Path Update Algorithms for Mobile Nodes in a Wireless Ad Hoc Sensor Network

Ritesh Madan & Dileep George
CS428 – Project Presentation
June 05, 2003
Assumptions

- A path exists between the two mobile agents
- Connectivity is preserved
- Position of the mobile agents do not change during a search. (i.e, they move slow enough)
- Mobile Agents do not act as routers
- Local-handoff is sufficient to maintain connectivity.
Scenario-Example
Objective:
To minimize total energy usage in searching for a path and communication over a path.

- Low data rate - can use ad hoc routing.
- High data rate – need energy efficient paths.

(Data rates - low and high w.r.t the avg velocity of the agents)
Local One-Hop Optimization

- Maintain Connectivity – Handoffs
- Local Optimization – “String Stretching”
Path Maintenance
Local Optimizations

Snapshot 3

Snapshot 4
É - Compactness

\[ \delta = \min_{u,v \in V} \frac{d(u, v)}{\hat{d}(u, v)} \]

\( d(u; v) \) is the distance between nodes \( u \) and \( v \)

\( \hat{d}(u; v) \) is the network distance

If a path of length \( l \) is known between \( u \) and \( v \), then the optimum path lies in the ellipse

\[ E = \{ x \in \mathbb{R}^2 \mid d(u, x) + d(x, v) \leq \min(l, \frac{1}{\delta} d(u, v)) \} \]
Path sub-optimality

Optimum Path

Path Extension

M1
x1

M2
y1

M1
x2

M2
y1

n
Path sub-optimality measures

\[ \hat{\imath} = \frac{|\bar{a}|_{\text{ext}}}{|+|_{\text{ext}}} \]

measures change in topology since last search

\[ \delta_{\text{ext}} = \min_{u,v \in P_{\text{ext}}} \frac{d(u,v)}{\hat{d}(u,v)} \]

measures compactness of the path
\[ E = \{ x \in \mathbb{R}^2 \mid d(u, x) + d(x, v) \leq \min(l, \frac{1}{\delta}d(u, v)) \} \]

\[ l_{\text{search}} = \min(l, \frac{1}{\delta}d(u, v)) \]

for \( 0 < \alpha \leq 1 \), let

\[ E_\alpha = \{ x \in \mathbb{R}^2 \mid d(u, x) + d(x, v) \leq \alpha l_{\text{search}} \} \]

\( \mathbf{E} \) close to 1 – high energy + high chance of finding better path
Choice of $e$

Search over an ellipse such that search energy (estimate) satisfies

$$\mathcal{E}_{\text{search}}(\alpha) \leq R\Delta t(\mathcal{E}(l) - \mathcal{E}(\alpha l_{\text{search}}))$$

Where ....

- $R$ = rate of data communication
- $\Delta t$ = time for which topology does not change
- $\mathcal{E}(l)$ = energy for communicating one bit over a path of length $l$ (need to have an estimate)
Proposed Algorithm

At each time step $t$

- If an agent becomes disconnected, do local handoff to maintain connectivity
- Do local optimization using one hop information
- If $\gamma$ is less than 0.90 (10% change since last path search), look for nodes on the path extension which give minimum value of compactness value.
  - For this pair of nodes, choose $\alpha$ by the criterion on previous slid
  - Update path between this pair
  - update path between the agents
Simulation Study

- Objectives
  - Gain intuition
  - Compare the performance of various schemes
  - Derive a set of design parameters
Simulation Setups

- Setup A: uniform node density in a 10x10 field
- Setup B: uniform node density except for two holes.
Setup B
Experiment 1

- Does the average length of path maintained by an algorithm vary with node density?
Average Path Length

Using Global Search

Using Local Updates
Experiment 2

Objective

- Compare the performance of different path update algorithms for a given scenario
Algorithms

When to Search?

Algorithm 1: Local Updates only
Algorithm 2: Uses only path compactness heuristic
Algorithm 3: Uses only % change in path length
Algorithm 4: Uses 2 and 3.
## Results (Setup A)

Setup A. Agents moved around randomly for 10000 hops.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Average Length of Path</th>
<th>Number of Searches</th>
<th>Number of Nodes Searched</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algo 1</td>
<td>16.95</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Algo 2</td>
<td>8.4135</td>
<td>109</td>
<td>21238</td>
</tr>
<tr>
<td>Algo 3</td>
<td>7.9867</td>
<td>365</td>
<td>36181</td>
</tr>
<tr>
<td>Algo 4</td>
<td>8.4105</td>
<td>104</td>
<td>29909</td>
</tr>
</tbody>
</table>
## Results (Setup B)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Average Length of Path</th>
<th>Number of Searches</th>
<th>Number of Nodes Searched</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algo 1</td>
<td>14.33</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Algo 2</td>
<td>8.77</td>
<td>106</td>
<td>19454</td>
</tr>
<tr>
<td>Algo 3</td>
<td>8.52</td>
<td>291</td>
<td>28433</td>
</tr>
<tr>
<td>Algo 4</td>
<td>8.882</td>
<td>89</td>
<td>16826</td>
</tr>
</tbody>
</table>

Setup B. Agents moved around randomly for 10000 hops
Sample Energy Calculation

- $E = C(L) \times T + C(S)$
  - $C(L)$ - Average cost of transmitting on path of length $L$
  - $T$ - Average time spent at a path
  - $C(S)$ – Average Cost of search
Energy Budget

Calculations done for simulation results on setup B
In Conclusion…

- We proposed and studied different algorithms for path maintenance and update.
- Performance of the algorithms depend on the scenario at hand.
- More simulations studies are required to characterize different environments and to derive design parameters.