

Overlapping Particle Swarms for Energy-efficient Routing in Sensor Networks

Brian K. Haberman · John W. Sheppard

Received: date / Accepted: date

Abstract Sensor networks are traditionally built using battery-powered, collaborative devices. These sensor nodes do not rely on dedicated infrastructure services (e.g., routers) to relay data. Rather, a communal effort is employed where the sensor nodes both generate data as well as forward data for other nodes. A routing protocol is needed in order for the sensors to determine viable paths through the network, but routing protocols designed for wired networks and even ad hoc networks are not sufficient given the energy overhead needed to operate them. We propose an energy-aware routing protocol, based on overlapping swarms of particles, that offers reliable path selection while reducing the energy consumption for the route selection process. Our Particle-based Routing with Overlapping Swarms for Energy-Efficiency (PROSE) algorithm shows promise in extending the life of battery-powered networks while still providing robust routing functionality to maintain network reliability.

Keywords Sensor Networks · Routing · Overlapping Swarms · Quality-of-Service

1 Introduction

One of the fundamental issues with wireless sensor networks is maintaining the infrastructure to move data from low-power sensors to data collection points. Unlike

Brian K. Haberman
Department of Computer Science
The Johns Hopkins University
3400 N. Charles Street
Baltimore, MD 21218
Tel.: +410-516-5298
E-mail: bkhabs@cs.jhu.edu

John W. Sheppard
Department of Computer Science
357 EPS Building
Montana State University
Bozeman, MT 59717
Tel.: +406-994-4835
E-mail: john.sheppard@cs.montana.edu

PC-based wireless networks (e.g., IEEE 802.11), sensor networks are composed of very low-power (i.e., battery-based) devices designed to last for months or years. Typical routing protocols designed for wired and wireless networks generally do not meet the resource constraints imposed by these types of networks. The focus of the research reported here is the development of a novel swarm intelligence (SI) algorithm [Kennedy and Eberhart, 1995] to address the problem of routing in wireless sensor networks. Rather than maintain a single swarm that encompasses the entire sensor network, our approach is based on overlapping swarms that localize information dissemination.

The use of the SI approach in this environment provides two interesting advantages. The first benefit is the ability to make path determinations through the network graph based on a variety of parameters (e.g., power, hop count). This flexibility allows the network to determine the least resource intensive path at any point in time. The second benefit is the ability to localize the distribution of these parameters between nodes. Localization reduces control traffic flooding throughout the network thus reducing energy consumption while still allowing nodes to make valid route selection decisions. This paper presents the Particle-based Routing with Overlapping Swarms for Energy Efficiency (PROSE) algorithm as a potential solution to conserving energy in sensor networks.

2 Problem Statement

Wireless sensor networks (WSNs) have several unique properties that distinguish them from other networks. Sensor nodes are battery-powered, small-scale (i.e., limited CPU capacity) devices equipped with one or more sensors designed to collect targeted information. Unlike traditional wired (e.g., Ethernet) networks, typical sensor networks do not have dedicated infrastructure for forwarding data, so each node participates in data forwarding. A deployment of sensor nodes involves a large-scale placement over some area of interest to collect data. That data is then sent to a data sink (referred to as a gateway). The gateway node is usually a higher-powered processing machine capable of storing larger amounts of data and/or transmitting the data over longer distances.

With this deployment model in mind, several network-related issues arise. The primary issue concerns path determination to the gateway node. Traditional network routing protocols require regular data exchanges between peer nodes in the network to maintain path information. However, with limited power budgets, sensor nodes need alternative methods of path determination that do not consume precious energy with routine routing traffic. On the other hand, sensor nodes in WSNs do not need to know a path to all other nodes in the network, as is the case with traditional data networks. Rather, sensor nodes only need to know the possible next-hop nodes leading to the destination gateways. This reduces the path computation problem by pruning large parts of the network from the set of potential data destinations.

Ideally, a sensor node's selection of a path to the destination gateway will take various critical variables into account. Rather than always choosing the "shortest" path (e.g., fewest number of hops) as in traditional routing protocols, a node interested in maximizing the global performance of the network will incorporate feedback from its nearest neighbors when selecting a next-hop. Traditional routing protocols for both wired and wireless networks typically require state messages to be flooded throughout the network (global information) for each node to make informed and correct path

selections that are optimal. However, we hypothesize that we can make decisions strictly on local information that improve the global performance of the network, even if the path selection is suboptimal based on traditional criteria such as hop count. The results of our experiments show that the power consumption can be spread across the network, enhancing the availability of the entire network.

3 Related Work

A variety of approaches have been proposed using traditional sensor network routing protocols. In-depth surveys of these approaches are available in [Al-Karaki and Kamal, 2004] and [Akkaya and Younis, 2005]. Directed Diffusion uses a data-centric (pull vs. push) approach rather than a network layer routing scheme [Intanagonwiwat et al., 2003]. That is, the data is named using attribute/value pairs and the nodes interested in the data (sinks) broadcast queries through the network searching for the data of interest. Source nodes holding data matching the attribute/value pairs send responses back to the sink. The sink then resends the original query at different time intervals to reinforce the selected path(s).

Shah and Rabaey attempt to address the energy efficiency in sensor network routing by applying a load balancing technique [Shah and Rabaey, 2002]. Unlike other approaches that simply use energy expenditure as a link metric in a shortest-hop style protocol, their approach uses a flooding technique to discover multiple low-energy routes from a source to a destination. When data begins to flow, it is routed over the set of routes in a stochastic manner to load balance the traffic.

Several Ad hoc On-demand Distance Vector (AODV) [Perkins and Royer, 1999] variations have been proposed that attempt to make AODV more energy efficient. Pappa et al. replace the traditional AODV route metric (hop count) with a ratio of each node's remaining energy level [Pappa et al., 2007]. As the route request transits the network, it accumulates the amount of energy remaining at the nodes along its path. When the request reaches the intended target, the destination selects the request with the lowest cost, which equates to the path with the lowest energy consumption. Nie and Comaniciu propose an AODV variant designed to save energy in CDMA networks [Nie and Comaniciu, 2006]. Their approach leverages two primary changes. First, they replace the AODV route metric with a power metric that takes the transmission energy per packet into account. This is accomplished by computing the transmission power needed for the initial packet's size. Secondly, they leverage a cross-layer design to allow the network layer routing to influence the directional antennas (beamforming) used within the CDMA network. Both approaches retain the route discovery and response paradigm of traditional AODV.

