Search-based Adaptive Routing Strategies for Sensor Networks

Ying Zhang and **Markus Fromherz**
Palo Alto Research Center
3333 Coyote Hill Road
Palo Alto, California 94304
{yzhang, fromherz}@parc.com

Abstract

Real-time or agent-centered search has been an active research area in AI for the past decade. These techniques have been applied to many applications including planning in robotic systems and routing in communication networks. Sensor networks are distinguished from traditional networks by characteristics such as deeply embedded routers, highly dynamic networks, resource-constrained nodes, and unreliable and asymmetric links. In this paper, we explore the space of search-based techniques for sensor networks, including piggybacked heuristics, heuristic estimation, promiscuous learning, indirect confirmation, and forward propagation. Performance evaluations for these techniques on real application scenarios are conducted on a routing simulator for sensor networks.

Introduction

Large-scale ad-hoc networks of wireless sensors have become an active topic of research (Estrin *et al.* 1999) (Kahn, Katz, & Pister 1999). Such networks share the following properties:

- *embedded routers* – each sensor node acts as a router in addition to sensing the environment;
- *dynamic networks* – nodes in the network may turn on or off during operation due to unexpected failure, battery life, or power management (Cerpa & Estrin 2002) (Chen *et al.* 2002) (Xu, Heidemann, & Estrin 2001); attributes associated with those nodes (locations, sensor readings, load, etc.) may also vary over time;
- *resource constrained nodes* – each sensor node tends to have small memory and limited computational power;
- *dense connectivity* – the sensing range in general is much smaller than the radio range, and thus the density required for sensing coverage results in a dense network;
- *asymmetric links* – the communication links are not reversible in general.

Applications of sensor networks include environment monitoring, traffic control, building management, object tracking, etc. Routing serves as a middleware component for sensor networks (Berkeley 2003). Routing in sensor networks, however, has very different characteristics than routing in traditional communication networks. First of all, address-based destination specification is replaced by a more general feature-based specification, such as geographic location (Karp & Kung 2000) (Yu, Govindan, & Estrin 2001) or information gain (Chu, Haussecker, & Zhao 2002). Secondly, routing metrics are not just shortest delay, but also energy usage and information density. Thirdly, in addition to peer-to-peer communication, multicast (one-to-many) and convergecast (many-to-one) are major traffic patterns in sensor networks. Even for peer-to-peer communication, routing is more likely to be source or destination driven than table-based (Royer & Toh 1999), and source/destination pairs often are dynamic (changing from time to time) or mobile (moving during routing).

Real-time or agent-centered search has been an active research area in AI for the past decade (Yokoo & Ishida 1999) (Koenig 2001). Routing in communication networks can be formulated as a weighted shortest path problem, which can be solved by dynamic programming off-line or real-time reinforcement learning on-line (Koenig & Simmons 1993). Q-learning, a type of reinforcement learning (Sutton & Barto 1998) (Kaelbling, Littman, & Moore 1996), has been applied to network routing, i.e., Q-routing (Boyan & Littman 1994) (Kumar & Miikkulainen 1998).

In this paper, we apply real-time search and reinforcement learning to ad-hoc routing in wireless sensor networks. We exploit the broadcast channel in wireless networks and set the communication to promiscuous mode for learning without extra control packets. We estimate initial heuristics from routing specifications and update heuristics while routing. We trade energy and latency by setting the forward propagation policy, and trade success rate and energy by the retransmission policy. Furthermore, routing metrics other than the shortest path, e.g., energy-aware routing, can be applied in this framework.

The rest of the paper is organized as follows. Section 2 presents the search-based adaptive routing strategies. An energy-aware routing specification is also developed. Section 3 analyzes the performance of these strategies in a simulation environment for sensor networks with real application scenarios.

Search-based Adaptive Routing Strategies

Sensor Network Structure

Consider a network as a graph $\langle V, E\rangle$, where V is the set of nodes and $E \subseteq V \times V$ is a set of connections, with $(v, w) \in E$ iff w can receive messages from v; v is called a *neighbor* of w and w is called a *co-neighbor* of v. For an asymmetric network, (v, w) does not imply (w, v). A *path* in the network is a sequence of nodes, $v_0, v_1, ..., v_d$, such that $(v_i, v_{i+1}) \in E$. For an ad-hoc sensor network, V or E will change from time to time, due to mobility, failures, battery life, or power management (Cerpa & Estrin 2002) (Chen *et al.* 2002) (Xu, Heidemann, & Estrin 2001).

Each node is in part defined by a set of attributes. An *attribute* is a data entity consisting of a *type* and a domain of values. An *attribute value* denotes the current value of an attribute. Attributes can be shared by neighbors — each node broadcasts its attributes initially and updates whenever there is a significant change.

Overall Framework

A routing specification may include destination specification in terms of constraints on attributes, local route constraints, and a global routing objective (Zhang & Fromherz 2004). The problem of routing can be considered as designing a policy at each node, so that the overall path from source v_0 to destination v_d — $v_0, v_1, ..., v_d$ — is *optimal*, i.e. $\min \Sigma o_{v_i}$, where o_v is the *local* cost function at node v and Σo_{v_i} is the global *additive* objective. Routing for additive objectives is a *weighted* shortest path problem.

