Deploying Long-Lived and Cost-effective Hybrid Sensor Networks

Wen Hu [a,d], Chun Tung Chou [b], Sanjay Jha [c,d], Nirupama Bulusu [e]

[a] *School of Computer Science and Engineering, The University of NSW, Sydney, NSW 2052, Australia*
{wenh}@cse.unsw.edu.au
Phone: +61-2-8374 5594 Fax: +61-2-8374 5531

[b] *School of Computer Science and Engineering, The University of NSW, Sydney, NSW 2052, Australia*
{ctchou}@cse.unsw.edu.au
Phone: +61-2-9385 7203 Fax: +61-2-9385 5995

[c] *School of Computer Science and Engineering, The University of NSW, Sydney, NSW 2052, Australia*
{sjha}@cse.unsw.edu.au
Phone: +61-2-9385 7383 Fax: +61-2-9385 5533

[d] *Networks and Pervasive Computing Program, National ICT Australia Limited, Bay 15, Locomotive Workshop, Australian Technology Park, Garden Street, Eveleigh NSW 1430, Australia*

[e] *Maseeh College of Engineering and Computer Science, Department of Computer Science, Portland State University, Portland, OR 97207-0751*
{nbulusu}@cs.pdx.edu
Phone: +1-503-725 2404 Fax: +1-503-7253211

Abstract

In this paper, we consider the problem of network deployment in hybrid sensor networks, consisting of both resource-rich and resource-impoverished sensor devices. The resource-rich devices, called micro-servers, are more expensive but have significantly greater bandwidth and energy capabilities compared to the low-cost, low-powered sensors. Such hybrid sensor networks have the potential to support the higher bandwidth communications of broadband sensor networking applications, as well as the fine-grained sensing that is made possible by smaller sensor devices. However, care must be taken to ensure that such systems are cost-effective to the end user. We investigate some fundamental questions for hybrid sensor network deployment — for a given number of micro-servers, what is the maximum lifetime of a sensor network and the optimal micro-server placement? What benefit can additional micro-servers add to the network, and how financially cost-effective is it to introduce these micro-servers? We propose a cost model and integer linear programming (ILP) problem formulation for minimizing energy usage/maximizing lifetime in a hybrid sensor network. Then, we prove that the integer linear optimization problem is NP-hard and introduce an efficient approximation algorithm using tabu-search technique. Our studies show that network lifetime can be increased dramatically with the addition of extra micro-servers; the locations of micro-servers can affect the lifetime of network significantly. Moreover, we propose a network performance-cost ratio model to analyze the cost-effectiveness of network and show that hybrid sensor network is financially cost efficient for a large case. Our optimization algorithm, together with the performance-cost ratio model, can be used to estimate the lifetime and financial cost of hybrid sensor network before actual deployment. [1]

Key words: Hybrid Sensor Networks, Energy, Lifetime, Deployment, Tabu Search

1 Introduction

This paper investigates the problem of *network deployment* in hybrid sensor/actuator networks. By hybrid sensor networks, we mean those networks consisting of both resource-rich and resource-impoverished sensor devices. The resource-rich devices, called *micro-servers*, are more expensive but have significantly greater bandwidth and energy capabilities compared to the low-cost, low-powered sensors. Such hybrid sensor networks have the potential to support the long-range and/or high-bandwidth communications required by data-intensive sensing applications using broadband networking standards such as 802.16 as well as the low-power, fine-grained sensing possible by smaller sensing devices. Examples of broadband sensor networking applications include time-elapsed imaging using video sensors for coastal monitoring, and speech analysis in home health care and cane-toad monitoring.

In the past couple of years, sensor networks research has addressed the development of sensor platforms [2], application domains [3], and communication paradigms [4–7]. However, they neither exploited hybrid device capabilities such as out-of-band data communication channels nor explored anycast services for sensor networks.

[1] This paper is a comprehensive extension of our earlier work in [1].

1.1 *Motivation for Hybrid Sensor Networks*

Historically, large scale networks have evolved to encompass myriad types of network devices. The Internet today combines different devices such as routers, servers and hosts. Even the routers can be classified into different categories (e.g. into core routers and edge routers). For large scale sensor networks that may have thousands of nodes in the future, it is more realistic to have hierarchical models of network devices rather than flat ones. Such a sensor network involves a hybrid of resource-rich specialized nodes in conjunction with small sensor devices [8]. The resource-rich nodes provide service such as (i) long-range data communications, (ii) persistent data storage, or (iii) actuation. Examples of actuation would be re-charging or replacing small nodes whose energy has been depleted, imagers which can take photos or video when activated by sensors, sprinklers used for precision agriculture which can sprinkle water in badly parched areas etc. The resource-rich node can act as a data sink, and we call it a *micro-server*. Fig. 1 shows the hierarchical view of a hybrid sensor network. Lower tier consists of numerous inexpensive sensors, e.g. MICA2 (See Fig. 2) from CROSSBOW [9]; and upper tier consists of many expensive but resource-rich micro-servers, e.g. STARGATE (See Fig. 2) from CROSSBOW.

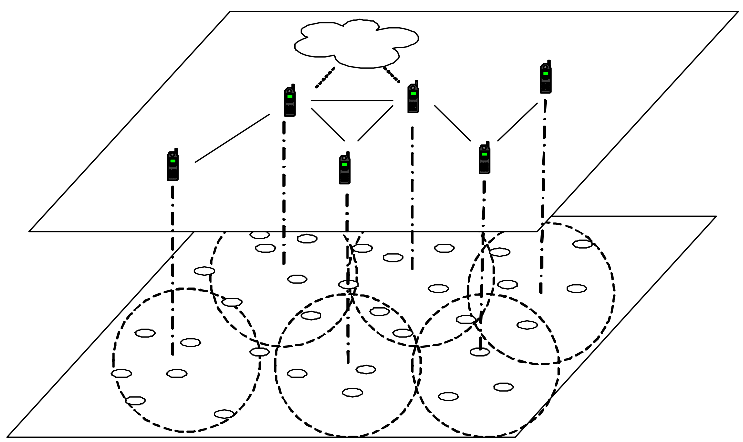

Fig. 1. An example of hybrid sensor network.

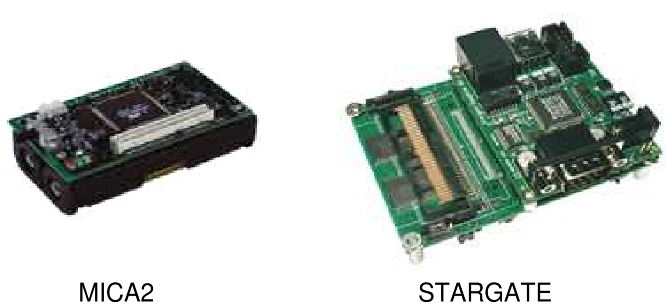

Fig. 2. MICA2 and STARGATE.

1.2 *Motivation for Data Anycast*

The key challenge in building Ad-Hoc multi-hop sensor networks from small, low-powered sensor nodes are scalability and energy-efficient mechanisms for data dissemination. Previously proposed data routing protocols [4–7] for sensor networks have not been designed to leverage the capabilities of hybrid devices. By exploiting resource-rich devices, the communication burden on smaller, energy, bandwidth, memory and computation-constrained sensor devices can be reduced. Consequently, these protocols may not be best suited for several applications of such hybrid sensor networks, which involve a multitude of mutually cooperative micro-servers.

Our thesis is that an *anycast* service, which routes sensor data to the nearest available micro-server, rather than to a single designated server, can provide significant improvements to the aforementioned data dissemination protocols for such applications and networks. The intuition is that you only care for the service, not which server provides it. The anycast service should be useful for several hybrid sensor applications.

Consider the case of mobile soldiers operating in a battlefield. The soldiers may be equipped with more powerful data transmitters (out of band higher-range radios) than sensors. It may be more effective to forward the information (e.g. enemy detection, land mine presence, convoy vehicles) to the nearest available soldier, who can forward it to the other soldiers, instead of sending it to all soldiers in the field. In a disaster recovery operation, several biochemical sensors may have been scattered, and multiple imagers (aerial or robotic) may be navigating the terrain. When biochemical sensors detect a toxic plume, this message just needs to go to the nearest imager (rather than a specific imager) which can act accordingly. In the example of Fig. 1, resource-impoverished MICA2 motes transfer data to one of the STARGATES, and the STARGATE can either handle the data or transfer it to interested parties using out-of-band transmission channel (e.g. WiFi) and other routing protocols, e.g. Ad-hoc On-demand Distance Vector (AODV) routing [10].

