Sensor Relocation with Mobile Sensors: Design, Implementation, and Evaluation

Jie Teng, Tim Bolbrock, Guohong Cao, and Tom La Porta
Pennsylvania State University

Ad-Hoc networks Seminar 2008-2009

Presented by
Mustafa Sofean

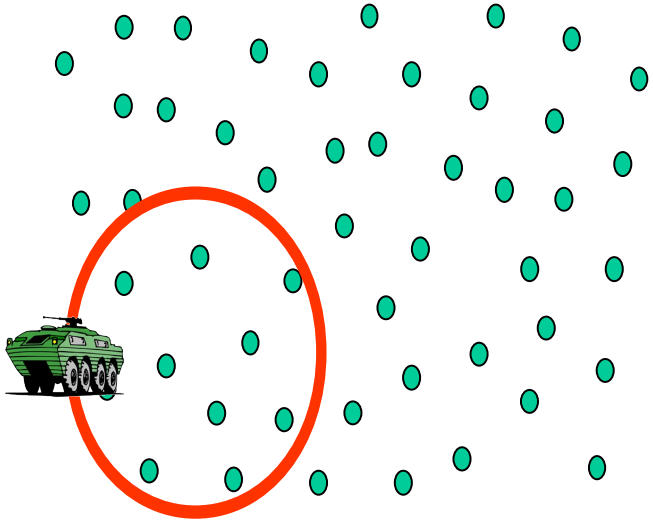
Freiburg University
February 16th, 2009

Outline

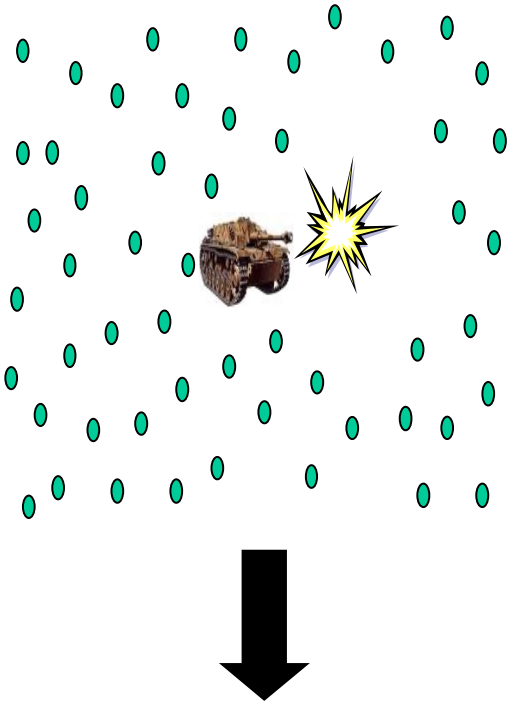
- Sensor Relocation Problem
- Sensor Relocation Application
- Cascaded Movement
- Distributed Cascaded Sensor Relocation Algorithm
- Prototype and Implementation
- Experimental Evaluations
- Conclusion

Introduction

New event



Sensor failure



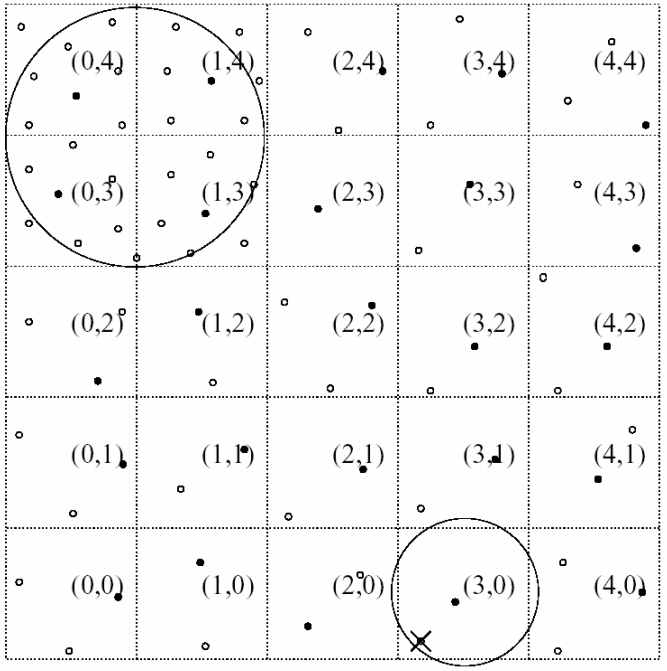
Sensor Relocation

APPLICATION REQUIREMENTS

- Time response.
- Energy efficiency.
- Dynamic reconfiguration.

Sensor Relocation

- Two phases to solve sensor relocation:-
  - Finding the redundant sensors and
  - Relocating them to the target location.



