Title
Intelligent Fluid Infrastructure for Embedded Networking

Authors
Kansal, Aman
Somasundara, Arun
Jea, David C
et al.

Publication Date
2004-05-05

Intelligent Fluid Infrastructure for Embedded Networks

Aman Kansal, Arun A Somasundara, David D Jea, Mani B Srivastava
Department of Electrical Engineering
University of California, Los Angeles (UCLA)
{kansal,arun,dcjea,mbs}@ee.ucla.edu

Deborah Estrin
Department of Computer Science, UCLA
destrin@cs.ucla.edu

ABSTRACT

Computer networks have historically considered support for mobile devices as an extra overhead to be borne by the system. Recently however, researchers have proposed methods by which the network can take advantage of mobile components. We exploit mobility to develop a fluid infrastructure: mobile components are deliberately built into the system infrastructure for enabling specific functionality that is very hard to achieve using other methods. Built-in intelligence helps our system adapt to run time dynamics when pursuing pre-defined performance objectives. Our approach yields significant advantages for energy constrained systems, sparsely deployed networks, delay tolerant networks, and in security sensitive situations. We first show why our approach is advantageous in terms of network lifetime and data fidelity. Second, we present adaptive algorithms that are used to control mobility. Third, we design the communication protocol supporting a fluid infrastructure and long sleep durations on energy-constrained devices. Our algorithms are not based on abstract radio range models or idealized unobstructed environments but founded on real world behavior of wireless devices. We implement a prototype system in which infrastructure components move autonomously to carry out important networking tasks. The prototype is used to validate and evaluate our suggested mobility control methods.

Categories and Subject Descriptors

C.2.4 [**Computer Communication Networks**]: Distributed Systems. C.2.1 [**Computer Communication Networks**]: Network Architecture and Design

General Terms

Algorithms, Design, Experimentation

Keywords

Controlled mobility, sensor networks, mobile router, data gathering

ACM MobiSYS'04, June 6–9, 2004, Boston, Massachusetts, USA.

1. INTRODUCTION

Mobile devices have traditionally been supported in wireless networks for allowing users with portable devices [27]. These methods are geared towards interactive computers where users carry their computing devices with them. Newer research efforts, especially those in the past decade [1,2], have started focusing on proactive computers where most of the components of the system are embedded in the user's environment and work autonomously, such as sensor networks. They may carry out pre-assigned tasks or work by anticipating user needs at run time. The resource constraints and design considerations in such systems differ significantly compared to legacy systems because of the need for a heightened level of autonomy and deeply embedded operation. Mobility in such systems takes on a completely different role, and can be actively exploited to enhance system performance. Previous work which considered utilizing mobility for networking, assumed mobility to be outside the control of the communications sub-layer. Mobility was either treated to be random, following partially predictable patterns or even deterministic but not controlled. This paper follows from previous work such as [4, 5, 6, 7, 8, 29, 36], which made use of external mobility for improving network performance. However, we introduce the new element of deliberate control over mobility which leads to significant performance improvement. We incorporate mobile components into the networking infrastructure, making the infrastructure fluid and demonstrate how this yields practical solutions to some hard networking problems in sensor networks.

2. KEY CONTRIBUTIONS

This paper argues for incorporating controllably mobile components into the networking infrastructure where such inclusion leads to significant performance advantage.

We first show what problems can be solved using mobility in embedded computing. Several research groups have shown that mobility is advantageous for network lifetime and data fidelity [5, 8, 29, 36]. We extend this foundation and motivate the need to add controllable mobile components into embedded wireless sensor networks for more predictable and larger gains. We contrast this deliberate mobility to previous research efforts that have attempted to exploit externally induced random or predictable mobility. While mobility may be used by several applications, the focus in this paper is on network layer functionality.

To illustrate real system parameters that enter the design process, we describe our prototype hardware, which uses mobile infrastructure and battery powered nodes. Our implementation is a sensor network with an autonomous mobile router. To the best of our knowledge this is the first prototype of an embedded network using an autonomous mobile networking device. Using real hardware for our test and evaluation ensures that our methods are directly usable and do not depend on any hidden idealistic assumptions.

Next, we present control primitives that the mobile networking components can use for autonomously managing their motion. These primitives allow the motion to be adjusted to run time dynamics of the system and adaptively improve with growing environmental learning. We outline several design considerations and tradeoffs that exist for mobility control in view of the resource and performance constraints specific to deeply embedded systems. To take a concrete example, we show how the trade-offs are balanced in our working prototype.

We then discuss the design of our communication protocol which enables support for mobile infrastructure and energy constrained nodes with long sleep durations. The communication protocol and sleep time decisions do not rely on idealized radio models such as the unit disc model, but work on the actual radio connectivity available in the specific deployment scenario. Again, we leave no hidden energy costs or latencies in our design by testing them on the prototype system.

The contributions described above serve as the outline for the paper from sections 4 to 8. The next section describes prior work and explicates the novelty of our approach.

3. RELATED WORK

Several research efforts have proved the feasibility of the technology required and illustrated the challenges in realizing the benefits of sensor networks for improving human productivity [1, 14, 33, 37].

One of the basic infrastructure service required for these systems is that of networking the distributed components in the face of severe energy constraints. Several methods to reduce energy consumption have been considered before in the presence of static devices alone [3, 23, 24, 25, 26]. Our approach is orthogonal to these methods. Further, our approach also gives other advantages mentioned later.

More recently, the use of mobility has been explored for improving the networking facilities in the system. We classify the type of mobility used into three categories: random, predictable and controlled. The network layer exploitation of random and predictable mobility has been studied before. Random mobility of all the nodes has been considered for improving data capacity in [30]. An algorithm for routing data among mobile users has been suggested in [4], where data is forwarded to nodes which have recently encountered the intended destination node. Random motion of mobile entities was also used to carry data in [6, 7], where the mobile entities were whales in [6] and zebras in [7]. However, in such cases the latency of data transfer cannot be bounded, and if the data is cleared from the buffer at the mobile agent acting as relay, delivery itself is in jeopardy. In [5, 36] a random walk model was used for the motion of a mobile relay to theoretically derive various parameters of interest, such as delay and data delivery ratio. The use of a predictable mobile agent was considered in [8] where a network node was mounted on a commuter bus and acted as a base station for collecting data from several static nodes along its path. In [29], possibility of changing the trajectories of mobile hosts in a disconnected ad hoc network to transmit messages among hosts is explored.

While some of the advantages of mobility can be realized in the above approaches, the full potential can be realized much more easily using a controlled mobile element. External mobile elements with appropriate trajectories may not be available in all situations. For instance, in current sensor network deployments such as those for ecological research and habitat monitoring [9, 10], no free mobile components are available which the system can use. Control over mobility is also necessary to get control over the reliability of data transfer and providing latency bounds. For mission critical or performance sensitive systems, the design cannot rely on the availability of mobile components from the deployment environment itself. We make the mobile components an integral apart of the infrastructure and give complete control over the mobility capabilities of these components to the infrastructure. In our work we exploit the new opportunities provided by this form of mobility and introduce various design challenges that exist for optimally utilizing it. Our system is thus categorized into the third category of systems that exploit mobility, i.e. systems using controlled mobility for networking, and to the best of our knowledge is the first in its class.

The use of physical mobility for data transfer may increase the latency in some situations. Networks with increased latency have been considered in Delay Tolerant Networking [11, 28] for interplanetary networks, for connectivity in rural regions where network penetration is low [38] or hostile regions where a fixed infrastructure is unsustainable. Our system can be considered as a new class of delay tolerant networks aimed at embedded devices where battery resources are scarce.

4. ADVANTAGES DUE TO MOBILE COMPONENTS

In this section we consider some of the benefits that can be realized using controlled mobility for networking, instead of using multi-hop wireless routing over static embedded nodes.

