Swarm Intelligence manuscript No.
(will be inserted by the editor)

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 particle swarms, 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 · Particle Swarms · Overlapping Swarms · Quality-of-Service

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

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 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 algorithm based on the Particle Swarm Optimization (PSO) (Kennedy et al, 2001) concept to address the problem of routing in wireless sensor networks. Rather than maintain a single particle swarm that encompasses the entire sensor network, our approach is based on overlapping swarms that localize information dissemination.

The use of the PSO 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 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.

More formally, we consider a graph $G = (V, E)$ with a set V of vertices and a set E of edges connecting them. Additionally, there exists a node $d \in V$ that is the destination for all sensor data collected. Each sensor node $s \in V$ must select a next-hop

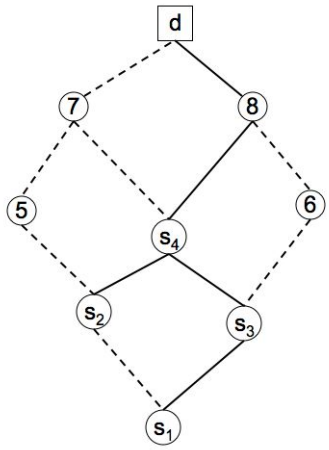

Fig. 1 Example Wireless Sensor Network

node y, where $(s, y) \in E$ and the objective is for (s, y) to be on a path to d (i.e., the gateway).

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. For example, consider the network shown in Figure 1. Assume we wish to select paths from nodes S_1, S_2, S_3, and S_4 to the destination gateway d. The solid segments represent paths that would be selected using traditional shortest-hop metrics and approaches. However, by failing to use alternative, available paths, this route would most likely lead to the depletion of the batteries at nodes along that best path. Instead, path selection should consider alternative paths that distribute the power consumption around the network and extend the battery life of the nodes along the popular path. For example, node S_4 can provide an update to S_2 indicating a reduced resource level. S_2 can incorporate the updated information into its path selection, and potentially choose an alternative path through node 5 in order to reduce the load imposed on S_4.

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 optimize the global performance of the network, even if the path selection is suboptimal based on traditional criteria such as hop count. If a node's path selection chooses the same path for all traffic, the nodes along that path consume more power and become the weak point in the network. 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

The application of Artificial Intelligence (AI) algorithms in general and PSO in particular to the problem of network routing has been sporadic. While many AI algorithms in general are designed to search a large state space, it appears that the potentially rapidly changing state space in data communications networks has resulted in limited application of these algorithms to networking problems. The following describes several approaches that have been explored in applying AI techniques to the problem of network path determination.

Several approaches based on AI techniques for network routing utilize the Q-learning algorithm (Watkins, 1989). Littman and Boyan (1993) describe a distributed reinforcement learning approach to network routing. In this approach, 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 (2006) developed a variation of the Q-learning approach applying a Q-value parametric approximator rather than evaluating the optimal action/value function. By approximating the value function using Least Squares Policy Iteration (LSPI), Wang and Wang argue that they can learn the set of actions in a model-free environment independent of initial parameters settings (e.g., learning rate) that will allow packet forwarding to follow the shortest hop path. Brown (2000) 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. The Brown approach focuses on learning a policy for when to turn a radio interface on/off and when to schedule packet transmissions in order to increase wireless link utilization. The limitation of Brown's approach comes in the simplistic model used (battery-powered client communicating to a base station with reliable power). While the approach provides significant power savings by controlling the radio's duty cycle, arbitrarily disabling the radio interface can impact other nodes' data traffic. 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 but with a goal of increasing

global reward. The actual 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. 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 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. In simulated experiments, they claim to be able to perform some amount of load balancing due to the probabilistic nature of using the information from ants. Even so, their approach focuses more on finding single best paths (based on hop count and path delay) between pairs of nodes. A PSO-based variation of AntHocNet 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. While the focus of their approach is on using PSO in sensor networks, the presented results are based on a single swarm and do not address energy efficiency.