Several AI-based approaches for network routing utilize the Q-learning algorithm [Watkins, 1989]. Littman and Boyan [Littman and Boyan, 1993] describe a distributed reinforcement learning approach to network routing where each node maintains statistics on the delivery time of data packets flowing through the network. Using this information, the algorithm applies a variation to the Q-learning algorithm to generate a policy that routes data traffic over the minimum number of hops. Peshkin and Savova propose a modification to this approach incorporating a Gradient Ascent Policy Search (GAPS) [Peshkin and Savova, 2002]. The GAPS approach forwards data packets in a stochastic fashion based on a learned policy reinforced by rewards provided by the network. Packets in transit carry information such as source, destination, routing history,

and origination time, which are used by intermediate nodes to either forward (based on hop count) or discard it. The rewards are used in a gradient ascent search to find the locally optimum policy.

Wang and Wang developed a variation of the Q-learning approach applying a Q-value parametric approximator rather than evaluating the optimal action/value function [Wang and Wang, 2006]. By approximating the value function using Least Squares Policy Iteration (LSPI), Wang and Wang argue that they can learn the set of actions directly from the environment (model-free), independent of initial parameter settings (e.g., learning rate) that will allow packet forwarding to follow the shortest hop path. Brown applies Q-learning in an approach that reduces power consumption and increases wireless channel utilization by replacing the number of hops metric with an energy-based metric [Brown, 2000]. This approach focuses on learning a policy for activating a radio interface and scheduling packet transmissions to increase wireless link utilization. Beyens et al. leverage data compression in their Q-learning based routing protocol [Beyens et al., 2005]. By combining data from multiple sensors they reduce the overhead associated with the data transmission. This approach provides locally optimal performance but does not increase the global performance of the network.

An approach to packet scheduling is proposed in [Melo and Coello, 2000] that applies fuzzy logic to the classification of traffic flows to meet Quality-of-Service (QoS) constraints. The classification serves as the basis for packet scheduling as data flows fill output queues and the router must determine a schedule for packet transmission that meets the QoS requirements. Their approach focuses on the classification and scheduling aspects of packet forwarding, but does not address actual route discovery.

The concept of Collective INtelligence (COIN), described in [Tumer and Wolpert, 2000], uses multi-agent systems to determine path selection through a network. In a COIN network, each agent employs a local utility function with a goal of increasing global reward. Path selection is derived from a nearest neighbor algorithm where the neighbor of interest is the one with the best global cost to the final destination. As with previously cited work, this approach focuses on selecting a single best path but does have the ability to select different paths to the same destination as long as that alternative selection benefits the collection of agents. The COIN approach comes closest to satisfying the constraints focused on in our project, albeit without specifically incorporating multiple objectives into their utility function.

A biologically inspired reinforcement learning approach to sensor network routing, called Ant Colony Routing, is described by GhasemAghaei *et al.* [GhasemAghaei et al., 2007]. The Ant Colony approach relies on the flow of artificial ants through the network to discover paths [Dorigo and Stützle, 2004]. Sources originate forward ants to discover the destination, which in turn generate backwards ants to optimize the path. As with most other approaches described here, Ant Colony Routing relies on the discovery of the minimum-hop path from a source to a destination and does not optimize global resource consumption (e.g., energy). A similar approach by Di Caro et al., however, utilizes Swarm Intelligence, based on Ant Colony Optimization, for mobile ad hoc networks [Di Caro and Dorigo, 1998][Di Caro et al., 2005]. Using a Monte Carlo learning process, AntHocNet leverages forward ants to explore complete paths from a source to a destination and backwards ants to reinforce the pheromone table entry for the destination at the intermediate nodes. Their approach focuses more on finding single best paths (based on hop count and path delay) between pairs of nodes. A variation of AntHocNet based on Particle Swarm Optimization (PSO) is described in [Zhang and Xu, 2006]. Their approach utilizes a swarm updating forward and backward agent/particles

similar to the AntHocNet approach. The distinction occurs in the particle state update function where a variety of QoS metrics are incorporated.

A variety of geographic routing solutions have been developed to conserve energy in battery-based networks. Traditional routing protocols require periodic hello (or beacon) messages to determine adjacent neighbors. Heissenbüttel et al. developed beacon-less routing (BLR) algorithm in which each node is unaware of its neighbors [Heissenbüttel et al., 2004]. In BLR, a node simply broadcasts any data packets it has enqueued. When a neighboring node receives the packet, it computes a timeout value based on its position relative to the transmitting node and the destination node. The node with the best location will have the smallest timeout value and will transmit the packet first. Other approaches, such as EBGR [Zhang and Shen, 2010] and IGF [Blum et al., 2003], implement the same idea within the IEEE 802.11 Media Access Control (MAC) layer. Many of these approaches integrate their energy conservation mechanisms within the lower layers of the protocol stack in order to ensure tight control over power consumption. Another limitation of these approaches is the need for some type of location information. In order to determine the proximity of other nodes, each device must have a GPS, location database, or some other localization service.

Within the PSO literature, several applications of multiple swarms have been proposed. In [Brits et al., 2007], a global swarm is fragmented into niche swarms to find multiple optimal solutions. The niches fragment from the original swarm based on the discovery of a candidate solution, fragmenting the global population. Each niche is said to converge once it finds a solution and maintains the solution through a sufficient number of training iterations. A key difference between nichePSO and PROSE is the level of interaction between swarms. Within nichePSO, sub-swarms do not share information whereas PROSE is predicated on information sharing between overlapping swarms. Wang et al. propose the use of multiple, non-overlapping particle swarms to optimize the coverage area of a sensor network [Wang et al., 2007]. The algorithm focuses on optimal placement of sensors (both stationary and mobile) to find the maximal coverage area. Given the high dimensionality of the n -dimensional particle vectors, their algorithm splits the swarm into n swarms, each focused on finding a solution to a single component of the solution vector. While their approach focuses on sensor networks, it does not address the routing of data within the sensor network. It strictly solves the sensor coverage aspect of deployment.

