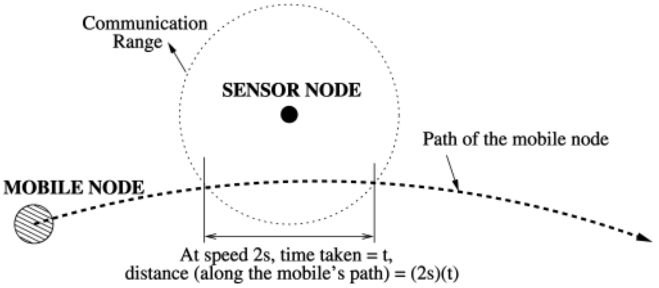


Fig. 12. Time spent by the mobile node in the nominal communication range of a static node depends on what length of its path intersects with the radio range of the static node.

7.3 Speed Control Options

In the algorithm presented above, the mobile node can be in either of only two states: moving at speed $2s$ or stopped. Another option may be considered for speed control, based on the use of more speed states. For instance, the nodes could be divided into three classes, K_1, K_2, and K_3, each containing k_1, k_2, and k_3 nodes, respectively, based on the amount of data[3] received at the mobile from the nodes in increasing order. The mobile node could then stop for nodes belonging to class K_1, move at an intermediate speed s for nodes belonging to the next better class K_2, and move at speed $2s$ for the best k_3 nodes. We show that no advantage may be gained by exploiting multiple speeds.

Consider a node (belonging to class K_2), its nominal transmission range, and a small segment on the path of the mobile node, as shown in Fig. 12. Suppose when the mobile node is moving at speed $2s$, it is in the connectivity range for time t. Thus, a length $2st$ of the mobile node's path passes within the connectivity range of this node. Suppose t_2 is the time alloted to each node in class K_2, i.e., when the mobile encounters a node in K_2, it will slow down to speed s and move at this speed for duration t_2, covering a distance st_2. If st_2 is smaller than the total distance for which connectivity is available, $2st$, then the increased time spent in range is given by the time spent in moving the distance $2st$ at speed s for time t_2 and at speed $2s$ for the remainder of the distance $2st - st_2$. This time, $t_{connected}$, is given by

$$\begin{aligned} t_{connected} &= t_2 + \frac{2st - st_2}{2s} \\ &= t_2/2 + t. \end{aligned} \tag{12}$$

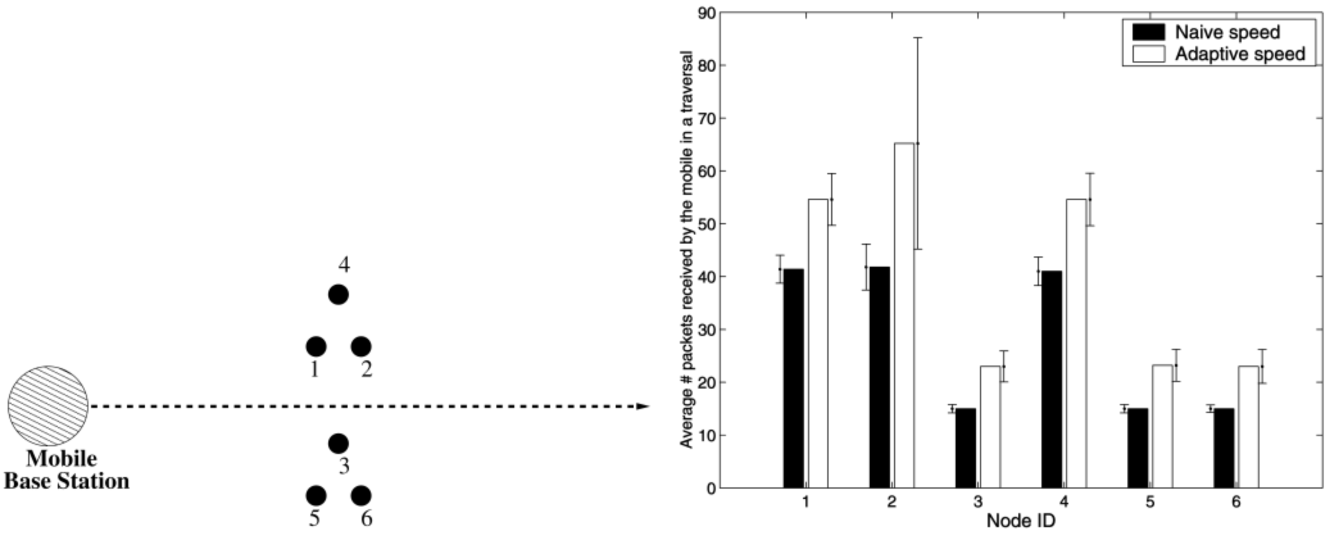
If, on the other hand, st_2 is greater than $2st$, then the time spent in the connectivity range is the time taken to cover a length $2st$ at speed s, given by:

$$t_{connected} = 2t. \tag{13}$$

Since st_2 is greater than $2st$ here, the value of $t_{connected}$ in (13) is smaller than that in (12). Now, for both of these cases, the extra time spent by the mobile node for carrying out the above slowing down procedure, compared to moving at speed $2s$ throughout, is $t_2/2$.

---

[3] Assume all nodes sample at the same rate; otherwise, percentages can be used.



(a) (b)

Fig. 13. Experimental results. (a) The network topology used for experiments. The topology emulates the presence of regions with high node density and regions with no nodes along the mobile node's path. (b) Experimental results comparing the performance of adaptive motion control with naive speed motion, using the network topology shown in (a).

As an alternative, consider moving at speed $2s$ and stopping for the same extra time spent $t_2/2$ when in the connectivity range of this node. For this alternative, the mobile node is in the connectivity range for $t + (t_2/2)$. This quantity is same as the value of $t_{connected}$ in (12) and greater than that found in (13). Thus, the mobile node can potentially collect as much or even more data from this static node when using only the stop and fast states instead of using the extra state of intermediate speed. Thus, using only two motion states to allocate differential time in congested regions is at least as good as or even better than using the extra speed. A similar argument can be drawn for the use of four or more speeds.

8 EXPERIMENTAL RESULTS

This section describes the evaluation of the prototype system and algorithms mentioned above. We created a topology as shown in Fig. 13a. The mobile base station moves on the path indicated. The path length is 12m in one direction. The path traversal latency is fixed at 1 minute and the naive speed used is $s = 20cm/s$ and, hence, $2s = 40cm/s$ is used as the fast speed for adaptive motion control. The static nodes were generating data at 1 packet per second.

The mobile node is made to traverse the path multiple times, first using the naive speed and then using the adaptive motion control algorithm discussed in the previous section. A traversal is defined as the mobile node covering the path in the forward direction (left to right in the figure) only. There was no data collection when the mobile node moved in the reverse direction as we assumed that, in a realistic setting, the ending point of the path would overlap with the starting point, forming a closed loop, and, hence, the reverse travel would be eliminated.

We compare the amount of data collected per traversal using the two motion control strategies. The results are shown in Fig. 13b. The X-axis shows the node ID, corresponding to Fig. 13a. The Y-axis shows the average number of packets collected by the mobile base station per path traversal. Note that while the total traversal time is the same for both motion strategies, the data collected is increased for all nodes with the adaptive motion control.

9 SIMULATION RESULTS

This section describes the simulation setup we used in the process of system development. In addition, we present simulation results involving the trade-off between speed and buffer size.

The simulator used is TOSSIM [42]. This simulator was used as a development tool for the communication protocol and motion control software as the same code which is executed by the simulator can be executed on the mote hardware. In our system prototype, the only difference is that the code for the mobile node is executed on a Stargate. TOSSIM does not allow different programs on different nodes, while we have to use a separate code on the mobile node from the static ones. To work around this limitation, we have both the mobile node and static node program developed together and each node branches to the appropriate section based on the node's identity (the identity of the mobile node is fixed to enable this). To simulate mobility, we use tython [43]. Tython allows modifying the TOSSIM deployment scenario, such as dynamically moving nodes, and we use this feature to simulate the motion of the mobile base station. With this, the connectivity of the nodes with the mobile base station changes dynamically as the mobile base station moves.

