Reactive Sink Mobility in Wireless Sensor Networks

Daniele Puccinelli, Matthew Brennan, and Martin Haenggi
Network Communication and Information Processing Laboratory
University of Notre Dame
Notre Dame, IN, USA
{dpuccine, mbrenna1, mhaenggi}@nd.edu

ABSTRACT

The ability of a sink node to move can greatly improve the fault tolerance and load balancing properties of a sensor network. Rather than assuming extensive mobility and trying to minimize the large-scale path loss between the mobile sink and the nodes, we focus on limited-scope, arbitrary mobility triggered in response to a form of network feedback. Due to multipath fading effects, limited mobility dynamically modifies the set of sink neighbors and distributes network traffic over a larger number of nodes. We illustrate the impact of this reactive sink mobility concept on data collection by implementing it on top of a novel gradient-based routing protocol. We use Berkeley motes to present a proof of concept as well as a performance evaluation of our approach, with a particular emphasis on the advantages in terms of robustness and lifetime.

Categories and Subject Descriptors: C.2.1 [Network Architecture and Design]: Wireless Communication

General Terms: Algorithms, Measurement, Experimentation.

Keywords: Wireless Sensor Networks, Mobility, Routing, Load balancing.

1. INTRODUCTION

1.1 Motivation

The standard use of a sensor network is *data collection*: a number of sensing nodes are deployed with the purpose of measuring physical phenomena. Data collection occurs in two steps: query dissemination, characterized by a one-to-many traffic pattern from the base station to the sensing nodes, and data retrieval, featuring a many-to-one traffic pattern from the nodes to the base station. The main drawback of a many-to-one traffic pattern is that the nodes with the best channel to the base station have a heavier workload than their peers, as they are called to relay traffic that they do not generate. This additional burden curtails their lifetime, disconnecting the base station from the rest of the network and therefore making the network useless; this is known as the *hot spot problem*. The one-hop neighbors of the sink are typically more affected by this problem (especially if multiple sources are concurrently active) and are therefore known as *critical nodes*. It is the aim of *load balancing* schemes to avoid the formation of hot spots, or at least reduce the gravity of the problem. Most routing schemes for wireless sensor networks employ a link reliability metric [1, 2, 3], which causes the nodes with the best channel to the sink to be overexploited. In this sense, load balancing can be seen as a way to preserve the resources of the *best* nodes.

1.2 Main contribution

We consider the load balancing problem in single-source networks (only one source is active at a given time). We introduce the concept of *reactive sink mobility*: the sink, a node with a dedicated wireless link to the base station (*e.g.*, higher power or higher rate) and renewable energy resources, moves opportunistically based on a form of feedback from the network. Mobility is seen not only as an advantage to exploit, but also as a resource to control. We show how sink mobility can boost the load balancing properties as well as the fault tolerance of the underlying routing protocol. An innovative element of our approach is the fact that we consider limited sink mobility, which we define as the ability of the sink to move around within an area of the order of a square wavelength (λ^2); since signal strength measurements are essentially uncorrelated over spatial displacements of the receiver of about $\lambda/2$, motion of such a limited scope is sufficient to obtain a significant spatial diversity benefit through induced multipath fading [4]. We illustrate the benefits of reactive sink mobility using a novel gradient-based routing protocol (partly based on our previous work [5]) with a cross-layer route selection metric that looks for the best compromise between reliability and load balancing.

2. RELATED WORK

2.1 Gradient-based routing

In *receiver-based* routing, a node estimates the cost of reaching the intended destination and includes it in the outgoing packet before broadcasting it. Only the neighbors that estimate a lower cost to the destination rebroadcast the packet, allowing it to descend a loop-free gradient towards the destination. Receiver-based routing follows the *gradient routing* (GRAd) paradigm [6], whose central idea

MobiOpp'07, June 11, 2007, San Juan, Puerto Rico, USA.

is a modification of the *route query and reply* mechanism of reactive routing. Route query comes in the form of sink-initiated query flooding that leads to a cost field centered at the sink. The cost field is set up as nodes estimate the cost of reaching the sink according to a given metric, *i.e.*, the distance in number of hops; this mechanism is often called *gradient setup*. Route reply is achieved by means of *reverse path routing*, as data packets descend the cost field from the sensing area to the sink. Widely cited examples of gradient-based routing are Directed Diffusion (DD) [7], Rumor Routing (RR) [8], and GRAdient Broadcast (GRAB) [9], which augments the gradient-based paradigm by employing Minimum Cost Forwarding (MCF) [10]. The cost field is obtained by means of flooding with backoffs based on link costs. Upon reception of a cost field setup packet from node i advertising its minimum cost to the sink, node j (upstream from node i) estimates the cost of link (i, j), computes an estimate of the minimum cost to the sink as the sum of the link cost and the cost advertised by node i, and sets its backoff timer based on said estimate. If another cost field setup packet is received from a different node, the procedure is repeated; if a smaller cost to the sink is estimated, the timer is reset accordingly. This mechanism ensures that the minimum cost is advertised. The result is the reduction of the redundancy introduced by flooding (only one broadcast per node is necessary).

2.2 Load balancing and sleep management

Several load balancing solutions exist as part of energy-aware routing schemes and use ideas from topology control and redundancy suppression. Multipath routing schemes also have load balancing properties, and backbone routing techniques that only keep a subset of the network active achieve load balancing by way of sleep management. In *adaptive duty cycling* schemes [11], a subset of the nodes go into sleep mode while an active backbone maintains connectivity. Adaptive duty cycling schemes come either in the form of novel MAC protocols or are designed to run on top of an existing MAC layer (as a basis for the network layer). In SPAN [12], each node makes periodic decisions on whether to sleep or to stay awake as a *coordinator*, *i.e.*, a node in the active backbone. The idea is to power off as many radios as possible without losing connectivity and capacity. In Adaptive Self-Configuring sEnsor Network Topologies (ASCENT) [13], *active nodes* are adaptively elected and required to stay awake to perform multihop routing while the other nodes keep their radios off and only periodically check whether they should become active. A sentry selection mechanism [14] is implemented in VigilNet [15], an application for energy-efficient surveillance. A sink-initiated diffusion tree is created for gradient-based routing, and asymmetric links are blacklisted to ensure the reliability of reverse paths to the sink.