4 Particle-Based Route Selection

The primary focus of our research is the development of a SI-based, layer-3 approach to network routing for wireless sensor networks that spreads the traffic load across distinct paths in the network to conserve critical battery life irrespective of the underlying transmission media. Specifically, an SI algorithm using overlapping swarms is developed and applied, optimizing a local route cost function for each node in the network. This cost function captures information about the number of hops from the node to the gateway and the residual energy of the nodes along a path to the gateway. This problem can be viewed as a multi-objective optimization problem in that the path selection mechanism considers: 1) local battery level, 2) neighbor battery level, and 3) hop count to the destination when the network path is selected at the time of data transmission. The goal is to maximize aggregate throughput by extending the lifetime of the entire network while keeping battery consumption minimal; however,

we also expect our approach to be applicable to more general Quality-of-Service (QoS) parameters.

The SI model is similar to the PSO method [Kennedy and Eberhart, 1995] based on the notion of individuals or agents collaborating as members of a social unit where the dynamics of that social unit have the appearance of a “swarm.” PSO operates by creating “particles” with states that represent solutions, strategies, or attributes of a function to be optimized. In the context of WSN routing, PROSE implements one particle in each node of the network, and defines a swarm to be relative to each node. Each swarm consists of the particle for that node and the particles for all of the node’s immediate neighbors. Thus, a set of overlapping swarms is defined.

By using overlapping swarms, we distinguish between PROSE and traditional PSO algorithms. In traditional PSO algorithms, the position of each particle in a swarm encodes a complete solution to the associated optimization problem. In PROSE, the swarms overlap to account for the fact local decisions are being made at each node, thus requiring each node to solve its own optimization problem. Each particle encodes a solution to an optimization problem local to the node at the center of the swarm.

More formally, PROSE is predicated on the following. Let $W = (N, E)$ represent a WSN where N corresponds to the nodes of the network and E corresponds to edges between nodes (*i.e.*, nodes able to communicate directly with each other). Additionally, there exists at least one node $s \in N$ that is a source and a node $d \in N$ that is the destination for all sensor data collected. Each sensor node $n_i \in N$ must select a next-hop node $n_j \in N$, where $(n_i, n_j) \in E$ and the objective is for (n_i, n_j) to be on a path from s to d (*i.e.*, the gateway). For each $n_i \in N$, define a swarm $S_{n_i} = \{n_i\} \cup \{n_j \mid (n_i, n_j) \in E\}$. Let p_i represent the particle associated with node n_i . Each p_i consists of a vector of information incorporated in to the route selection process. The *state* of p_i is its current selection of a “best” next-hop node to reach the sink node. As an example, consider the scenario where n_x has three neighboring nodes (n_a , n_b , and n_c). The particles for those nodes can be represented as i:(4, 0.5), a:(3, 0.6), b:(4,0.3), c:(5,0.6). Each term in the vector is a tuple of hop count and energy for the node. The vector is then incorporated into a probabilistic decision process based on the SI model. Each particle p_i knows its current state (denoted **present**_{*i*}), remembers its best state seen so far (denoted **pbest**_{*i*}), and is aware of the best state for all of the particles in the local swarm (denoted **lbest**_{*i*}). These *best states* represent the best solutions selected for the local particle and the particles within its swarm. Each node signals its best state to the other members of its swarms. It should be noted that, over time, **pbest**_{*i*} and **lbest**_{*i*}, which are local measures, may become suboptimal. However, this is acceptable since PROSE is attempting to optimize global, rather than local, performance. For our sensor network environment, these states represent the current power level of each neighbor’s battery and the hop count to the destination through each neighbor as a single objective measure (discussed in Section 5.3). The results shown in Section 6 indicate that PROSE improves global residual energy levels using local information.

The concept of swarms within a network routing paradigm shares many characteristics with network neighborhoods described in other WSN literature. Both approaches attempt to leverage localized interactions in order to minimize their impact on the network as a whole. Additionally, both techniques try to utilize superior peers to route traffic to the intended destination. However, the SI-based swarm approach differs from traditional neighborhood-based approaches by incorporating stochastic decision making rather than deterministic logic. The use of randomness within swarms allows them

```

function initProse()
  initializeDataStructures()
  start proseRcvThread()
  WHILE (TRUE)
    sendPSAMessage()
    sleep(msgInterval)
  ENDWHILE

```

Fig. 1 PROSE Initialization (leveraging the PSA message sending function defined in Figure 3).

to recover from local optima and incorporate a measure of exploration in the route selection process.

Similar to PSO, each particle also has a velocity vector (denoted \mathbf{v}_i) associated with it to determine how the state will change from one step to the next. The velocity vector is used to update the underlying state. The velocity is adjusted based on the communal experience of the swarm using:

$$\begin{aligned}
 \mathbf{v}_i &= \omega \times \mathbf{v}_i \\
 &+ d_1 \times \text{rand}() \times (\mathbf{pbest}_i - \mathbf{present}_i) \\
 &+ d_2 \times \text{rand}() \times (\mathbf{lbest}_i - \mathbf{present}_i)
 \end{aligned} \tag{1}$$

where $\text{rand}()$ represents the generation of two independent uniform random numbers in the range (0..1) that prevents particles from settling into a local optimum, d_1 and d_2 are constants used to tune performance, ω is an inertia weight to manage stability, and vector operations are component-wise. Once the particle's velocity has been updated, it then updates its present state ($\mathbf{present}_i$). When a particle updates its velocity using Equation 1, it includes \mathbf{lbest}_i as a component. Since each particle n_i is both the centroid of the swarm S_{n_i} and a member of $|nbr(n_i)| - 1$ other swarms, we define \mathbf{lbest}_i to be the best state (minimum objective value) the particle has seen in all of its swarms (*i.e.*, the best of the best). More formally,

$$\mathbf{lbest}_i = \arg \min_{n_i \in S_{n_i}} f(\mathbf{lbest}_{n_i}) \tag{2}$$

5 The PROSE Algorithm

The PROSE algorithm is composed of four high-level operations. These operations interact at each node to learn, disseminate, and use the available paths through the network. These components are:

1. Particle State Advertisement
2. Particle State Update
3. Best Neighbor Selection
4. Route Selection

While these operations are described in the following sub-sections, the general setup and logic flows for PROSE are shown in Figures 1, 2, and 3.

```

function proseRcvThread()
  initializeRcvSocket()
  WHILE (TRUE)
    msgPSA ← receiveMessage()
    particleStateUpdate(msgPSA)
    updateVelocity() // Equation 4
    IF (local state changed)
      computeLBest()
      sendPSAMessage()
    ENDIF
  ENDWHILE

```

Fig. 2 PROSE Receive Thread.

```

function sendPSAMessage()
  neighbor ← findBestNeighbor()
  new(PSAMessage) // Allocate memory for PSAMessage
  PSAMessage.batteryLevel ← Current battery level
  PSAMessage.hopCount ← getHopCount(neighbor)
  PSAMessage.nextHop ← neighbor
  sendMessage(PSAMessage)

```

Fig. 3 Algorithm for sending PSA messages.