Our previous work, [11] shows that the use of an anycast protocol provides significant gains in network performance, such as system lifetime and data latency, when added to existing data dissemination protocols, such as Directed Diffusion [5]. In this paper, we choose to analyze anycast as follows.

1.3 *Hybrid Sensor Network Deployment: Problems and Contributions*

In this paper, we investigate some fundamental questions on hybrid sensor network deployment to support anycast communication.

- *Given a number of micro-servers, how does the placement of them affect the life time of network?*
- *What is the benefit of introducing additional micro-servers into network? Is it cost effective to introduce these extra micro-servers?*

To answer these two questions, we formulate an integer programming problem to study how the placement of micro-servers affect the lifetime of a hybrid sensor network using anycast communication. This optimization problem allows us to study the cost-benefit of using multiple micro-servers. Our cost model accounts for the variation in the cost and capability of network resources in a hybrid sensor network, such as bandwidth and energy consumption, as well as the spatio-temporal variation in network events. In particular, we find that the cost-effectiveness of micro-servers increases with the size of the network, thus making hybrid sensor networks a scalable solution. Although we study network deployment in the context of anycast communication, a similar methodology can also be applied to distributed storage and computation in hybrid sensor networks.

The rest of this paper is organized as follows. Section 2 provides an overview of the anycast communication model and the other related work. Section 3 proposes an integer linear programming formulation of the network deployment problem and prove the problem is NP-hard. Section 4 introduces a tabu-search algorithm to solve the problem efficiently. Section 5 presents an analysis to compare the lifetime differences and a cost analysis of different scenarios. Section 6 discusses our conclusions.

2 Related Work

In this section, we provide an overview of our anycast mechanism and the other related work.

2.1 Tree-Based Data Anycast

In this section, we provide an overview of our anycast mechanism which motivates the network deployment problem addressed in this paper.

We assume a *hybrid* sensor network which consists of both resource-rich micro-server nodes and low-power sensor nodes. Further we assume that there are multiple micro-servers (sinks) interested in the same data. Data needs to only reach one sink, thus motivating an anycast service. We assume that sensor network applications can handle small amount of data loss; and therefore

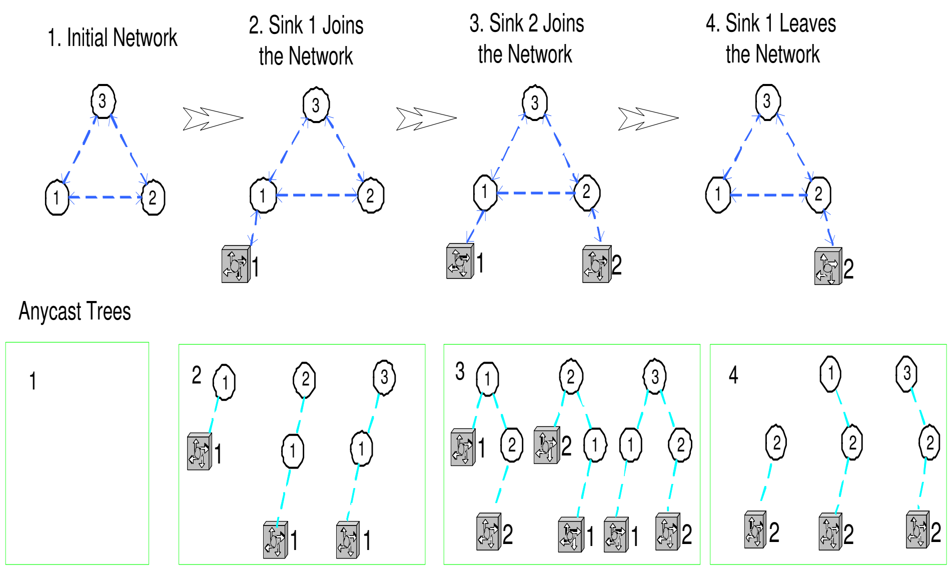

Fig. 3. Illustration of the anycast mechanism. The lower boxed pictures show the structure of each anycast tree as two sinks join and leave a sensor network.

anycast does not need to explicitly provide reliable data delivery.

We want to provide an anycast service that is scalable, self-organizing, robust, simple and energy-efficient. To implement this, we adopted a shared tree approach. Corresponding to each event source, a *shortest-path tree* rooted at the source is constructed. Sinks form the leaves of the tree. Sinks can dynamically join or leave the anycast tree. Although this approach requires more network state, it is a good approach to handling dynamics, as it simultaneously maintains paths to all sinks. By eliminating the need to discover paths to alternate sinks each time a sink leaves, it can reduce worst-case latency (when sinks fail) and does not require synchronization among sinks. Fig. 3 illustrates how the structure of each anycast tree evolves when two sinks join and leave a sensor network. Details of the anycast mechanism are described in [11].

An important metric in determining the performance of the anycast scheme is the number and placement of micro-servers (resource-rich nodes), relative to low-powered sensor nodes. The number of micro-servers must be sufficient to meet system lifetime objectives, as well as other application-governed objectives (e.g. message delivery latency), without exceeding resource cost thresholds. Moreover, the number of micro-servers chosen depends on parameters such as the occurrence pattern (frequency, spatial distribution) of sensor events in the system. In Section 3, we propose a problem formulation for resource provisioning, *i.e.* placement of micro-servers and sensors, incorporating all these factors.

2.2 Related Work in Deployments and Lifetime Optimizations of Sensor Networks

Although previous work has considered optimal sensor deployment in the context of homogeneous sensor networks [12–14], network deployment has not been previously considered in the context of hybrid sensor networks.

Energy consumption model and the network lifetime analysis in homogeneous sensor networks have been studied in [15,16] respectively. In [17], the authors analyze heterogeneous deployments of sensor networks and shows how they impact the coverage aging process of a sensor network. In [18], the authors try to maximize the amount of information collected by all the nodes within required lifetime of two-tired sensor network. Power-aware base station positioning in sensor networks problem has been investigated in [19] recently. In [20], the authors try to meet the lifetime requirement of sensor network by adding additional sinks into the network. Network lifetimes are calculated by simulations. The upper layer sensor locations' problem in two-tiered sensor networks has also been studied in [21]. However, their networks consist of homogeneous sensor devices; therefore, their research focuses on the lifetime constraint of upper-layer sensors.

However, previous work do not consider the problem with any routing protocol in hybrid sensor networks. The key difference between our work and these prior studies is that we focus on the deployment, e.g. network lifetime and financial cost, of hybrid sensor networks that use tree-based anycast as routing algorithm. In addition to the energy constraints in sensors, we also consider the energy constraints in micro-servers. Moreover, our work also provide a comprehensive study on the impacts of heterogeneity and the financial cost-effectiveness of hybrid sensor networks.

3 Cost Model and Optimization

In this section, we propose a model to investigate how the number of micro-servers and their placement affect the lifetime of a hybrid sensor network and prove that the problem is NP-hard. In this paper, we define network lifetime as the cumulative active time of the network until the time when the depletion of the first sensor or micro-server happens.

3.1 Cost Model

We model a sensor network as a graph $G = (V, E)$, where V is a set of vertices 1, 2, ... n and E is a set of edges. A sensor or micro-server i is located at (i_x, i_y). Given the transmission range of sensor (R), $e_{i,j}$ is an edge if equation (1) holds.

$$\sqrt{(i_y - j_y)^2 - (i_x - j_x)^2} \leq R \tag{1}$$

For a given G that has n vertices, assuming that each vertex can hold either a sensor or a micro-server, our placement problem is to decide where the micro-servers should be placed so that the lifetime of the network can be maximized. In order to formulate the placement problem, we define the following parameters:

- A number of events r_i can be detected by either a sensor or a micro-server at location (i_x, i_y) within each time unit.
- It costs e_1 units of energy for a sensor to sense an event, and then, to transmit the related data packets of the event.
- It costs E_1 units of energy for a micro-server to sense an event.
- It costs e_2 units of energy for a sensor to forward (both receive and transmit) the data packets of an event.
- It costs E_2 units of energy for a micro-server to receive the data packets of an event. Depending on the application, micro-servers can either handle the event locally or co-ordinate with each other. Therefore, we do not consider transmission in the upper layer, i.e. the communications among micro-servers.
- The initial energy of a sensor is B^{sensor} units.
- The initial energy of a micro-server is B^{server} units.
- The shortest distance (hop-count) between vertex i and vertex j is d_{ij}.
- The network lifetime is L.
- The lifetime of sensor or micro-server at vertex k is L_k.
- $\lambda = \frac{1}{L}$
- $\lambda_k = \frac{1}{L_k}$

Sensors use their energy for two purposes — (i) sensing, and (ii) relaying packets from a data source to a micro-server. In order to have the second type of energy consumption captured in the optimization model succinctly, we define the indication function γ_{ij}^k as follows:

$$\gamma_{ij}^k = \begin{cases} 1 & \text{if vertex } k \text{ is on the transmission path from vertex } i \text{ to vertex } j \text{ and } k \neq j \text{ (Note that the requirement that } k \neq j \text{ is required because the last node in the path does not have to re-transmit.)} \\ 0 & \text{otherwise.} \end{cases}$$

The values of γ_{ij}^k depend on the network's routing algorithm (e.g. tree-based anycast) and can be calculated in advance.