Given a routing specification, one can define a cost function on each node, called *Q-value*, indicating the minimum cost from this node to the destination. In this framework, Q-value is piggybacked to all the packets sent from this node to the destination. For a distributed sensor network, the cost is initially unknown, and an initial estimation is made according to the type of message or by an initial flooding from the destination. Furthermore, a node also stores its neighbors' Q-values, *NQ-values*, which are estimated initially according to the neighbors' attributes and updated when packets are received from neighbors.

The search-based strategies typically consist of an *initialization* phase, a *forwarding* phase, and a *confirmation* phase. Learning happens in all phases. For each packet sent out from a node, the current Q-value of the node with respect to that destination is attached. All the nodes are set to be in promiscuous listening mode. Learning is triggered whenever a node hears a packet, and its own Q-value is updated by

$$Q_m \leftarrow (1-\alpha)Q_m + \alpha(o_m + \min_n NQ_m(n)) \qquad (1)$$

where $0 < \alpha \leq 1$ is the learning rate, $o_m > 0$ is the local objective or cost, and n is a neighbor of this node.

The forwarding phase is a policy improvement phase, deciding which neighbor to pass the packet to according to its current estimates. In this search-based routing strategy, a node holding the current packet forwards it to the neighbor for which it has the best NQ-value. This strategy has similarities to Q-routing (Boyan & Littman 1994). However, we use the broadcast channel of wireless networking, so that value iteration in reinforcement learning is realized in a cost effective way. In other words, when a packet is sent, all co-neighbors, not just the designated receiver, will hear the message and make use of the Q-value sent with the packet. Furthermore, implicit packet confirmation is used, so that if the packet (identified by its message and sequence IDs) is not received (overheard) from the forwarded node v within a certain time period (because of a dead node or an asymmetric link), the corresponding NQ-value is updated to be the largest among the neighbors, e.g., $NQ(v) = \max_n NQ(n) + c$, where c is the estimated link cost, so that v will not be selected next time. This is extremely important for dynamic and asymmetric networks. The neighborhood management (removing dead nodes from the neighbor list) and the symmetric network assumption, which are problematic for most search-based methods, can be lifted to a certain degree.

Forwarding phase:

```
received $(m, Q)$ at $w$ from node $u$ do
  $r_u \leftarrow r_u + 1$;
  if $new(m)$ then
    for all $v$ with $(v, w) \in E_m$ do
      $NQ_m(v) \leftarrow Q^0_m(v)$;
    end
    $Q_m \leftarrow Q^0_m(w)$;
  end
  if $destination_m(w)$ then
    $Q_m \leftarrow 0$;
    broadcast $(m, 0)$;
    return;
  end
  $\beta \leftarrow 1 - (l_u/(l_u + r_u))$;
  $NQ_m(u) \leftarrow (1-\beta)NQ_m(u) + \beta Q$;
  $Q_m \leftarrow (1-\alpha)Q_m + \alpha(o_m + \min_v NQ_m(v))$;
  if $designated(m)$ then
    $v \leftarrow$ argmin$_n NQ_m(n)$; (random tie break)
    send$(m, Q_m)$ to $v$;
  end
end
```

Confirmation phase:

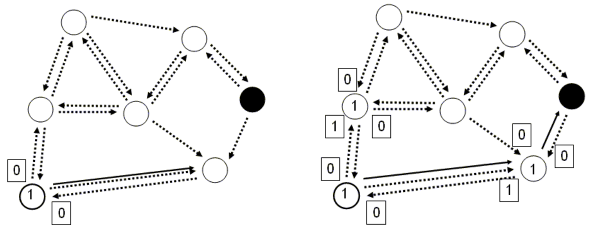
```
timeout $(m$ to $v)$ do
  $l_v \leftarrow l_v + 1$;
  $NQ_m(v) \leftarrow \max_n NQ_m(n) + c$;
  $Q_m \leftarrow (1-\alpha)Q_m + \alpha(o_m + \min_v NQ_m(v))$;
end
```

Figure 1: Framework of search-based routing strategies

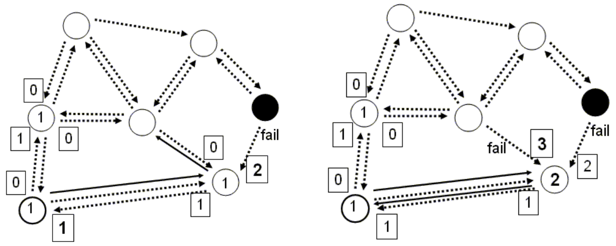
The pseudo code for the forward and confirmation phases is illustrated in Figure 1, where $Q^0_m(n)$ is an initial estimate for node n, according to the routing specification of the message and the attribute values of this node. Details will be presented in the section "Initial Heuristic Estimation". Variable r_u indicates the total number of received packets from node u and l_u indicates the total number of lost packets to node u.

When a transmission to node u fails, the link cost c to neighbor u can be set to 1, or it can be set dependent on the number of lost packets, such as e^{l_u/r_u}.