The first advantage of our approach is an increase in system lifetime. In sensor networks, the sustainable lifetime is severely limited by the battery capacity of the embedded static sensor nodes, as replacement or recharging of batteries is not feasible, either because physically reaching the nodes is not possible, or because the cost of carrying out the operation is prohibitive. A mobile agent can help conserve energy at the static embedded nodes by reducing the number of packets transmitted by the embedded nodes. Consider a sensor network with static nodes only. Data from all nodes to the user location travels over a multi-hop wireless path with intermediate nodes relaying the data towards the destination. The radio is a major energy consumer in embedded sensor nodes [34]. The energy spent in relaying traffic could be saved if data were transmitted over fewer hops. This can be achieved by having a mobile device reach locations close to the sensor nodes such that data can be sent to the mobile device over a very small number of hops. The mobile device can then travel to the user end and deliver the data. The mobile device is not energy constrained because it can return to base and recharge itself. But the embedded sensor nodes or any static infrastructure provided for these nodes is energy constrained, and through saving energy at these nodes we are able to increase the system lifetime.

To get a quantitative estimate, we consider a sample topology, shown in Fig 1. The topology shown only represents the connectivity of the nodes and not their geographic locations. On the left is the topology with no mobile infrastructure. The user node where all data is received is labeled 'Base'. Paths from various nodes to the sink have hop counts between one and five. Now consider a fluid infrastructure along the dashed line as shown in Fig 1 on the right. The dashed line shows the path of the mobile device traversing the network. Now, all nodes can reach this infrastructure over one or two hop paths.

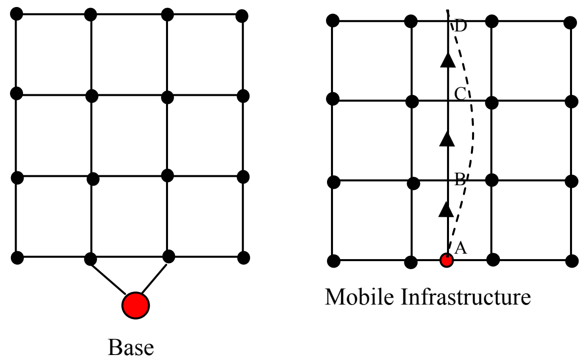


Figure 1. Sample network topology considered to simulate the advantage due to mobile infrastructure.

For getting an exact number of packets transmitted for communication from all the nodes to the sink, we assume a specific data gathering protocol to be running on the network: directed diffusion [12]. In this protocol an 'Interest' message is broadcast by the sink and propagated by the nodes which hear it. All nodes respond with data to this interest message via the nodes from which they heard the interest. Diffusion specific parameters were: interest transmission period = 60s, interest expiry timer = 75s, sample generation period = 5s, samples per data packet = 4.

We simulated the above topologies in TOSSIM [13] with diffusion. First the directed diffusion protocol was implemented in TOSSIM and used for simulating the static network. For simulating the routing behavior with fluid infrastructure, the mobile device was simulated to be present at four locations, marked A to D in the figure, one after the other. This mimics the connectivity of the mobile router with the static nodes. Time spent at each of the locations A to D was 60 seconds. After stopping at D, the mobile device re-starts from A. The exact communication protocol changes with the presence of the mobile device and we discuss that in a later section. Figure 2 presents the total number of packets transmitted by the nodes at a particular hop distance from the base, both when data is communicated wirelessly using multi-hop paths and when the fluid infrastructure is used.

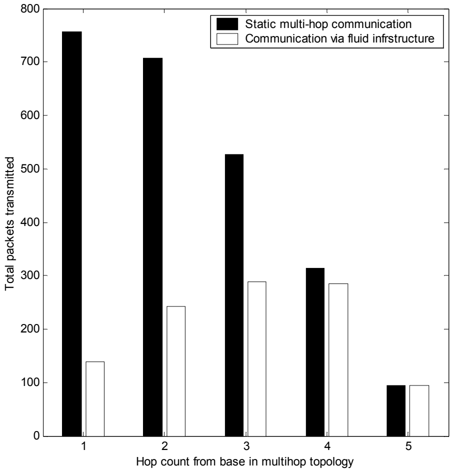


Figure 2. Comparing number of packets transmitted in multi-hop routing and when using mobile router.

The simulation was conducted for 16 minutes. Clearly, the number of packets transmitted in the presence of the fluid infrastructure is significantly lower. As the energy consumed by a node increases with the number of packets transmitted, the resultant gains in system lifetime is substantial.

A second advantage comes in the form of data fidelity. It is well known that the probability of error increases with increasing number of hops that a data packet has to travel. If we reduce the number of hops, this immediately reduces the probability of error. This not only increases the quality of data received but further reduces the energy spent at the static nodes by reducing the retransmissions required due to errors.

A third advantage due to the mobile router is that for certain scenarios data rate can be increased, or latency can be reduced. The reduction in latency may seem counter intuitive as one expects data to travel faster over a wireless link than over a physical mobile device. However, the addition of new resources: that of carrying data physically in a mobile node, increases the network capacity. The following example illustrates this.

Consider a typical sensor node, the mica2 mote [14], which we also use in our prototype. Suppose we have a two hop mote network as shown in Figure 3 with two motes, labeled A and B. The mote radio range is taken to be 10m for this example; actual range varies with environment. The mote has 512KB of flash memory. Suppose we have stored sensor data worth 500KB on each mote, say from sensor readings. The mote radio has a baud rate of 38.4Kbps according to the datasheets [14]. Let us try to give advantage to the multi-hop routing case and assume that an ideal MAC protocol is available such that no bandwidth is wasted in collisions. Also assume that the channel does not have any errors; this gives further advantage to the multi-hop case. Let $T(X,Y,Z)$ represent the time taken to transfer the data generated at node X from node Y to Z. Now, the minimum time, T_{ideal}, taken to transfer data from both the motes to the base station will be

$$T_{ideal} = T(A,A,B) + T(A,B,base) + T(B,B,base).$$

With the mote radio and ideal bandwidth mentioned above, this evaluates to 312.5 seconds. Now consider the case when fluid infrastructure is used. To ensure that we are not using any unrealistic assumptions in this case, we will use only the characteristics of our implemented mobile router, described in section V. Our prototype mobile router has a maximum speed of 388cm/second. Assume we use it at 200cm/s to keep navigation challenges tractable. Since the mobile device is not energy constrained, it has an 802.11 wireless connection with the base. We will assume that this connection is used only when the router is physically present at the base. (Using this higher rate connection from a distance could have given further benefits to the mobile router.) Suppose the mobile router is stationed at the base initially. Let the time taken to move from point X to point Y be denoted $D(X,Y)$. Then the time taken to transfer all the data from A and B to the base is,

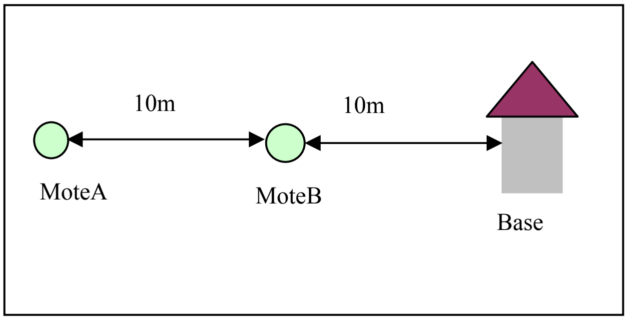


Figure 3. Example topology to compare transmission times for multihop routing and mobile router.

$$\begin{aligned}T_{mobile} = {} & D(base,A) + T(A,A,mobile\_router) + D(A,B) + \\ & T(B,B,mobile\_router) + D(B,base) + \\ & T(\{A,B\},mobile\_router,base).\end{aligned}$$

The time T_{mobile}, using the mote radio parameters and the characteristics of our mobile router, evaluates to 228.8 seconds. Note that this is 83.7 seconds less (26.7% lower) than the case when all the data was sent wirelessly. Thus, the bandwidth of the network has been increased. The data-rate increase would be even more dramatic when the multi-hop network spans several hops and each mote is connected to several neighbors instead of just one. Considering the fact that MAC collisions would waste a non-negligible fraction bandwidth in the multi-hop routing case and the increased probability of errors due to traversing multiple hops would cause more retransmissions than in the mobile router case, the mobile router has a definite win. Experiments show that the manufacturer specified radio data-rate of 38.4Kbps is actually not observed and achieved rates are lower. As the number of radio channel uses is reduced with the mobile router, this gives further advantage to the mobile router. These benefits are a result of the extra data carrying capacity added to the system in the form of physically carrying the data in a mobile node instead of transporting it wirelessly. The extra 802.11 radio for communicating with the base is not important for the increase in data capacity. Even if the same mote radio is used for transferring data to the base from the mobile router, the advantage due to the mobile component manifests itself at 4 or more hops. However, we feel having a higher bandwidth connection between the mobile component and the base is likely in most systems as the mobile device is not energy constrained and a higher bandwidth radio can be used.