In the realm of traditional sensor network routing protocols, a variety of approaches have been proposed. 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 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 in order to reinforce the selected path(s). Traditional routing algorithms are built on a push paradigm where the source initiates the operation. Directed Diffusion uses a pull paradigm which conceptually leads to energy savings due to data aggregation within the network.

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 in order to load balance the traffic. While the goals and end results are similar, the primary difference between this approach and PROSE is that the amount of control traffic generated in order to discover the routes is much lower in PROSE.

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 (discussed later in Section 6.2).

Within the PSO literature, several applications of multiple swarms have been proposed. In (Brits et al, 2007), the global swarm is fragmented into niche swarms in order 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, Wang, and Ma (2007) propose the use of multiple, non-overlapping particle swarms to optimize the coverage area of a sensor network. 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. In addition, their approach is an off-line algorithm where PROSE is an on-line approach.

4 Particle Swarm Route Selection

The primary focus of our research is the development of a PSO-based approach to network routing for wireless sensor networks that spreads the traffic load across distinct paths in the network to conserve critical battery life. Specifically, a PSO 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 of the entire network while keeping battery consumption minimal. It should be noted that while the focus here is on energy consumption, we expect our approach to be applicable to more general Quality-of-Service (QoS) parameters.

The PSO model is 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. According to Kennedy and Eberhart, PSO dynamics exhibit five essential features of a biological swarm (Kennedy and Eberhart, 1995)(Kennedy et al, 2001):

1. *Proximity* (behavior is based on particles in close proximity to one another) - PROSE focuses information exchange to occur only between direct neighbors
2. *Quality* (PSO maintains individual and global best estimates for behavior in the swarm) - PROSE maintains best estimates for each particle and each swarm
3. *Diversity of response* (changes to both local and global best estimates result in diverse changes to behavior of individual particles) - PROSE particles react to changes in the best estimates based on their current state and their calculated velocity
4. *Stability* (social behavior changes only occur when the global best "state" changes) - PROSE updates are driven by the velocity update which is controlled by the difference between a particle's current state and the best state
5. *Adaptability* (the social behavior actually does change over time) - PROSE selects new paths through the network as the resource consumption changes the network's dynamics.

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 these overlapping swarms, we make a significant distinction between PROSE and traditional PSO algorithms. In traditional PSO algorithms, the position of each particle in a swarm encodes a 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. Thus, the position of each particle encodes a solution to a specific optimization problem from the perspective of the node at the center of the swarm.

More formally, PROSE provides a modified PSO-based algorithm 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). For each $n_i \in N$, define a swarm $S_{n_i} = \{n_i\} \bigcup \{n_j \mid n_j \in nbr(n_i)\}$ where $nbr(n_i) = \{n_j | (n_i, n_j) \in E\}$. Each particle n_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 $lbest_i$). 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.

Each particle also has a velocity vector (denoted v_i) associated with it to determine how the state will change from one step to the next based on this feedback. The velocity vector is used to update the underlying state. The velocity is adjusted based on the

communal experience of the swarm using:

$$v_i = v_i + d_1 \times rand() \times (pbest_i - present_i) + d_2 \times rand() \times (lbest_i - present_i) \quad (1)$$

where $rand()$ represents the generation of a uniform random number in the range (0..1) that prevents particles from settling into a local optimum, and d_1 and d_2 are constants used to tune performance. Once the particle's velocity has been updated, it then updates its present state ($present_i$). When a particle updates its velocity using Equation 1, it includes $lbest_i$ 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 $lbest_i$ 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,

$$lbest_i = \min_{n_i \in S_{n_i}} lbest_{n_i} \quad (2)$$

5 The PROSE Algorithm

Our use of particle swarms for network route calculation relies on four primary functions. Those functions are:

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

5.1 Particle State Advertisement

In order to maintain the swarm, each particle must advertise its current state periodically to the other members of the swarm. The periodic state advertisements provide updates to the other particles in the swarm on the best path available to them 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. The Particle State Advertisement (PSA) message informs neighboring particles of the components that comprise the scalar 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. The logic for this function is shown in Algorithm 1.