As was noted in Section 5.3, the amount of buffer space required at the static nodes has a dependence on the speed of the mobile node. If the mobile moves at a faster speed and makes more frequent data collection trips, a smaller

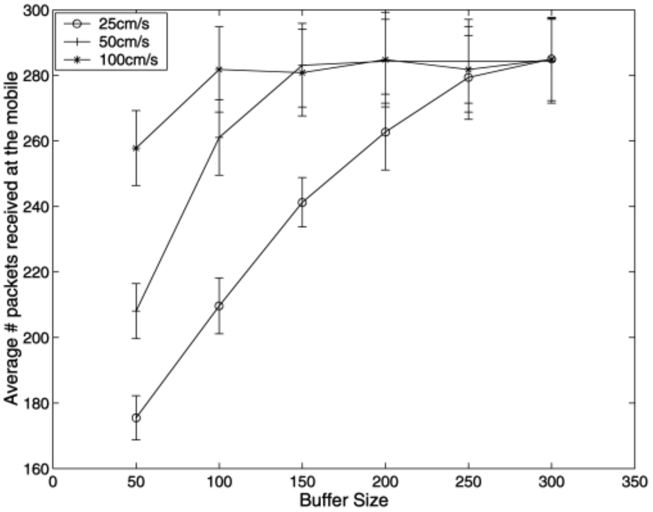


Fig. 14. Simulation results showing the trade-off between buffer size and speed. The error bars represent the standard deviation observed in the results averaged over 20 random topologies.

buffer is expected to be required at the static nodes and, in particular, at the cluster heads. We study the performance of our proposed communication protocol at different speeds for the mobile node and at different buffer sizes at the static nodes. The simulation set up consists of 50 nodes placed in a $100m \times 100m$ area. The transmission radius was set at $r = 25$ to ensure node connectivity with high probability at the above node density with random deployments. The simulations are averaged over 20 random topologies each. The mobile base station moved from $(50, 0)$ to $(50, 100)$, vertically traversing the square deployment region. On reaching the top, it would immediately come back to the starting position as if the path is a closed loop. The nodes are generating data at the rate of one packet per second.

Fig. 14 shows the results obtained. Three speeds of 25cm/s, 50cm/s, and 100cm/s are considered. The X-axis shows the buffer size on the static nodes. This is in units of space required to store one packet worth of data. As the noncluster head nodes would be continuously sending data to their cluster heads, this buffer space is of relevance for cluster heads only. Also, this buffer space indicates the size of each queue shown in Fig. 9 in the cluster head. The Y-axis shows the average number of packets received per node at the mobile base station per 400s (traversal time at slowest speed), averaged over multiple traversals. Since the total time for data collection is kept constant at 400s at all speeds, the number of traversals varies with the speed used, allowing one traversal at 25cm/s, two traversals at 50cm/s, and four traversals at 100 cm/s.

We see that a small buffer is sufficient to collect the same amount of data at higher speeds than was collected with a much larger buffer size at slower speeds. This reduction in the buffer size requirement comes at a cost: The mobile moves a longer distance, which, in turn, costs more energy. However, equipping the static nodes with a larger memory requires a larger energy drain at the static nodes. Hence, if the mobile base station has a higher energy capacity or rapid recharging is available, a smaller buffer would be preferred for the static nodes.

10 CONCLUSIONS

In this paper, we explored the possibility of using a mobile device for data transfer in sensor networks. The motivation for doing this was to save energy in the embedded sensor nodes and increase the useful service time of a deployed system. The key intuition was that using a mobile node to establish shorter data routes reduces the data relaying overhead, especially at the nodes close to the data egress point in the network. We showed analytical calculations which help develop this intuition further. We then described many of the challenges in the design of such a system, both for networking and for motion control, and presented the design choices made in our prototype. The network protocols we designed can even be used in other systems which do not use controllably mobile nodes but rely on available mobile elements in the deployment environment which follow approximately fixed trajectories, such as public transportation vehicles. The energy and lifetime advantages may come at a cost of increased data latency in many cases and this approach is advantageous for applications where latency is acceptable.

An important conclusion to be drawn from this investigation is that, for the same total amount of data communicated, moving data physically may sometimes be more energy efficient than transmitting it over a wireless route, as was shown in Fig. 2, even when the energy consumed by the mobile node for locomotion is provided by the deployed system itself. Further, this approach changes the points of energy consumption in the system and makes adding energy to the system, on a long term basis, easier than in a static network with no mobile components. Energy expenditure at a small number of mobile nodes is used to reduce the energy expenditure in a much larger number of sensor network components. Thus, the total energy may not be provided up front, but is added to the system over a period of time, such as through environmental energy harvesting. We expect this approach to be advantageous in many long term sensor network deployments.