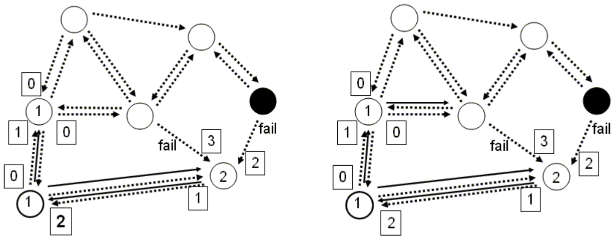
The algorithm assumes neither a static nor a symmetric network. Even with a static network, the routes may vary from time to time as a result of learning.



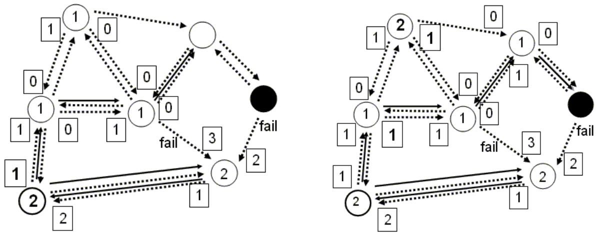
(1) starting from source; (2) sending to its best neighbor



(3) and (4) fail due to asymmetric links



(5) and (6) try another path



(7) and (8) arriving at the destination

Figure 2: An example of the real-time search-strategy

Figure 2 illustrates the search process through an example, where dotted arrows indicate the neighborhood structure. The dark circle indicates the destination. Values inside the circles indicate the estimated Q-values at the time, and values inside the squares indicate the NQ-values. In this example, $o_m = 1$, $\alpha = 1$, and initially there is no heuristic about the destination, i.e., $Q^0_m = 0$. Note that learning happens in the neighborhood of nodes, in addition to the designated node. The cost field is built up while packets are searching for the destination.

Complexity Analysis

For distributed routing approaches, one can define two types of complexity. *Message complexity* measures the total number of packets sent in the network. *Time complexity* measures the maximum number of hops from the source to a destination node. Theoretical complexities can be easily derived if the network is static and symmetric, i.e., related attributes in the routing specification do not change during routing and the set of neighbors is the same as the set of co-neighbors. For an asymmetric network, a packet can be lost if it is sent to a neighbor who is not a co-neighbor. One may engage more complex neighborhood management so that only the subset of neighbors who are co-neighbors is maintained in the neighbor list, in which case, from the routing algorithm point of view, it is a symmetric network.

The value of Q_m is called *admissible* or *under-estimated* if $0 \leq Q_m \leq Q^*_m$, where Q^*_m is the optimal value of Q_m from this node to the destination. It is easy to see that: if Q^0_m is admissible, then Q_m stays admissible given the update rule. Furthermore, $Q_m(v) \leq o^{max}_m d$, where o^{max}_m is the maximum value of the local objective function o_m, and d is the distance from v to the destination.

Theorem 1 *Given a static and symmetric network $\langle V, E\rangle$ and admissible initial estimate Q^0_m for message m, and assuming that the destination is not empty and there is a path from the source to the destination, both the message complexity and time complexity of the search-based routing are $\mathcal{O}(nd)$, where n is the number of nodes in the network and d is the diameter of the network. Furthermore, it approaches the optimal route after delivering at most n packets, and the message complexity of converging to an optimal route is bounded by $\mathcal{O}(nd)$.*

Proof. The proof is similar to that of (Koenig & Simmons 1993). With induction, one can show that $\alpha Q^t_m(v_t) + \sum_{v\in V} Q^0_m(v) + \alpha o^{min}_m t \leq \alpha Q^0_m(v_0) + \sum_{v\in V} Q^t_m(v)$ given the update rule $Q^{t+1}_m(v_t) = Q^t_m(v_t) + \alpha(o_m(v_t) + Q^t_m(v_{t+1}) - Q^t_m(v_t))$; i.e., $Q^{t+1}_m(v_t) - Q^t_m(v_t) \geq \alpha(o^{min}_m + Q^t_m(v_{t+1}) - Q^t_m(v_t))$, where o^{min}_m is the minimum value of the function o_m. Therefore, $\alpha o^{min}_m t \leq \sum_{v\in V}(Q^t_m(v) - Q^0_m(v)) + \alpha(Q^0_m(v_0) - Q^t_m(v_t))$, i.e., $\alpha o^{min}_m t \leq n o^{max}_m d + \alpha o^{max}_m d$. We have $t \leq \frac{o^{max}_m}{\alpha o^{min}_m} d(n+1)$.

A node is called *done* iff $Q_m(v) = Q^*_m(v)$. One can see that at least one node will be done after delivering a packet on a non-optimal path. Let t_i be the number of hops for delivering the i'th packet. We have $\alpha o^{min}_m t_i \leq \sum_{v\in V}(Q^{t_i}_m(v) - Q^{t_{i-1}}_m(v)) + \alpha Q^{t_{i-1}}_m(v_{t_{i-1}})$. Let T be the total number of hops for delivering n packets. We have $\alpha o^{min}_m T \leq \sum_{v\in V}(Q^T_m(v) - Q^0_m(v)) + \alpha \sum_i Q^{t_{i-1}}_m(v_{t_{i-1}}) \leq (1+\alpha) o^{max}_m nd$, i.e., $T \leq 2\frac{o^{max}_m}{\alpha o^{min}_m} nd$. ■

The worst case message and time complexities for this routing approach are not as good as those for simple flooding, which are $\mathcal{O}(n)$ and $\mathcal{O}(d)$, respectively. However, if the number of packets is large enough, the message complexity will reduce to $\mathcal{O}(d)$ on optimal routes. Since d is in general much smaller than n, this routing approach can be more efficient. Notice that if there is no path from the source to any destination node, this algorithm will not stop. In practice, one may use the maximum number of hops or the maximum Q_m value to stop further search for a destination node.