Algorithm 1 sendPSAMessage

```
neighbor ← findBestNeighbor()

Allocate memory for PSAMessage
PSAMessage.batteryLevel ← Current battery level
PSAMessage.hopCount ← getHopCount(neighbor)
PSAMessage.nextHop ← neighbor

sendMessage(PSAMessage)
```

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 in order 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 a later section.

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. This function can be represented as shown in Algorithm 2.

Algorithm 2 parsePSAMessage

```
sender ← source of PSAMessage

if sender not in Neighbor Table then
   addNeighbor(sender)
end if

batteryLevel(sender) ← PSAMessage.batteryLevel
hopCount(sender) ← PSAMessage.hopCount + 1
nextHop(sender) ← PSAMessage.nextHop
```

It should be noted that the particle state update step must increment the hop count in order to accurately reflect the additional hop in the path to the destination that traverses the neighbor sending the PSA. The neighboring particle's next-hop is also maintained for loop detection during the route selection step.

5.3 Best Neighbor Selection

Ultimately a PSO algorithm focuses on optimizing an objective function defined over some state space. 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 $Consumed\_battery\_level_i$. The specific objective function optimized is shown in Equation 3.

$$f(i) = C_1 \times Hopcount_i + C_2 \times Consumed\_battery\_level_i \tag{3}$$

As the objective function, Equation 3 is used to determine which of a node's known neighbors is the *best*. That is, it uses this function to determine which neighbor currently represents the best next-hop towards the sink. 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 Algorithm 3.

Algorithm 3 findBestNeighbor

```
objectiveLevel ← MAXINT
for all neighbors in neighborTable do
  currentObjective ← C1 × hopCount(neighbor) + C2 × batteryLevel(neighbor)

  if currentObjective < objectiveLevel then
    bestNeighbor ← neighbor
    objectiveLevel ← currentObjective
  end if
end for
```

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 $present_i$ using Equation 4.

$$present_i = present_i + v_i \tag{4}$$

5.4 Route Selection

Routing protocols in general can be categorized as either *proactive* or *reactive*. Proactive routing protocols attempt to maintain consistent routing information for every node in the network. When the topology changes, route updates are propagated throughout the network in order to maintain the consistent view. Reactive protocols, on the other hand, create route entries for destination nodes only when needed (typically driven by the arrival of a data packet). Once a route to the target destination is established, a maintenance procedure preserves the route entry until it is no longer needed (timed out) or a topology change makes the route invalid (route failure).

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 node, based on the particle's current state, whenever a data packet needs forwarding and a route entry does not already exist. That is, if a route to the intended destination is already in the route table, it is used. Otherwise, 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 (Algorithm 3) 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 loops from forming. The route selection mechanism is outlined in Algorithm 4. 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.

Algorithm 4 findBestRoute

```
continue ← TRUE
while continue do
  bestNeighbor ← findBestNeighbor(present_i)

  if nextHop(bestNeighbor) ≠ self then
    continue ← FALSE
  else
    mark bestNeighbor unusable
  end if
end while
```

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 which find paths through the network. PROSE eliminates the need for 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 PSO-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. 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 are defined in Table 1. The following sections describe the simulation environment and the experimental setup for each experimentation phase.

Table 1 PROSE user selected parameter settings

Experiment	d_1	d_2	C_1	C_2
Hop Count	0.33	0.67	1.0	0
Energy	0.33	0.67	1.0	0.01

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 AODV

The AODV protocol is a routing protocol originally designed for mobile *ad hoc* networks (MANETs). It operates using a request/reply model for discovering routes to a target destination when needed. It is considered an efficient protocol since it only establishes a path to a destination when there is data traffic for that node and discards the route once the data flow terminates.

When a source node has data for a particular destination, it first checks its routing table to see if a route exists. If it does not exist, the node broadcasts a Route Request (RREQ) packet across the network. Each node receiving the RREQ adds a route to the source of the packet to its routing table. If the receiving node is the target destination or has an active route to the target destination, it responds to the RREQ with a Route Reply (RREP) packet. Otherwise, it rebroadcasts the RREQ packet.

