Assigning Sensors to Competing Missions

H. Rowaihy[1], M. Johnson[2], T. Brown[2], A. Bar-Noy[2] and T. La Porta[1]

Problem and Model

- A sensor network may be tasked with multiple simultaneous missions, each requiring multiple sensors
- Missions can compete for sensing resources,
- We need scheme to *decide which sensors should be assigned to which missions?*

Model:

- Missions have *demands* and *profits*
- Sensors have *utilities* to the different missions – contribute to demand
- Utility is a measure of quality of information
- Mission considered to be successful if a threshold of its demand is met

Two settings:

1. **Static:** all missions are known beforehand – solve the problem once
2. **Dynamic:** missions arrive over time – resolve the problem with each mission arrival or departure

Goal: Maximize the profit achieved from successful missions

Centralized Schemes

- A base-station collects information about all sensors in the field and decides on the assignments. (Rerun for each event in dynamic environments)
- Two schemes:

1. **Greedy:**
   - Order missions according to profit.
   - Start from highest profit and try to satisfy missions in order.
   - If a mission cannot reach success threshold do not assign any sensors
   - Fast but not optimal (results show that this comes within 1.2% of optimal)
2. **Polynomial Time Approximation Scheme (PTAS):**
   - Utilize the fact that sensors and missions exist on a geometric field.
   - Only a subset of sensors apply to a mission (depending on the sensing range)
   - Divide the filed into smaller slices and solve each independently.
   - Time consuming and may require brute-force in the smaller slices

Distributed Schemes

- Centralized solutions can provide high quality solutions because of their global view **but** they have high communication cost
- This especially true in a dynamic environment (base station continually gather data about sensors in the field)
- *Distributed* schemes use local information about sensors in a neighbourhood → lower communication cost
- Each mission has a *mission leader*
- Mission leaders simultaneously perform a local assignments of sensors

- Two schemes:

1. **Dynamic Proposal**
   - Leader send advertisement message
   - Nearby sensors that are not assigned propose to mission with utility
   - Currently assigned sensors check the effective profit and may decide to propose
   - One level reassignment is allowed (a mission with higher effective profit can preempt a mission with lower profit)
   - A mission preempted by a new one will not preempt on other missions (this limits interruption of ongoing missions and communication overhead)
   - Sensors chosen greedily based on utility
2. **Energy-aware Dynamic Proposal**
   - *Dynamic proposal* selects sensors based only on utility and not energy
   - Greedy might not be the best if network lifetime is important
   - Instead leader choose sensors based on utility and remaining energy

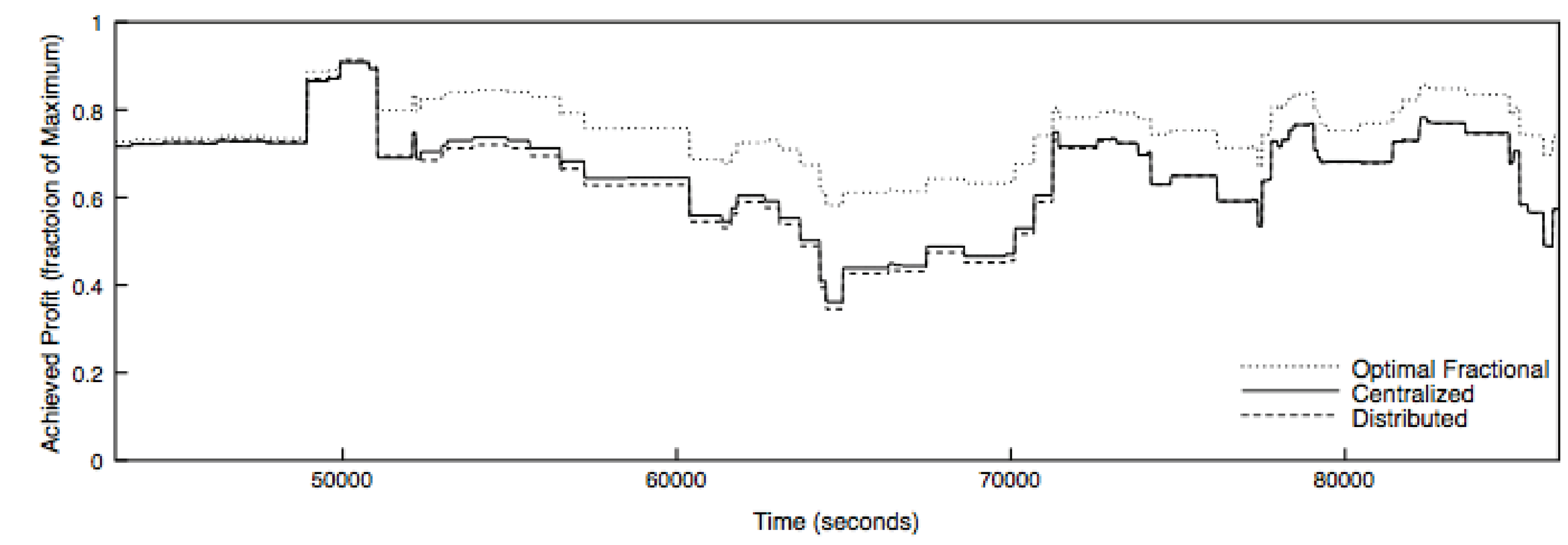
$$f(U,E) = U \times E^{\beta}$$

   - Even out usage of sensors to extend lifetime
   - Periodically check if nodes have depleted energy and if reassignment is needed