The decision variables are x_i as:

$$x_i = \begin{cases} 1 & \text{if the device at vertex } i \text{ is a sensor} \\ 0 & \text{otherwise (a micro-server).} \end{cases}$$

With anycast routing, a sensor will be transmitting to the closest micro-server. To enforce this in the problem formulation, we define an auxiliary variable z_{ij}:

$$z_{ij} = \begin{cases} 1 & \text{if the micro-server at vertex } j \text{ is the closest micro-server to the sensor at vertex } i \\ 0 & \text{otherwise.} \end{cases}$$

The objective of the optimization is choose the locations of the m micro-servers so as to maximize the lifetime of the network. Therefore, the problem can be formulated as:

$$Minimize \quad \lambda \tag{2}$$

subject to:

$$r_k e_1 x_k + \sum_{i=1}^{n} \sum_{j=1}^{n} (\gamma_{ij}^k r_i z_{ij}) e_2 - B^{sensor} \lambda_k \leq 0, \forall k \tag{3}$$

$$r_k E_1 - r_k E_1 x_k + \sum_{i=1}^{n} (r_i z_{ik}) E_2 - B^{server} \lambda_k \leq 0, \forall k \tag{4}$$

$$d_{ij} w_{ij}^k \leq d_{ik} - d_{ik} x_k, \forall i, j, k \tag{5}$$

$$w_{ij}^k \leq z_{ij}, \forall i, j, k \tag{6}$$

$$z_{ij} - x_k \leq w_{ij}^k, \forall i, j, k \tag{7}$$

$$\gamma_{ij}^k z_{ij} - x_k \leq 0, \forall i, j, k \tag{8}$$

$$\sum_{i=1}^{n} x_i = n - m \tag{9}$$

$$z_{ij} - 1 + x_j \leq 0, \forall i, j \tag{10}$$

$$\sum_{j=1}^{n} z_{ij} = 1, \forall i \tag{11}$$

$$\lambda \geq \lambda_i, \forall i \tag{12}$$

$$x_i \in \{0, 1\}, \forall i \tag{13}$$

$$z_{ij} \in \{0, 1\}, \forall i, j \tag{14}$$

$$w_{ij}^k \in \{0, 1\}, \forall i, j, k. \tag{15}$$

Constraints (3) and (4) model, respectively, the energy consumption of a sensor and a micro-server. The details as to how these constraints are derived can be found in the Appendix.

Constraints (5) to (7) enforce that a sensor delivers packets only to the closest micro-server. For details on derivation, see Appendix. Constraint (8) ensures a micro-server cannot be an intermediate node of a path. Constraint (9) limits that there are m micro-servers in the network. Constraint (10) ensures that only a micro-server can be the end point (sink) of disseminated data. Constraint (11) enforces that a sensor sends packets to one micro-server only. Constraint (12) says the lifetime of the network is the smallest lifetime of all the sensors and micro-servers. Constraints (13, 14, 15) define the scopes of variables x_i, z_{ij} and w_{ij}^k.

Remark: Although the above formulation uses the mean spatial data rate r_k to determine the locations of the micro-servers. It can be given a more general interpretation. Given a temporal-spatial data rate distribution $r_k(t)$ at time t, if the lifetime is sufficiently long and the distribution has finite mean and variance, we can apply Central Limit Theorem and Gaussian distribution to argue that the spatial data rate at vertex k is less than r_k^ϵ with probability $(1 - \epsilon)$. By using r_k^ϵ in our formulation instead, we can obtain a lifetime guarantee with probability $(1 - \epsilon)$.

3.2 Proof of NP-Hardness

We will prove that the Integer Programming problem introduced in Section 3.1 is a NP-hard problem by transform it to a well-known NP-hard problem of *p-median* problem [22].

We consider a special case of our problem where only one sensor k has energy limitation (all the other sensors and micro-servers have no energy limitation), and packets can be delivered to any of micro-server regardless of the distance between the source and the micro-server. Since the network lifetime equals to the lifetime of sensor k, the objective function (2) and equation (12) can be rewritten as:

$$Minimize \quad \lambda_k \tag{16}$$

.

As equation (3), we can further rewrite the objective function as:

$$Minimize \quad r_k e_1 + \sum_{i=1}^{n}\sum_{j=1}^{n}(\gamma_{ij}^{k} r_i e_2 z_{ij}) \tag{17}$$

.

The constraints of the model can be rewritten as:

$$\sum_{i=1}^{n} x_i = n - m \tag{18}$$

$$z_{ij} - 1 + x_j \leq 0, \forall i, j \tag{19}$$

$$\sum_{j=1}^{n} z_{ij} = 1, \forall i \tag{20}$$

$$x_i \in \{0, 1\}, \forall i \tag{21}$$

$$z_{ij} \in \{0, 1\}, \forall i, j \tag{22}$$

.

The above model is a *p-median* problem [22] where $p = n - m$. Therefore, our problem is NP-hard.

4 A Tabu Search Algorithm

Since the combinatorial optimization problem introduced in Section 3 is NP-hard, it is very inefficient to solve the problem and achieve optimized solution.

From our experience, we find that the maximum network size that the state-of-art commercial optimization package CPLEX [23] can handle efficiently is 20 nodes. Thus, the results produced by CPLEX are not very helpful for the deployment of a reasonable size network. We therefore develop an heuristic solution based on tabu search [24].

```
int tsStable = 0;
int stabilityLimit = 500;

while(tsStale < stabiliyLimit) {

    if(bestGain(x, best, obj) >= 0) {    //intensification
        randomMoveOneOfTheBest(x);
    } else {                              //diversification
        randomMoveAllMicroservers(x);
    }
    if(obj > best) {                      //better result found
        best = obj;
        tsStable = 0;
    } else {
        tsStable = tsStable + 1;
    }

    update_tabu_list(tabu_list_from, tabu_list_to);
}

bestGain(x, best, obj) {
    old = obj;
    soFarBest = -1;

    for each neighbour of current microservers {
        getlifetime(x, obj);

        if(obj > best) {        //aspiration level condition
            update(x);
            soFarBest = obj;
        } else if(inTabulist(x)) {
            continue;
        } else {
            if(obj > soFarBest)
            soFarBest = obj;
        }
    }
    return old - obj;
}
```

Fig. 4. A tabu-search algorithm for sensor network lifetime Optimization Model.

4.1 Tabu Search

The tabu search is conducted within a neighborhood of the current solution. We have tested a number of different ways of defining the neighborhood and our experience shows that the following works best: during a local search, we vary the location of one micro-server at a time; if the current location of the micro-server is at grid k, then its neighborhood N_k is defined as all the other grids in the network:

$$N_k = \{1, 2, ..., k-1, k+1, ...n\} \tag{23}$$

Our tabu-search algorithm (Fig. 4) defines two tabu lists. The first one records the vertices that micro-servers can not move to for a number of iterations I_t. The second one records the vertices that micro-servers can not leave for another number of iterations I_f. The value of I_t and I_f should be large enough to avoid cycles (we tuned them as $3/4 \times n$ and $1/2 \times m$ respectively in our experiments).

The algorithm tries to find out a local maximum by calculating the lifetime of each possible single move in *intensification* stage. When the gain is negative, the algorithm explores the unexplored area in *diversification* stage by random movement. Note that it will not move to recent locations since they are recorded in tabu-lists unless aspiration level condition is satisfied. The *aspiration level condition* is defined as a new best lifetime found. The algorithm terminates when the objective function has not improved for the number of *stabilityLimit* iterations. The *stabilityLimit* parameter is defined as a large integer (e.g. 500) to ensure the robustness of the algorithm.

4.2 Algorithm Benchmark

To validate the tabu-search algorithm, we compared its results with those from CPLEX for a 20-grid network (the maximum grid size that CPLEX can handle efficiently). The results, see Table 1, showed that our tabu-search algorithm achieved the same optimal results as CPLEX, but in much shorter time.