5.1 Particle State Advertisement

To create and maintain its local swarm, each particle must advertise its current state periodically to the other members of the swarm. A node's local swarm is comprised of all of its direct neighbors learned through the periodic PSA message exchange. The periodic state advertisements provide updates to the other particles in the swarm on the best path available through the advertising particle. Additionally, if a particle undergoes a significant change in its state, it will advertise its new state on demand to expedite changes in its swarm (e.g., if a node's energy level dropped 10% since the last periodic PSA message). This dampening reduces the chance of route oscillations since the information being exchanged only changes when a particle's best state changes. The PSA message informs neighboring particles of the components that comprise the state of the particle, namely its battery level and current hop count to the data sink. The receiver of the PSA will compute the scalar state based on the advertised information (Figure 3).

The PSA message is the key component in reducing the amount of control traffic sent over the entire network. As each particle's state changes, the PSA message only advertises the particle's *best* information from one swarm to another. This reduces the amount of overhead messaging that is normally transmitted to affect route selection. The function call to *findBestNeighbor*() returns information on the particle seen as being the current best choice to reach the destination. Further description of the selection process occurs in Section 5.3.

```

function particleStateUpdate(PSAMessage)
  sender ← source of PSAMessage
  IF (sender not in Neighbor Table)
    addNeighbor(sender)
  ENDIF
  batteryLevel(sender) ← PSAMessage.batteryLevel
  hopCount(sender) ← PSAMessage.hopCount + 1
  nextHop(sender) ← PSAMessage.nextHop
   $v_i \leftarrow v_i + d_1 \times \text{rand}() \times (pbest_i - present_i)$ 
     $+ d_2 \times \text{rand}() \times (lbest_i - present_i)$ 

```

Fig. 4 Algorithm for updating particle state.

5.2 Particle State Update

The central component of our PROSE algorithm is the particle state update process. Particle state update occurs in two steps—incorporating information from neighboring states into its own state, and adjusting the state based on the particle update equation shown in Equation 1. We describe the first here and discuss the second in the context of neighbor selection in Section 5.3 below. Each PROSE particle in a swarm maintains a swarm neighbor table that tracks the state of its neighboring particles. Upon reception of a PSA message, a particle parses the message and updates/adds the sending particle’s information in its state table (shown in Figure 4). In addition, Equation 2 is invoked to update \mathbf{lbest}_i .

5.3 Best Neighbor Selection

In the PROSE algorithm, each node in the network optimizes a local function representing the anticipated impact traffic will have on global network availability to determine where to forward packets it has received. The state of each particle i in each swarm contains two attributes— $Hopcount_i$ and $ConsumedBattery_i$. The objective function optimized is simply:

$$f(i) = C_1 \times Hopcount_i + C_2 \times ConsumedBattery_i \quad (3)$$

As the objective function, Equation 3 is used to determine which neighbor currently represents the best next-hop towards the sink. The weighted aggregation approach is employed over other approaches (*e.g.*, dominance-based) in order to avoid expensive processing in the sensor nodes. Since we are trying to reduce both energy consumption and the hop count of the path, the best neighbor will have the minimum objective value of all known neighbors. This process is represented in Figure 5.

Once the best neighbor is found, the current node updates its velocity using Equation 1. With the new velocity, the particle then updates its state $\mathbf{present}_i$ as follows:

$$\mathbf{present}_i = \mathbf{present}_i + \mathbf{v}_i \quad (4)$$

```

function findBestNeighbor()
  objectiveLevel  $\leftarrow$  MAXINT
  FORALL (neighbor in neighborTable)
    currObj  $\leftarrow$   $C_1 \times \text{hopCount}(\text{neighbor})$ 
      +  $C_2 \times \text{ConsumedBattery}(\text{neighbor})$ 
    IF (currObj < objectiveLevel)
      bestNeighbor  $\leftarrow$  neighbor
      objectiveLevel  $\leftarrow$  currObj
    ENDIF
  ENDFOR

```

Fig. 5 Algorithm for finding the best neighboring node.

```

function findBestRoute()
  continue  $\leftarrow$  TRUE
  WHILE (continue)
    bestNeighbor  $\leftarrow$  findBestNeighbor(presenti)
    IF (nextHop(bestNeighbor)  $\neq$  self)
      continue  $\leftarrow$  FALSE
    ELSE
      mark bestNeighbor unusable
    ENDIF
  ENDWHILE

```

Fig. 6 Algorithm for deciding the next best hop along the route.

5.4 Route Selection

In PROSE, a reactive model is used where route entry creation is under the control of the particle's current state (**present**_{*i*}), which represents where the particle resides in a state space. It does not represent which node in the neighbor table is the current best selection. Rather it represents the particle's vector through state space trying to reach **lbest** and **pbest**_{*i*}. The combination of the random component in Equation 1 and the overlapping swarms allows energy levels to be propagated through the network so that the reactive route selection is not entirely local, even though the PROSE message exchanges are completely local.

With PROSE, the route selection mechanism chooses a next-hop based on the particle's current state, whenever a data packet needs forwarding and a route entry does not already exist. Our approach relies on the current state (**present**_{*i*}) of the particle and the information in the neighbor table to select a next-hop for packet forwarding. The *findBestNeighbor*() function call (Figure 5) searches the neighbor table for the best objective value relative to **present**_{*i*}. This neighbor is selected as the next-hop for the pending data packet. A neighbor is excluded (marked as *unusable*) from use if it has indicated that its next-hop to the sink is the node performing the route lookup, thus preventing local (tight) loops from forming. The route selection mechanism is outlined in Figure 6. The route entry remains in the route table as long as there is data flowing through the node to the destination. The route entry times out after a period of inactivity or the residual energy of the current next-hop in the route entry drops below a threshold value.