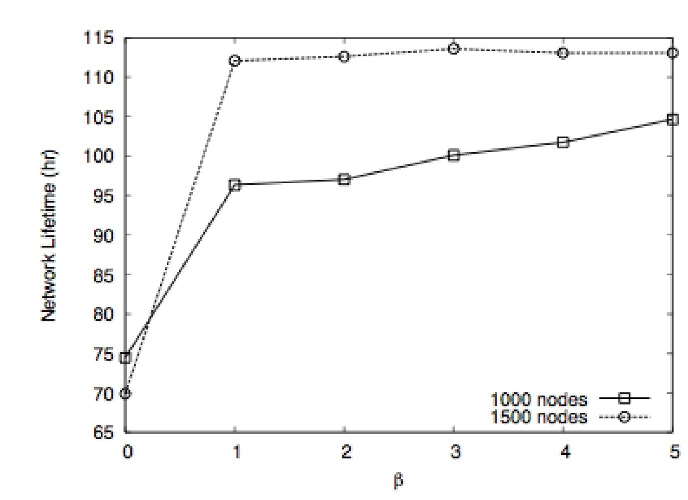
Simulation Results

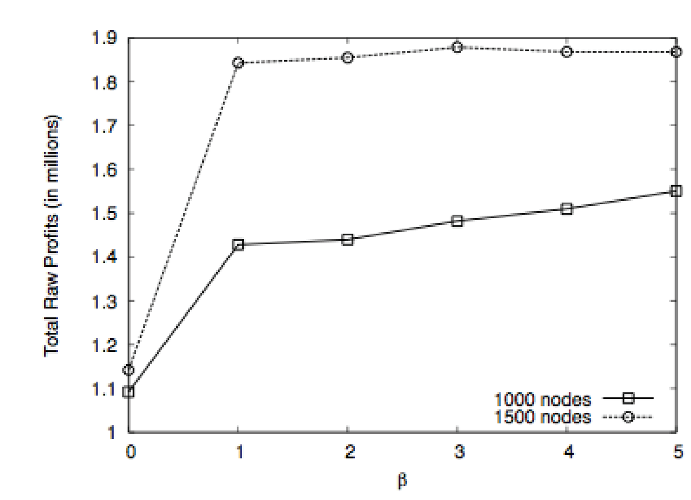
Setup:

- Field size = 400m x 400m
- Sensing Range = 30m
- Avg. mission demand = 2, max = 6 (exponential)
- Avg. mission profit = 1 (exponential)

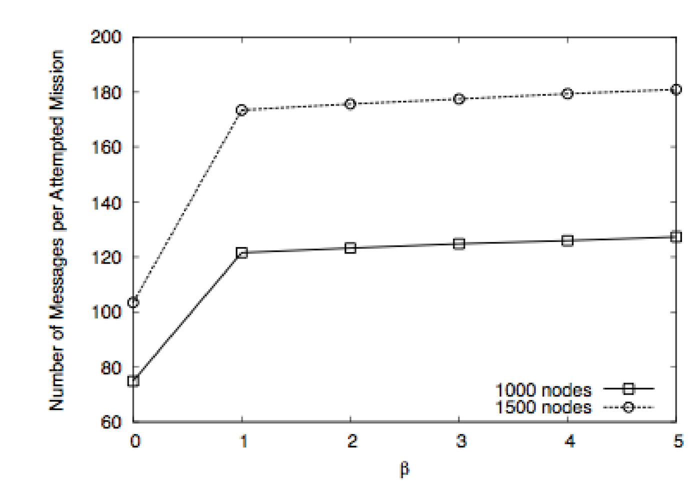


- Number of Nodes = 500
- Avg. mission lifetime = 3 hours (exponential)
- Arrival rate = 3 missions/hour (Poisson)
- Results for a period of 12 hours (12 – 24)
- **Distributed closely matches centralized**





- Initial energy = 10 active hours
- Nodes = 1000 and 1500
- **Lifetime increases by 40% (1000) and 70% (1500)**
- Communication overhead increases due to periodic updates



Demo

- Demo shows operation of the dynamic proposal scheme
- Field size is 355m x 185m
- 350 nodes deployed uniformly over the field
- Missions arrive over time with an average of 10 missions/hour
- Success threshold = 50%

- Sensors are shown as circles (gray = unassigned, colored = assigned)
- Missions are shown with squares
  - The color indicates the satisfaction level
    - 100% satisfied
    - 90% - 100% satisfied
    - 80% - 90% satisfied
    - 70% - 80% satisfied
    - 60% - 70% satisfied
    - 50% - 60% satisfied
    - Failed
  - Mission numbers and assigned sensors are shown with the same color

- The simulator shows the current time in seconds and last the event
- The lower left corner shows number of active missions
- Lower right corner shows percentage of achieved profit
  - Percentage can be low if there are unsuccessful missions

[1] Penn State University
[2] City University of New York
[*] This work has been submitted to *Infocom'08*

Supported by US-ARL and UK-MoD under the ITA Program