We have also applied our tabu search algorithm to larger grid sizes. For example, for a grid size of 100 and 10 micro-servers, it takes about 8 minutes and 48 seconds to obtain a solution.

Furthermore, we compared the results of our tabu-search algorithm and the best of a large number, e.g. 1,000, random micro-server placements at a 150-node network (see Fig. 11) in Table 2. It shows that our tabu-search algorithm can achieve results which are up to 40 percent better than those of the best of 1,000 random micro-server placements.

Table 1
Results of CPLEX and tabu-search algorithm at a 20-grid network.

Number of Micro-servers	Lifetime		Computation Time (seconds)	
	CPLEX	Tabu-search	CPLEX	Tabu-search
1	16901	16901	105.94	0.16
2	22641	22641	633.74	0.51
3	25531	25531	900.5	1
4	25531	25531	732.22	2.22
5	25531	25531	1618.37	8.75
6	29268	29268	342.95	40.71

Table 2
Results of tabu-search algorithm and the best of 1000 random micro-server's placements at a random 150-node network.

Number of Micro-servers	Lifetime		Ratio
	Tabu-search	Random	
1	6703.91	6703.91	100%
2	11215	11215	100%
3	16901.4	14457.8	117%
4	22641.5	18461.5	123%
5	25531.9	22641.5	113%
6	29268.3	25531.9	115%
7	34285.7	29268.3	117%
8	34285.7	29268.3	117%
9	41379.3	29268.3	141%
10	41379.3	34285.7	121%

5 Results and Analysis

The mathematical model introduced in Section 3 enables us to study the effect of the number of micro-servers and their placements on the network lifetime of hybrid sensor networks utilizing anycast routing. Moreover, this model also allows us to study the financial cost effectiveness and in particular to determine the most cost-effective combination of sensors and micro-servers in a hybrid sensor network. Furthermore, our scalability studies show that the cost effectiveness of hybrid sensor networks increases with the size of the

Table 3
Simulation parameters

Parameter	Value
Initial energy of a sensor	6,000 Joules
Initial energy of a micro-server	60,000 Joules
Energy to sense and transmit the data packets of an event	35 mJ
Energy to sense an event for a micro-server	25 mJ
Energy to forward the packets generated by an event for a sensor	6 mJ
Energy to receive the packets generated by an event for a micro-server	6 mJ

network.

We used three different network sizes (50, 100, 150) and two types of network topologies (grid and random), in our studies. The parameters that we used for our study is showed in Table 3. Note that the sensing and transfer energy figures are taken from [8]. The reason why we chose 6,000 J for the sensor is that this is the energy found inside two AA batteries. We used two different traffic patterns. The first traffic pattern, a uniform traffic pattern, had five events taking place at each sensing location within each time unit. The second traffic pattern, a non-uniform traffic pattern, had r_k events taking place at a sensing location k per unit time where r_k is an uniformly distributed integer in $[0, 10]$.

5.1 Network lifetime analysis

In order to study the effect of the number of micro-servers and micro-server placement on the lifetime of the network. For a given number of micro-servers, we find:

(1) The micro-server placement, that will give the maximum lifetime using the tabu-search algorithm described in Section 4, will be referred to as “the best”.
(2) The lifetime resulted from random/blind placement of the micro-servers. This is calculated by generating 29 random placements according to uniform distribution. The mean lifetime of these 29 placements will be referred to as “random (mean)”, and the worst lifetime of these placements will be referred to as “the worst”. “random (mean)” is the expected network lifetime of random/blind micro-server deployment.

For the 150-grid (15 columns and 10 rows) case, Fig. 5, 6 plot lifetime for the best, the worst and random (mean) placement against different number of micro-servers with uniform and non-uniform traffic patterns, respectively. The figures show that network lifetime can be improved by placing micro-servers at optimal locations. For example, when two micro-servers are deployed, the best micro-server placements can extend the network lifetime by about four folds comparing to the worst, and by more than about 100% comparing to the random (mean) placements. This demonstrates the need to optimize the locations of the micro-servers.

Fig. 5, 6 also show that, with optimal placement, additional micro-servers can improve network lifetime significantly. For example, network lifetime improves by more than 80% with the addition of the second micro-server when the traffic pattern is uniform.

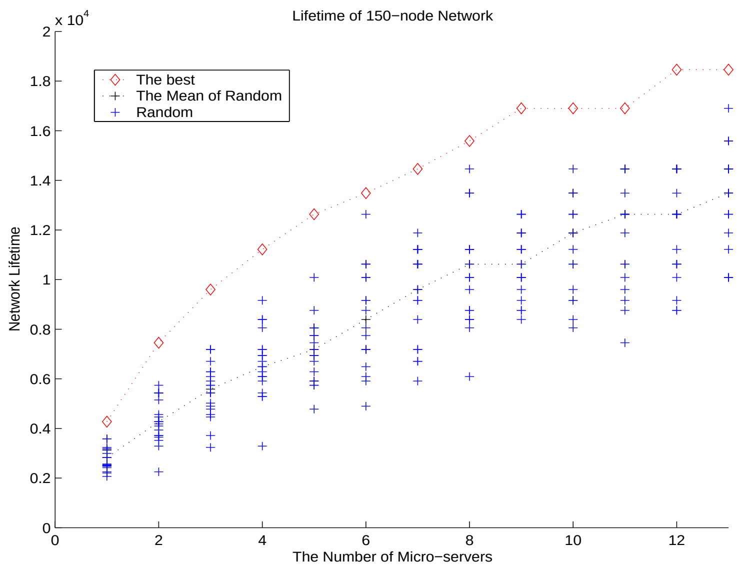

Fig. 5. Network life time of a 150 grid network with different number of micro-servers. (uniform traffic pattern)

We further investigated the effect of micro-server placement in a general network. We generated a random topology of 150 nodes where the nodes are located inside an area of $320m \times 240m$ and the transmission range R of nodes is 40m. Fig. 7, 8 show the best micro-server locations of 3 and 4 micro-servers scenarios respectively.

Similar to grid topology, Fig. 9 shows the optimal micro-server location can improve network lifetime significantly. Moreover, Fig. 9 shows that our tabu-search algorithm, compared to random micro-server placement, performs significantly better in non-uniform topology than in grid topology. There are two reasons for such performance difference. Firstly, there are a large number of local optima “plateaus” in grid topology, which makes the probability higher for random algorithm to have a “good” solution. Secondly, our tabu-search

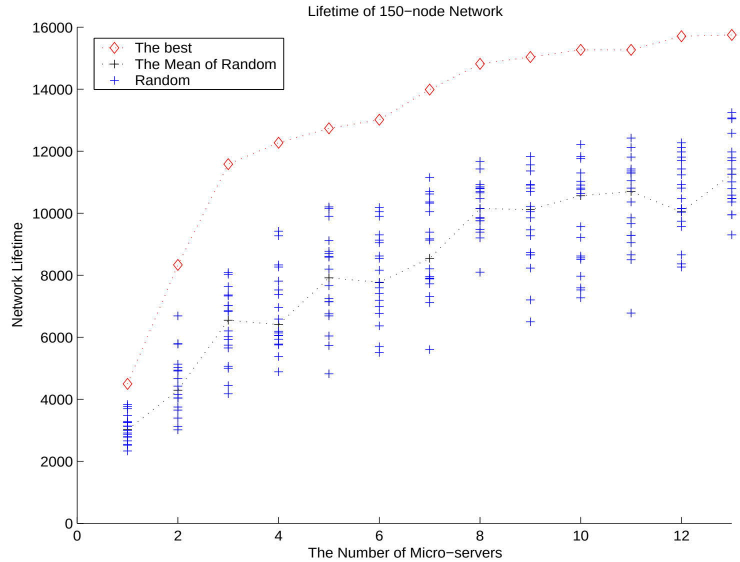

Fig. 6. Network life time of a 150 grid network with different number of micro-servers. (non-uniform traffic pattern)

algorithm can achieve better result ("global peaks") compared to the local optima ("plateaus") achieved by the best of random algorithm. However, the differences between "global peaks" and "plateaus" are not as large in grid topology as the difference in non-uniform topology.

Moreover, we investigated the performance of our algorithm in different network topologies. We generated 20 random topologies of 150 nodes inside an area of $320 \times 240m$. We calculated "the best", "the worst" and "random(mean)" lifetimes of these networks with 4 micro-servers. Fig. 10 shows that "the best" micro-server placement can extend network lifetime by around 2.5 times comparing to "the mean", and by more than 5 folds compared to "the worst". Fig. 11, 12 show two of the topologies and related micro-server placements. This demonstrates the robustness of our algorithm.