2.3 Sink mobility

The concept of a mobile base station is probably the most intuitive solution to the hot spot problem. Since hot spots arise in the vicinity of the base station, making the base station mobile distributes the problem across the network. In [16], the use of multiple mobile base stations is advocated, and the need for residual node energy monitoring is underscored. The base stations are normally static, and only move at fixed intervals. The minimization of two different objective functions (total energy consumption and peak node energy consumption) is used to determine the location of the base station during each stationary period. In the context of mobile base stations, it is natural to jointly consider routing and mobility. The analysis in [17] determines that the best mobility strategy consists in following the periphery of the network (the same pattern suggested in the base station relocation scheme in [16]) and studies how routing can leverage on the trajectory of the base station. The results are encouraging: joint mobility and routing would increase the network lifetime by as much as 500%. It should be noted, however, that the analysis is based on the disc model [18]. It is also shown that even an arbitrary trajectory of the base station extends the network lifetime. MobiRoute, a routing protocol based on [17], has been proposed in [19] as a superset of MintRoute [20]. One of the goals of MobiRoute is to convince the community that the benefits of controlled mobility offset its costs. Similarly to [21], lifetime maximization is cast in the form of a linear program: lifetime is modeled as the sum of the sojourn times of the mobile base station at each anchor point, and is used as an objective function to be maximized subject to the constraint that the energy consumption at each anchor point does not exceed the sum of the initial energy levels available at each node in the network.

Many mobile sink approaches aim at one-hop data transfers between the nodes and the sink. Data is typically transfered when the trajectory of the sink brings it as close as possible to a given node (the underlying idea is the minimization of the large-scale path loss). The possibility of using a mobile node as an information sink was first indicated in [22]. In [23], data collection from a sparse sensor network is performed with *data mules*, mobile sinks moving in an arbitrary fashion and collecting data opportunistically whenever they are near a given static sensing node. A similar idea is also at the center of [24], where mobile nodes are used as *message ferries* that permit message exchanges between disconnected nodes, and in the SENMA architecture [25]. It is worth mentioning that data mules are already being used in existing sensor network deployments [23].

Controlled mobility has been suggested in [26]. The main motivation is to save the energy of the sensor nodes and increase the lifetime of the network. It is shown that sink mobility can avoid the hot spot problem, and it is underlined that using a mobile sink reduces the data relaying overhead. The maintenance paradigm of a system with mobile sinks is rather different from the one of a standard wireless sensor network: whereas it is not possible nor desired to renew the energy resources of many sensing nodes, it is easier to change the batteries of a few mobile sinks. This concept forms the basis for our assumption that the sink is not as resource-constrained as the other nodes. Motion planning is considered jointly with routing: nodes that cannot communicate directly with the sink use geographically-constrained (local) multihop routing. The underlying idea is clustering: the mobile sink only communicates with designated cluster heads. It is underscored that it is not realistic to expect a sink to be able to move anywhere within a deployment area: rugged terrain and unfavorable physical phenomena may make it impossible.

3. MOBILITY-AIDED ROUTING

In order to demonstrate and assess reactive sink mobil-

ity we need an underlying routing scheme. We present a gradient-based reactive routing protocol that integrates route discovery into the connectivity discovery process and comprises of the following steps.

- **Step 1: sink-initiated connectivity discovery**. The sink initiates connectivity discovery by means of interest dissemination, *e.g.*, by broadcasting a setup packet S_S^p containing initialization state information. Said setup packet includes the sender address (sink address), the bottleneck downstream link quality L_S, the bottleneck remaining energy in the downstream nodes V_S, hop distance from the sink H_S (one hop). The bottleneck information (link quality and remaining energy) is initialized to 1. All other nodes initialize their depth D to ∞.

- **Step 2: connectivity discovery by the neighbors of the sink**. The (upstream) neighbors of the sink receive the setup packet from the sink. Neighbor k measures $l_{1,k}$ (by way of a form of *link estimation*) and discards the incoming setup packet based on a given *link blacklisting policy*. Otherwise, node k sets its depth to $D_k = H_S = 1$, measures its own remaining energy v_k, and creates setup packet S_k^p with $L_k = \min(l_{1,k}, L_S) = l_{1,k}$,$V_k = \min(v_k, V_S) = v_k$, and $H_k = H_S + 1 = 2$. Setup packet S_k^p is broadcast after a backoff period T_k. This mechanism prevents the neighbors from transmitting at the same time (the race condition issue pointed out in [15]).

- **Step 3: connectivity discovery and route selection by the rest of the network**. Upon reception of setup packet S_j^p sent by its downstream neighbor j, node i only processes S_j^p if link (j, i) is accepted by the link blacklisting policy and $H_j < D_i$ (to avoid routing loops). Depth D_i is estimated to be $H_j + 1$ and a metric $M_{i,j}$ is computed by node i as

$$M_{i,j} = \eta \min(l_{j,i}, L_j) + \zeta \min(v_i, V_j) + \frac{1 - \eta - \zeta}{D_i}, \quad (1)$$

  with $0 \leq \zeta \leq 1$ and $0 \leq \eta \leq 1 - \zeta$. Node i must now wait until it receives setup packets from all its downstream neighbors. Therefore, a timer is set to fire after a backoff period T_i. This mechanism also prevents node i from transmitting at the same time as other nodes at depth D_i. Let $\mathcal{Y}_i$ be the set of all nodes whose setup packets are received and processed by node i; when the timer fires, node i broadcasts setup packet S_i^p with $L_i = \min(l_{j^\star,i}, L_{j^\star})$, $V_i = \min(v_i, V_{j^\star})$, and $H_i = H_{j^\star} + 1$, where

$$j^\star = \underset{j \in \mathcal{Y}_i}{\operatorname{argmax}} M_{i,j} \quad (2)$$

  is the downstream neighbor of node i offering the best metric or, equivalently, the minimum cost to the sink. The notation is clarified in Figure 1.