For dynamic networks some of the properties may not hold. Here we study two cases. Let $\langle V^i, E^i \rangle$ be the network and o_m^i be the local objective function at the time of routing the i'th packet of message m, respectively.

Theorem 2 *Let $V^{i+1} \subseteq V^i$ or $o_m^{i+1} \geq o_m^i$. If the initial estimates are admissible, the search-based strategy guarantees delivery for a symmetric network if a path exists. It also guarantees convergence to an optimal route if the change rate of the network is slower than the convergence rate.*

Proof. If $V^{i+1} \subseteq V^i$, there are fewer nodes left in the graph; if $o_m^{i+1} \geq o_m^i$, the optimal values increase. In both cases, Q_m^* increases. Therefore, Q_m keeps being under-estimated, satisfying the condition of delivery and convergence. $\blacksquare$

This type of dynamic network occurs often in practice, such as when nodes fail.

If the dynamic change of the network breaks the admissibility condition of the heuristic estimation, the search-based strategy may still guarantee the delivery for symmetric networks.

Theorem 3 *The search-based strategy guarantees delivery in $\mathcal{O}(nd)$ if the initial over-estimate is bounded by $\mathcal{O}(d)$.*

Proof. Given an initial estimate Q_m^0 on some node v, the global effect of v on all other nodes is bounded by $Q_m^0 + o_m^{max} d$, i.e., if Q_m^0 is bounded by $\mathcal{O}(d)$, Q_m is bounded by $\mathcal{O}(d)$. According to the analysis in Theorem 1, $\alpha o_m^{min} t \leq \sum_{v \in V}(Q_m^t(v) - Q_m^0(v)) + \alpha(Q_m^0(v_0) - Q_m^t(v_t))$, t is bounded by $\mathcal{O}(nd)$. $\blacksquare$

This type of dynamic network occurs when, for example, a new node joins the network. However, breaking the admissibility condition may cause convergence to a non-optimal route.

In asymmetric networks, packets can be lost. Implicit confirmation plays the role of adjusting a neighbor's Q-values so that the same link will not be used again. In general, search-based routing strategies only find the optimal symmetric paths rather than optimal paths in asymmetric networks.

Initial Heuristic Estimation

For the real-time search type of routing strategies, if the initial estimates are close to optimal, the actual time of delivering a single packet and the actual time of convergence to optimal routes are close to the minimum.

Destination constraints are used to estimate the minimum number of hops to the destination, while objective functions are used to estimate the minimum cost along the path. Let C be a constraint and $s(C)$ be the degree of satisfaction of C; $s(C)$ is zero iff C is satisfied. Let ΔC be the maximum change possible for $s(C)$ in one hop. The minimum number of hops based on constraint C is $h(C) = s(C)/\Delta C$. Given the destination constraint as a Boolean formula of logical **and**'s and logical **or**'s, the estimation can be computed recursively as follows. If $C = C_1 \wedge C_2$, then $h(C) = \max(h(C_1), h(C_2))$. If $C = C_1 \vee C_2$, then $h(C) = \min(h(C_1), h(C_2))$. For a geographical destination specification, h can be estimated based on the distance to the destination and the radio range. For example, if destination constraints are geographical, $C : (x = x_d) \wedge (y = y_d)$, and the maximum radio range is R, given the current location (x, y), we can obtain $h(C) = \max(|x - x_d|/R, |y - y_d|/R)$. For other types of destination specifications, e.g., an address specification C, there are no heuristics for the number of hops to the destination, so $h(C) = 0$ for admissible estimation. Therefore, geographical routing is more efficient if the locations of nodes are known.

Initialization phase:

```
initialization for $m$ at node $w$ do
  $Q_m \leftarrow 0$;
  if $destination_m(w)$ then
    broadcast $init(m, 0)$;
  end
end

received $init(m, Q)$ at $w$ from node $u$ do
  $NQ_m(u) \leftarrow Q$;
  $Q_m \leftarrow (1 - \alpha)Q_m + \alpha(o_m + \min_v NQ_m(v))$;
  if not $duplicated(init(m, Q))$ then
    broadcast $init(m, Q_m)$;
  end
end
```

Figure 3: Initialization from destination flooding

Given the estimate of the number of hops h to the destination, an admissible additive estimate Q^0 can be obtained as $o_{min} h$, where o_{min} is the minimum value of o over the network. A non-admissible estimation may be more efficient overall, but convergence to optimality cannot be guaranteed.