Sensor Relocation

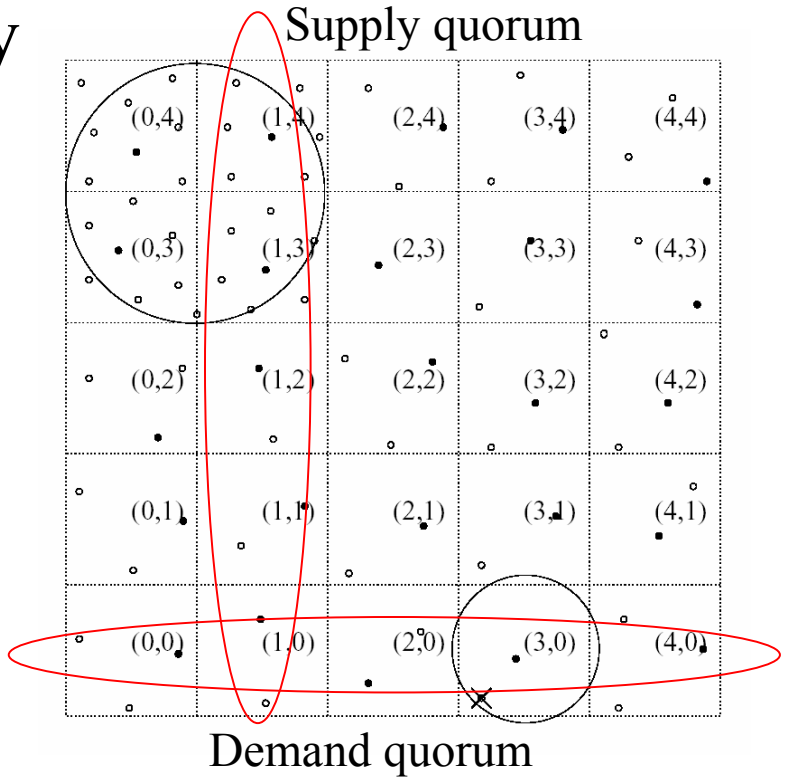
- Redundant Sensor Discovery

- **Grid-Quorum system**

  - **demand quorum**:
    grids in a column
  - **supply quorum:**
    grids in a row

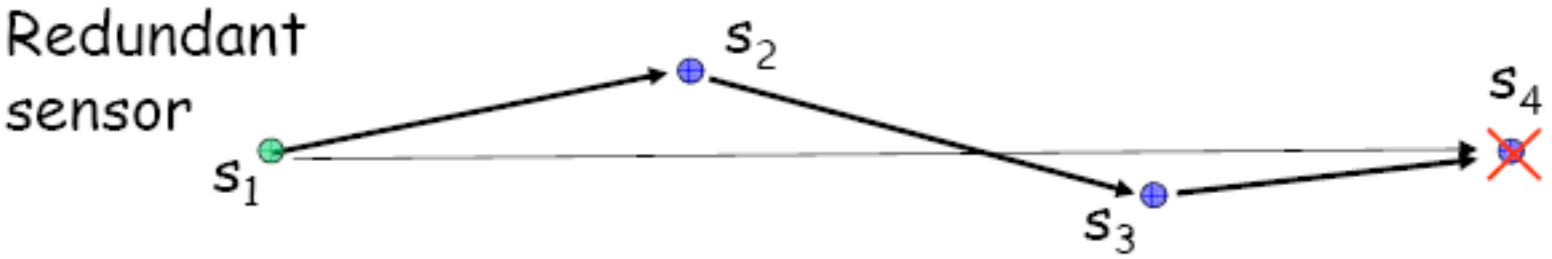
  - need sensor: search demand quorum

  - have redundant sensor: notify supply quorum



Sensor Relocation

- **Direct moving**
  - Long delay.
  - Too much energy .



Use cascaded movement

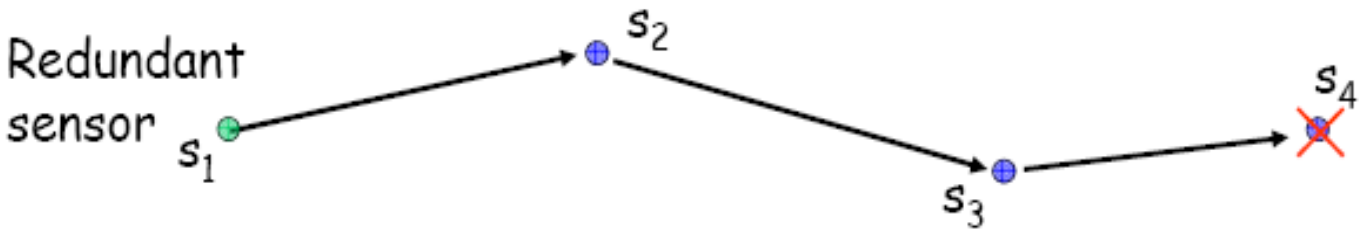
How to choose the cascading nodes?

- Relocation Delay?
- Energy balance?

Cascaded Movement

- Satisfy the requirement of relocation delay.