Of course, the transport latency advantage discussed above is easily achievable only when the motion of the networking device is controlled by the system itself and no such advantage may be observed when external mobile entities, moving randomly or predictably, are used.

A fourth advantage due to the mobile routing device is that sparse and disconnected networks can be handled. This is useful in several situations. The static devices can reduce their transmission ranges to the lowest value required to reach the mobile infrastructure, even if it fragments the static network topology. This will save transmit energy at the sensor nodes. Another possibility occurs when the sensor network is sensing a phenomenon for which the sensors need to be deployed at such a low density that is not enough to guarantee connectivity. Relationships between sensing range and communication range derived in [24], show that a communication range equal to twice the sensing range is required for ensuring connectivity. Thus in situations where sensing range is larger than the radio range, several relay nodes may be needed just for connectivity. If a mobile component is available, such relay nodes can be done away with and communication energy spent by the embedded portion of the network can be drastically reduced. Another situation where the static network may be fragmented arises when the environment in which the network is deployed has dense occlusions to wireless communication and a reliable multi-hop topology cannot be established. Here, a mobile device can position itself to successfully establish communication with the embedded devices and help them communicate.

There are other advantages of using a mobile component apart from those in networking. We mention some of them briefly here, but do not directly pursue them in this paper. It was shown in [36] that time synchronization error increases with increasing number of hops between two nodes. Since in our design, the number of hops between the base and the embedded sensors is reduced, much finer time synchronization is possible than in a multi-hop case. Security can be enhanced as the data does not traverse multiple hops across potentially compromised nodes. Another use of controlled mobility was recently proposed for calibrating a localization system [22]. Mobile components can also support other system activities such as delivering required resources [15, 32]. These issues illustrate that the use of mobile components is a rich area for exploration in various dimensions of system design. We consider some aspects of using mobility for networking in this paper. Further research is required to fully exploit controlled mobility for enhanced system design.

With the above intuition to motivate the benefits of a fluid infrastructure, we design a working system with an autonomous mobile data gathering device. Since the infrastructure must operate autonomously, we need methods to control the motion of the mobile routing device. Further, power control, topology management and communication strategies are significantly impacted by the fluid nature of the infrastructure. Before presenting our design of the exact algorithms for handling some of these issues, we present our prototype system on which we experimentally illustrate and verify our proposals. Among the numerous network design aspects mentioned above, we focus on data delivery enhancement in this paper.

5. SYSTEM ARCHITECTURE

We assume a network application where several static devices are embedded for measuring a specific environment of interest and a small number of mobile nodes are added to establish a fluid infrastructure and do not themselves collect data about the environment. We add only one mobile node in our prototype network. The implemented system consists of two classes of devices:

1. Static embedded sensor nodes, referred to as motes, which are Crossbow Mica2 sensor nodes [14] in our prototype, and
2. A mobile router, which is our fluid infrastructure connecting the motes to the base. (The base could be an autonomous controller or a human user based system which receives all the data collected by the motes.)

A picture of the prototype system is shown in Figure 4, with the various hardware components marked. The motes are an off-the-shelf sensor node as mentioned above. We describe our construction of the mobile router in more detail. The mobile router is composed of a traction platform, a processor board and a mote. The three components are described below.

5.1 Traction Platform and Control Interface

We use a rugged terrain robot, the Packbot [16] as our mobile platform, with our own control interface. Packbot is a robot developed by iRobot [17] as an unmanned ground vehicle (UGV).

It is suitable for our project because of its belt based traction system allows it to easily maneuver over rocks and rubble, climb up stairs, and navigate narrow, twisting passages. The traction system is geared for our final objective of running it in natural environments such as forests and ecosystem reserves. The current experiments are performed in a more benign environment – a flat courtyard, to minimize navigational challenges allowing us to focus on the networking issues alone.

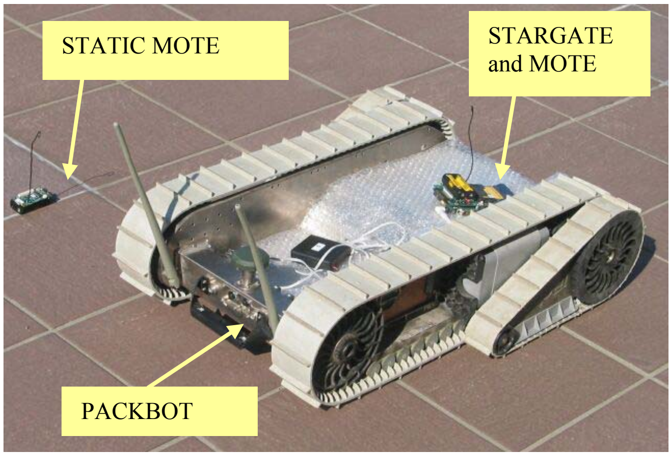


Figure 4. Prototyped system seen in action. The mobile router moves to the static devices for data collection.

The Packbot is commanded by sending it appropriate command packets on its 802.11 interface. The driver for sending the robot specific messages has been integrated into a generic development environment, which we call Simple Interface for Robots (SIR), to assist controlling the robot movement. The control commands are sent by the Stargate controller, described later.

SIR is still an ongoing application interface library. The core design principle is make the interface simple to use and easy to integrate. Unlike the complicated proprietary interfaces that usually come with robots, SIR focuses mainly on controlling the mobility and reduces the API to two functions, robotAdd() and robotAct(). It hides the low level details but still ensures enough functionality required for motion control. Currently SIR runs on two platforms, Linux and TinyOS [19]. It also supports two different robots: Packbot and Amigobot [18].

To operate the Packbot using SIR, specific instructions determined from the robot capabilities are provided: moving forward or backward, rotating clockwise or counterclockwise, and flipping. The instructions are specified with parameters required for it. For example, the SIR command robotAct(id, 'f', 30) asks robot earlier added with identification ‘id’ to execute command f = "move forward" with parameter "30" which represents the specified speed in cm/s. Combining the above actions with timer, the moving distance and turn angle can be decided, allowing complete 2-dimensional navigation of the robot. For user convenience, advanced abstracted functions wrapped as robotAct(id, 'm', 30, 1.0, 1.0) are also provided. This command will cause the robot to execute command m = “move” to coordinate (1.0m, 1.0m) in 2D space at a speed of 30 cm/s.

5.2 Processing Platform

The processing platform added to the robot is a Stargate [20] node which is a xScale processor based computing device, running Linux. An 802.11 card is added to it for communicating with the base and for sending control packets to the robot's wireless interface. A mote is attached to the Stargate to be used as mote network interface card (NIC) for communication with the static motes in the network. A block diagram of the software implemented on the Stargate is given in Figure 5. The complete localization and navigation component for arbitrary environments is not yet implemented. We assume an obstacle free environment for the current experiments, and control the motion as described with our mobility control algorithms, in Section VI. This processing platform controls the robot, collects data from the mote NIC and executes our intelligent motion control algorithms to help improve the data collection process.