- **Step 4: connectivity discovery and route selection completion**. The data source (node 0) operates in the same manner, but after the set backoff period T_0 (necessary to ensure the termination of the route discovery phase), relays data packets to node

$$h^\star = \underset{h \in \mathcal{Y}_0}{\operatorname{argmax}} M_{0,h}. \quad (3)$$

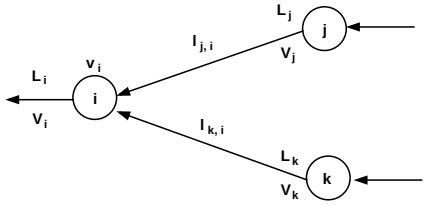

Figure 1: If the route through j has a higher metric than the route through i, *i.e.*, $M_{i,j} > M_{i,k}$, then i sets its state to $L_i = \min(l_{j,i}, L_j)$ and $V_i = \min(v_i, V_j)$; otherwise, $L_i = \min(l_{k,i}, L_k)$ and $V_i = \min(v_i, V_k)$.

At this point, data packets may be routed to the sink: the source starts sending data according to a given *data packet injection policy*. Any node w receiving a data packet relays it to node

$$x^\star = \underset{x \in \mathcal{Y}_w}{\operatorname{argmax}} M_{w,x}. \quad (4)$$

When the sink receives a data packet, it relays it to the base station. Route maintenance is feedback-triggered. If the *maintenance request conditions* are met, steps 1 through 4 are repeated. State update is performed during route maintenance. The particular maintenance request conditions as well as the *maintenance actions* depend on the implementation of the protocol, but in general they are based on a form of feedback from the network, such as the packet delivery rate measured by the sink. This basic routing paradigm can be enriched with a *sleep management policy* whereby only a subset of the nodes in the network is kept active at a given time. The route selection metric fuses cross-layer information: link reliability (physical layer), remaining energy (hardware), and hop count (network layer).

We now present an example of how the basic protocol framework works. We represent a network with an undirected weighted graph; we assign a weight to all links and relays based on typical values measured in real sensor networks during our experiments. Link weights quantify link quality (*e.g.*, estimated packet delivery rate), whereas node weights quantify the remaining lifespan of a given node (*e.g.*, remaining energy). In a connectivity graph, nodes not connected by an edge may not communicate at the rate and power level used by the hardware. Applying our scheme to the connectivity graph in Figure 2 results in the connectivity graph displayed in Figure 3 which only indicates the links used by the protocol. A note on the protocol: in general, the presence of the depth term in (1) (with D_i at the denominator) is only necessary if the depth estimate D_i changes during the route discovery process due to the particular values of the backoffs T_i. In the example in Figure 3 (and, later, Figure 5) we assume the connectivity discovery process to always be breadth-first, so the depth term does not influence the metric computation.

The reliability and the energy-efficiency of the downstream data retrieval process can be improved with controlled sink mobility. Differently from most studies on mobile sinks (surveyed in Section 2.3), we only assume limited-scope sink mobility. From a load balancing perspective, sink mobility reduces the hot spot problem, redistributing the load over a larger number of nodes (different nodes at different

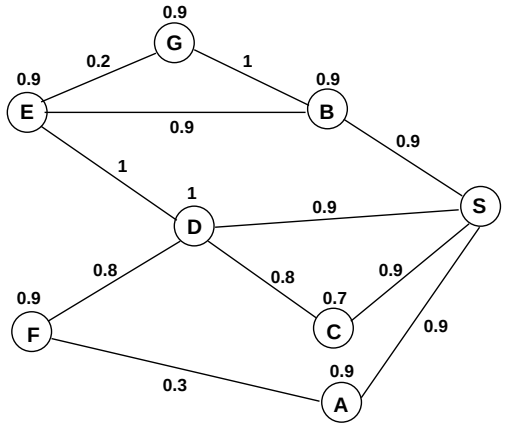

Figure 2: Connectivity graph of our sample network. Link quality is indicated near the edges and the fraction of remaining energy is shown on top of the nodes.

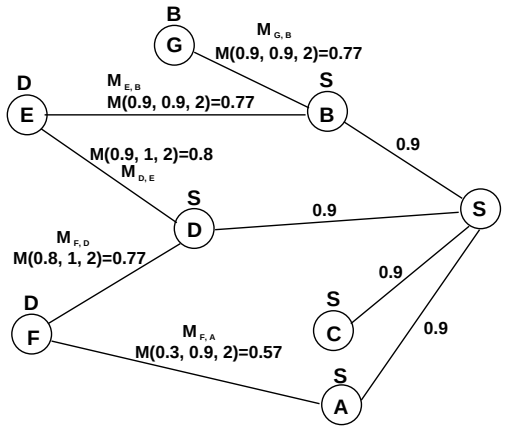

Figure 3: Connectivity graph visualizing the route selection process. Only the links used by our routing protocol are visualized.