For search-based routing, the initialization phase is to establish a neighborhood structure, i.e., each node has a list of its neighbors. To do so, each node sends out a few "hello" packets with its attributes. If the destination is known at initialization time, which is common for applications of sensor networks (e.g., a base station), initialization can be instead a flooding from the destination. Each packet carries a Q-value of the node, and Q-values are propagated through the network. The pseudo code for flooding initialization is shown in Figure 3.

Forward Propagation

In the search-based routing framework there are no extra control packets other than for initialization. However, if there are no heuristics, it may take a long time for the first packet to find the destination. Forward propagation is a mechanism using extra control packets to speed up the learn-

Forwarding phase:

```
received $(m, Q)$ at $w$ from node $u$ do
  ...
  $Q_m^{old} \leftarrow Q_m$;
  $Q_m \leftarrow (1-\alpha)Q_m + \alpha(o_m + \min_v NQ_m(v))$;
  if $designated(m)$ then
    $v \leftarrow$ argmin$_n NQ_m(n)$; (random tie break)
    send$(m, Q_m)$ to $v$;
  elseif $|Q_m - Q_m^{old}| > \delta$
    broadcast $(m, Q_m)$
  end
end
```

Figure 4: Fragment for forward propagation

Confirmation phase:

```
timeout ($m$ to $v$) do
  $l_v \leftarrow l_v + 1$;
  $NQ_m(v) \leftarrow \max_n NQ_m(n) + c$;
  $Q_m \leftarrow (1-\alpha)Q_m + \alpha(o_m + \min_v NQ_m(v))$;
  if $(m.resend > 0)$
    $m.resend \leftarrow m.resend - 1$;
    $v \leftarrow$ argmin$_n NQ_m(n)$; (random tie break)
    send$(m, Q_m)$ to $v$;
  end
end
```

Figure 5: Confirmation with retransmission

ing process. Figure 4 shows a fragment of forward propagation, where δ is a parameter that can be set to control the number of extra packets. Forward propagation would only work to a certain degree, since collisions are introduced if there are more packets in the air.

Retransmission

In this routing framework, a packet can be lost due to asymmetric links or node failures. We use implicit packet confirmation, i.e., instead of sending a separate packet for acknowledgement, a packet is expected to be sent out by the designated node (even at the destination, in which case a broadcast is issued). If the sender has not heard the packet for a certain time after sending the packet, it can be assumed that the packet is lost. The packet can be retransmitted to increase the success rate. Figure 5 shows the confirmation with retransmission, where $m.resend$ is the number of "resends" left. One can set $m.resend$ to N initially for maximum retransmission. Similar to forward propagation, retransmission only works to the degree that the network is not overloaded.

Energy-awareness

The search-based routing framework is very general and can be applied to different routing metrics such as, for example, energy-aware routing. Shortest path routing does save the total used energy for one packet; however, it will use up the energy on the path quickly, and, in the long term, induce longer and longer paths, and eventually the network will become disconnected. Instead of using the shortest path as a routing metric o, where $o = 1$, one can use $o = ku + 1$, where u is the used energy and k is a constant. In this case, routes will prefer those nodes with less used energy.

Performance Evaluations

In this section, we analyze the effects of various strategies in the search-based routing framework using a simulator that is particularly designed for sensor networks.

Network Model

Prowler (Simon 2003), written in Matlab, is an event-driven simulator that can be set to operate in either deterministic mode (to produce replicable results while testing an algorithm) or in probabilistic mode that simulates the nondeterministic nature of the communication channel. Prowler consists of a *radio propagation model* and a *MAC-layer model*.

The radio propagation model determines the strength of a transmitted signal at a particular point of the space for all transmitters in the system. Based on this information the signal reception conditions for the receivers can be evaluated and collisions can be detected. The transmission model is given by:

$$P_{rec,ideal}(d) \leftarrow P_{transmit}\frac{1}{1+d^{\gamma}}, \quad where\ 2 \leq \gamma \leq 4 \qquad (2)$$

$$P_{rec}(i,j) \leftarrow P_{rec,ideal}(d_{i,j})(1+\alpha(i,j))(1+\beta(t)) \qquad (3)$$

where $P_{transmit}$ is the signal strength at the transmission and $P_{rec,ideal}(d)$ is the *ideal* received signal strength at distance d, α and β are random variables with normal distributions $N(0, \sigma_\alpha)$ and $N(0, \sigma_\beta)$, respectively. A network is asymmetric if $\sigma_\alpha > 0$ or $\sigma_\beta > 0$. In (3), α is static depending on locations i and j only, and β is dynamic which changes over time. A node j can receive a packet from node i if $P_{rec}(i,j) > \Delta$ where Δ is the threshold. Furthermore, an additional parameter p_{error} models the probability of a transmission error by any unmodeled effects. The MAC-layer simulates the Berkeley motes' CSMA protocol, including random waiting and back offs.

Performance Metrics

Various performance metrics can be used for comparing different routing strategies. In this paper, we only look at the following metrics:

- *latency* – the time delay of a packet from the source to the destination;
- *success rate* – the total number of packets received at the destinations vs. the total number of packets sent from the source;
- *life-time* – it is not easy to measure the life-time of a network. We introduce here a new metric, *used energy variance* $\sigma_u^2 = \frac{\sum(u_i - \bar{u})^2}{N}$, where $\bar{u}$ is the average used energy, i.e., $\frac{\sum u_i}{N}$, and N is the total number of nodes in the network. And $E - (\bar{u} + \sigma_u)$ is used to approximate the life-time metric, where E is the total initial energy at each node (full battery charge).