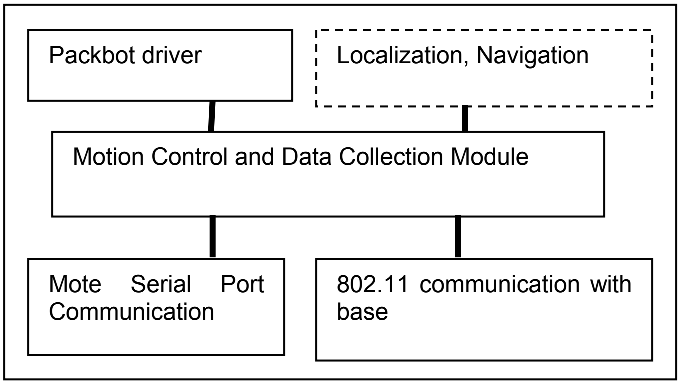


Figure 5. Software architecture of prototyped mobile router.

5.3 Mote

The mote is used as a network interface to the static nodes which communicate over the mote radio. It relays any data received from its radio to the Stargate over a serial port connection, and transmits any packets received from the Stargate to its radio. The mote can also be used as a data transfer interface to the base, but since radio energy is not a significant portion of the Packbot's battery consumption, a higher rate radio has been used (Power consumption of the Packbot is approximately 60W varying with speed, while the Stargate processor board with the 802.11 card and mote consumes only 1.7W).

6. ADAPTIVE MOTION CONTROL

The motion control algorithms used by the mobile components will depend on two factors – the constraints on the mobility imposed by the terrain and the data collection performance parameters which are important for the higher layer application or user collecting the data.

Consider first the mobility constraints placed by the terrain. The terrain may be such that the mobile device can move to all nodes directly. Here, the mobile device may chose to visit each node or may visit a small subset of the nodes and the remaining nodes transfer their data to this visited subset using multi-hop routing. Various trade-offs in the energy used by static devices and the time taken by the mobile device to complete its patrol of the network exist here. In other situations, the terrain may not allow free mobility to arbitrary locations. Our first application for the fluid infrastructure is a data gathering system for measuring physiological, environmental and ecological variables from sensors installed in a natural ecosystem. Installation of a fixed wired infrastructure is not feasible in this ecosystem; only compact wireless sensors can be embedded. In such terrain, both due to the dense vegetation restricting the mobility of the mobile device and to minimize disturbance to the natural habitat, we decided to move the robot only along the trail passing through the ecosystem; this trail is also used by human users to collect data manually. Thus, we design our motion control algorithms for the situation that the path traversed by the mobile agent is fixed.

The second issue requires various performance parameters to be tuned. The algorithm can be designed to achieve several objectives depending on the application priorities.

First, the lifetime of the system may have to be maximized. In this case the mobile router should visit each node individually and come within the closest possible distance to it. The embedded energy constrained device should have to use the least amount of energy in transmitting its data.

Second, the total amount of data collected could be an important parameter for certain applications. This happens when the static devices are generating data at a rapid rate and the mobile router must transfer the maximum possible data from the static devices to the base. The motion control algorithm would then try to discover a hybrid topology that uses both multi-hop paths and the mobile infrastructure such that data transfer rate is maximized.

In other applications latency may be a hard constraint. So a third performance metric to optimize for could be data transfer delay and the system would wish to maximize the amount of data collected within the given latency constraint.

Fourth, the static devices may also have a finite buffer constraint and the mobile device would then need to collect the data before buffers overflow. This may mean that certain data has to be transmitted via multi-hop routing, or may require the mobile device to use higher speeds and more frequent traverses through the network than required just from latency constraints alone.

The latency and data-rate will also depend on the energy supply of the mobile router itself. The mobile router needs some time at the base to charge its battery. This time may not be a factor in latency if the data is not collected continuously. In this case, there would be some intervals of no activity, when the mobile router may recharge its full battery and operate unhindered when data is actively being collected. If however, data is continuously collected, the recharge time adds to the data latency.

We now address some of the design issues in planning the motion of the mobile device and determine speed control primitives for studying one of the trade-offs, that of latency sensitive data collection, in two scenarios – networks with potential for MAC collisions among the static nodes and in disconnected sparse networks.

6.1 Influence of Speed on Data Collection

Before designing the motion control algorithms, we need to study the influence of the speed of mobile device on data collection. Is there an ideal speed for the mobile router? Intuitively, if at speed ***s***, the mobile router can cover the trail in time *t*, and get *n* packets, then at speed *2s*, it can cover the trail twice in the same time *t*, getting *n/2* packets in each round, effectively getting *n* packets overall. We wished to verify this experimentally because if the system behaves in this manner, any speed can be used, depending on the other performance trade-offs such as time for one traversal of the network, data success rate, or energy recharge-time of the robot. The experiment, variations of which also served for detailed debugging and verification of the developed prototype software, was performed as follows. To test the influence of speed alone and to suppress the effect of other parameters which may affect the radio communication, we placed two motes such that at any point the mobile router will be in range with at most one node. The radio range of the motes was reduced for this to happen within a smaller open area. Thus packet loss will occur only due to the mobile router going out of range of the nodes, and we did not want to have any collisions between the transmissions from two nodes. Figure 6 shows the arrangement of the static nodes.

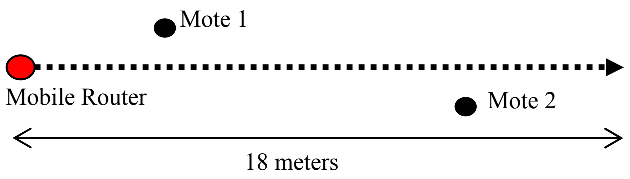


Figure 6. Experimental Topology for studying influence of speed.

The two nodes were continuously transmitting packets, so that the amount of data collected depends essentially on the time for which the mobile router is in range. The mobile router counts the number of received packets. We ran this experiment for seven different speeds (30 cm/s – 150 cm/s). The results at each speed are averaged over three rounds of the trail. The trail length was 18m, and a round is defined as the mobile router going 18m from the base and coming back the same distance. There was no packet reception in the reverse path (because eventually we expect to have a closed path as a trail). The results are plotted in Figure 7. As can be seen, the number of packets received per unit time at each speed is almost same. This suggests that the data delivery when the robot is within radio range does not depend strongly on the speed of the robot for the speed values which our system uses. Hence we do not account for any influence of speed in collision free radio channel for our motion control algorithms.

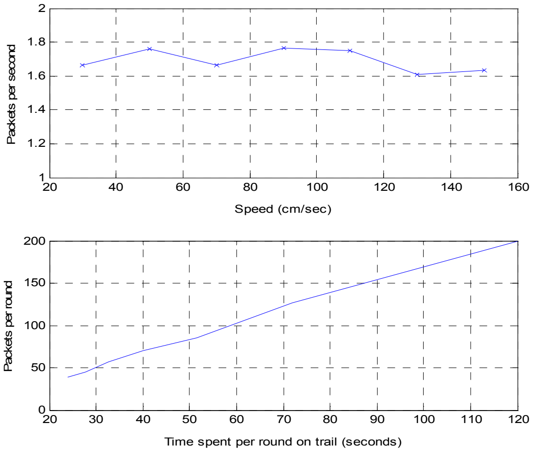


Figure 7. Data collection at different speeds. Data rate, (packets per unit time) does not strongly depend on speed.

6.2 Latency Sensitive Data Collection

Suppose a constraint on the maximum latency of the time for reporting sensor readings is specified. The maximum time that the mobile router can spend in completing one round across the network is then equal to this latency. The objective becomes maximizing the data collected in this round. Let the latency constraint specified be *T*. Given this acceptable round time, and knowing the length of the trail, the speed at which mobile router has to move can be decided. A naïve approach would be for the mobile router to move at this speed. But we propose methods to achieve better performance than obtained by the naïve approach.