ACKNOWLEDGMENTS

This material is based upon work supported by the UCLA Center for Embedded Networked Sensing (CENS) and the US National Science Foundation (NSF) under Grant Nos. ANI-00331481 and CNS-0306408. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the US National Science Foundation.

REFERENCES

[1] W.J. Kaiser, G.J. Pottie, M. Srivastava, G.S. Sukhatme, J. Villasenor, and D. Estrin, "Networked Infomechanical Systems (NIMS) for Ambient Intelligence," Technical Report 31, Center for Embedded Networked Sensing, Univ. of California, Los Angeles, 2003.

[2] A. Cerpa, J. Elson, D. Estrin, L. Girod, M. Hamilton, and J. Zhao, "Habitat Monitoring: Application Driver for Wireless Communications Technology," *Proc. ACM SIGCOMM Workshop Data Comm. Latin Am. and the Caribbean,* 2001.

[3] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, "Wireless Sensor Networks for Habitat Monitoring," *Proc. ACM Int'l Workshop Wireless Sensor Networks and Applications (WSNA),* 2002.

[4] R.C. Shah, S. Roy, S. Jain, and W. Brunette, "Data MULEs: Modeling a Three-Tier Architecture for Sparse Sensor Networks," *Proc. IEEE Workshop Sensor Network Protocols and Applications (SNPA),* 2003.

[5] T. Small and Z. Haas, "The Shared Wireless Infostation Model—A New Ad Hoc Networking Paradigm (or Where there Is a Whale, there Is a Way)," *Proc. ACM MobiHoc,* 2003.

[6] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. Peh, and D. Rubenstein, "Energy-Efficient Computing for Wildlife Tracking: Design Tradeoffs and Early Experiences with Zebranet," *Proc. Architectural Support for Programming Languages and Operating Systems (ASPLOS),* 2002.

[7] A. Chakrabarti, A. Sabharwal, and B. Aazhang, "Using Predictable Observer Mobility for Power Efficient Design of Sensor Networks," *Proc. Second Int'l Workshop Information Processing in Sensor Networks (IPSN),* 2003.

[8] A. Kansal, A. Somasundara, D. Jea, M. Srivastava, and D. Estrin, "Intelligent Fluid Infrastructure for Embedded Networks," *Proc. Second Int'l Conf. Mobile Systems, Applications, and Services (MobiSys),* 2004.

[9] W. Zhao, M. Ammar, and E. Zegura, "A Message Ferrying Approach for Data Delivery in Sparse Mobile Ad Hoc Networks," *Proc. ACM MobiHoc,* 2004.

[10] J. Luo and J.-P. Hubaux, "Joint Mobility and Routing for Lifetime Elongation in Wireless Sensor Networks," *Proc. 24th Ann. Conf. Sponsored by IEEE Comm. Soc. (Infocom),* 2005.

[11] B. Zhang, G. Sukhatme, and A. Requicha, "Adaptive Sampling for Marine Microorganism Monitoring," Technical Report CRES-04-006, Center for Robotics and Embedded Systems, Univ. of Southern California, 2004.

[12] V. Bokser, C. Oberg, G. Sukhatme, and A. Requicha, "A Small Submarine Robot for Experiments in Underwater Sensor Networks," *Proc. IFAC Symp. Intelligent Autonomous Vehicles,* 2004.

[13] A. LaMarca, D. Koizumi, M. Lease, S. Sigurdsson, G. Borriello, W. Brunette, K. Sikorski, and D. Fox, "PlantCare: An Investigation in Practical Ubiquitous Systems," *Proc. Fourth Int'l Conf. Ubiquitous Computing (UbiComp),* 2002.

[14] M. Rahimi, H. Shah, G.S. Sukhatme, J. Heidemann, and D. Estrin, "Studying the Feasibility of Energy Harvesting in a Mobile Sensor Network," *Proc. IEEE Int'l Conf. Robotics and Automation,* 2003.

[15] W. Merrill, L. Girod, J. Elson, K. Sohrabi, F. Newberg, and W. Kaiser, "Autonomous Position Location in Distributed, Embedded, Wireless Systems," *Proc. IEEE CAS Workshop Wireless Comm. and Networking,* 2002.

[16] S. Ganeriwal, A. Kansal, and M. Srivastava, "Self-Aware Actuation for Fault Repair in Sensor Networks," *Proc. IEEE Int'l Conf. Robotics and Automation (ICRA),* Apr. 2004.