This approach reduces the amount of routing state maintained across the entire network. This reduction occurs not only for the forwarding state maintained in each node, but also in the amount of control traffic generated within the network. Typical reactive routing protocols, like AODV, generate localized messages (*i.e.*, HELLO messages) that facilitate localized neighbor discovery as well as flooded route discovery messages to find paths through the network. PROSE does not need network-wide discovery messages since the overlapping swarms propagate the information needed to find a path.

6 Experimentation

Given the differences mentioned between traditional routing protocols and a swarm-based approach, we divided our experimentation into three phases. The first phase of testing focused on testing the capability of PROSE to make sound route selection decisions. This was accomplished by focusing on finding the minimum hop count path to the destination. For this first phase, the user-defined variables in Equations 1 and 3 are set as follows: $d_1 = 0.33$, $d_2 = 0.67$, $\omega = 1$, $C_1 = 1.00$, and $C_2 = 0.00$. The second and third phases tested the capability of PROSE to reduce overall energy consumption across the network by incorporating energy and hop count in the objective function. The user-defined variables in Equations 1 and 3 for the last two phases are set to: $d_1 = 0.33$, $d_2 = 0.67$, $\omega = 1$, $C_1 = 0.99$, and $C_2 = 0.01$. The following sections describe the setup and simulation environment for each experimentation phase.

6.1 Simulation Environment

All of the simulation results presented here were collected using the Network Simulator-2 (ns-2) [USC/ISI, 2007], version 2.32. The PROSE algorithm was tested against the Ad hoc, On-Demand Distance Vector (AODV) routing protocol [Perkins and Royer, 1999] included in the ns-2 distribution, the Directed Diffusion routing protocol [Intanagonwiwat et al., 2003] also included in the ns-2 distribution, and the energy-aware AODV variant described in [Pappa et al., 2007]. All nodes used in the simulation environment operated a single IEEE 802.15.4 radio interface with an omni-directional antenna. The traffic sources generated a Constant Bit Rate (CBR) traffic stream to simulate sensor data collected at set intervals for PROSE, AODV, and the energy-aware AODV while the data sink queried sensors on the same intervals for Directed Diffusion. For the energy-based routing, the standard ns-2 energy model was utilized to simulate the energy consumption in each node with each node being initialized with 10 Joules of energy.

6.2 NS-2 Energy Model

The energy model in ns-2 provides an abstract mechanism for simulating energy consumption within a node. It maintains the total energy level of the node based on an initial energy level and a discrete set of user-configurable, action/energy settings. The action/energy variables allow a simulation to control the energy expended per action. For networking nodes, a simulation can assign an energy expenditure level for packet

transmission, packet reception, active/sleep transition, and operating in an idle mode. For the simulations discussed here, the default values provided for the IEEE 802.15.4 MAC layer were used for packet transmission and reception (281.8 mWatts).

It should be noted that this energy model is simplistic in nature. It does not support parameterization on operations such as variable packet transmission power. Such flexibility would allow for a higher fidelity model of wireless networks [Xue et al., 2007]. However, given the simplicity of the energy model, it provides a consistent model that allows for direct comparisons of resource consumption between multiple protocols operating under the same conditions.

6.3 Shortest Hop Emulation

The first step in evaluating PROSE focused on emulating a traditional minimum-hop metric. That is, the objective function selected next-hop neighbors to minimize the number of nodes a message traversed to reach the destination. This was accomplished by setting C_1 to 1 and C_2 to 0 in Equation 3. This test provides insight into the route selection capabilities of PROSE in comparison to a baseline routing protocol.

This minimum-hop variation of PROSE was first tested against AODV in a 100 node (arranged in a 10×10 grid) sensor network illustrated in Figure 7. Within this network, one node acted as the destination gateway and three nodes acted as sources. The topology of the network is representative of a sensor deployment that would be found in warehouse environments, albeit with fewer data transmitters. Each of the sensors (shaded nodes in the figure) generated CBR traffic destined for the destination gateway node (labeled 0). The simulation was run 100 times for each routing protocol (PROSE and AODV). During each simulation run (lasting 90 seconds), the routing tables of each node were tracked to determine the paths selected. The energy level of each node was also collected at the end of each simulation run even though energy utilization was not the focus of this initial test.

The AODV simulation runs focused all the traffic along a common path (Figure 7). Such traffic localization leads to increased energy expenditure along the popular path (*i.e.*, under PROSE, these nodes had a significantly higher residual energy level than under AODV). The PROSE simulation runs demonstrated better path diversity (Figure 8) through the network while only using hop-count as the objective metric. The selection of parallel paths through the dense network distributed the workload among a larger population of nodes while still selecting minimum hop paths. These results demonstrate that PROSE is capable of making good routing decisions using a standard routing metric while still distributing the work across more of the network.

Additional minimum-hop tests were carried out with two smaller networks using the same number of sources/sinks, CBR traffic pattern, simulation runs, and run duration. In an 11-node network there was not a statistically significant difference in residual energy between AODV and PROSE. In a 21-node network, AODV had a statistically significant higher residual energy level.