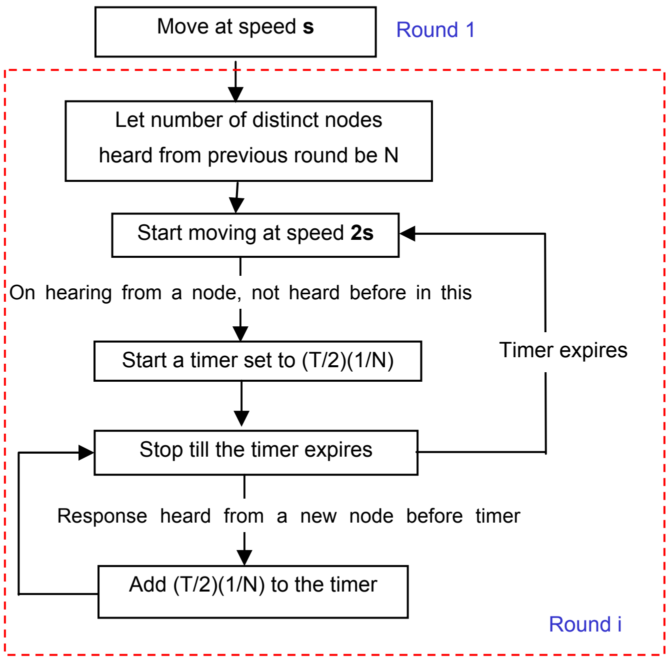
We make the following observations about practical network deployments which form the basis for our proposed improvements. The network is assumed to be randomly deployed. This will cause certain nodes to be in the range for longer duration, such as being very close to the path of the mobile router. Also, at certain locations, more than one node may be in range of the router, and communication bandwidth will be divided among them, suffering further due to MAC collisions. Further, the wireless channel may introduce more errors at particular nodes at certain times, causing data-rate to be reduced for these nodes.

The key insight is that the data collected can be increased if the mobile router moves slower when faced with a situation where more time is needed to collect the data and speeds up where high data-rates are achieved. We provide a run time algorithm for the mobile router to adaptively adjust its speeds in response to the run time dynamics of the network and improve the system performance.

To ensure practical applicability, we design an algorithm that is not based on the idealistic disc model for radio range. This model does not hold in practice as is observed in several experimental studies such as [31]. We also do not use geographic information such as used in [21] about the location of the static nodes as location may not be available or the error in location estimate may well be of the same order of magnitude as the radio range of the static devices; for instance the GPS localization error and the mote radio range are of the same order of magnitude. In fact our algorithm does not even require the router to know the number of static nodes on the path. This number may change based on new node installation, node damage or battery deaths, and will not affect our algorithm. The mobile router estimates the number of nodes based on node IDs in received data.

Suppose from the latency constraint *T* for round traversal time, the naïve speed calculated is *s*. At speed *2s*, the router can complete the round in time *T/2*. This leaves an extra time *T/2* for collecting data in regions where the wireless bandwidth is congested or data transmission is poor due to other reasons. In our scheme, the router adaptively learns where such regions occur, based solely on radio connectivity. We consider two approaches to manage the extra time spent in such locations. The first is to stop at locations where static nodes are found waiting with data. The other is to move slower in regions where data collection is moderately poor and stop in regions where data loss is severe. The precise methods to carry out this speed adaptation are described below. The communication specific details for both these methods are explained in the next section.

The first algorithm for speed control is called Stop to Collect Data (SCD), and its flow chart is presented below.



Algorithm 1. ***Stop to collect data*** **(SCD)**

As can be noted from the algorithm, no external information about node locations or time synchronization are required. The algorithm does not deadlock regardless of packet loss or channel errors. We will compare the performance of this algorithm to that of moving at constant speed in our experiments.

The second algorithm to control the motion of the mobile router adaptively decides to slow down or stop depending on the communication characteristics of the deployed network. Hence, it needs to learn the data transfer characteristics to control its motion. To achieve this, the mobile router maintains state information. The processing platform on the router itself is not severely energy constrained and the required processing capability can be provided here. The Stargate processor board used in our prototype is sufficient for this. The state information is acquired based on the data received from the static nodes. In every data packet that a static node transmits to the mobile router, the remaining number of data samples it wishes to transfer is included in the header. Based on the first and last packet heard from a static node, the router can calculate the percentage, *d*, of total samples that were received.

Define the following:

the number of unique samples received from a node = n,

number of remaining samples count as per the 1st packet = s1

number of remaining samples count as per the last packet = s2

Then,

$$d = \frac{n}{s1 + (s2 - s1) + n}$$

The formula also accounts for the data samples collected by the sensor after having sent the first packet. The router maintains this percentage for every node ID that it hears from. Let the percentage of data samples (which is 4 times the number of packets, as we are sending 4 samples in each packet) successfully received from a node be denoted d_i.

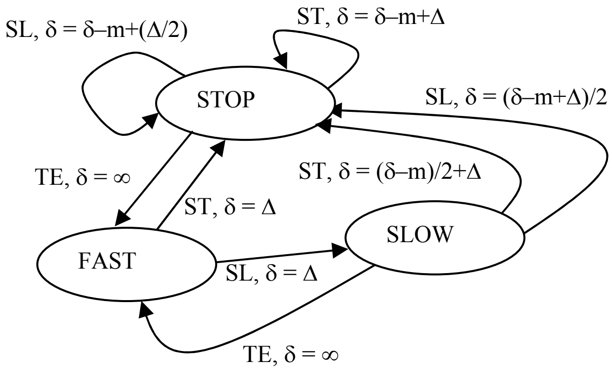


Figure 8. Traction-state diagram of the mobile router, with timer calculation.

Choose two thresholds, ω_1 and ω_2, $(0 < \omega_1 < \omega_2 < 100)$ on the data delivery success percentage. Denote the class of nodes for which $d_i < \omega_1$ by N1 and the class of nodes for which $\omega_1 < d_i < \omega_2$ by N2. The mobile router will stop where it encounters a node of class N1, and will slow down to speed *s* when it encounters a node of class N2. An encounter is defined as the event when a packet is first received from a node. The router will move at speed *2s* elsewhere. This method of taking a decision on encountering a node keeps the algorithm free of location information. Suppose the router found n1 nodes in class N1 and n2 nodes in class N2. Calculate:

$$\Delta = \frac{T/2}{n1 + (n2/2)}$$

As can be seen from the above expression, extra time given to a node of class N1 is double that of N2. This is because if the mobile router moving at speed *2s* stops for Δ, time lost is Δ, whereas if it slows down to speed *s* for Δ, time lost is Δ/2.

The router can be in three states, moving fast, moving slow or stopped. Define the following events:

1. SL: Encounter a node of class N2
2. ST: Encounter a node of class N1
3. TE: Current state timer expired.

Let *M* denote the timer. Let *m* denote the time elapsed since the timer was reset. Let δ denote the duration to which a timer is reset; δ can be different at every reset. With these events, the exact traction state of the mobile router itself can be modeled as shown in the state diagram in Figure 8. The exact durations to which the timer is reset corresponding to each event, are also shown. The transitions in the state diagram mean that on happening of an event (SL, ST, TE), the action to be taken is (setting δ to indicated value and moving to a new state). For instance, when the mobile router is in SLOW state, and gets event SL, it moves to the STOP state with new value of the timer $(\delta - m + \Delta)/2$. The transition is to STOP and not SLOW because due to the presence of two nodes in range the bandwidth is being shared and if the router goes to SLOW again (i.e. increases the duration of its existing SLOW state), it may happen that the node which requested the SL initially may pass out of range by the time it gets its share of bandwidth at SLOW speed. The total time is divided by 2 as the time lost due to stopping is double that due to slowing. If a TE event is received, it reverts to FAST, and sets the timer δ to ∞ as it has to move at speed *2s* until a new encounter.

The events that cause the router to transition between these states are based solely on the specific characteristics of network operation and no design time knowledge about the topology is needed. Like the SCD algorithm this algorithm too does not fail in the presence of wireless channel errors. The precise methods used and the exact details of how the latency constraint is managed in the prototype implementation are presented in the algorithm specification below.