[17] R.K. Harle and A. Hopper, "Building World Models by Ray Tracing within Ceiling Mounted Positioning Systems," *Proc. Fifth Int'l Conf. Ubiquitous Computing (UbiComp),* 2003.

[18] M. Grossglauser and D.N.C. Tse, "Mobility Increases the Capacity of Ad Hoc Wireless Networks," *IEEE/ACM Trans. Networking (TON),* vol. 10, no. 4, Aug. 2002.

[19] S. Diggavi, M. Grossglauser, and D. Tse, "Even One-Dimensional Mobility Increases Ad Hoc Wireless Capacity," *Proc. IEEE Int'l Symp. Information Theory (ISIT),* 2002.

[20] H. Dubois-Ferriere, M. Grossglauser, and M. Vetterli, "Age Matters: Efficient Route Discovery in Mobile Ad Hoc Networks Using Encounter Ages," *Proc. ACM MobiHoc,* 2003.

[21] S. Jain, R.C. Shah, G. Borriello, W. Brunette, and S. Roy, "Exploiting Mobility for Energy Efficient Data Collection in Sensor Networks," *Proc. Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt),* 2004.

[22] A. Gamal, J. Mammen, B. Prabhakar, and D. Shah, "Throughput-Delay Trade-Off in Wireless Network," *Proc. 23rd Conf. IEEE Comm. Soc. (Infocom),* 2004.

[23] Q. Li and D. Rus, "Sending Messages to Mobile Users in Disconnected Ad-Hoc Wireless Networks," *Proc. MobiCom,* 2000.

[24] D. Goldenberg, J. Lin, A.S. Morse, B. Rosen, and Y.R. Yang, "Towards Mobility as a Network Control Primitive," *Proc. ACM MobiHoc,* 2004.

[25] A. Somasundara, A. Ramamoorty, and M. Srivastava, "Mobile Element Scheduling for Efficient Data Collection in Wireless Sensor Networks with Dynamic Deadlines," *Proc. 25th IEEE Int'l Real-Time Systems Symp. (RTSS),* 2004.

[26] "MICA2 Wireless Measurement System," datasheet: http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/6020-0042-0%5_A_MICA2.pdf, 2004.

[27] A. Cerpa, N. Busek, and D. Estrin, "SCALE: A Tool for Simple Connectivity Assessment in Lossy Environments," Technical Report 0021, Center for Embedded Networked Sensing (CENS), Univ. of California Los Angeles, Sept. 2003.

[28] "CC1000 Single Chip Very Low Power RF Transceiver," product datasheet, http://www.chipcon.com, 2004.

[29] B. McBride, R. Longoria, and E. Krotkov, "Measurement and Prediction of the Off-Road Mobility of Small Robotic Ground Vehicles," *Proc. Performance Metrics for Intelligent Systems (PerMIS),* 2003.

[30] L. Kleinrock and J.A. Silvester, "Optimum Transmission Radii for Packet Radio Networks or Why Six Is a Magic Number," *Proc. IEEE National Telecomm. Conf.,* 1978.

[31] T. Hou and V. Li, "Transmission Range Control in Multihop Packet Radio Networks," *IEEE Trans. Comm.,* vol. 34, pp. 38-44, Jan. 1986.

[32] F. Xue and P.R. Kumar, "The Number of Neighbors Needed for Connectivity of Wireless Networks," *Wireless Networks,* vol. 10, no. 2, pp. 169-181, 2004.

[33] E.J. Duarte-Melo and M. Liu, "Data-Gathering Wireless Sensor Networks: Organization and Capacity," *Computer Networks (COMNET),* special issue on wireless sensor networks, vol. 43, no. 4, pp. 519-537, Nov. 2003.

[34] "Packbot, The Next Step in Unmanned Tactical Mobile Robots," http://www.packbot.com, 2004.

[35] "X-Scale Single Board Computer and Wireless Networking Platform," http://www.xbow.com/Products/XScale.htm, 2004.

[36] G. Dissanayake, P. Newman, S. Clark, H. Durrant-Whyte, and M. Csorba, "A Solution to the Simultaneous Localization and Map Building (SLAM) Problem," *IEEE Trans. Robotics and Automation,* vol. 17, no. 3, June 2001.