The RREP message propagates back to the sender via the route entries established by the traversal of the RREQ. At each node in the path, the source of the RREP is added to the routing table and then the RREP is sent to the next-hop. Once the RREP reaches its destination, a bi-directional route exists in the network to support communications between the source node and the target destination.

While AODV was not originally designed for use in sensor networks, it has been widely applied to that domain due to its route management efficiency. Nodes unaffected by a particular data flow are not burdened with maintaining routing information for either the source of destination of the flow. In addition, the route maintenance based on active data flows allows forwarding state to be managed without additional, explicit signaling. Several variations of AODV have been proposed which incorporate energy conservation into the route selection mechanism (Pappa et al, 2007)(Nie and Comaniciu, 2006).

Table 2 NS-2 energy model settings

Initial Energy	Transmit Energy	Receive Energy	Transition Energy	IDLE State
10 Joules	281.8 mWatts	281.8 mWatts	N/A	N/A

6.3 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 how much energy is expended per action. For networking nodes, a simulation can assign an energy expenditure level for:

- Transmitting a packet
- Receiving a packet
- Transitioning from active to sleep state
- Operating in an IDLE state

For the simulations carried out in our work, the default values provided for the IEEE 802.15.4 MAC layer were used (Table 2).

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.

6.4 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 2. 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 in order to determine the paths selected. Additionally, the energy level of each node was 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 2). Such traffic localization leads to increased energy expenditure along the popular path. The PROSE simulation runs demonstrated better path diversity (Figure 3) through the

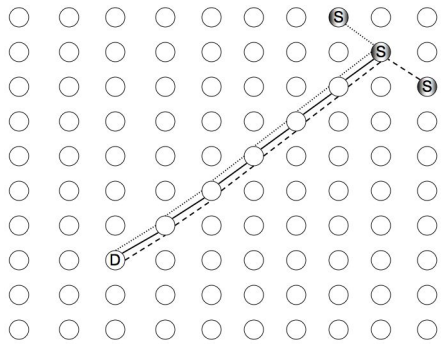

Fig. 2 100-node sensor network

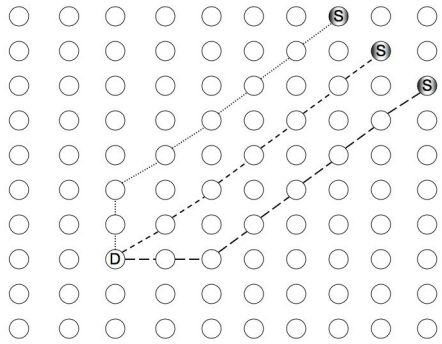

Fig. 3 100-node PROSE path selection

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.

6.5 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 objective function. For the following simulation runs, the variables C_1 and C_2 are set to 1.0 and 0.01, respectively. These values for C_1 and C_2 were selected by a factorial 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. 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 2. Three sets of 100 simulation runs were carried out

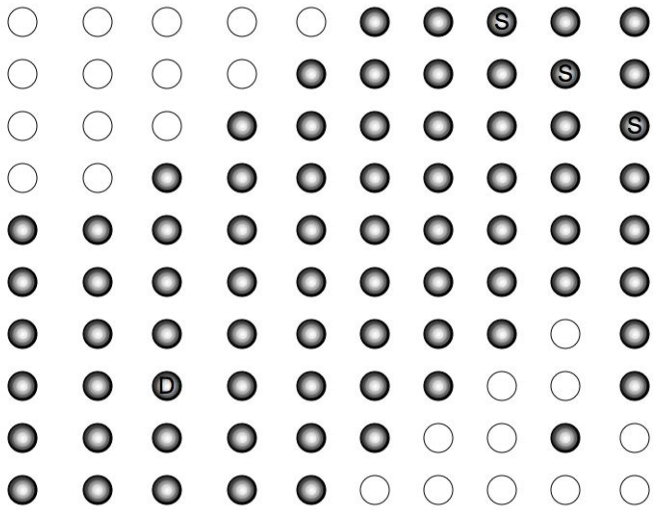

Fig. 4 AODV Energy levels in 100-node, 90-second test

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

In Figure 4, 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 3. 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