Algorithm 2. **Adaptive Speed Control** (ASC)

```
1. Round 1: Move at constant speed. Determine
   the sets N1 and N2 for this round.
2. Repeat in every round: If n1+n2=0, move at
   speed s, else:
      a. Set δ = ∞, and start moving at
         speed 2s, transmitting INTEREST
         packet at periodic intervals.
         Listen for responses from static
         nodes.
      b. If any of the events ST, SL or TE
         occurs change speed and timer value
         according to state diagram of
         Figure 8.
      c. Round will finish in time less than
         or equal to T. Now transfer
         collected data to base over the
         802.11 link.
```

6.3 Latency Sensitive Communication in Sparse Networks

Another speed control problem arises in sparse networks. A sparse network is a network where the devices are too far apart to ensure connectedness of the induced topology graph and the graph is split up into multiple fragments, potentially as many fragments as the total number of nodes. The static devices are energy constrained and hence want to use a small radio range, which means that each device is in communication range of the mobile router only for a limited portion of the router's path. There exist transects in the path where there are no nodes in range for the mobile router. Clearly, the data collected in a given latency constraint can be increased if the router could stop when connected to a static node and move faster where there is no node in range. We propose using the SCD algorithm mentioned earlier for this scenario.

The proposed motion control algorithms can be used with the vanilla directed diffusion [12] based communication protocol. However, there are several issues which arise due to the presence of the fluid infrastructure and further performance gains can be achieved, both in terms of data quality and lifetime of the network, if the communication protocols are optimized for the mobile router. We discuss these issues next.

7. COMMUNICATION PROTOCOLS FOR FLUID INFRASTRUCTURE

Typically, ad hoc wireless networks use on-demand routing protocols such as AODV or DSR. However such protocols are not well suited for our application. In these protocols, the source node needs to know the address of the destination. In most sensor networking applications, the individual node addresses are not of great concern and only the data attributes are of interest. This property has been exploited in directed diffusion [12] to address packets based on data attributes. Hence we base our communication protocol on directed diffusion.

Previous networks using mobile nodes for carrying data [5, 6, 7] are designed for random mobility and are based on the presence of multiple mobile peers. These methods cannot be used for our system since we introduce an infrastructure oriented design for reliable communication. The protocol in [8] is designed for a predictable mobile router. It learns the locations at which it is in communication with individual sensor nodes. This method requires the mobile agent to know its precise position on the path and its localization error should be significantly less than the radio range. Further, this approach is fragile in case of network and channel dynamics. Also, none of the above methods exploit the fact that mobility can be controlled and hence are not directly applicable to our system.

Directed diffusion is a communication protocol designed for collecting data from a large number of sensor nodes. It is summarized here, since our method builds on this successfully tested scheme. The network is composed of sensor nodes, which are data sources and one or more data collection devices, called sinks. A sink expresses interest in particular sensor data satisfying some constraint. The INTEREST packets are broadcast from 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 forwarded. The nodes which hear the interest store it in their interest-cache, and rebroadcast it after decreasing the TTL field. As the interest propagates through the nodes, it establishes reverse paths towards the sink (called gradients), along which the data in response to the interest will be returned to the sink. Whenever a node has data, it will check its interest cache to see if this data matches any of the constraints received in an interest. If so, it sends the data to the node from which it heard the interest. This node will in turn consult its interest-cache, and the packet is relayed subsequently to the sink. To handle network dynamics and changing topologies, there is an expiration time for the interests.

However, directed diffusion cannot be directly used in the presence of fluid infrastructure because an end-to-end path does not exist continuously between the sources and the destinations. Hence we need to modify the protocol for this scenario. In our system, the static motes are the sources, and the mobile router is the single sink.

The first observation is that the static nodes should send their data directly to the mobile router if possible, and otherwise use multi-hop communication to reach the nearest node which can communicate with the mobile router. There is no design time knowledge of the radio propagation characteristics of the deployment environment and the mobile router cannot know a-priori what the maximum hop distance is between the router's trail and a node. Consider the scenario shown in Figure 9.

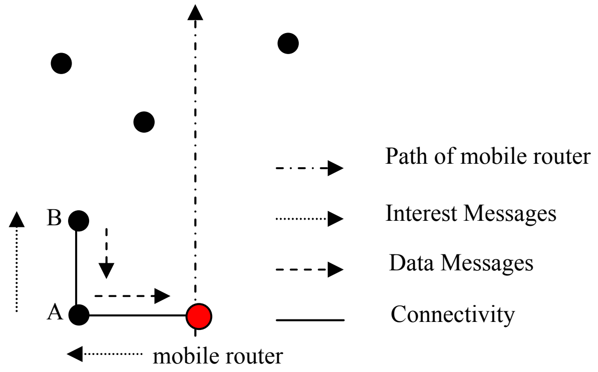


Figure 9. The communication protocol needs to restrict multi-hop communication but support nodes which are not in direct range of the router's trail.

The solid lines represent which nodes are in communication range; the node locations shown are arbitrary. Node A will broadcast the interest heard from the mobile router, which will be heard at node B. This will trigger node B to send data to node A, which will in turn relay it to the mobile router. It is possible that node B will have a direct connection to the mobile router after some time. If such is the case, we need to save the energy that node A spent in forwarding B's data. However, node A has no way of knowing if another node can reach the router only through it or will have a direct path. Our first modification to directed diffusion takes care of this. The first round of the mobile router is used to learn the connectivity of the mobile router. For every interest received, each node records if the interest came from the mobile router or another static node. If a node hears an interest from the mobile router, it will not respond to or rebroadcast any subsequent interests from another static node. Such a node will only rebroadcast interests directly received from the router. Other nodes will respond to and rebroadcast every unique interest message received. A node which does not receive interest from the router, may receive interest messages from two different nodes on the router's trail. Diffusion takes care of this.

The second observation is that the mobile router will be in range of a source only for small time duration. There may be a duration for which the interest has not expired but the mobile router has moved out of range. The data transmitted in this duration will be lost. It may be important for the static node to know what data was successfully received. We use an acknowledgement-retransmit scheme to handle this. The mobile router sends acknowledgement to the nodes it receives data from. The node will send the next packet only after it gets acknowledgement from the mobile router. There is a retransmit timer at the motes, after expiration of which the mote will send the same data again. This also takes care of lost packets due to channel errors, collisions etc. Eventually, the interest message on the motes will expire, leaving the interest cache empty, and the mote will stop sending out packets. When the mote again hears an interest, it may either resume sending data where it left or send other data according to the new interest. The modified behavior of the static devices compared to the raw diffusion protocol is summarized in the algorithm below in view of the above two modifications.

Algorithm 3. Communication protocol at static node

```
1. FLAG = NOT_ON_TRAIL
2. If an INTEREST message is heard:
     a. If FLAG == ON_TRAIL and INTEREST is
        not from router, break
     b. Start INTEREST_EXPIRY timer
     c. If the INTEREST message is from the
        mobile router set FLAG=ON_TRAIL
     d. Decrement TTL, If TTL > 0
        rebroadcast INTEREST message
3. If there is a valid INTEREST in the
   interest-cache
     a. Decrement INTEREST_EXPIRY
     b. Transmit data in buffer to node
        from which interest message was
        heard, and start retransmit timer
     c. If timer expires and no ACK is
        heard, retransmit data and reset
        timer
     d. If ack is heard, decrement number
        of remaining samples and goto set 3
4. Collect sample every sample period. If
   buffer overflows, discard oldest data.
```

7.1 Communication Protocol Issues

Another observation regarding the communication protocol mentioned above is that since some nodes which have direct communication with the router have to forward data from some other nodes as well, which are not directly queried by the router and since connection with the router exists for a limited time, the forwarding nodes can pre-fetch data from nodes for which they forward data. This can be achieved as follows. From the second round onwards, if a node hears a response to an interest transmitted by it, it knows that there are other nodes which forward their data through it. This node can then initiate a local diffusion. It sends a local interest message with TTL = 1. Nodes which receive this message may themselves have deeper nodes in the tree for which they transfer data. Such nodes would have initiated a local diffusion with TTL = 1 as well. These nodes will then respond to an interest when their local diffusion is complete. This way, the nodes need not have explicit knowledge about the network topology and any length of path to the mobile router can be pre-cached. When the mobile router visits, the data from all nodes in the region can be quickly transferred.