5.2 The impacts of heterogeneity

Hybrid sensor networks can extend the network lifetime in two aspects: injecting extra energy into the system by adding additional micro-servers, and shortening the data transmission paths. In this section, we study the impacts of these two aspects.

For the random 50-node network topology shown in Fig. 13, we analyzed the relationships between network lifetimes and initial total network energies for a number of scenarios, which are summarized in Table 4. We used a fixed value of 6,000 Joules as B^{sensor} and varied the values of B^{server} from 6,000 Joules to

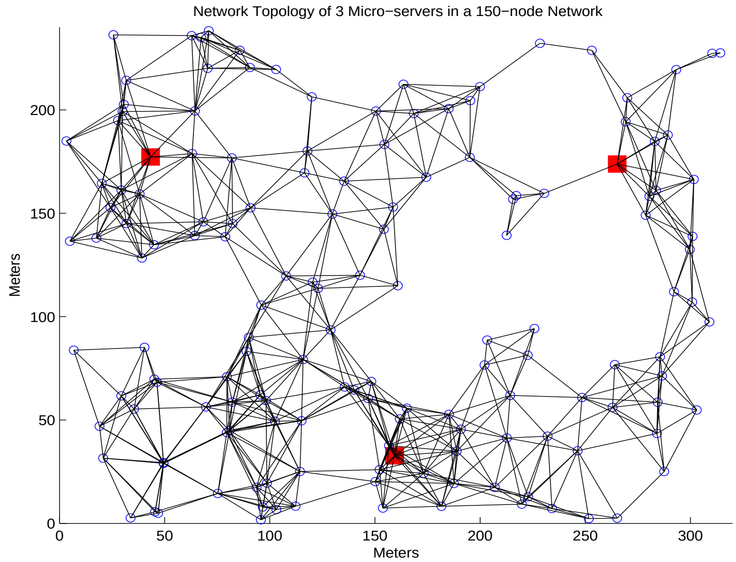

Fig. 7. Network topology and the best 3 micro-server placements of a random 150-node network. (The micro-server is indicated by a square.)

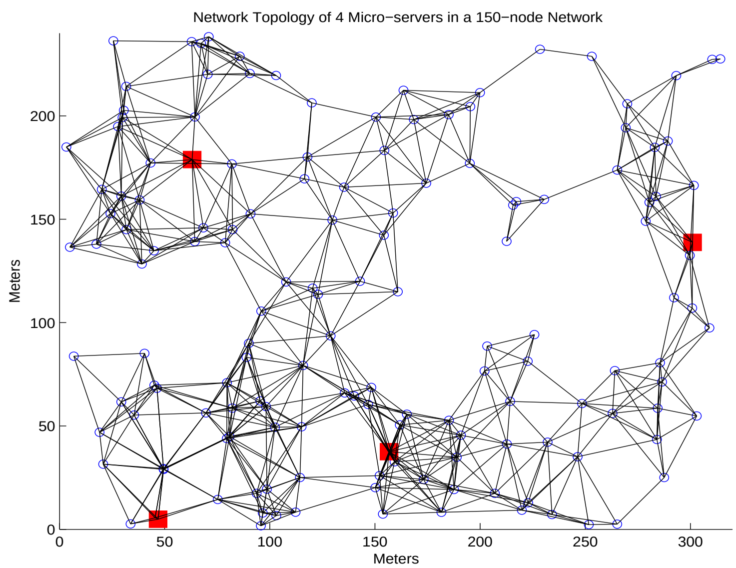

Fig. 8. Network topology and the best 4 micro-server placements of a random 150-node network. (The micro-server is indicated by a square.)

246,000 Joules in our experiments. The scenarios that we used for comparisons are defined as follows:

Traditional: there are 49 sensors with initial energy of B^{sensor} and one micro-server with initial energy of B^{server}. In this scenario, the increased network energies will be allocated to one micro-server only.

Homogeneity: there are 46 sensors with initial energy of $(49B^{sensor} + B^{server})/50$ and four micro-servers with initial energy of $(49B^{sensor} + B^{server})/50$; the total initial network energy equals to that in "Traditional". In this scenario,

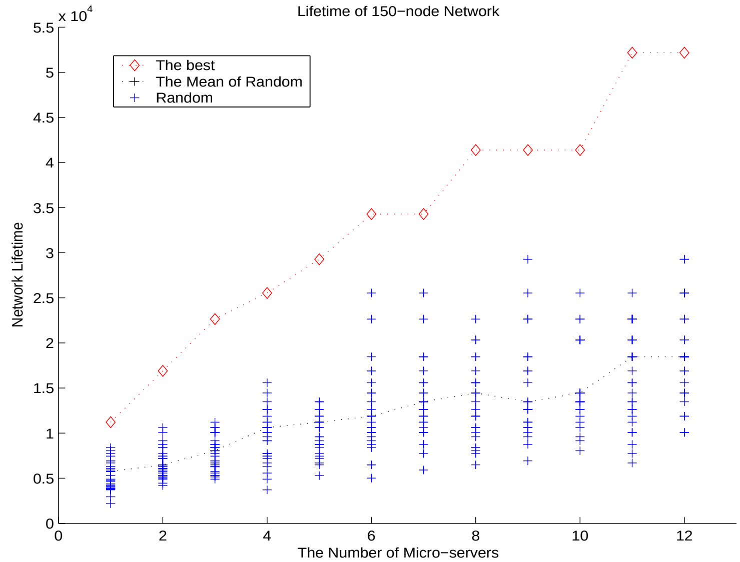

Fig. 9. Network lifetime of a random 150-node network with different number of micro-servers.

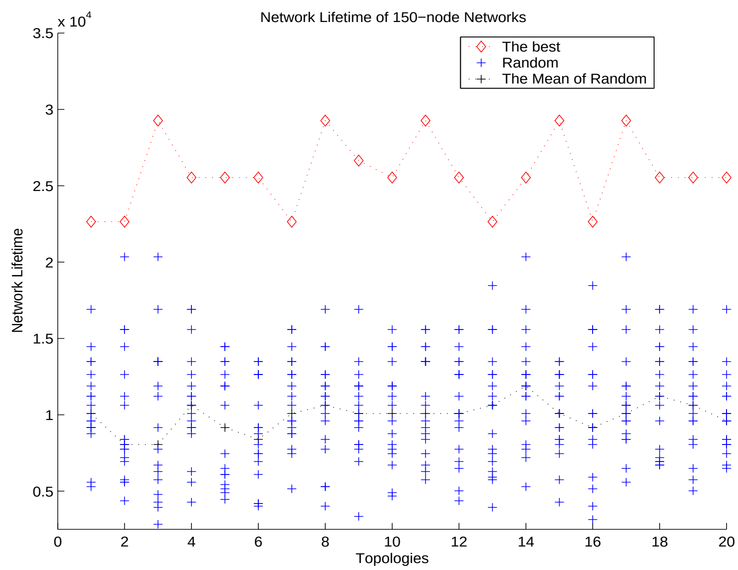

Fig. 10. Network lifetime of 20 random 150-node network with 4 micro-servers.

the increased network energies will be allocated to all devices, i.e. sensors and micro-servers, evenly. We chose this scenario to analyze the impacts of different initial total network energies to network lifetime.

Heterogeneity I: there are 46 sensors with initial energy of B^{sensor} and four micro-servers with initial energy of $(B^{server} + 3B^{sensor})/4$; but the total initial network energy equals to that in "Traditional". In this scenario, the increased network energies will be allocated to all micro-servers evenly.

Heterogeneity II: there are 46 sensors with initial energy of B^{sensor} and four micro-servers with initial energy of B^{server}; and the total initial network energy is more than that in "Traditional". In this scenario, the increased

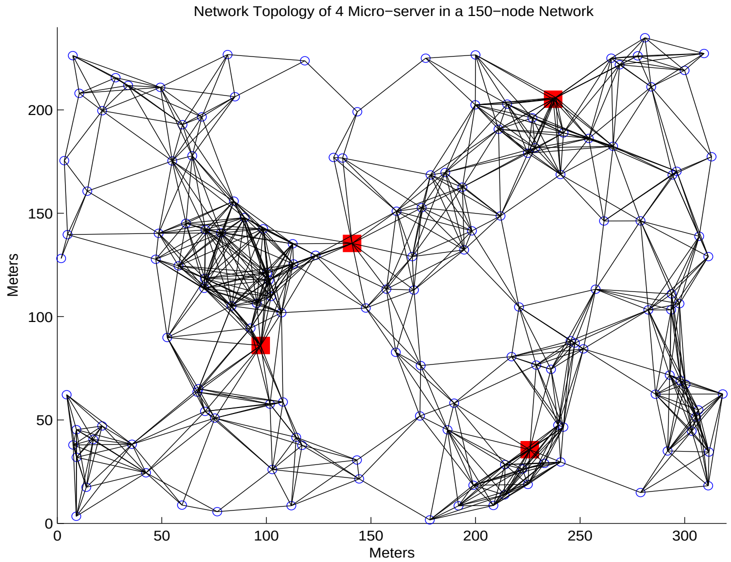

Fig. 11. Network topology and the best 4 micro-server placements of a random 150-node network. (The micro-server is indicated by a square.)