$$D(i,j)/V_i - (t_j - t_i) \le T_j$$

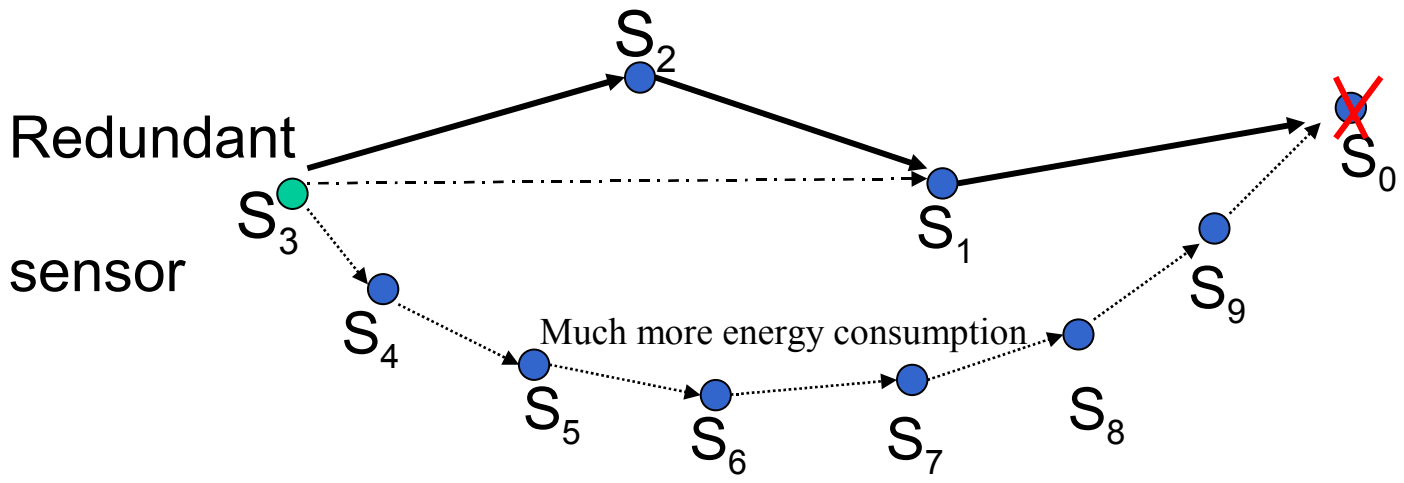


- $D(s_3,s_4) \le V_3 * T_4$
- $D(s_2,s_3) \le V_2 * (T_3 + t_3)$
- $D(s_i,s_j) \le V_i * (T_j + t_j)$

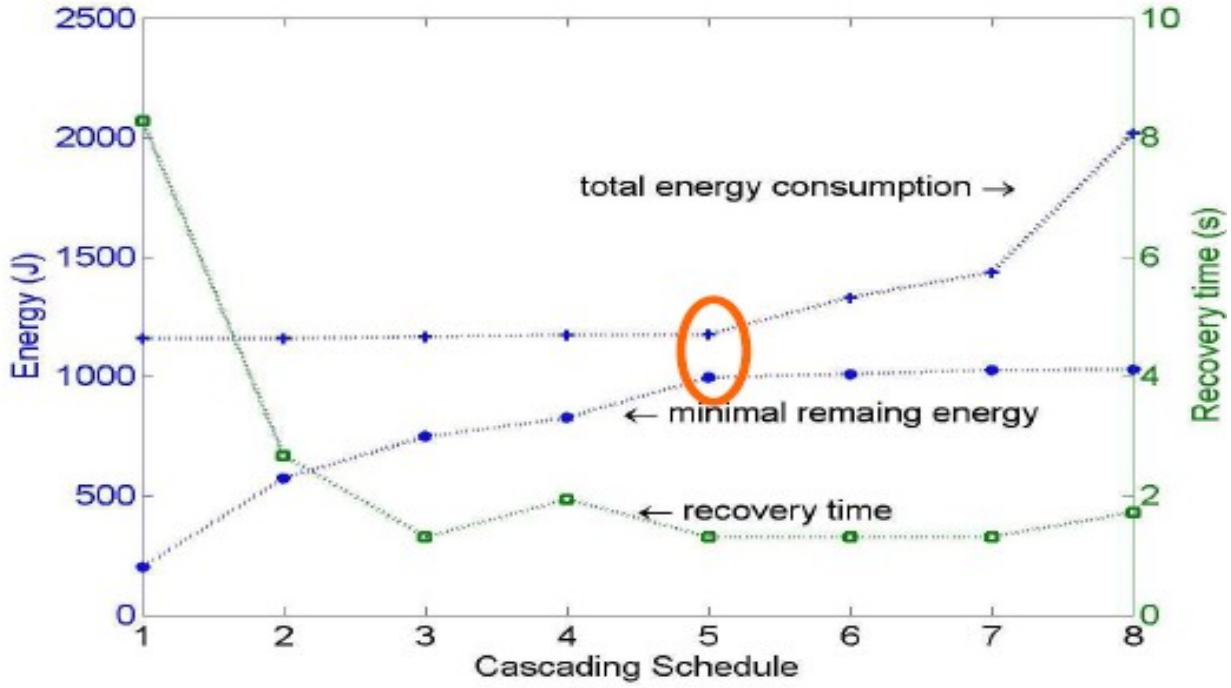
- T_i : the recovery delay constraint of S_i
- t_i : The departure time of S_i's movement
- D(i,j): distance between s_i and s_j
- V_i : the moving speed of S_i

Cascaded Movement

- Minimize total energy consumption $\mathbf{E}_{total}$?
- Maximize minimum remaining energy $\mathbf{E}_{min}$?



Cascaded Movement



$$\mathbf{Min(E_{total} - E_{min})}$$

Cascaded Movement

ModifiedDijkstra(Graph G(V,E), Vertex s_0)

Initialization: $S = \{s_0\}$, $Q = V$
DeleteEdge(s_0,0)
while not Empty (Q)

1. Let $\mathcal{F} = \{\langle s_k, s_l\rangle \mid \langle s_k, s_l\rangle \in S \times Q, d_{lk} \leq (T_k + t_k) * speed\}$
2. Find $\langle s_i, s_j\rangle \in \mathcal{F}$ such that $\forall \langle s_k, s_l\rangle \in \mathcal{F}, d_{ji} \leq d_{lk}$
3. s_j.predecessor = s_i
4. $t_j = T_i + t_i - d_{ji}/speed$
5. $P'_j = P_j - d_{ji}$
6. Add s_j to S
7. DeletedEdge(s_j, t_j)