Apart from reducing energy by limiting the number of transmissions, further improvements can be obtained in lifetime by utilizing the sleep mode. Sleep mode utilization is generally referred to as topology management and several schemes such as [23, 24, 25, 26] have been suggested for this purpose with static nodes. We present a topology management method designed for the embedded devices communicating through a mobile router. The radio is a major power consuming component, and the motes must turn it on only when the mobile router will be in their range. Suppose the period of the mobile router to complete one round is T. This number can be made known to the mote at design time, piggybacked on the interest message, or can be learnt at run time. The duration for which the mobile router is within range is learnt at run time as it is not fixed at design time. The mote may start a timer when it first hears a packet from the mobile router. The mote will continue to receive ACK packets from the mobile router as long as the router stays in range. It records the value of the timer for each packet received, overwriting the timer value stored when the previous packet was received. After some time the mote will stop receiving ACKs from the mobile router. The timer value of the last packet received gives the time duration t for which the mobile router was in range. The minimum time the router takes to return to a mote is $T/2$, it could be more depending on how the speed varies in a particular run. The estimate of t can have some variation due to speed control. Thus the mote can switch off the radio for an amount of time equal to $(T/2) - t - \tau$, where τ is the error margin in the estimate of t. The sleep protocol will also change when pre-caching is activated.

As mentioned earlier, the time synchronization between the base and sensor nodes can be significantly improved since the number of hops is much lower. Thus, it would be beneficial for the system to utilize the data packets exchanged to execute a time synchronization protocol. In our prototype, the nodes in addition to the data, send the number of samples remaining to be transmitted in their buffers as specified in the speed control algorithm. The router notes the time at which the sample is received. From the number of samples remaining at the mote, it can calculate how many sampling periods have elapsed since the sample was collected. Using this and the sampling rate it can back calculate the time of the current sample.

The above discussion suggests that there are several design issues to be considered in the communication protocols and topology management. We showed the energy advantage and capacity improvement in Section 4, we now evaluate the data delivery performance with our proposed motion control schemes and communication protocol. The interaction of sleep management and local diffusion with motion control primitives is yet to be studied.

8. EXPERIMENTAL RESULTS

In this section we present the experimental results of the performance evaluation of our proposed speed control methods on our prototype system. For our present experiments, we use a straight trail (with no packet reception in reverse path) as the deployment scenario because we do not yet have the navigation equipment for traversing a complex closed trail currently. The experiments were done outdoors. The speed control methods will be the same on a more complex trail as well.

The first experiment is carried out on a dense network topology where the static nodes do form a connected network. The topology is shown below in Figure 10. The communication protocol specific parameters are listed in Table 1.

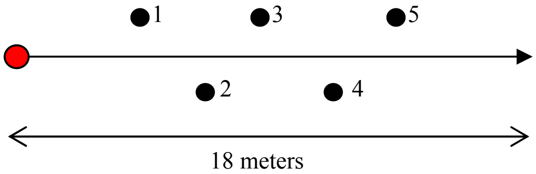


Figure 10. Topology for experiment on a connected network.

Figure 11 shows the performance in terms of the amount of data collected within the equal latency constraint for the naïve approach of constant speed, and the two speed control algorithms proposed above. For the trail length used, we fixed a latency constraint of 72 seconds to keep *s* and *2s* within feasible speed limits for the robot. This yields a naïve speed of 50cm/sec. For the ASC and SCD algorithms a fast speed of 100cm/s is used and ASC uses a slow speed of 50cm/sec. For ASC ω_1 was 25% and ω_2 was 75%. All results plotted are averaged over seven rounds. While both speed control algorithms perform better than the naïve approach, the relative performance among the two does not follow a specific trend.

Table 1. Data Collection Parameters

Parameter	**Value**
Implementation platform	TinyOS
Interest Repetition Interval	2s
Interest Expiry Time	10s
Sampling Period	1s
Retransmit timer	1s
Buffer size	150 Bytes
MAC random backoff	0 to 500ms

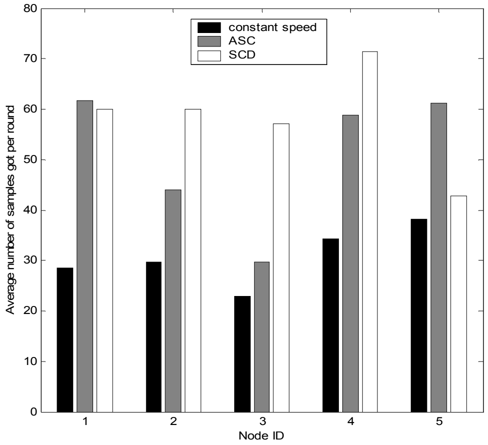


Figure 11. Performance of three speed control methods on connected topology.

The second experiment is performed on a topology with two groups of nodes, where nodes within a group are within wireless collision range of each other, leading to wastage of bandwidth due to MAC level contention. The groups themselves are far apart and do not interfere. The topology is shown in Figure 12. The data collection parameters are as before and the results are again averaged over seven rounds.

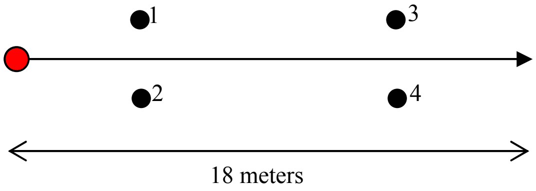


Figure 12. Topology for experiment on a disconnected network with two MAC contention groups.

Performance is plotted in Figure 13. The figure shows that the both the adaptive speed control methods perform better than the naïve approach to move at constant speed. The performance gains vary at different nodes. In this case SCD performs better than ASC as moving even at a slow speed may take the router outside the range of the colliding nodes.

A third experiment was performed for the sparse network case. Here, no multi-hop routes exist and communication is not possible without the presence of the mobile router. The topology used is as shown in Figure 6. Two nodes were placed well outside the range of each other. Thus, there existed a segment in the trail when the mobile router was not in range of either node. The SCD algorithm was employed. As expected, controlling the speed helped improve the amount of data collected within the latency constraint. The performance is shown in figure 14 for both the nodes.

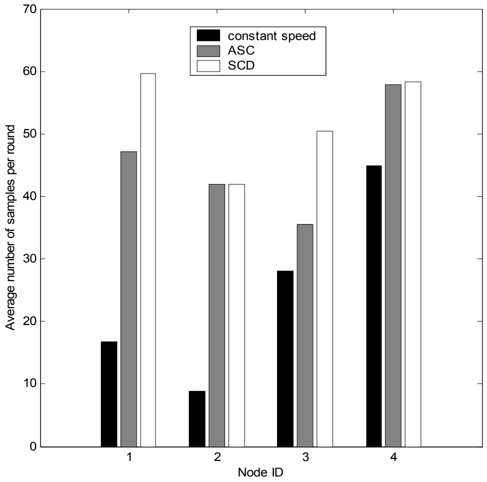


Figure 13. Performance comparison on the topology of Figure 12.

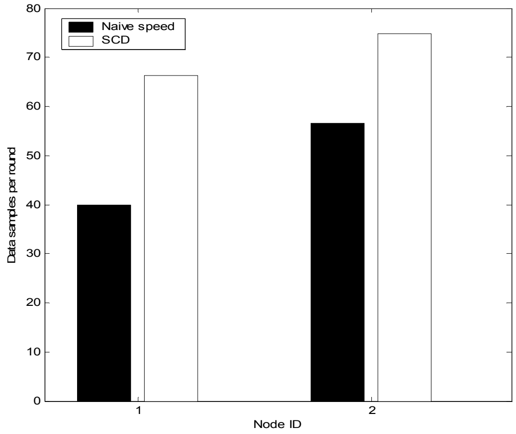


Figure 14. Performance comparison on sparse network topology.