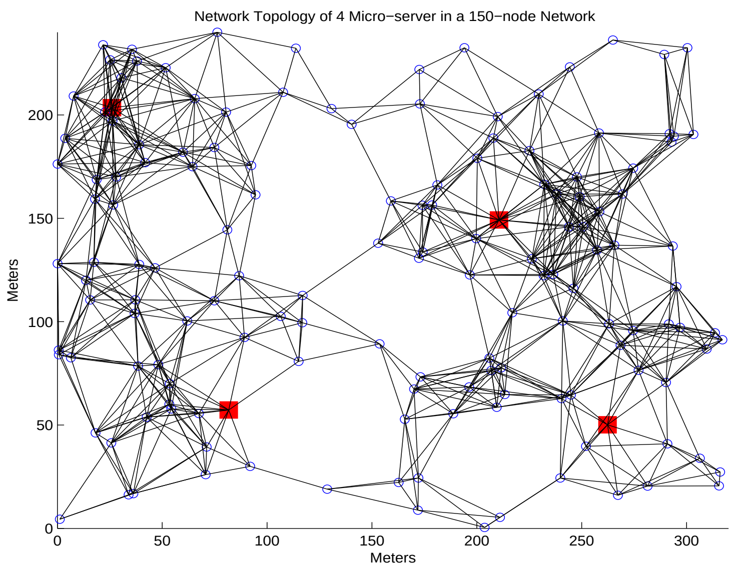

Fig. 12. Network topology and the best 4 micro-server placements of a random 150-node network. (The micro-server is indicated by a square.)

network energies will be allocated to all micro-servers evenly.

Note that we chose four micro-servers arbitrarily in “Homogeneity”, “Heterogeneity I” and “Heterogeneity II”. However, the number of micro-servers won’t have impact on general result.

Fig. 14 plots the network lifetimes with different initial micro-server energy B^{server}. Although network lifetimes increase with the injection of additional energy in all cases, the lifetimes of both “Heterogeneity” cases increase at sig-

Table 4
The numbers and initial energies of sensors, micro-servers and network in four test scenarios.

	Sensors	Micro-servers	Total Initial Network Energy
Traditional	$49 \times B^{sensor}$	$1 \times B^{server}$	$49B^{sensor} + B^{server}$
Homogeneity	$46 \times (49B^{sensor} + B^{server})/50$	$4 \times (49B^{sensor} + B^{server})/50$	$49B^{sensor} + B^{server}$
Heterogeneity I	$46 \times B^{sensor}$	$4 \times (B^{server} + 3B^{sensor})/4$	$49B^{sensor} + B^{server}$
Heterogeneity II	$46 \times B^{sensor}$	$4 \times B^{server}$	$46B^{sensor} + 4B^{server}$

B^{sensor}: the initial energy of sensor

B^{server}: the initial energy of micro-server

nificantly faster rates. It is the locations of energy-injection, i.e. micro-servers' locations, rather than the energy-injection itself that have much greater impacts on network lifetime. Namely, the impact of shorter transmission paths contributes much more to the longer network lifetimes than that of additional energy-injection. In "Traditional", "Heterogeneity I" and "Heterogeneity II", since extra energies are allocated to the micro-servers only, the bottleneck of network lifetime increasing is the lifetime of sensor after some thresholds.

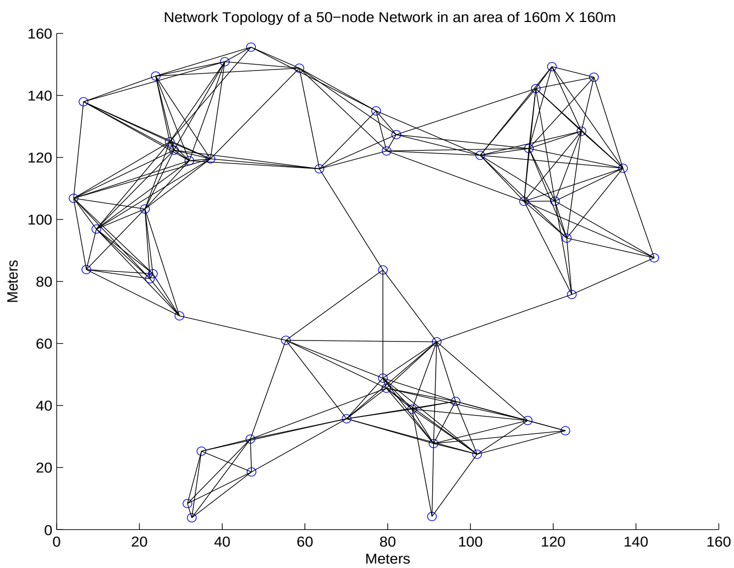

Fig. 13. A random network topology of 50 nodes.

5.3 Financial cost-effectiveness analysis

It is obvious that the network lifetime increases with the number of micro-servers. An important question is how cost-effective this is. We define the performance cost ratio of a hybrid sensor network with m micro-servers as

$$L_m = \frac{L}{(n-m)c_s + mkc_s} \tag{24}$$

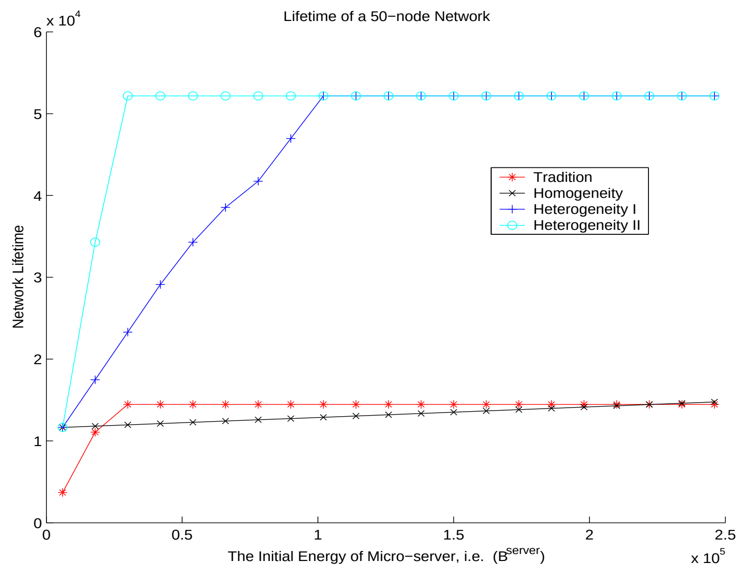

Fig. 14. Features network lifetime VS. initial total network energy.

where L is network lifetime and the denominator is the network cost. The cost consists of $n-m$ sensors at cost c_s and m micro-servers at cost kc_s where k represents the ratio of the cost of a micro-server to a sensor. If we use the current costs of Mica Mote and STARGATE, then $k=5$. However, this can change in the future. In our studies, we used k from 5 to 200. Note that the network cost here is the hardware cost of the network, and the communication cost has been taken care of by our optimization model (the network lifetime, equations (3) and (4)).

As a basis of comparison, we normalized the performance cost ratio with respect to that with only one micro-server; namely, we defined $N_{L_m} = \frac{L_m}{L_1}$. Note that N_{L_m} is independent of unknown parameter c_s. Therefore, if N_{L_m} is larger than 1 or 100%, then network L_m (which has m micro-servers) is financially more cost-effective than L_1 (which has one micro-server).

Fig. 15, 16 plot the values of the normalized performance ratio for 150-node grid networks with uniform traffic patterns and non-uniform traffic patterns respectively. Fig. 17 plots the values of the normalized performance ratio for a random 150-node network shown in Fig. 11 with uniform traffic patterns. The figures show that hybrid sensor networks are cost effective for a wide range of k. For example, in a 150-grid network with uniform traffic pattern, for $k=5$ and $m \in [3,14]$, the cost-effectiveness of these networks are more than twice that of a single micro-server network. The figures also show that hybrid sensor network is more financially cost effective when the network topology is non-uniform than it is grid. For example, when $k=10$, in a 150 grid network with uniform traffic pattern, the network lifetime per unit cost when there are four micro-servers is about 2.2 longer than that when there is one micro-server; while in a non-uniform network with uniform traffic pattern, the network lifetime per unit cost when there are four micro-servers is about 2.8 longer than that when there is one micro-server. Namely, hybrid sensor network is scalable with network topology complexity.