end

Cascaded Movement

- Algorithm to calculate the best cascading schedule

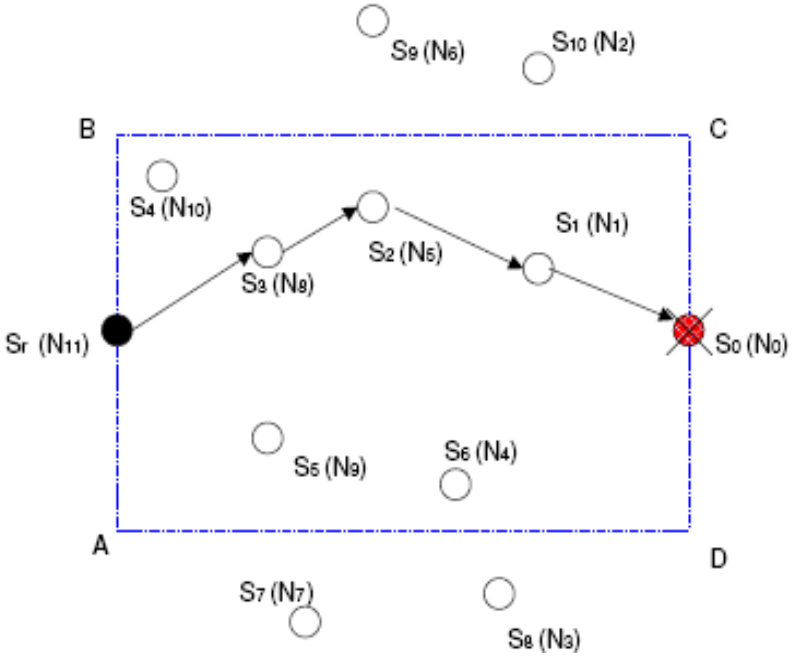
Initialization: $E = 0, Emin = -2, E' = 0, Emin' = -1$
while (1)
1. find the shortest cascading schedule using the Modified Dijkstra's algorithm
2. record the minimum remaining power as $Emin'$
3. delete all edges $s_i s_j$ if $P_i - d_{ij} \leq Emin'$
4 **if** $E' - Emin' < E - Emin$ **then**
$E = E', Emin = Emin'$
else
return the previously calculated schedule

Distributed Cascaded Sensor Relocation Algorithm

- The grid head‘s S_0 broadcasts

  $(T_0, t_0, S_r, E_0, E_{min0})$

  - S_2 can take S_1‘s place within T_1
  - S_2‘s remaining energy after moving is no larger than the minimum remaining energy in the last schedule.



After that: $E_2 = d_{21} + E_1$.

$E_{min2} = min(P_2 - d_{21}, E_{min1})$

S_2 *Remember its predecessor* s_1

B*roadcasts* $(T_2, t_2, S_r, E_2, E_{min2})$

Distributed Cascaded Sensor Relocation Algorithm

- **How the node make correct decision?**
  - ***If*** $E_{current} < E_{previous(received\ message)}$ ***Then***
    it reportcasts the update version

    High message overhead?

  - Wait for a period of time *T'*

    If *T' is low* ***then***

    not enough information is received

    If *T' is high* ***then***

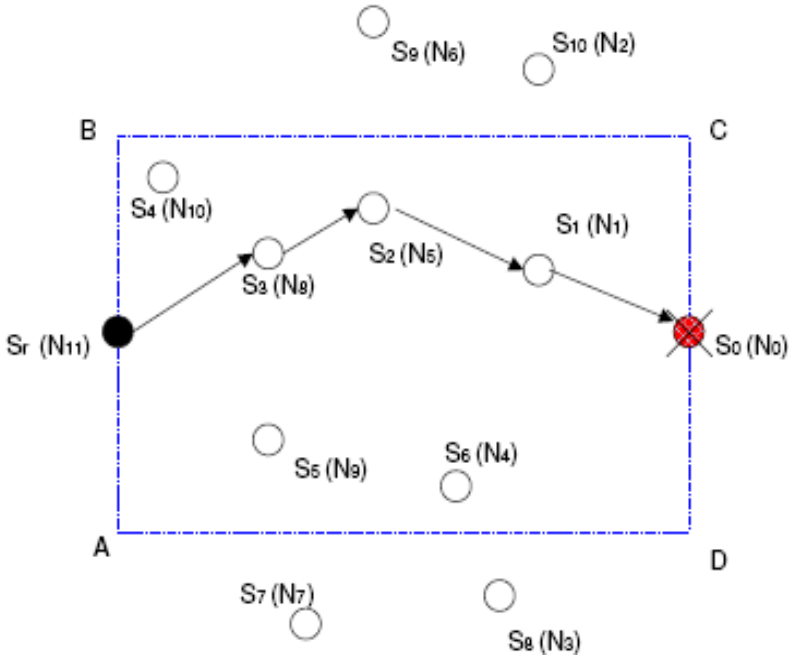
    The delay may be increased

    Hard to decide the time threshold?

Distributed Cascaded Sensor Relocation Algorithm

Primary search area and Waiting list:

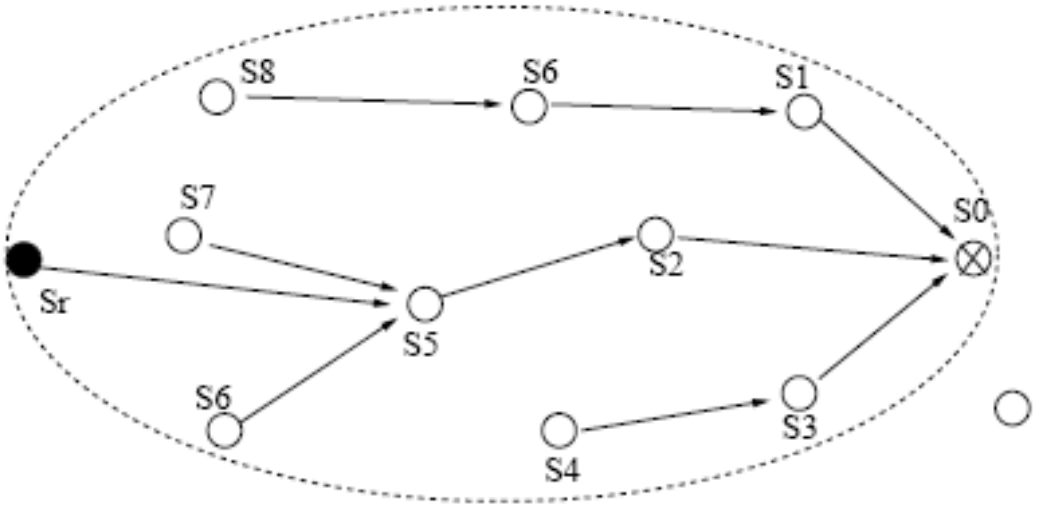
- Bsed on the location of S_0 and S_r.
- Can be in any shape



- **waiting list**
  - The neighbors of the node.

Distributed Cascaded Sensor Relocation Algorithm

- **Message Piggybacking and Processing**
  - The successor of a node may not be its communication neighbor.
  - Each node uses ***a waiting list message queue.***
  - The message includes: S_{org},T_{org},t_{org},E_{org} and E_{min}



- *2 byte for every field =10Bytes*
- **Mica2 has 4KB data memory.** ***Not feasible to cache all the Msg's?***
- **processin the messages in batch .**

Prototype and Implementation

- **The prototype of mobile sensors network**

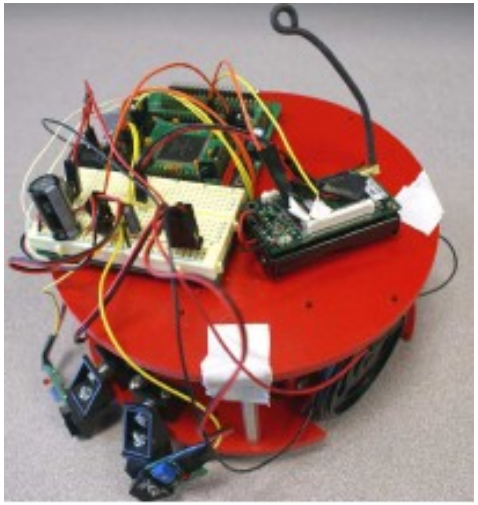
The Hardware

- **Mica2 mote**
  - The third generation mote built for WSN.
  - Microprocessor 4MHZ.
  - 4 KB RAM, 128 KB code space.
  - 868/916 MHz RFM radio.
  - 512KB flash EEPROM.
  - Radio range is 500 ft.

- All computation related to applications is done by the motes.

The Hardware

- **Mobile node**



Consist of **Mica2** and

Robot platform

- Two 6 plastic bases.
- The lower base consists of the motor, odometry encoders, wheels, and batteries.
- On the upper base the Mica2.
- Microcontroller

(16-bit CPU,256KB flash EEPROM,4 KB EEPROM and 12 KB RAM)

The Software

- **The robot program**
  - sending and receiving messeges.
  - Handling the navigation queue.
  - Calling for odometery updates.
- **The robot control program**
  - **TinyOS component.**