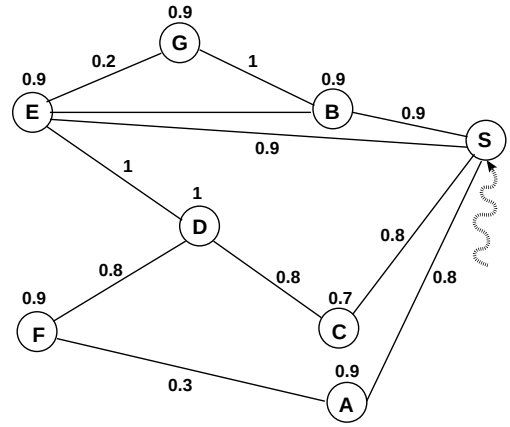

Figure 4: Sink mobility modifies the connectivity graph with respect to Figure 2.

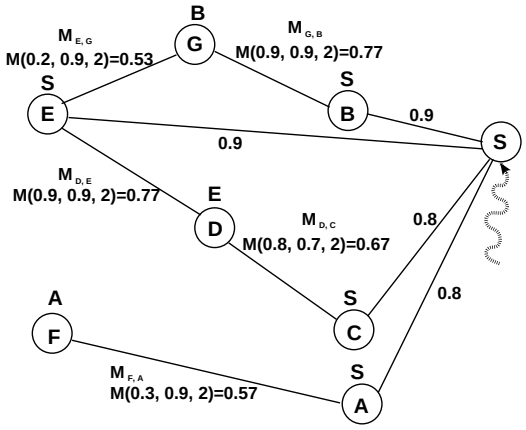

Figure 5: As a consequence of the changes in the connectivity graph shown in Figure 4, a different route is selected by our protocol.

times). We consider mobility as an advantage that can be controlled and exploited. Specifically, sink mobility can be triggered by some form of feedback from the network. As we mentioned in Section 2.3, studies such as [26] consider the problem of routing to a mobile sink and cope with the added complexity of an ever-changing topology. We adopt a completely different approach and use sink mobility to opportunistically modify the set of sink neighbors; we move the sink in discrete steps to modify the set of the sink neighbors and rerun a route discovery process to handle the changes in connectivity (with no added complexity at the network layer).

As an example, a small displacement of the sink in Figure 4 modifies the connectivity properties of the sink neighbors with respect to Figure 2. The impact on route selection is considerable, as can be surmised by comparing Figures 3 and 5. Sink mobility inherently favors load balancing, since it dynamically modifies the set of sink neighbors, the critical nodes that are typically subject to a heavier workload due to their favorable position. It is evident from this example that a connectivity discovery process must be initiated anytime the sink moves to a new position, as state information needs to be updated to properly mirror the modified topology. In general, for purposes of route maintenance, connectivity discovery can be run whenever the sink moves, whenever a node wakes up, and/or whenever any node requests it. With the aggressive use of sleep mode management techniques, we do not anticipate this form of periodic controlled flooding to bear significant cost.

One possible form of feedback is the packet delivery rate measured by the sink: if it falls below a certain threshold, the sink can move arbitrarily and run a new connectivity discovery process. It is reasonable to assume that the sink knows how many packets and at what rate the source intends to send; for example, the source might respond to a query disseminated by the sink with meta-data describing the nature of its sensed data along with the number of packets it is going to send and the transmission rate. If the drop in the

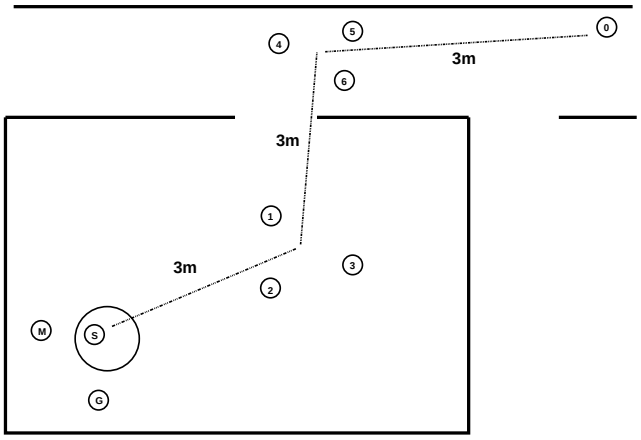

Figure 6: Layout of our proof-of-concept experiment. Node 0 is a data packet source whose goal is to relay its packets to the sink, node S, which is placed on a motorized rotating platform controlled by node M (mote-controlled setup). All nodes are elevated (up to 1m), whereas node 0 is on the floor.

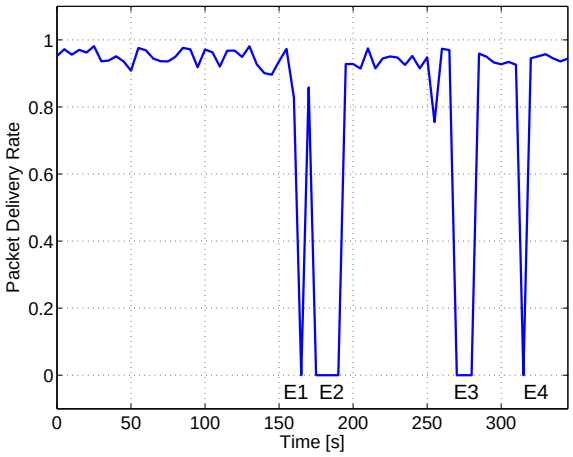

Figure 7: Proof of concept of mobility-aided feedback-based routing. In this figure, the packet delivery rate measured by the sink is averaged over each burst.

packet delivery rate is due to a node failure or a topology change in the environment (such as an obstacle shadowing a link and thus causing heavy losses), the sink can find a different route to draw data from the source. Another possible form of feedback can be derived from the self-monitoring of the nodes: nodes with a high workload can ask the sink to change its position and thus modify the network topology, so that traffic can be rerouted and other nodes can serve as sink neighbors. For instance, the displacement of the sink S in Figure 4 with respect to its position in Figure 2 could be dictated by the failure of node D and the consequent drop in the packet delivery rate measured by S, or simply by a request of D. Either way, the effects of sink mobility in this example are that packets from E can now go straight to S rather than over D, packets from D (if D still works) must hop over to E rather than going straight to S, and packets from F are rerouted over A.