Moreover, to achieve maximum cost-effectiveness, the figures show that different number of micro-servers should be used as the values of k change. For example, in a 150 grid network with uniform traffic pattern, if $k = 5$, the lifetime of network can be extended by more than 230% at the same cost ratio if twelve micro-servers are used compared to just one micro-server is used; if $k = 50$, network lifetime can be extended by about 50% at the same cost ratio if three micro-servers are used compared to just one micro-server is used. Not surprisingly, the performance decreases as the value of k increases (when micro-server becomes much more expensive than sensor).

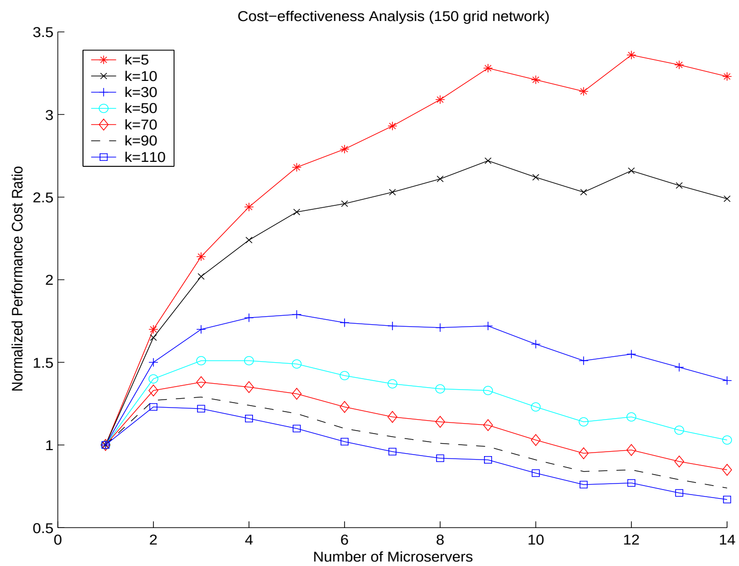

Fig. 15. The normalized performance cost ratio N_{L_m} at a 150 grid network with uniform traffic patterns.

Furthermore, we found that cost-effectiveness increases with network size. We have plotted N_{L_m} for $k = 50$ for different grid network sizes with uniform traffic patterns in Fig. 18. It shows that the larger the network, the more financially cost-effective it is to add additional micro-servers into the network. For example, the network lifetime per unit cost can be extended by more than 40% when the second micro-server is added to a 150 grid network, while the lifetime can be only extended by about 20% or 10% respectively when the second micro-server is added to a 50 or 100 grid network. We had also performed experiments with non-uniform traffic patterns and random network topologies. They shown similar results.

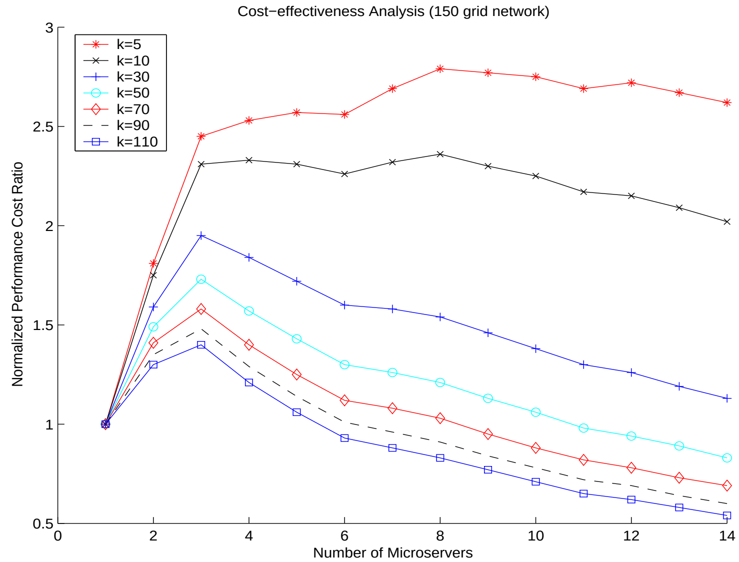

Fig. 16. The normalized performance cost ratio N_{L_m} at a 150 grid network with non-uniform traffic patterns.

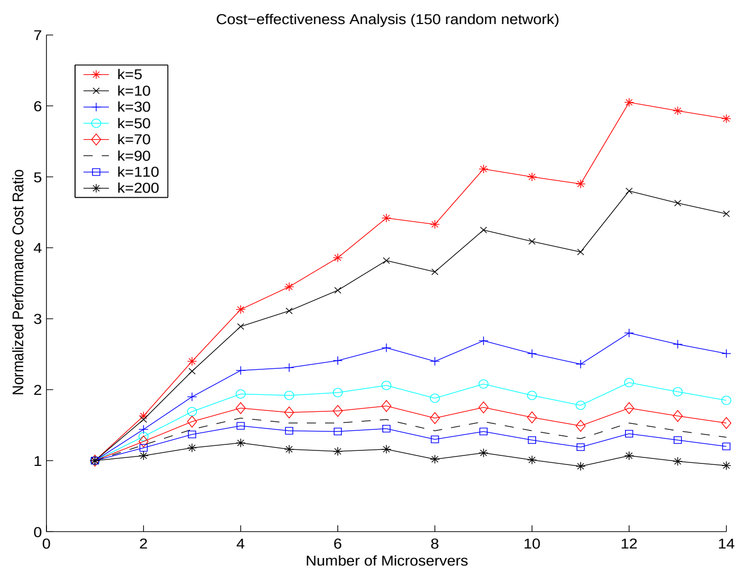

Fig. 17. The normalized performance cost ratio N_{L_m} at a 150 random network with uniform traffic patterns.

6 Conclusions

In this paper, we considered the problem of network deployment for hybrid sensor networks, consisting of both resource-rich and resource-impoverished sensor devices.

We model the sensor network as a graph. We proposed an integer linear programming formulation to maximize network lifetime, proved that it is NP-

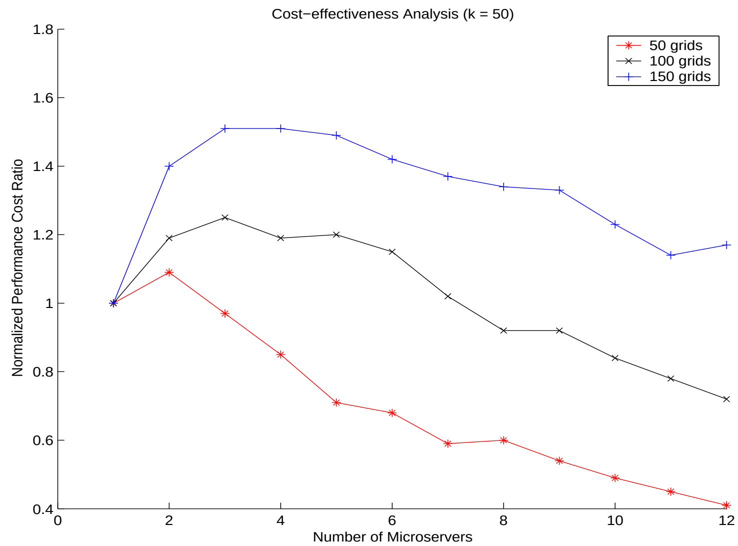

Fig. 18. The performance cost ratio N_{L_m} at 50, 100 and 150 grid networks with uniform traffic patterns. ($k = 50$)

hard, and introduced a tabu-search algorithm to answer some fundamental questions related to hybrid sensor network deployment — for a given number of micro-servers, what is the maximum lifetime of a sensor network and what is the optimal micro-server placement? What benefit can additional micro-servers add to the network, and how cost-effective is it to introduce these micro-servers?

Our extensive studies show that network lifetime can be increased dramatically with the addition of extra micro-servers; the locations of micro-servers can affect the lifetime of network significantly. We also proposed a network performance-cost ratio model and showed that a maximum performance cost ratio can be achieved. In particular we find that the cost-effectiveness of micro-servers increases with network size, thus making hybrid sensor networks a scalable solution. Although we studied network deployment to support anycast communication, a similar methodology could be applied to deployment for distributed computation and storage in hybrid sensor networks.