9. CONCLUSIONS AND FUTURE DIRECTIONS

We discussed several advantages and design considerations for incorporating a controlled mobile element into the networking infrastructure. Our fluid infrastructure design saves significant energy in the embedded sensor nodes, thus improving system lifetime and utility. The prototype system has clearly established the feasibility of using a mobile networking device for data gathering and some of the advantages of doing so. The test system constructed is now being used for research on several interesting issues in the use of fluid infrastructure. We elaborated on a subset of the design issues in the development of such an infrastructure, focusing on motion control and communication protocol design. The proposed motion control methods yield significant advantage in terms of data quality.

Our results are very promising and provide sensor network designers with a new method to solve important problems in achieving improved network lifetime, data fidelity, time synchronization and bandwidth utilization. One of the components which we are in the process of adding to our fluid infrastructure is navigational support. Interesting design issues exist when the latency constraint allows for a smaller duration to collect data than the time it takes to complete a full traversal of the network deployment trail. Here, a trade-off must be made in the extent to which data travels wirelessly and using the mobile device. We also mentioned that when the time taken by the robotic platform for recharging is an issue in latency, new trade-offs in network lifetime and data latency would come into play. Optimizing the communication protocol design for low energy operation utilizing topology management techniques is another future direction. Future work also includes considering scenarios where the terrain allows the mobile router to move on several paths or when multiple mobile components are available.

10. ACKNOWLEDGEMENTS

This paper is based on research supported in part by the Intel Corporation, by the NSF Center on Embedded Networked Sensing and by the Office of Naval Research.

11. REFERENCES

[1] G. J. Pottie and W. J. Kaiser. Wireless integrated network sensors. In *Communications of the ACM*, May 2000.

[2] David Tennenhouse. Embedding the Internet: Proactive computing. In *Communications of the ACM*, May 2000.

[3] Ya Xu and John Heidemann and Deborah Estrin. Geography-informed energy conservation for Ad Hoc routing. In *Proc. ACM Mobicom* July 2001.

[4] Henri Dubois-Ferriere, Matthias Grossglauser, Martin Vetterli. Age Matters: Efficient Route Discovery in Mobile Ad Hoc Networks Using Encounter Ages. In *ACM Mobihoc* June 2003.

[5] Rahul C Shah, Sumit Roy, Sushant Jain and Waylon Brunette. DataMULEs: Modelling a Three Tiered Architecture for Sparse Sensor Networks. In *First IEEE International Workshop on Sensor Network Protocols and Applications (SNPA)*, May 2003.

[6] Tara Small and Zygmunt J. Haas. The shared wireless infostation model: a new ad hoc networking paradigm (or where there is a whale, there is a way). In *ACM Mobihoc*, June 2003.

[7] Philo Juang, Hidekazu Oki, Yong Wang, Margaret Martonosi, Li Shiuan Peh and Daniel Rubenstein. Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with ZebraNet. In *Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, October 2002.

[8] A Chakrabarty, A Sabharwal and B Aazhang. Using Predictable Observer Mobility for Power Efficient Design of a Sensor Network. In *Second International Workshop on Information Processing in Sensor Networks (IPSN)*, April 2003.

[9] Alberto Cerpa, Jeremy Elson, Deborah Estrin, Lewis Girod, Michael Hamilton and Jerry Zhao. Habitat monitoring: Application driver for wireless communications technology. In *ACM SIGCOMM Workshop on Data Communications in Latin America and the Caribbean*, 2001.

[10] Alan Mainwaring, Joseph Polastre, Robert Szewczyk, David Culler and John Anderson. Wireless Sensor Networks for Habitat Monitoring. In *First ACM Workshop on Wireless Sensor Networks and Applications (SNPA)*, September 2002.

[11] Delay Tolerant Networking Research Group. www.dtnrg.org.

[12] Chalermek Intanagonwiwat, Ramesh Govindan and Deborah Estrin. Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks. In *Mobicom*, August 2000.

[13] Philip Levis, Nelson Lee, Matt Welsh, and David Culler. TOSSIM: Accurate and Scalable Simulation of Entire TinyOS Applications. In *ACM SenSys*, November 2003.

[14] Mica2 motes. Product Datasheet. http://www.xbow.com/Products/Wireless_Sensor_Networks.htm.

[15] A. LaMarca, W. Brunette, D. Koizumi, M. Lease, S. B. Sigurdsson, K. Sikorski, D. Fox, and G. Borriello. PlantCare: An Investigation in Practical Ubiquitous Systems. In *Ubicomp*, September 2002.

[16] Packbot, The Next Step in Unmanned Tactical Mobile Robots. www.packbot.com.

[17] iRobot. www.irobot.com.

[18] ActivMedia Robotics. www.amigobot.com.

[19] TinyOS: a Component-based OS for the networked sensor regime. http://webs.cs.berkeley.edu/tos/.

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

[21] Dragos Niculescu and Badri Nath. Trajectory based forwarding and its applications. In *Mobicom*, September 2003.

[22] J Scott and M Hazas. User-Friendly Surveying Techniques for Location-Aware Systems. In *Ubicomp*, October 2003.

[23] Curt Schurgers, Vlasios Tsiatsis, Saurabh Ganeriwal, Mani Srivastava. Optimizing Sensor Networks in the Energy-Latency-Density Design Space. In *IEEE Transactions on Mobile Computing*, January-March 2002.

[24] Xiaorui Wang, Guoliang Xing, Yuanfang Zhang, Chenyang Lu, Robert Pless, and Christopher Gill. Integrated Coverage and Connectivity Configuration in Wireless Sensor Networks. In *ACM SenSys*, November 2003.

[25] Benjie Chen, Kyle Jamieson, Hari Balakrishnan, Robert Morris. Span: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks. In *ACM Mobicom*, July 2001.

[26] Alberto Cerpa and Deborah Estrin. ASCENT: Adaptive Self-Configuring sEnsor Networks Topologies. In *Infocom*, June 2002.

[27] A.K. Salkintzis. A Survey of Mobile Data Networks. In *IEEE Communication Surveys*, 3rd Quarter 1999.

[28] Kevin Fall. A Delay-Tolerant Network Architecture for Challenged Internets. In *Sigcomm*, August 2003.

[29] Qun Li and Daniela Rus. Sending messages to mobile users in disconnected ad-hoc wireless networks. In *ACM Mobicom*, August 2000.

[30] Matthias Grossglauser and David Tse. Mobility Increases the Capacity of Ad-hoc Wireless Networks. In *Infocom*, April 2001.

[31] Jerry Zhao and Ramesh Govindan. Understanding Packet Delivery Performance in Dense Wireless Sensor Networks. In *ACM Sensys*, November 2003.

[32] Mohammed Rahimi, Hardik Shah, Gaurav S. Sukhatme, John Heidemann and D. Estrin. Studying the Feasibility of Energy Harvesting in a Mobile Sensor Network. In *IEEE Int'l Conference on Robotics and Automation*, May 2003.

[33] Deborah Estrin, Ramesh Govindan and John Heidemann. Embedding the Internet: introduction. In *Communications of the ACM*, May 2000.

[34] V. Raghunathan, C. Schurgers, S. Park and M. Srivastava. Energy aware wireless microsensor networks. In *IEEE Signal Processing Magazine*, March 2002.

[35] Jeremy Elson, Lewis Girod and Deborah Estrin. Fine-Grained Network Time Synchronization using Reference Broadcasts. In *Proceedings of the Fifth Symposium on Operating Systems Design and Implementation (OSDI)*, February 2002.

[36] S. Jain, R. Shah, W. Brunette, G. Borriello and S. Roy. Exploiting Mobility for Energy Efficient Data Collection in Sensor Networks. In *IEEE Workshop on Modeling and Optimization in Mobile Ad hoc and Wireless Networks (WiOpt),* March 2004.

[37] M Srivastava, R. Muntz, and M. Potkonjak. Smart kindergarten: sensor-based wireless networks for smart developmental problem-solving enviroments. In *Proceedings of the Seventh ACM Annual International Conference on Mobile Computing and Networking (MobiCom)*, pages 132 ñ 138, July 2001.

[38] AS Pentland, R Fletcher, and A Hasson. Daknet: rethinking connectivity in developing nations. *IEEE Computer*, 37(1):78ñ83, January 2004.