4. PROOF OF CONCEPT

In this paper, we only consider one-dimensional motion, and we assume the control signal to be binary (*move* or *do not move*). We use arbitrary motion to gauge the baseline performance improvement (in terms of packet delivery rate and load balancing) that can be achieved with limited-scope reactive sink mobility. Figure 6 shows the layout of a 10-mote experiment aimed at providing a proof of concept of the potential of the proposed approach. Node 0 is a data packet source whose goal is to relay its packets to the sink, node S, which is placed on a motorized rotating platform controlled by node M (mote-controlled setup). Node S relays data packets over to the base station (wired to a CrossBow MIB510 gateway), node G. We use MICAz nodes, devices built around a 4MHz Atmel microprocessor with 128KB of programmable memory and 4KB of data memory. MICAz is equipped with a CC2420, an IEEE 802.15.4-compliant radio operating at 2.4GHz, the same frequency used by 802.11b/g. CC2420 provides two measurements that be used for link estimation: Received Signal Strength (RSS) and Link Quality Indicator (LQI). The former is an estimate of the received power and typically ranges from -35 to -90dBm, whereas the latter represents an estimate of the chip error rate [27] and ranges from 50 (low packet delivery rate) to 110 (high packet delivery rate). Sink mobility is provided by a custom-made mote-controlled rotating platform. The motorized rotating platform is controlled by mote M wired to the driver of a stepper motor. The rotating platform is built around a 2-phase high-torque stepper motor (HT23-396 from Applied Motion Products, Inc.) directly coupled with a shaft. An MD2S-P-L microstepping motor driver is used to send commands to the motor, and a custom interface connects a MICAz mote to said driver. Whenever the sink S (placed on the rotating platform at a distance r_S from the center) wants to move, it notifies node M which sets the table to rotate by an angle ϕ so at step k that node S experiences a net displacement of $d = 2r_S \sin(\frac{\phi}{2})$. In all the experiments described in this paper, $r_S = 7.5$cm and $\phi \approx 2.9$rad, so that $d \approx \lambda/6$.

We have implemented feedback-based on-demand mobility on top of a gradient-based routing protocol based on the framework described in Section 3. The data injection policy requires the source to send a burst of P packets every $T_b = 3$s (at a rate $1/T_P$); we use $P = 100$, $T_P = 25$ms. Note that we use this particular data injection policy just to present and assess the reactive mobility concept; in practice, the protocol can be used with the data injection policy that best suits the particular application. Link estimation is performed based on the RSS of a single control packet [3], and the link blacklisting policy requires the acceptance of a packet over link (j, i) if node i measures an RSS larger than -85dBm. In our experiments, we have verified that this form of blacklisting typically identifies an overlay network with symmetric links. The remaining energy of a node is estimated based on the square of the battery voltage. The packet size is 19 bytes for data packets and 14 bytes for control traffic. The backoff period (in ms) is $T_i = T_{ds} + 10i$, where T_{ds} is a downstream processing delay (to ensure that all the setup packets coming from the downstream neighbors are received and processed) equal to 100ms (which has been

empirically verified to work for our network size). As for the route selection metric, we set $\eta = \zeta = 1/3$.

The sink is normally static and computes the packet delivery rate based on the sequence numbers of the received data packets (feedback from the network). In all experiments, all nodes other than the sink are static. The maintenance request condition is a packet delivery rate of 0 over $T_{\rm b}$, which sets the maintenance step counter k_m to 1. The maintenance step counter is incremented every $T_{\rm b}$ while the packet delivery rate remains at 0, and the maintenance actions are route rediscovery with no sink motion if k_m is odd, and sink motion if k_m is even. Figure 7 shows the packet delivery rate as estimated by node S based on packet sequence numbers over a 10-packet window. A number of events happen in the system, causing it to react. Namely:

- E1: node 3 is turned off; the system reacts with a new route discovery process.
- E2: node 4 is turned off; the sink cannot find a new route from its current position and moves to different positions until it is able to find a route.
- E3: node 2 is turned off; the sink cannot find a new route from its current position and moves to a different position.
- E4: the motion of two people in the surroundings of the sink cause the packet delivery rate to drop to 0, and the sink keeps using the same route without moving.

In the experiment, nodes 5 and 6 are never selected; node 3 carries most of the traffic until it is turned off. The average data packet delivery rate for the whole experiment (excluding the time spent searching for new routes, during which no data transmissions are performed) is about 95%. Considering that we employ no link layer acknowledgments, no retransmissions, no handshakes (such as Request To Send/Clear To Send), no channel coding, and no buffering at the relays, the performance of our protocol is very promising.