Table 3 Residual energy analysis (Energy - 100 node network, 90 seconds)

Protocol	Average Residual Energy	Standard Deviation	Two-tailed t-test p-value
PROSE	9.76 J	0.114	
AODV	8.93 J	0.996	2.89E-13
EA-AODV	9.64 J	0.278	0.0001
Dir. Diff.	9.78	0.039	0.51

Table 4 Residual energy analysis (Energy - 100 node network, 900 seconds)

Protocol	Average Residual Energy	Standard Deviation	Two-tailed t-test p-value
PROSE	9.39 J	0.29	
AODV	6.01 J	3.29	7.9E-18
EA-AODV	7.64 J	2.28	1.39E-12
Dir. Diff.	9.20 J	0.48	0.00018

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 4).

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.

Another observation made during the 900-second simulation run was the distinct pattern of energy loss under AODV. In Figure 5, 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 5 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 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.

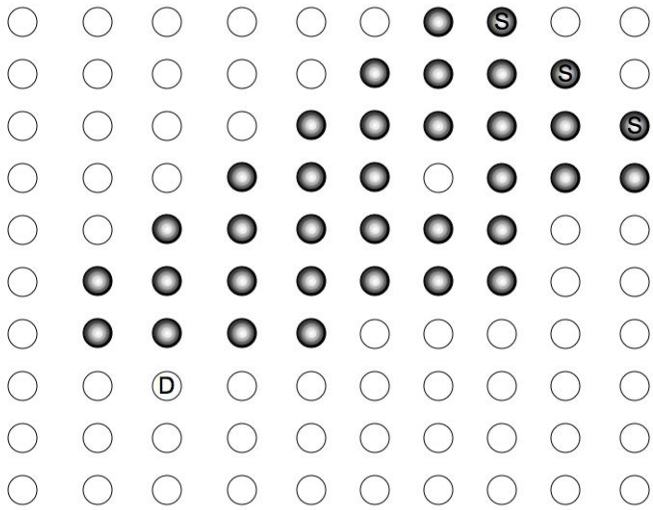

Fig. 5 AODV energy loss pattern in 100-node, 900-second simulation runs

Table 5 Residual energy analysis (Energy - 100 node network, 6000 seconds)

Protocol	Average Residual Energy	Standard Deviation	Two-tailed t-test p-value
PROSE	7.78 J	1.760	
AODV	5.94 J	2.795	2.12E-11
EA-AODV	6.82 J	2.285	5.52E-08
Dir. Diff.	7.21 J	2.092	0.0002

The primary observation for this third set of simulations is the average residual energy level shown in Table 5. 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 6 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

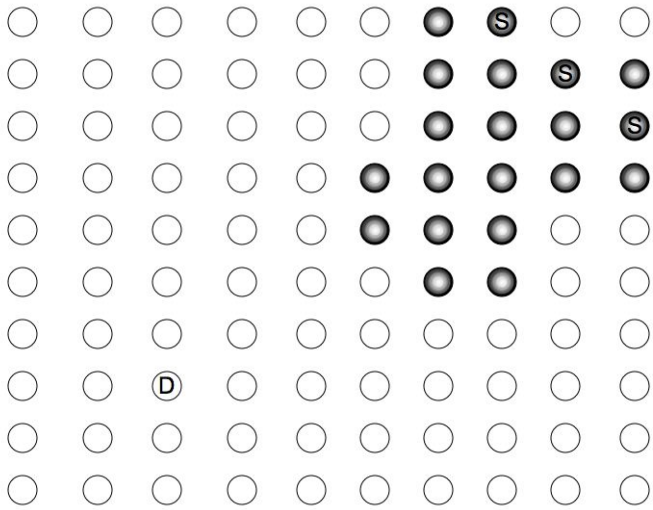

Fig. 6 AODV nodes failing in 100-node, 6000-second simulation runs

Table 6 Residual energy analysis (Energy - 76 node network, 6000 seconds)

Protocol	Average Residual Energy	Standard Deviation	Two-tailed t-test p-value
PROSE	7.32 J	1.174	
EA-AODV	6.24 J	1.944	1.85E-09
Dir. Diff.	6.51 J	1.884	1.08E-06

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. It should be noted that 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.