6.4 Energy-based Routing

Given the capability demonstrated by PROSE in selecting minimum-hop paths, the second phase of experimentation focused on incorporating energy levels into the objec-

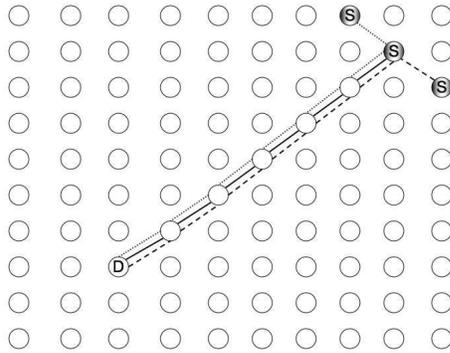


Fig. 7 100-node sensor network

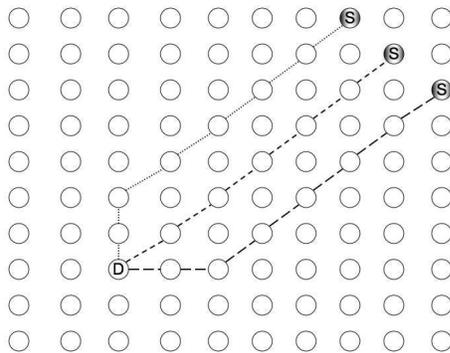


Fig. 8 100-node PROSE path selection

tive function. For the following simulation runs, the variables C_1 and C_2 are set to 0.99 and 0.01, respectively. These values for C_1 and C_2 were selected by a parametric analysis designed to evaluate the influence of individual variables and their interactions. The selected values provided sufficient weight to both the hop count and consumed energy levels while still resulting in distinct values. Intuitively, the higher weight for the hop count ($C_1 \gg C_2$) aided in the prevention of loops while still providing sufficient weight to the energy level. The hypothesis is that we will see statistically significant improvement in the difference between average residual energy with the sensor network operating with the PROSE algorithm versus the AODV, energy-aware AODV (EA-AODV), and Directed Diffusion algorithms.

The experimentation with the energy-based objective function primarily utilized the network illustrated in Figure 7. Three sets of 100 simulation runs were carried out where the only difference between the sets was the duration of a simulation run. The simulation run times were 90, 900, and 6000 seconds. These time values were chosen by iteratively running simulations and examining the resulting trace files for energy depleted nodes. As with the hop-count experiment, each node is initialized with 10 J of energy.

The first set of simulation runs lasted 90 seconds each. As with the hop-count experimentation on this network size, AODV routed all of the sensor data traffic along

Table 1 Residual energy analysis

Node Count	Duration	Protocol	Ave. Res. Energy	St. Dev.	Two-tailed t -test p -value
100	90 sec	PROSE	9.76 J	0.114	
100	90 sec	AODV	8.93 J	0.996	2.89E-13
100	90 sec	EA-AODV	9.64 J	0.278	0.0001
100	90 sec	Dir. Diff.	9.78	0.039	0.51
100	900 sec	PROSE	9.39 J	0.29	
100	900 sec	AODV	6.01 J	3.29	7.9E-18
100	900 sec	EA-AODV	7.64 J	2.28	1.39E-12
100	900 sec	Dir. Diff.	9.20 J	0.48	0.00018
100	6000 sec	PROSE	7.78 J	1.760	
100	6000 sec	AODV	5.94 J	2.795	2.12E-11
100	6000 sec	EA-AODV	6.82 J	2.285	5.52E-08
100	6000 sec	Dir. Diff.	7.21 J	2.092	0.0002
76	6000 sec	PROSE	7.32 J	1.174	
76	6000 sec	EA-AODV	6.24 J	1.944	1.85E-09
76	6000 sec	Dir. Diff.	6.51 J	1.884	1.08E-06

the same path. On the other hand, PROSE varied the paths it selected based on energy consumption utilizing a diverse set of nodes to conserve energy.

In Figure 9, the non-shaded nodes had higher residual energy levels when operating with the AODV protocol. The shaded nodes had higher residual energy levels with our PROSE algorithm. The energy analysis for this simulation is shown in Table 1. The paired, two-tailed t -test p -value of 2.89E-13 indicates a high statistically significant difference between PROSE and AODV. Also of note is the low (0.114) value of the standard deviations within the PROSE samples. The residual energy level is also significantly higher for PROSE than it is for the energy-aware AODV (t -test p -value of 0.0001). This indicates that PROSE is outperforming AODV even when the routing metric in EA-AODV incorporates energy expenditure.

The comparison of PROSE to Directed Diffusion shows that the average residual energy levels are not significantly different. In these shorter-term experiments PROSE is providing essentially the same energy savings as Directed Diffusion.

While the energy conservation of all the protocols differed, the packet delivery ratio (number of packets transmitted by the sources / number of packets received by the sink) for all protocols was over 94%. This shows that all approaches are providing reliable transport across the sensor network.

This first simulation run shows promising initial results, but does not demonstrate behavior of the protocols once nodes begin depleting their batteries. To observe the protocols behaviors under these conditions, the next set of simulation runs increased the simulation run time to 900 seconds. The ten-fold increase in simulation run time shows PROSE has a dramatic increase in performance over AODV, a significant difference with the energy-aware AODV, and an increasing advantage over Directed Diffusion (t -test p -value of 0.00018) with respect to average residual energy (Table 1).

Given the superiority of PROSE over the two AODV protocols in the 90-second experiments, the increased performance in the 900-second experiments were expected. The encouraging result for PROSE is its ability to outperform Directed Diffusion. The statistically significant difference in residual energy can be attributed to the larger amount of network-wide control traffic needed by Directed Diffusion to establish a route from source to destination.

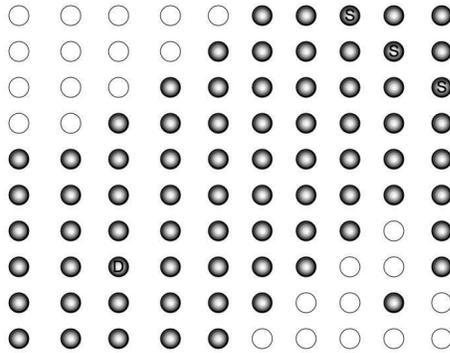


Fig. 9 AODV Energy levels in 100-node, 90-second test

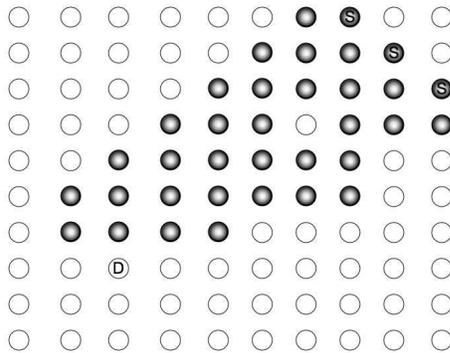


Fig. 10 AODV energy loss pattern in 100-node, 900-second simulation runs

Another observation made during the 900-second simulation run was the distinct pattern of energy loss under AODV. In Figure 10, the shaded nodes all exhibited an average residual energy level at least 2 Joules less than the average for AODV. This energy loss pattern can be attributed to the AODV approach to overloading the “popular” path. The single non-shaded node surrounded by shaded nodes in Figure 10 is an interesting case. This node, while falling along the shortest-hop path, appears to have been bypassed by AODV once surrounding nodes began failing. The ns-2 traces indicate this node fell out of RF transmission range once its adjacent nodes began to fail. Since it did not have to route traffic under AODV, it conserved its energy levels. All the other protocols exhibited a more evenly dispersed loss of energy.