5. PERFORMANCE EVALUATION

5.1 Protocol specifications and experimental setup

For the performance analysis, we have modified some of the features of the protocol used for the proof of concept in Section 4. Link estimation is still RSS-based, but the link blacklisting policy is now stricter: a packet over link (j, i) is accepted if node i measures an RSS larger than -80dBm and an LQI above 100 (roughly corresponding to an estimated packet delivery rate of at least 0.95). The backoff period T_i $(i > 0)$ is now drawn from a uniform distribution between 0 and 250ms. T_0 is set to 300ms. The maintenance request condition is a packet delivery rate estimate lower than 80% or the assignment of an excessive workload to a sink neighbor. Workload is estimated based on the number of packets relayed and is deemed to be excessive if more than 500 packets in a row are relayed by the same node. Control traffic is not considered for workload estimation due to its negligible size with respect to data traffic (nodes relay one control packet per route discovery process and at least 100 data packets). A route is not modified if it yields a packet delivery rate of 80% or above, but sink neighbors subject to an excessive workload are allowed to *cheat*, *i.e.* propagate fake state information upstream. For example, a node i in these conditions would propagate $L_i = 0$ (unusable channel), $V_i = 0$ (no remaining energy) and $H_i = \infty$ (infinite distance to the sink). This mechanism ensures that node i will not be selected by any of its upstream neighbors at the next route discovery process, unless it is the only node available (or all sink neighbors are cheating). Relay i $(i > 0)$ monitors its own activity (number of packets relayed since the last sleep period, whose long-term time average is indicated by A_i) and go into sleep mode for $T_{\rm sleep}$=10s if inactive (as a relay) for at least 10s.

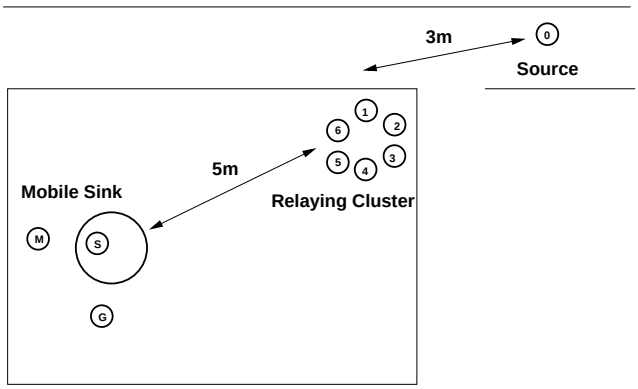

Figure 8: Layout for the fault tolerance and load balancing assessment (experiments in 5.2 and 5.3.1).

5.2 Fault tolerance

We use the experimental setup in Figure 8 to assess the fault tolerance of our network with and without reactive mobility. The source 0 can communicate with the sink S by way of the network formed by nodes 1 through 6, whose antennae are about half a wavelength (6.25cm for MICAz) apart. With respect to the setup in Figure 8, every time an integer multiple of 1000 packets has been received by the sink, we emulate node failure by shutting down the relay that gave the most significant contribution (in terms of successfully relayed packets) in the last 1000-packet window. Figure 9 shows the packet delivery rate as measured by the sink (averaged over the 1000 packets in between successive emulated node failures) for 4 different experiments (2 different static sink positions and 2 different starting positions for the mobile sink). Depending on the position of the static sink, as few as 2 node failures may impede communication. With sink mobility, even 5 node failures (out of 6 available relays) do not compromise the reliability of the system.

5.3 Load balancing

5.3.1 Experiment 1

For the assessment of load balancing we first use the network shown in Figure 8. We choose five static sink position and have the network relay at least 1000 data packets for each position; we then turn on node M and have the network relay 5000 data packets to a reactively mobile sink. Our cross-layer route selection metric favors short routes; in this particular setup, all routes are over two hops. We define the amount of unbalance as $\beta = (N\phi - 1)/(1 - \phi)$,

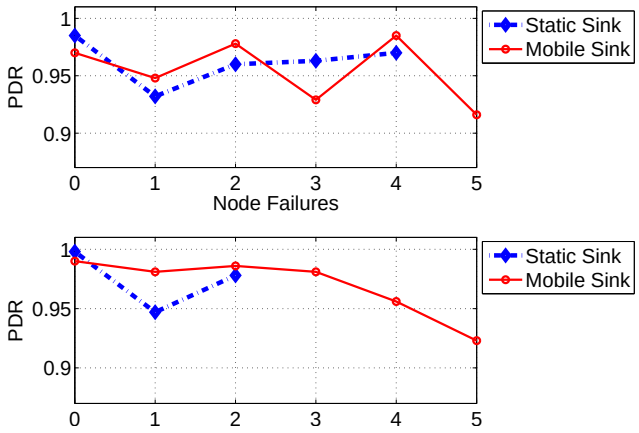

Figure 9: Sink mobility significantly improves fault tolerance, as shown by three experiments with and without sink mobility (experiment in 5.2).

Node	Stat.1	Stat.2	Stat.3	Stat.4	Stat.5	Mob.
1	0	0	25.6%	0	0	12.4%
2	45.5%	0	22.9%	0	0	17.4%
3	1.8%	30.6%	7.3%	0	46.7%	32.1%
4	0	15.3%	0	60.6%	36.9%	11.7%
5	8.2%	0	44.2%	0	0	7.2%
6	44.5%	54%	0	39.4%	16.5%	23.5%
β	3.17	4.87	2.96	6.69	3.38	1.36
PDR	99%	98.8%	96.7%	98.3%	97.4%	98.3%

Table 1: Load balancing experiment in 5.3.1.

Node	Stat.1	Stat.2	Stat.3	Stat.4	Stat.5	Mob.
1	13.7%	17.2%	34.8%	6.4%	31.7%	21.5%
2	51.3%	43.6%	48.1%	3.8%	54.2%	30%
3	24.2%	11%	23.3%	10.1%	22.7%	32.4%
4	36.6%	38%	7.5%	17.7%	14.1%	24.5%
5	34.8%	34.8%	18.9%	12.7%	19.1%	24.5%
6	15.6%	15.6%	30.7%	74.6%	39.1%	29.4%
β	4.27	2.87	3.63	13.69	4.92	1.4
PDR	97.5%	93.3%	96.1%	97%	96.3%	96.6%

Table 2: Load balancing experiment in 5.3.2.