6.6 Alternative Grid Topology

Performance results in (Shah and Rabaey, 2002) are based on an alternative grid-like topology where the nodes are clustered into two quadrants with a small number of intermediate nodes that link them together. The last set of experiments reported here are based on a reproduction of that topology (without any nodes being mobile). The reader is directed to (Shah and Rabaey, 2002) for a more complete description of the physical topology. Due to time constraints, the average results are based on 10 simulation runs for each protocol where each run lasts for 6000 seconds. Given the performance of baseline AODV in the previous experimental phase, it was not included in this set of experiments.

Table 6 shows that PROSE has statistically significant higher residual energy than Directed Diffusion and the energy-aware AODV algorithms. As with the previous experiments, these results show that PROSE not only has a higher average residual

energy, the standard deviation indicates that the residual energy is more evenly dispersed across the nodes in the network. While the averages for EA-AODV and Directed Diffusion appear to be similar and much closer than their results shown in the previous section, the difference is statistically significant (t-test p-value of 0.00026).

Given the relatively few paths available between the two quadrants of the network, it is particularly encouraging that PROSE outperformed Directed Diffusion by a substantial margin in this particular topology and continued to maintain its considerable margin over EA-AODV. As noted before, this is due to the elimination of any end-to-end route discovery messages in PROSE. This reduces the load on nodes that are not in the path between the source and destination nodes. In the more restrictive topology used in this phase, the elimination of this control traffic precludes nodes on the fringes of the network from having to process route discovery messages. This is borne out by analyzing the average residual energy for the eight nodes that join the two quadrants. In these eight nodes, the average residual energy favors PROSE by only 0.074 Joules and is not statistically significant (*i.e.*, the t-test p-value is 0.733). Thus the bulk of the energy savings must have been garnered in the nodes not directly involved in the packet forwarding between quadrants. This indicates PROSE takes better advantage of having more options for routing than the other protocols, even when such a bottleneck exists.

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. **Additional Topologies** - The operation of PROSE should be examined under a wider variety of topologies. Variations should include network density, inter-node distances, multiple destinations, increased number of sources, and mobility.
2. **Variation in swarm size** - 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). Research should be done to measure protocol performance with other swarm sizes to determine an optimal swarm size (e.g., all two-hop neighbors).
3. **Time to fragment analysis** - The PROSE algorithm was not simulated until complete network disconnection. A study is needed to determine the time to network failure for PROSE similar to results reported in (Shah and Rabaey, 2002).
4. **Applicability to mobile environments** - The simulations reported in this paper do not incorporate mobile nodes. Future work will incorporate mobile nodes within the simulation in order to determine the impact of node movement on the PROSE algorithm.
5. **Alternative objective functions** - Other statistics, such as RF parameters, can be incorporated into the objective function to offer a finer granularity of control over a particle's state providing a generalized framework for QoS-based routing.

8 Conclusions

Our research shows promise in the application of Particle Swarm Optimization 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 in order 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.