[37] J. Neira, J. Tardos, J. Horn, and G. Schmidt, "Fusing Range and Intensity Images for Mobile Robot Localization," *IEEE Trans. Robotics and Automation,* vol. 15, no. 1, Feb. 1999.

[38] "TinyOS: A Component Based OS for the Networked Sensor Regime," http://webs.cs.berkeley.edu/tos/, 2003.

[39] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks," *Proc. ACM MobiCom,* 2000.

[40] K. Fall, "A Delay-Tolerant Network Architecture for Challenged Internets," *Proc. ACM SIGCOMM,* 2003.

[41] J. Zhao and R. Govindan, "Understanding Packet Delivery Performance in Dense Wireless Sensor Networks," *Proc. First ACM Conf. Embedded Networked Sensor Systems (SenSys),* 2003.

[42] P. Levis, N. Lee, M. Welsh, and D. Culler, "TOSSIM: Accurate and Scalable Simulation of Entire TinyOS Applications," *Proc. First Int'l Conf. Embedded Networked Sensor Systems (SenSys),* 2003.

[43] M. Demmer and P. Levis, "Tython—Scripting for TOSSIM," Network Embedded Systems Technology Winter 2004 Retreat, Jan. 2004.

Arun Somasundara received the BE degree in computer engineering from the Karnataka Regional Engineering College, Surathkal, India, followed by the ME degree in computer science and engineering from the Indian Institute of Science, Bangalore, India, and the PhD degree at the University of California, Los Angeles. His research focused on wireless sensor networks, in particular, the effect of mobility in improving the system performance. He is currently with the Broadcom Corporation in San Jose, California.

Aman Kansal received the BTech degree in electrical engineering and the MTech degree in communications and signal processing from the Indian Institute of Technology, Bombay, in 2001 and 2002, respectively. He is currently pursuing the PhD degree in electrical engineering at the University of California, Los Angeles. He has been an active researcher in the area of voice over Internet, wireless networks, Bluetooth personal access, and pervasive computing. His current research focuses on mobile sensor networks and energy neutral systems for networked and embedded computing. He was a recipient of the Microsoft Innovation Award in 2001 for prototyping a mobile telephony system for airplanes (team effort) and several prizes in technology development at the national level. He was awarded the NTSE scholarship by the Government of India and the graduate research fellowship by the Regents of the University of California. He served as a cochair (organization) of the ACM Mobisys Workshop on Applications of Mobile and Embedded Systems 2004 and as a reviewer for numerous conferences. He is a student member of the IEEE.

David Jea received the BS degree in computer science and information engineering from the National Chiao-Tung University, Hsinchu, Taiwan, Republic of China, in 1998 and the MS degree in electrical engineering from the University of California, Los Angeles (UCLA), in 2005. He is currently a PhD student with the Electrical Engineering Deparment at UCLA. His current research interests include calibration of sensor networks and data mules applications. He is a student member of the IEEE.

Deborah Estrin received the BS degree from the University of California, Berkeley in 1980 and the PhD degree from the Massachusetts Institute of Technology in 1985, both in electrical engineering and computer science. She is currently is a professor of computer science at the University of California, Los Angeles (UCLA), where she holds the Jon Postel Chair in Computer Networks and is the founding director of the US National Science Foundation Science and Technology Center for Embedded Networked Sensing (CENS). Dr. Estrin has been instrumental in defining the research agenda for wireless sensor networks. Her research focuses on technical challenges posed by autonomous, distributed, physically coupled systems. She is particularly interested in environmental monitoring applications and is on the National Ecological Observatory Network (NEON) design team. Earlier in her career, she contributed to the design of Internet routing protocols. She is a fellow of the IEEE and a member of the IEEE Computer Society.

Mani Srivastava received the PhD degree in electrical engineering and computer science from the University of California, Berkeley in 1992. Currently, he is a professor in the Electrical Engineering Department at the University of California, Los Angeles (UCLA). He is also associated with UCLA's Center for Embedded Networked Sensing (CENS), a US National Science Foundation Science and Technology Center. Prior to joining UCLA, he worked at Bell Labs Research. His current interests are in embedded sensor and actuator networks, wireless and mobile systems, embedded systems, power-aware computing and communications, and pervasive computing. More information about him and his research group is available at his Networked and Embedded Systems Lab's Web site, http://nesl.ee.ucla.edu. He is a senior member of the IEEE and a member of the IEEE Computer Society.