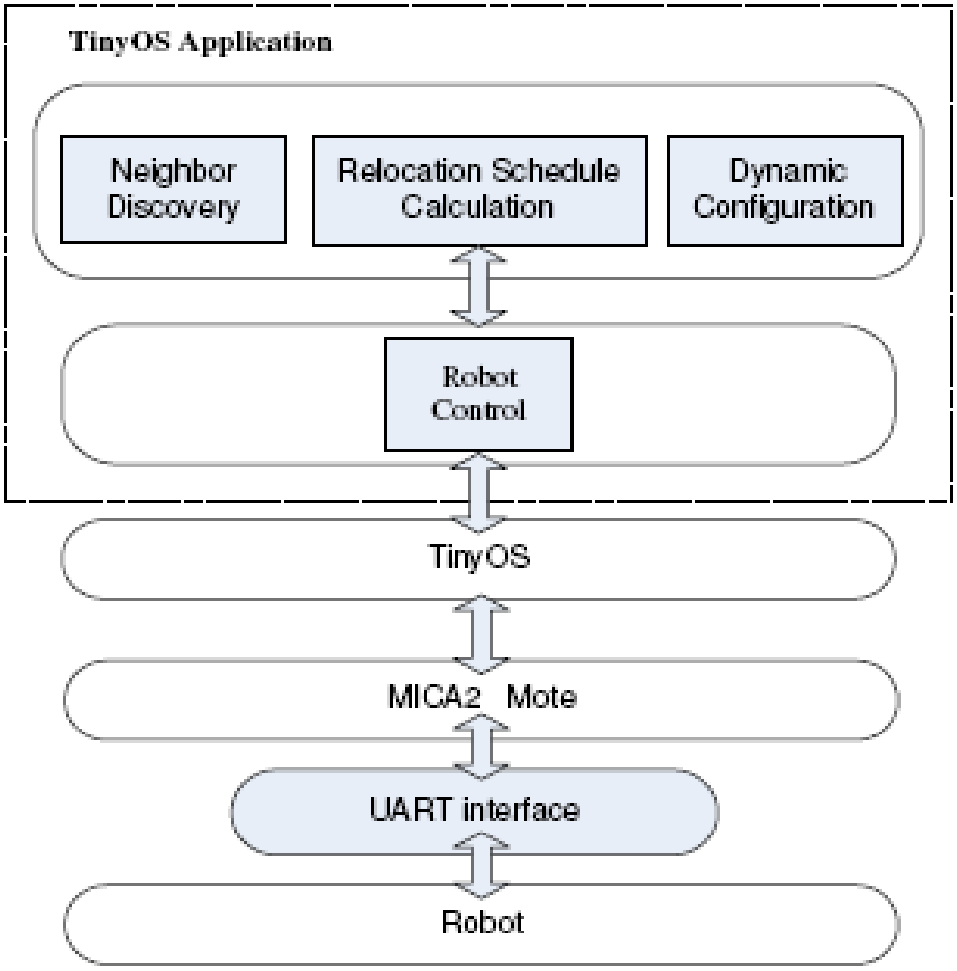
The Software

- **The sensor relocation algorithm**
  - **Neighbor Discovery**

    A node sends its ID, its location and neighbor list**.**
  - **Network Reconfiguration.**

    eg. Change the parameters of recovery delay.
- **Program on the base station**
  - Laptop connected with a mote for debugging/visualizing

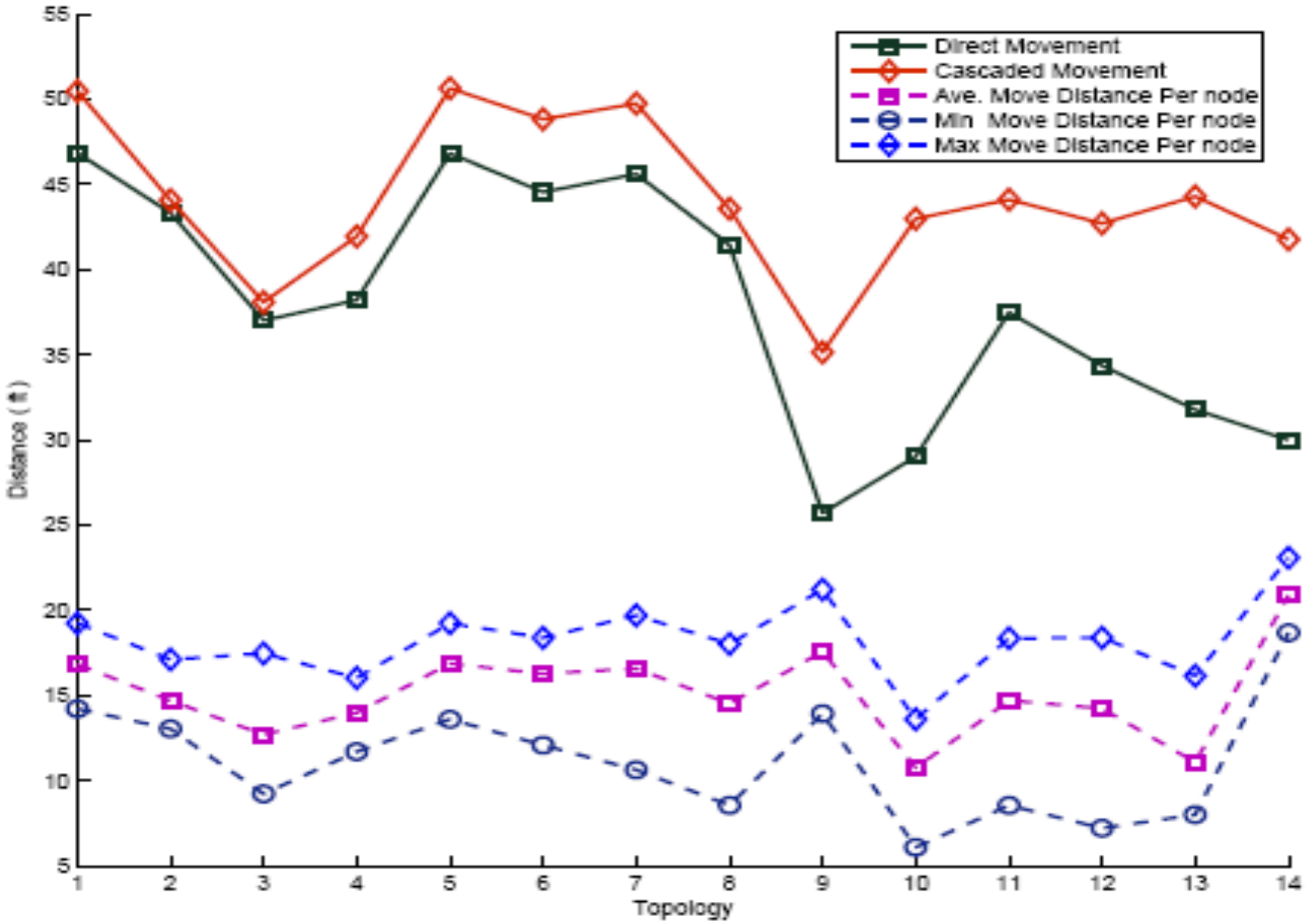
Mobile Sensor Relocation Application Architecture