7 Acknowledgment

National ICT Australia is funded through the Australian Government's *Backing Australia's Ability* initiative, in part through the Australian Research Council. Nirupama Bulusu is supported by research and equipment grants from the National Science Foundation (award CISE-RR-0423728), Tektronix, the PSU Foundation, and the Maseeh College of Engineering and Computer Science.

References

[1] W. Hu, C. T. Chou, S. Jha, N. Bulusu, Deploying long-lived and cost-effective hybrid sensor networks, in: Proceedings of The First Workshop on Broadband Advanced Sensor Networks. (BaseNets 2004), San Jose, CA, October 25,2004.

[2] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, K. Pister, System architecture directions for networked sensors, in: Proceedings of the ninth international conference on Architectural support for programming languages and operating systems, ACM Press, Cambridge, Massachusetts, United States, 2000, pp. 93–104.

[3] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh, D. Rubenstein, Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with zebranet, in: Proceedings of the 10th international conference on Architectural support for programming languages and operating systems, ACM Press, San Jose, California, 2002, pp. 96–107.

[4] S. Madden, M. J. Franklin, J. M. Hellerstein, W. Hong, Tag: a tiny aggregation service for ad-hoc sensor networks, ACM SIGOPS Operating Systems Review 36 (SI) (2002) 131–146.

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

[6] F. Ye, H. Luo, J. Cheng, S. Lu, L. Zhang, A two-tier data dissemination model for large-scale wireless sensor networks, in: Proceedings of the eighth annual international conference on Mobile computing and networking, ACM Press, Atlanta, Georgia, USA, 2002, pp. 148–159.

[7] J. Liu, F. Zhao, D. Petrovic, Information-directed routing in ad hoc sensor networks, in: Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications, ACM Press, San Diego, CA, USA, 2003, pp. 88–97.

[8] R. Kumar, V. Tsiatsis, M. B. Srivastava, Computation hierarchy for in-network processing, in: Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications, ACM Press, San Diego, CA, USA, 2003, pp. 68–77.

[9] Crossbow technology, inc. http://www.xbow.com.

[10] N. W. Group, Ad hoc on-demand distance vector (aodv) routing, IETF RFC 3561.
URL `http://www.ietf.org/rfc/rfc3561.txt`

[11] W. Hu, N. Bulusu, S. Jha, A communication paradigm for hybrid sensor/actuator networks, in: Proceedings of the IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2004), Barcelona, Spain, 2004.

[12] A. Howard, M. J. Mataric, G. S. Sukhatme, Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem, in: Proceedings of the 6th International Symposium on Distributed Autonomous Robotics Systems (DARS02), Fukuoka, Japan, 2002.

[13] A. Howard, M. Matadd, G. Sukhatme, An incremental self-deployment algorithm for mobile sensor networks, algorithm for mobile sensor networks. Autonomous Robots, Special Issue on Intelligent Embedded Systems.

[14] N. Heo, P. Varshney, An intelligent deployment and clustering algorithm for a distributed mobile sensor network, Systems, Man and Cybernetics, 2003. IEEE International Conference on, (2003) 4576 – 4581.

[15] W. Heinzelman, A. Chandrakasan, H. Balakrishnan, An application-specific protocol architecture for wireless microsensor networks, IEEE Transactions on Wireless Communications (2002) 660 – 670.

[16] M. Bhardwaj, A. Chandrakasan, Bounding the lifetime of sensor networks via optimal role assignments, in: Proceedings of the 21st Conference of the IEEE Communications Society (INFOCOM), New York, 2002.

[17] J.-J. Lee, B. Krishnamachari, C.-C. J. Kuo, Impact of heterogeneous deployment on lifetime sensing coverage in sensor networks, in: First IEEE International Conference on Sensor and Ad hoc Communications and Networks (SECON), Santa Clara, CA, 2004.

[18] Y. Hou, Y. Shi, H. Sherali, Rate aladdress in wireless sensor networks with network lifetime requirement, in: Proceedings of ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc), Tokyo, Japan, 2004.

[19] A. Bogdanov, E. Maneva, S. Riesenfeld, Power-aware base station positioning for sensor networks, in: Proceedings of the 23rd Conference of the IEEE Communications Society (INFOCOM), Hong Kong, 2004.

[20] E. I. Oyman, C. Ersoy, Multiple sink network design problem in large scale wireless sensor networks, in: Proceedings of the International Conference on Communications (ICC), Paris, France, 2004.

[21] J. Pan, Y. T. Hou, L. Cai, Y. Shi, S. X. Shen, Topology control for wireless sensor networks, in: MobiCom '03: Proceedings of the 9th annual international conference on Mobile computing and networking, ACM Press, New York, NY, USA, 2003, pp. 286–299.

[22] P. Mirchandani, R. Francis, Discrete Address Theory, Wiley, New York, 1990.

[23] Ilog inc. http://www.ilog.com.

[24] F. Glover, M. Laguna, Tabu search, In Handbook of Applied Optimization, P. M. Pardalos and M. G. C. Resende (Eds.) (2002) 194 – 208.

Appendix: Derivation of the optimization model

Constraints (3) and (4)

The energy of a sensor is used for sensing and relaying packets. If the device at vertex k is a sensor with lifetime L_k, we have

$$r_k e_1 x_k L_k + \sum_{i=1}^{n}\sum_{j=1}^{n}(\gamma_{ij}^{k} r_i z_{ij}) e_2 x_k L_k - B^{sensor} \leq 0, \forall k \quad (.1)$$

where the first and second terms in the above equation model energy consumption for, respectively, sensing and packet relaying. Note that the x_k term is used to ensure that the above inequality is active only when the device at vertex k is a sensor. Note also that the second term is only active when the sensor at vertex i uses micro-server at vertex j (indicated by $z_{ij} = 1$) and the transmission path from vertex i to vertex j includes vertex k (indicated by $\gamma_{ij}^{k} = 1$).

If the device in vertex k is a micro-server, its lifetime L_k obeys

$$\begin{aligned} & r_k E_1 L_k (1 - x_k) + \sum_{i=1}^{n}(r_i z_{ik}) E_2 (1 - x_k) L_k \\ & - B^{server} \leq 0, \forall k \end{aligned} \quad (.2)$$

Note that the $(1 - x_k)$ term is used to ensure that this inequality is active only when the device at vertex k is a micro-server.

By definition, $\lambda_k = \frac{1}{L_k}$, constraints (.1) and (.2) can be rewritten as:

$$r_k e_1 x_k + \sum_{i=1}^{n}\sum_{j=1}^{n}(\gamma_{ij}^{k} r_i z_{ij}) e_2 x_k - B^{sensor} \lambda_k \leq 0, \forall k \quad (.3)$$

$$r_k E_1 (1 - x_k) + \sum_{i=1}^{n}(r_i z_{ik}) E_2 (1 - x_k) - B^{server} \lambda_k \leq 0, \forall k \quad (.4)$$

Constraint (.3) is not linear. Consider $\gamma_{ij}^{k} z_{ij} x_k$ which is a factor in the second term of (.3). In Table .1, we compare the value of $\gamma_{ij}^{k} z_{ij} x_k$ against that of $\gamma_{ij}^{k} z_{ij}$ for all the 8 possible combinations of its constituent variables, we find that they only differ in row 7. However, this combination is excluded by constraint (8). Thus, we can replace constraint (.3) by (3).

Similarly, we use constraint (10) to remove the nonlinear term in constraint (.4) to obtain (4).

Table .1
The values of $\gamma_{ij}^k z_{ij} x_k$ and $\gamma_{ij}^k z_{ij}$. They have different values only at row 7.

γ_{ij}^k	z_{ij}	x_k	$\gamma_{ij}^k z_{ij} x_k$	$\gamma_{ij}^k z_{ij}$
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	0	0
1	0	0	0	0
1	0	1	0	0
1	1	0	0	1
1	1	1	1	1

Constraints (5, 6, 7, 15)

The requirement that a sensor uses the closest micro-server as its sink can be enforced by the inequality

$$d_{ij} z_{ij}(1 - x_k) \leq d_{ik}(1 - x_k), \forall i, j, k \tag{.5}$$

This ensures that a sensor at vertex i will only use the micro-server at vertex j if the hop count d_{ij} is less than the hop count to all other micro-servers. This constraint is nonlinear but can be linearized by defining $w_{ij}^k = z_{ij}(1 - x_k)$ and introducing the following additional constraints:

$$w_{ij}^k \leq z_{ij} \tag{.6}$$
$$w_{ij}^k \leq 1 - x_k \tag{.7}$$
$$w_{ij}^k \geq z_{ij} - x_k \tag{.8}$$

This shows how constraints (5, 6, 7, 15) are derived. Note that we do not need to include (.7) because it is implied by (.6) and (10) together.