The final energy-based simulation run increased the simulation time to 6,000 seconds. These simulation runs allowed for a sufficient number of node failures leading to network fragmentation under AODV. That is, no usable path existed from the sources to the destination by the end of the simulation run. As nodes fail, the routing protocols must discover alternate paths to the destination (since they are all reactive protocols). Under AODV, these alternate paths are discovered once a failure is detected on the previously chosen path. Upon this discovery, the AODV protocol re-performs the route discovery mechanism based only on the hop-count metric. PROSE, on the other hand, dynamically changes routes throughout the simulation run based on the objective function calculation, spreading the depletion of energy evenly across a more

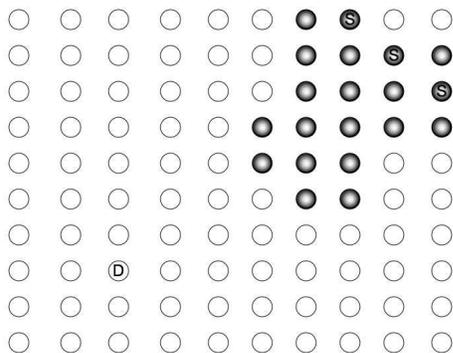


Fig. 11 AODV nodes failing in 100-node, 6000-second simulation runs

diverse set of nodes. Directed Diffusion and the energy-aware AODV also spread the depletion across a wider variety of nodes. However, their routes are only changed when the route discovery process is invoked.

The primary observation for this third set of simulations is the average residual energy level shown in Table 1. The average residual energy level for AODV is a misleading indicator of the protocol's behavior with respect to energy. The average over the whole network does not reflect the points of failure within the simulation runs. The shaded nodes in Figure 11 indicate those nodes that consistently exhausted their batteries, leading to network fragmentation. The pattern of nodes indicates the fragmentation that can occur within the network leading to AODV being unable to route data traffic to the destination node. The loss of viable paths between the sources and destination led to AODV having a packet delivery ratio under 75%. PROSE, on the other hand, exhibited no complete battery drain and was able to retain a high (>95%) packet delivery ratio.

PROSE, Directed Diffusion, and EA-AODV did not encounter any complete failure of nodes in this experiment. For each of these protocols, the energy expenditure was more evenly distributed across the network. But, while Directed Diffusion and the EA-AODV maintained a >95% packet delivery ratio, there was significantly less residual energy (*t*-test *p*-values of 0.0002 and 5.52E-08 respectively). The energy consumption differences between PROSE, Directed Diffusion, and the EA-AODV can be traced to the amount of control traffic needed to establish and maintain routes through the network and the dynamic re-routing when a node's energy level drops below a threshold. PROSE is able to establish a route without any end-to-end route discovery exchange (*e.g.*, AODV RREQs and RRSPs). This reduces the amount of additional traffic needed to establish routes and conserve the battery level. Additionally, the ability to re-route a packet when a node's energy level drops below a threshold allows PROSE to distribute the routing workload more evenly.

The application of an energy-based objective measure shows promise across all the test topologies employed so far. The first node failure under PROSE did not occur until the simulation was allowed to run for over 11000 seconds and that failure only occurred in a single simulation run.

7 Future Work

These initial experiments show promise in the use of Particle Swarm Optimization techniques leveraging overlapping swarms in network routing. Going forward there are several areas of continued research that should be explored.

1. The operation of PROSE will be examined under a wider variety of topologies. Variations will include factors such as network density, inter-node distances, multiple destinations, increased number of sources, mobility, multi-hop loop detection, and dynamic power adjustment.
2. The current PROSE algorithm uses a swarm size defined as all nodes in direct RF contact with the root of the swarm (also called one-hop neighbors). Future studies will measure protocol performance with other swarm sizes to determine an optimal swarm size (e.g., all two-hop neighbors).
3. PROSE experiments were not run to complete network disconnection. We will explore the time to network failure while using PROSE, similar to results reported in [Shah and Rabaey, 2002].
4. The simulations reported here do not incorporate mobile nodes. Future work will incorporate mobile nodes within the simulation to determine the impact of node movement on the PROSE algorithm.
5. Other statistics, such as RF parameters, will be incorporated into the objective function to offer a finer granularity of control over a particle's state, thus providing a generalized framework for QoS-based routing.

8 Conclusion

Our research shows promise in the application of Swarm Intelligence to the problem of routing in sensor networks. Of key interest is the reduction of the routing control traffic while still maintaining a high packet delivery ratio. The energy savings garnered with this reduction increases the useful lifetime of the sensor network. PROSE also shows significant performance improvements over two representative energy-conserving routing protocols.

The results measuring residual energy on the 10×10 grid, where there is a large number of available paths, show that PROSE outperforms EA-AODV by a substantial margin but has a similar performance to Directed Diffusion. Even so, in both cases, the advantage held by PROSE over the other two protocols is statistically significant. With the more restrictive 76-node, two-quadrant topology, fewer alternate paths are provided. In this case, PROSE still provides a substantial, statistically significant improvement in residual energy over both protocols. While the two topologies have similarities, they provide vastly different challenges in route selection due to a bottleneck existing in one but not the other. In both scenarios, PROSE outperforms Directed Diffusion and EA-AODV, thus suggesting its potential for superior performance in a wide variety of contexts.

The application of overlapping swarms for sensor network routing allows particles to optimize routes based on energy levels to extend the lifetime of the network. The results of our work indicate that the PROSE approach may provide a generalized QoS-routing framework for sensor and other types of networks. This is supported by the ability to replace the objective function based on parameters of interest (e.g., hop

count versus hop count and energy) without changing any other aspect of the protocol. By extending network lifetime, sensor network users are subject to less administrative overhead such as replacing depleted nodes. Our PROSE approach is just a first step in investigating PSO-based solutions to hard networking problems.