Evaluation of Search-based Routing Strategies

Our simulations are all performed using Prowler under the default radio model, with $\sigma_\alpha = 0.45$, $\sigma_\beta = 0.02$ and $p_{error} = 0.05$. Each test is performed with 10 runs and then averaged. The network is a 7×7 sensor grid with small random offsets. The transmit signal strength is set to 1, so that the maximum radio range is about $3d$, where d is the distance between two neighbor nodes in the grid. Figure 6 shows an instance of the connectivity of such a network.

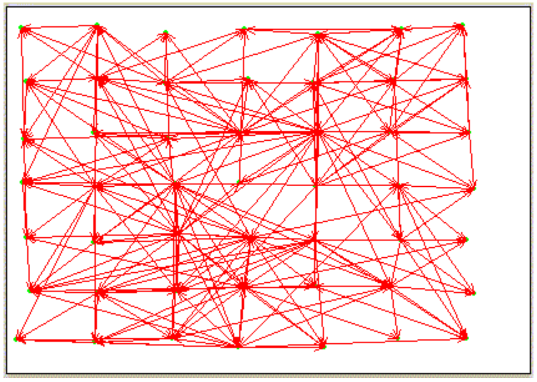


Figure 6: Instance of the radio connectivity in the 7x7 mesh

Let the source be at the upper right corner and the destination be at the bottom left corner. Let the source rate be 1 packet per second. Assume that the destination is known at initialization time, and flooding from the destination (Figure 3) is used for initialization. Figure 7 shows the success rate using the retransmission strategy (Figure 5). The success rate increases with the number of retransmissions. (Note that this may not be true, due to collisions, when the source rate is high.) Figure 8 shows the latency using the forward propagation strategy (Figure 4). In general, the smaller δ, the more forward propagation happens, and the shorter the delay. However, forward propagation introduces extra load and may reduce success rates due to collisions.

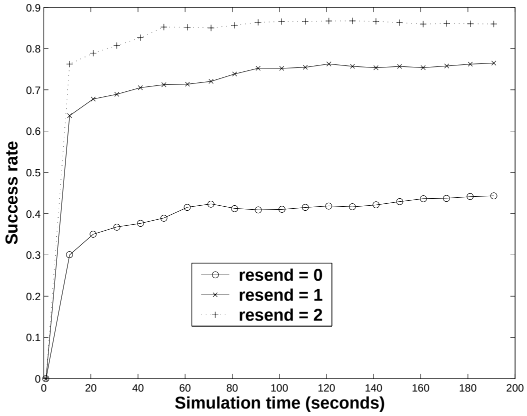


Figure 7: Comparisons of success rate with resend = 0, 1, 2

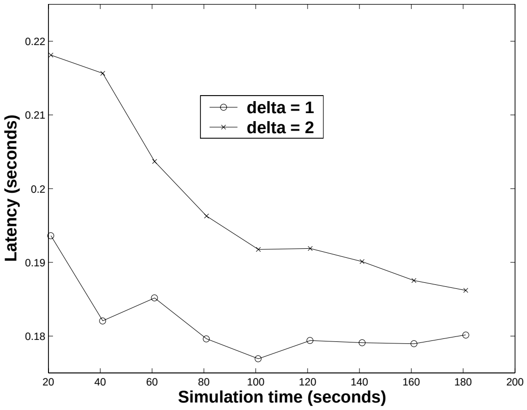


Figure 8: Comparisons of latency with forward propagation $\delta = 2, 1$

Effects of Initial Heuristics

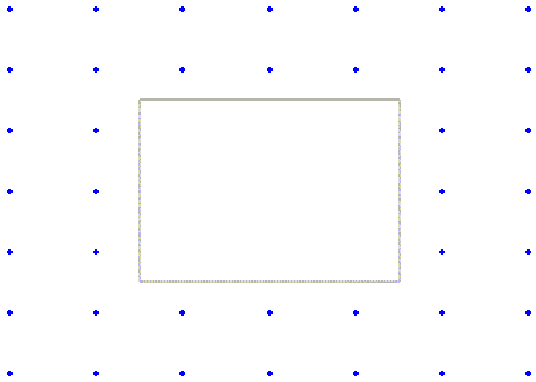


Figure 9: Example of a sensor network with a hole

Consider a 7x7 mesh with a hole of size 3x3 in the middle (Figure 9). Again let the source be at the upper right corner and the destination be at the bottom left corner. Let the source rate be 1 packet per second. Three initialization strategies are compared: 1) no hint; 2) geographical destination specification; and 3) known destination (flooding from destination). Without a hint, packets initially search for the destination while building up the potential field with Q-values. Figure 10 shows the Q-value field after sending out 10 packets without initial heuristics, i.e., $Q^0 = 0$. If the geographical location of the destination is known, the initial Q-values at each node is given by $\max(|x - x_d|/R, |y - y_d|/R)$ where R is 3 for this case. Figure 11 displays the success rates of these three initializations. Without a hint about the destination, the success rate is initially low, but after several packets the field is built up and the success rate increases. Figure 12 displays the latency using these three initializations. Without a hint, the paths tend to be long, resulting in high latency. Geographical routing performs quite well due