Acknowledgements 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

Akkaya K, Younis M (2005) A survey on routing protocols for wireless sensor networks. Ad Hoc Networks 3(3):325 – 349

Al-Karaki J, Kamal A (2004) Routing techniques in wireless sensor networks: a survey. Wireless Communications, IEEE 11(6):6–28

Beyens P, Peeters M, Steenhaut K, Nowe A (2005) Routing with compression in wireless sensor networks: a q-learning approach. In: Proceedings of the Fifth Symposium on Adaptive Agents and Multi-Agent Systems, AAMAS, Paris, France

Brits R, Engelbrecht A, van den Bergh F (2007) Locating multiple optima using particle swarm optimization. Applied Mathematics and Computation 189:1859–1883

Brown T (2000) Low power wireless communication via reinforcement learning. Advances in Neural Information Processing Systems (NIPS) 12:893–899

Di Caro G, Ducatelle F, Gambardella L (2005) Swarm intelligence for routing in mobile ad hoc networks. Swarm Intelligence Symposium, 2005 SIS 2005 Proceedings 2005 IEEE pp 76–83

GhasemAghaei R, Rahman A, Gueaieb W, El Saddik A (2007) Ant colony-based reinforcement learning algorithm for routing in wireless sensor networks. Instrumentation

and Measurement Technology Conference Proceedings, 2007 IMTC 2007 IEEE pp 1–6

Intanagonwiwat C, Govindan R, Estrin D, Heidemann J, Silva F (2003) Directed diffusion for wireless sensor networking. Networking, IEEE/ACM Transactions on 11(1):2–16

Kennedy J, Eberhart R (1995) Particle swarm optimization. Neural Networks, 1995 Proceedings, IEEE International Conference on 4:1942–1948 vol.4

Kennedy JF, Eberhart RC, Shi Y (2001) Swarm Intelligence. Morgan Kaufmann Publishers

Littman M, Boyan J (1993) A distributed reinforcement learning scheme for network routing. In: Proceedings of the International Workshop on Applications of Neural Networks to Telecommunications. Alspector, J., Goodman, R. and Brown, T. X. (Ed.), Lawrence Erlbaum Associates, Hillsdale, NJ, pp 45–51

Melo A, Coello JMA (2000) Packet scheduling based on learning in the next generation internet architectures. iscc 00:773

Nie N, Comaniciu C (2006) Energy efficient aodv routing in cdma ad hoc networks using beamforming. EURASIP J Wirel Commun Netw 2006(2):14–14

Pappa K, Athanasopoulos A, Topalis E, Koubias S (2007) Implementation of power aware features in aodv for ad hoc sensor networks. a simulation study. Emerging Technologies and Factory Automation, 2007 ETFA IEEE Conference on pp 1372–1375

Perkins C, Royer E (1999) Ad-hoc on-demand distance vector routing. Mobile Computing Systems and Applications, 1999 Proceedings WMCSA '99 Second IEEE Workshop on pp 90–100

Peshkin L, Savova V (2002) Reinforcement learning for adaptive routing. Neural Networks, 2002 IJCNN '02 Proceedings of the 2002 International Joint Conference on 2:1825–1830

Shah R, Rabaey J (2002) Energy aware routing for low energy ad hoc sensor networks. Wireless Communications and Networking Conference, 2002 WCNC2002 2002 IEEE 1:350–355 vol.1

Tumer K, Wolpert D (2000) Collective intelligence and braess' paradox. In: Proceedings of the AAAI Conference on Innovative Applications of Artificial Intelligence, pp 104–109

USC/ISI (2007) The network simulator ns-2. URL http://www.isi.edu/nsnam/ns/

Wang P, Wang T (2006) Adaptive routing for sensor networks using reinforcement learning. cit 0:219

Wang X, Wang S, Ma JJ (2007) An improved co-evolutionary particle swarm optimization for wireless sensor networks with dynamic deployment. Sensors 7:354–370

Watkins C (1989) Learning from delayed rewards. PhD thesis, King's College, Cambridge

Xue Y, Lee HS, Yang M, Kumarawadu P, Ghenniwa H, Shen W (2007) Performance evaluation of ns-2 simulator for wireless sensor networks. Electrical and Computer Engineering, 2007 CCECE 2007 Canadian Conference on pp 1372–1375

Zhang X, Xu W (2006) QoS Based Routing in Wireless Sensor Network with Particle Swarm Optimization, vol 4088/2006, Springer Berlin / Heidelberg, pp 602–607