Acknowledgment

We would like to thank Steve Butcher, Ben Mitchell, Ian Nowland, Chuck Robertson, Pat Donnelly, Omar Zaidan, and the anonymous reviewers for their feedback and suggestions that greatly improved the initial version of this work.

References

- K. Akkaya and M. Younis. A survey on routing protocols for wireless sensor networks. *Ad Hoc Networks*, 3(3):325 – 349, 2005.
- J.N. Al-Karaki and A.E. Kamal. Routing techniques in wireless sensor networks: a survey. *IEEE Wireless Communications*, 11(6):6–28, 2004.
- P. Beyens, M. Peeters, K. Steenhaut, and A. Nowe. Routing with compression in wireless sensor networks: a q-learning approach. In *Proceedings of the Fifth Symposium on Adaptive Agents and Multi-Agent Systems*, Paris, France, 2005. AAMAS.
- B. Blum, T. He, S. Son, and J. Stankovic. Igf: A state-free robust communications protocol for wireless sensor networks. Tech. report, University of Virginia, 2003.
- R. Brits, A.P. Engelbrecht, and F. van den Bergh. Locating multiple optima using particle swarm optimization. *Applied Mathematics and Computation*, 189:1859–1883, 2007.
- T.X. Brown. Low power wireless communication via reinforcement learning. *Advances in Neural Information Processing Systems (NIPS)*, 12:893–899, 2000.
- G. Di Caro and M. Dorigo. Antnet: Distributed stigmergetic control for communications networks. *Journal of Artificial Intelligence Research*, 9:316–365, 1998.
- G. Di Caro, F. Ducatelle, and L.M. Gambardella. Swarm intelligence for routing in mobile ad hoc networks. *Proceedings of the 2005 IEEE Swarm Intelligence Symposium, SIS 2005*, pages 76–83, 2005.
- M. Dorigo and T. Stützle. *Ant Colony Optimization*. The MIT Press, 2004.
- R. GhasemAghaei, A. Rahman, W. Gueaieb, and A. El Saddik. Ant colony-based reinforcement learning algorithm for routing in wireless sensor networks. *Proceedings of the 2007 IEEE Instrumentation and Measurement Technology Conference, IMTC 2007*, pages 1–6, May 2007.
- Marc Heissenbüttel, Torsten Braun, Thomas Bernoulli, and Markus Wälchli. Blr: Beacon-less routing algorithm for mobile ad hoc networks. *Computer Communications*, 27(11):1076–1086, 2004.
- C. Intanagonwivat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva. Directed diffusion for wireless sensor networking. *IEEE/ACM Transactions on Networking*, 11(1):2–16, Feb 2003.
- J. Kennedy and R. Eberhart. Particle swarm optimization. *Proceedings of the IEEE International Conference on Neural Networks*, 4:1942–1948 vol.4, 1995.
- M. Littman and J. Boyan. A distributed reinforcement learning scheme for network routing. In *Proceedings of the International Workshop on Applications of Neural Net-*

-
- works to Telecommunications*, pages 45–51, Hillsdale, NJ, 1993. Lawrence Erlbaum Associates.
- Alencar Melo and Juan Manuel Adan Coello. Packet scheduling based on learning in the next generation internet architectures. *Fifth IEEE Symposium on Computers and Communications, ISCC 2000*, 00:773, 2000.
- N. Nie and C. Comaniciu. Energy efficient aodv routing in cdma ad hoc networks using beamforming. *EURASIP Journal on Wireless Communications and Networking*, 2006(2):14–14, 2006.
- K. Pappa, A. Athanasopoulos, E. Topalis, and S. Koubias. Implementation of power aware features in aodv for ad hoc sensor networks. a simulation study. *IEEE Conference on Emerging Technologies and Factory Automation, ETFA*, pages 1372–1375, Sept. 2007.
- C.E. Perkins and E.M. Royer. Ad-hoc on-demand distance vector routing. *Proceedings of the Second IEEE Workshop on Mobile Computing Systems and Applications, WMCSA '99*, pages 90–100, 1999.
- L. Peshkin and V. Savova. Reinforcement learning for adaptive routing. *Proceedings of the International Joint Conference on Neural Networks, IJCNN '02*, 2:1825–1830, 2002.
- R.C. Shah and J.M. Rabaey. Energy aware routing for low energy ad hoc sensor networks. *2002 IEEE Wireless Communications and Networking Conference, WCNC '02*, 1:350–355 vol.1, 2002.
- K. Tumer and D. Wolpert. Collective intelligence and braess' paradox. In *Proceedings of the AAAI Conference on Innovative Applications of Artificial Intelligence*, pages 104–109, 2000.
- USC/ISI. The network simulator ns-2, 2007. URL <http://www.isi.edu/nsnam/ns/>.
- Ping Wang and Ting Wang. Adaptive routing for sensor networks using reinforcement learning. *Sixth IEEE International Conference on Computer and Information Technology, CIT'06*, 0:219, 2006.
- Xue Wang, Sheng Wang, and Jun-Jie Ma. An improved co-evolutionary particle swarm optimization for wireless sensor networks with dynamic deployment. *Sensors*, 7(3): 354–370, 2007.
- Chris Watkins. *Learning from Delayed Rewards*. PhD thesis, King's College, Cambridge, 1989.
- Yunjiao Xue, Ho Sung Lee, Ming Yang, P. Kumarawadu, H.H. Ghenniwa, and Weiming Shen. Performance evaluation of ns-2 simulator for wireless sensor networks. *2007 Canadian Conference on Electrical and Computer Engineering, CCECE 2007*, pages 1372–1375, 2007.
- Haibo Zhang and Hong Shen. Energy-efficient beaconless geographic routing in wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 21:881–896, 2010. ISSN 1045-9219. doi: <http://doi.ieeecomputersociety.org/10.1109/TPDS.2009.98>.
- X. Zhang and W. Xu. *QoS Based Routing in Wireless Sensor Network with Particle Swarm Optimization*, volume 4088/2006, pages 602–607. Springer Berlin / Heidelberg, 2006.