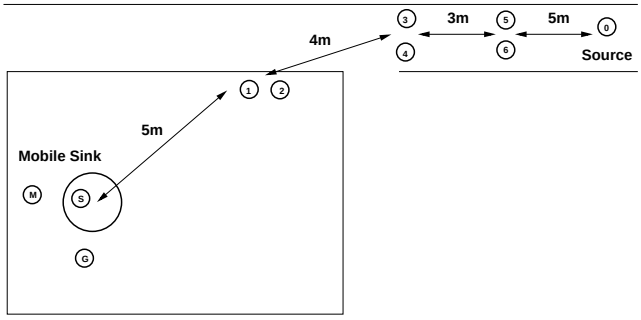

Figure 10: Layout for the load balancing assessment (in 5.3.2): the source 0 and nodes 3-6 are in a hallway and the rest of the network is inside an office.

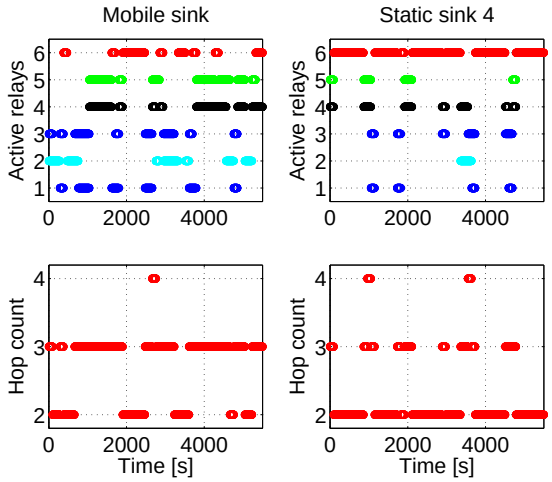

Figure 11: Load balancing assessment (5.3.2). The static sink results correspond to the column Stat.4 in Table 2.

where $\phi = \max_{i=1}^{N} A_i$ and N is the total number of relays. In a fully balanced network, each relay forwards $1/N$ of the traffic and $\beta = 0$. In a completely unbalanced network, one relay forwards all traffic and β diverges. Table 1 shows the activity (percentage of total traffic relayed) of each node, along with the amount of unbalance and the packet delivery rate. Reactive sink mobility yields the lowest amount of unbalance. Note that static positions 1, 2, and 4 yield a slightly higher packet delivery rate than the one obtained with sink mobility at the price of the heavy exploitation of one or two particular nodes.

5.3.2 *Experiment 2*

We now consider the network in Figure 10. The source is now farther away, and the 6 relays are now paired in 3 clusters. Table 2 compares the performance of our gradient-based routing scheme with and without reactive mobility for 5 different sink positions. Figure 11 compares the performance of sink mobility to a scenario with a static sink corresponding to column Stat.4 in Table 2. Again, reactive sink mobility yields the lowest amount of unbalance. From the particular position occupied by the static sink, only node 6 guarantees the desired packet delivery rate. However, our routing protocol partially compensates for the excessive workload inflicted on node 6 by mainly using 2-hop routes and allowing node 6 to periodically go into sleep mode. With mobility, it is no longer necessary to heavily leverage on node 6; however, going through any other node nearly always requires 3 or more hops, due to the network topology. In this case, a larger hop count is needed to preserve a given node; however, our scheme often reassigns the role of relay to different nodes in order to allow the exploitation of sleep modes and improve load balancing.

6. CONCLUSIONS

We have presented a scheme for the on-demand exploitation of sink mobility for reliable and load-balanced data collection in wireless sensor networks. After presenting and illustrating a gradient-based routing protocol based on a cross-layer route selection metric, we have proposed a novel

approach to the exploitation of sink mobility. Our on-demand sink mobility concept does not add any complexity at the network layer; on the contrary, it streamlines data collection by enhancing the reliability and load balancing properties of our routing protocol. We have provided a proof of concept and an experimental evaluation of the properties of our mobility-aided feedback-based scheme using the performance of our routing scheme without sink mobility as a baseline. As our experimental evidence demonstrates, limited sink mobility increases the fault tolerance of a sensor network and enhances the existing load balancing properties of the underlying routing scheme. Sink mobility dynamically modifies the set of the neighbors of the sink: the depth of a node in the routing tree varies as a function of the position of the sink, and shorter, more reliable, or less energy-costly routes can be found. It should not be overlooked that our mobility-aided solution significantly increases load balancing with the use of reliability feedback: routes are rediscovered based on the packet delivery rate measured by the sink. Sink mobility could also be triggered by load-based feedback; this is a feature that we plan to implement in a MICAz testbed to better understand the tradeoff between reliability and load balancing. Sleep mode management is an integral part of the underlying routing scheme; comparing how different sleep scheduling techniques fit in our framework will be the subject of future investigations.

Acknowledgments

The support of NSF (CNS 04-47869) and CRANE/DTRA (N00164-07-8510) is gratefully acknowledged.

7. REFERENCES

[1] M. Yarvis, W. Conner, L. Krishnamurthy, J. Chhabra, B. Elliott, and A. Mainwaring. Real-world Experiences with an Interactive Ad Hoc Sensor Network. In *International Conference on Parallel Processing Workshops (ICPP'02)*, Vancouver, BC, Canada, August 2002.

[2] D. De Couto, D. Aguayo, J. Bicket, and R. Morris. A High-Throughput Path Metric for Multi-Hop Wireless Routing. In *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking (MobiCom'03)*, San Diego, CA, USA, 2003.

[3] B. Chen, K. Muniswamy-Reddy, and M. Welsh. Ad-Hoc Multicast Routing on Resource-Limited Sensor Nodes. In *Second International Workshop on Multi-hop Ad Hoc Networks: from Theory to Reality* , Florence, Italy, May 2006.