Advantages of the Implementation

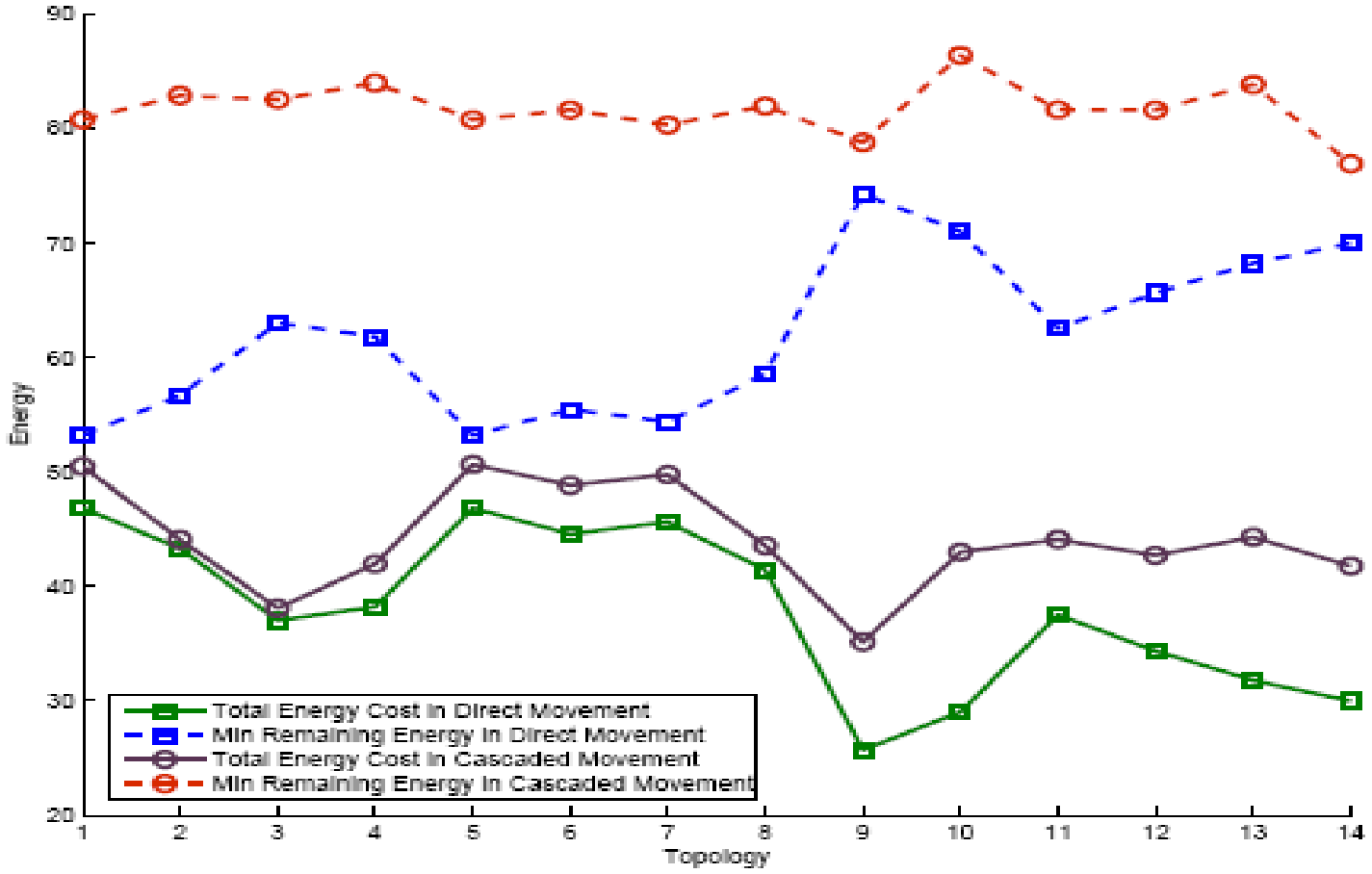
- Energy Efficiency.
- Simplicity.
- Flexibility.
- Contention Reduction.

Experimental Evaluations



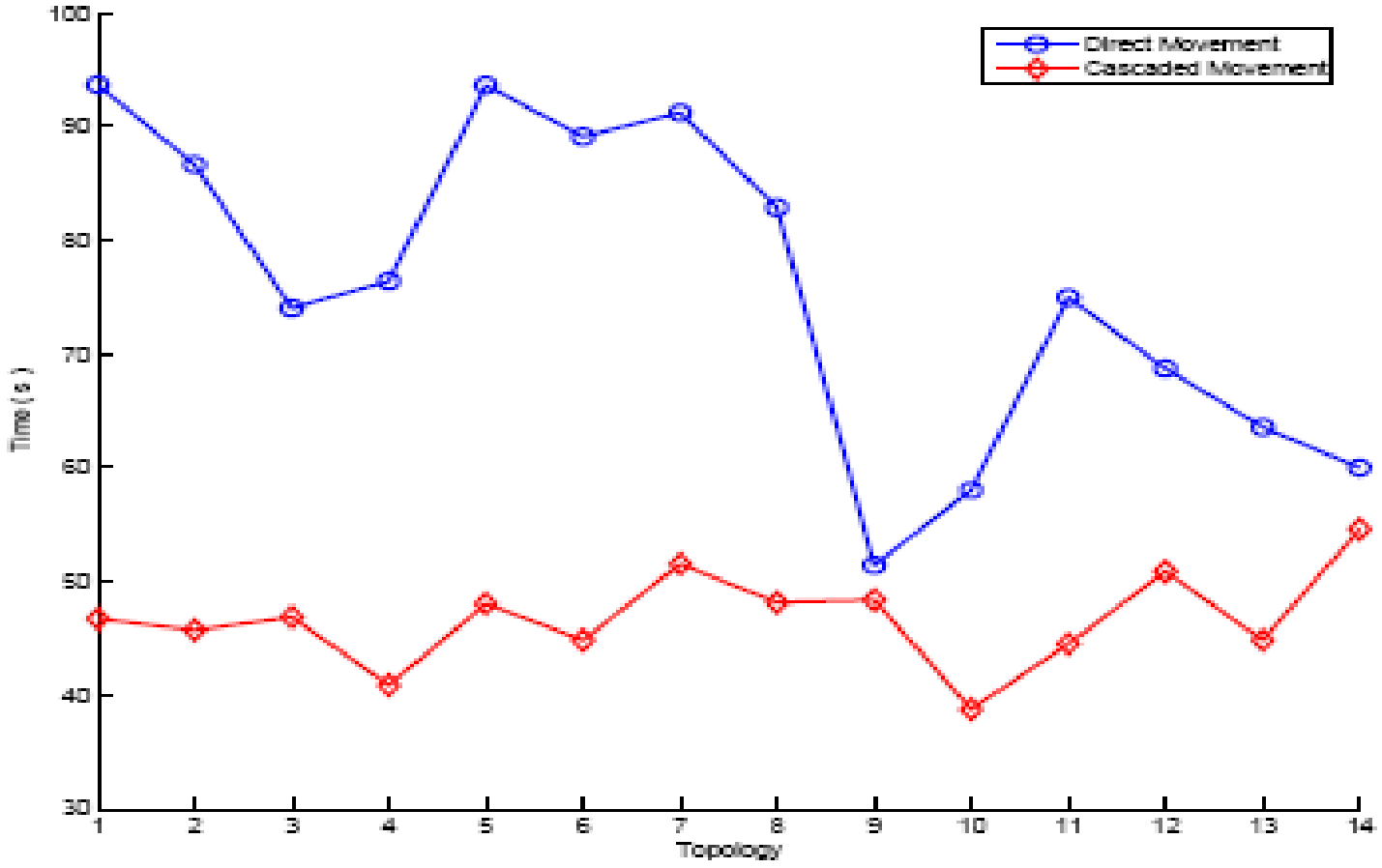
Comparison of the ***moving distance*** in Cascaded Movement and Direct Movement