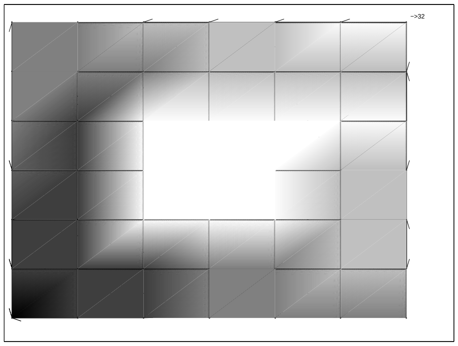


Figure 10: Potential field after 10 packets when no hint initially

to good heuristics at the beginning.

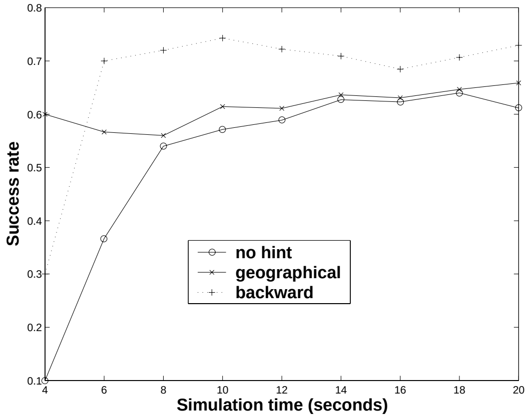


Figure 11: Comparisons of success rate with different initial heuristics

Energy-aware Strategies

Figure 13 shows the life-time metric of an energy-aware strategy where $o = ku + 1$. It shows that with the energy-aware objective in routing, the life-time is better (longer) than that of the shortest path strategy ($o = 1$).

Pursuer Evader Game

Finally, we use a real application to test the performance of the search-based routing algorithms. The application, Pursuer Evader Game (PEG) (Sastry *et al.* 2003), is to use the sensor network to detect an evader and to inform the pursuer about its location. The communication problem in this task is to route packets sent out by one of the sensor nodes (who detects the evader) to the mobile pursuer. The source is changing from node to node, following the movement of the evader, and the destination is mobile. Assume that the evader and the pursuer move at about the same speed. For the average speed s, we set the source rate to $s/0.2$. Let the speed be 0.1, 0.2, and 0.3. Figures 14 and 15 show the success rate and latency with respect to these speeds. The figures show that there is only minor performance degradation due to the dynamic and mobile aspects of the network. When the speed is higher, there is a slightly higher latency in packet delivery.

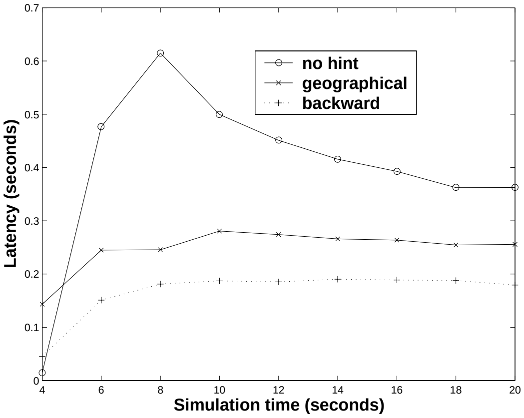


Figure 12: Comparisons of latency with different initial heuristics

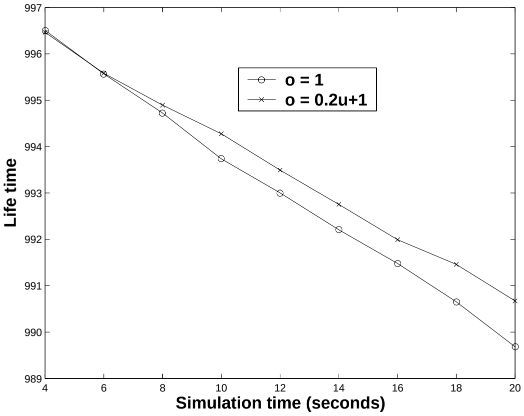


Figure 13: Life-time metrics: energy-aware routing strategy vs. shortest path

Conclusions

In this paper, we have studied search-based routing strategies for sensor networks. By exploiting the broadcast channel of wireless sensor networks, reinforcement learning can be done in a cost effective way. With implicit confirmation, these strategies work for asymmetric and dynamic networks as well. Theoretic analysis of search-based strategies has also been done, and performances of various strategies have been evaluated in a simulation environment.

Other routing strategies have also been developed (Zhang & Fromherz 2004). Comparisons with other routing strategies will be discussed in another paper. The search-based routing strategies have also been implemented on Berkeley motes. The details of the implementation are outside the scope of this paper.