[4] D. Puccinelli and M. Haenggi. Spatial Diversity Benefits by Means of Induced Fading. In *Third IEEE International Conference on Sensor and Ad Hoc Communications and Networks (SECON'06)*, Reston, VA, USA, September 2006.

[5] D. Puccinelli, E. Sifakis, and M. Haenggi. A Cross-Layer Approach to Energy Balancing in Wireless Sensor Networks. In *Workshop on Networked Embedded Sensing and Control (NESC'05)*, Notre Dame, IN, USA, September 2005.

[6] R. Poor. Gradient Routing in Ad Hoc Networks. 2000. Available at `http://www.media.mit.edu/pia/Research/ESP/texts/poorieeepaper.pdf`.

[7] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks. In *Proceedings of the 4th ACM International Conference on Mobile Computing and Networking (Mobicom'98)*, Dallas, TX, USA, August 2000.

[8] D. Braginsky and D. Estrin. Rumor Routing Algorithm For Sensor Networks. In *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, Atlanta, GA, USA, September 2002.

[9] F. Ye, G. Zhong, S. Lu, and L. Zhang. GRAdient Broadcast: A Robust Data Delivery Protocol for Large Scale Sensor Networks. *ACM Wireless Networks (WINET)*, 11(2), March 2005.

[10] F. Ye, A. Chen, S. Liu, and L. Zhang. A scalable solution to minimum cost forwarding in large sensor networks. In *Proceedings of Tenth International Conference on Computer Communications and Networks*, Scottsdale, AZ, USA, 2001.

[11] D. Ganesan, A. Cerpa, W. Ye, Y. Yu, J. Zhao, and D. Estrin. Networking issues in wireless sensor networks. *Journal of Parallel and Distributed Computing (Special issue on Frontiers in Distributed Sensor Networks)*, 64:799–814, December 2003.

[12] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris. Span: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks. *Wireless Networks*, 8:481–494, 2002.

[13] A. Cerpa and D. Estrin. ASCENT: Adaptive Self-Configuring sEnsor Networks Topologies. *IEEE Transactions on Mobile Computing - Special Issue on Mission-Oriented Sensor Networks*, 3(3):272–285, July-September 2004.

[14] Q. Cao, T. Abdelzaher, T. He, and J. Stankovic. Towards optimal sleep scheduling in sensor networks for rare-event detection. In *Proceedings of the 4th international symposium on Information Processing in Sensor Networks (IPSN'05)*, Los Angeles, CA, USA, April 2005.

[15] T. He, S. Krishnamurthy, J. Stankovic, T. Abdelzaher, L. Luo, R. Stoleru, T. Yan, L. Gu, and G. Zhou. VigilNet: An Integrated Sensor Network System for Energy-Efficient Surveillance. *ACM Transactions on Sensor Networks*, 2(1):1–38, February 2006.

[16] S. Gandham, M. Dawande, R. Prakash, and S. Venkatesan. Energy efficient schemes for wireless sensor networks with multiple mobile base stations. In *Proceedings of the IEEE 2003 Global Communications Conference (GLOBECOM'03)*, San Francisco, CA, USA, Nov 2003.

[17] J. Luo and J.-P. Hubaux. Joint Mobility and Routing for Lifetime Elongation in Wireless Sensor Networks. In *"Proceedings of the 24th Annual Conference of the IEEE Communications Societies (INFOCOM'05)"*, Miami, FL, USA, March 2005.

[18] E. N. Gilbert. Random plane networks. *Journal of SIAM*, 9:533–543, October 1961.

[19] J. Luo, J. Panchard, M. Piorkowski, M. Grossglauser, and J.-P. Hubaux. MobiRoute: Routing towards a Mobile Sink for Improving Lifetime in Sensor Networks. In *International Conference on Distributed Computing in Sensor Systems (DCOSS'06)*, San Francisco, CA, USA, June 2006.

[20] A. Woo, T. Tong, and D. Culler. Taming the Underlying Challenges of Reliable Multihop Routing in Sensor Networks. In *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys'03)*, Los Angeles, CA, USA, November 2002.

[21] Z. Wang, S. Basagni, E. Melachrinoudis, and C. Petrioli. Exploiting Sink Mobility for Maximizing Sensor Networks Lifetime. In *Proceedings of the 38th Hawaii International Conference on System Sciences*, Hilton Waikoloa Village, HI, USA, January 2005.

[22] K. Sohrabi, J. Gao, V. Ailawadhi, and G. Pottie. Protocols for self-organization of a wireless sensor network. *IEEE Personal Communications Magazine*, 7(5):16–27, October 2000.

[23] R. C. Shah, S. Roy, S. Jain, and W. Brunette. Data MULEs: Modeling and analysis of a three-tier architecture for sparse sensor networks. In *Ad Hoc Networks Journal*, volume 1, pages 215–233. Elsevier, Sep 2003.

[24] W. Zhao, M. Ammar, and E. Zegura. A message ferrying approach for data delivery in sparse mobile ad hoc networks. In *The Fifth ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'04)*, Tokyo, Japan, May 2004.

[25] P. Venkitasubramaniam, S. Adireddy, and L. Tong. Sensor Networks with Mobile Access: Optimal Random Access and Coding. *IEEE Journal on Selected Areas in Communications*, 22(6):1058–1068, August 2004.

[26] A. Somasundara, A. Kansal, D. Jea, D. Estrin, and M. Srivastava. Controllably Mobile Infrastructure for Low Energy Embedded Networks. *IEEE Transactions on Mobile Computing*, 8(8):958–972, August 2006.

[27] K. Srinivasan and P. Levis. RSSI is Under Appreciated. In *Third Workshop on Embedded Networked Sensors (EmNets'06)*, Cambridge, MA, USA, May 2006.