Experimental Evaluations



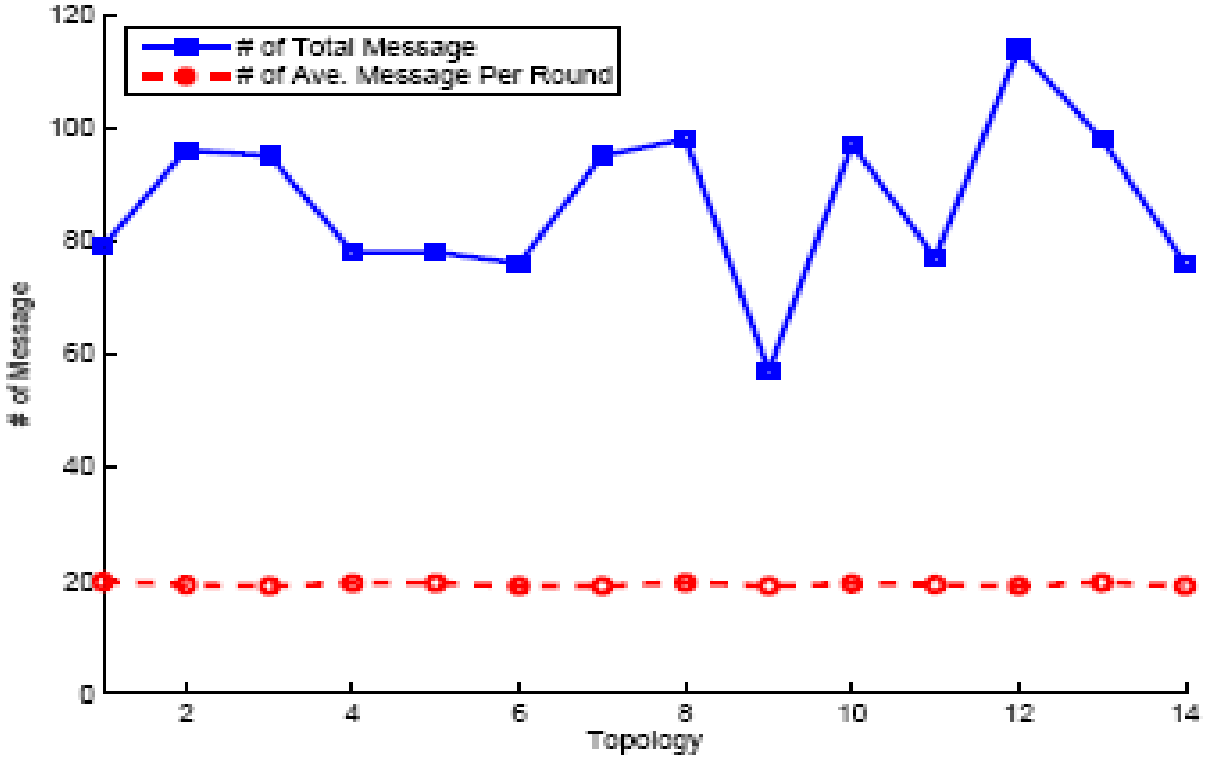
Comparison of ***energy cost*** in Cascaded Movement and Direct Movement

Experimental Evaluations



Comparison of ***total time cost*** in cascaded movement and direct movement

Experimental Evaluations



- The ***message complexity*** in cascaded movement

Conclusion

- We presented the sensor relocation problem in mobile sensor and the soloution:-
  - Redundant Sensor Discovery .
  - Cascaded movement.
- Distributed sensor relocation algorithm.
- Software and hardware design to implement the relocation application.
- Finally we present the results of experimental evaluations.

Reference

- Guiling Wang, Guohong Cao, Tom La Porta, and Wensheng Zhang “Sensor Relocation in Mobile Sensor Networks”

Thank You

Questions

?