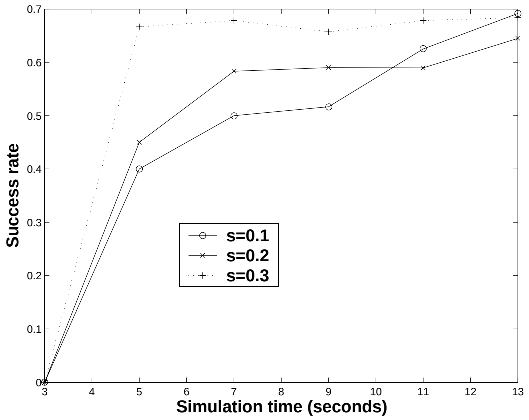


Figure 14: Comparisons of success rate with different speeds

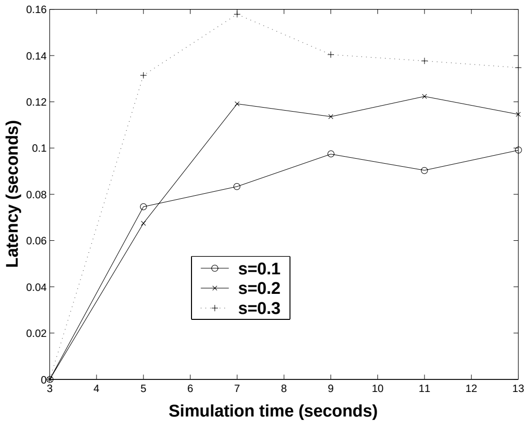


Figure 15: Comparisons of latency with different speeds

Acknowledgments

This work is funded in part by Defense Advanced Research Project Agency contract # F33615-01-C-1904.

References

Berkeley, U. 2003. Berkeley wireless embedded systems. http://webs.cs.berkeley.edu/.

Boyan, J. A., and Littman, M. L. 1994. Packet routing in dynamically changing networks: A reinforcement learning approach. In J. D. Crowan, G. Tesauro, J. A., ed., *Advances in Neural Information Processing Systems*, volume 6. Morgan Kaufmann Publishers, Inc. 671–678.

Cerpa, A., and Estrin, D. 2002. Ascent: Adaptive self-configuring sensor networks topologies. In *Proc. IEEE InfoComm*.

Chen, B.; Jamieson, K.; Balakrishnan, H.; and Morris, R. 2002. Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. *ACM Wireless Networks Journal* 8(5).

Chu, M.; Haussecker, H.; and Zhao, F. 2002. Scalable information-driven sensor querying and routing for ad hoc heterogeneous sensor networks. *Int. Journal on High Performance Computing Applications*.

Estrin, D.; Govindan, R.; Heidemann, J.; and Kumar, S. 1999. Next century challenges: Scale coordinations in sensor networks. In *Proc. ACM MobiComm*, 263–270.

Kaelbling, L.; Littman, M.; and Moore, A. 1996. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research* (4):237 – 285.

Kahn, J. M.; Katz, R. H.; and Pister, K. S. J. 1999. Next century challenges: Mobile networking for smart dust. In *Proc. ACM MobiComm*.

Karp, B., and Kung, H. T. 2000. GPSR: Greedy perimeter stateless routing for wireless networks. In *Proc. 6th Int'l Conf. on Mobile Computing and Networks (ACM Mobicom)*.

Koenig, S., and Simmons, R. G. 1993. Complexity analysis of real-time reinforcement learning applied to finding shortest path in deterministic domains. In *National Conference on Artificial Intelligence*, 99–105.

Koenig, S. 2001. Agent-centered search. *AI Magazine* 22(4):109–131.

Kumar, S., and Miikkulainen, R. 1998. Confidence-based q-routing: An on-line adaptive network routing algorithm. In *Proceedings of Artificial Neural Networks in Engineering*.

Royer, E., and Toh, C. 1999. A review of current routing protocols for ad hoc mobile wireless networks. *IEEE Personal Communications*.

Sastry, S.; S. Schaffert, L. S.; Sharp, C.; and Sinopoli, B. 2003. Nest challenge problem: Midterm, final demo paln. DARPA NEST Program Demonstration Plan, http://dtsn.darpa.mil/ixo/nest/day2/UCB1Nest_PI_07_10.ppt.

Simon, G. 2003. Probabilistic wireless network simulator. http://www.isis.vanderbilt.edu/projects/nest/prowler/.

Sutton, R. S., and Barto, A. G., eds. 1998. *Reinforcement Learning: An Introduction*. Cambridge, MA: The MIT Press.

Xu, Y.; Heidemann, J.; and Estrin, D. 2001. Geography-informed energy conservation for ad hoc routing. In *Proc. ACM MobiComm*.

Yokoo, M., and Ishida, T. 1999. Search algorithms for agents. In Weiss, G., ed., *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. Cambridge, MA: The MIT Press. chapter 4, 165–199.

Yu, Y.; Govindan, R.; and Estrin, D. 2001. Geographical and energy aware routing: a recursive data dissemination protocol for wireless sensor networks. Technical report ucla/csd-tr-01-0023, UCLA Computer Science Department.

Zhang, Y., and Fromherz, M. 2004. Message-initiated constraint-based routing for wireless ad-hoc sensor networks. In *Proc. IEEE Consumer Communication and Networking Conference*.