Maximizing Lifetime of Sensor-Target Surveillance in Wireless Sensor Networks

Hai Liu, Xiaowen Chu, Yiu-Wing Leung
Computer Science, Hong Kong Baptist University

†Xiaohua Jia, ‡Peng-Jun Wan
†Computer Science, City University of Hong Kong
‡Computer Science, Illinois Institute of Technology

Abstract—**The paper addresses the maximal lifetime problem in sensor-target surveillance networks. Given a set of sensors and targets in an Euclidean plane, each sensor can watch all targets within its surveillance range and each target should be watched by at least one sensor at any time. The problem is to schedule the sensors to watch the targets and forward the sensed data to the base station, such that the lifetime of the surveillance network is maximized, where the lifetime is the duration that all targets are watched and all active sensors are connected to the base station. We propose an optimal solution to achieve the maximal lifetime. Our solution consists of three steps: 1) compute the maximal lifetime of the surveillance network and find a workload matrix and data flows by using the linear programming technique; 2) decompose the workload matrix into a sequence of schedule matrices by using the perfect matching technique; 3) determine the sensor-target surveillance trees based on the above obtained schedule matrices and data flows, which specify the active sensors and the routes to pass sensed data to the base station. The proposed optimal solution is illustrated by a numeric example.**

Keywords-maximum lifetime; sensor networks; scheduling;

I. INTRODUCTION

A sensor-target surveillance network normally consists of a set of sensor nodes (sensors for short), a set of targets and the base station (BS). Sensors are utilized to watch targets and collect sensed data to the BS. In manufacture and transportation, sensors are usually used to monitor temperature, humidity, or biomedical value of some hot spots of a region or a building/container. For example, sensors are employed to monitor the cargo containers which carry dangerous gas/liquid during the long journey of shipment or during the storage at a port. The sensors self-organize into a multi-hop network which is connected to the BS. The sensors periodically sample the air and forward the sensed data to the BS by multi-hop transmissions. Alarm messages will be sent once value of the reported data exceeds a predetermined threshold which implies leakage of the gas/liquid. In these applications, locations of targets are usually static and sensors are utilized to monitor the targets. Another example is to use noncontact infrared sensors to monitor temperature of rotating machines in the manufactory. Overheating of the machines will be detected and reported to the BS for further analysis and subsequence actions.

Since sensors are powered by batteries and have stringent power budget, they are required to coordinate with each other to monitor the targets in turn, and find energy-efficient routes to forward the sensed data to the BS. In the paper, we study the maximal lifetime problem in sensor-target surveillance networks. We assume that each sensor has an initial energy reserve, a fixed surveillance range and an adjustable transmission range which is limited by the maximum transmission range of the sensor. A sensor, e.g., acoustic, magnet and temperature sensor, can watch all the targets within the surveillance range, and each target should be watched by at least one sensor at any time. The problem is to schedule a subset of sensors to be active at a time to watch all the targets and find the routes for the active sensors to forward data to the BS, such that the lifetime of the entire surveillance network is maximized. The lifetime is the duration up to the time when there exists one target that can no longer be watched by any sensors or data cannot be forwarded to the BS any more due to the depletion of energy of the sensor nodes.

The rest of the paper is organized as follows. Section II is related work which states the difference between our work and existing solutions. The formal definition of the problem is given in section III. In section IV, we present our optimal solution which consists of three steps. A numeric example is presented in section V. We conclude our work in section VI.

II. RELATED WORK

There are three major techniques for maximizing the lifetime of wireless sensor networks: the use of energy efficient routing, the introduction of on/off modes for sensors, and the integrated solutions of the above two techniques.

Energy efficient routing: Extensive research has been done on energy efficient data gathering and information dissemination in sensor networks. Some well-known energy efficient protocols were developed, such as Directed Diffusion [7] and LEACH [5]. Recently, a clustering architecture to improve the lifetime of two-tiered sensor networks was studied in [1]. Both single-hop and multi-hop routing were considered and the problem was formulated as integer linear programming (ILP). An optimal flow routing algorithm for upper-tier aggregation and forwarding nodes in two-tier sensor networks was proposed in [6]. The maximum lifetime can be achieved in both cases that the flows are splittable and non-splittable. Construction of a data gathering tree to maximize the lifetime of sensor networks was proved to be NP-complete in [15]. The proposed approximation algorithm starts with an arbitrary tree and iteratively reduces the load on overloaded nodes. Constructing data gathering trees in both grid and general graphs was studied in [16]. Authors proposed a Minimal Steiner Tree based algorithm which provides a constant approximation ratio for grid graphs, and a randomized algorithm which guarantees a polylogarithmic performance bound for general graphs.

On/off scheduling: Another important technique used to prolong the lifetime of sensor networks is the introduction of

978-1-4244-4148-8/09/$25.00 ©2009
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2009 proceedings.

switch on/off modes for sensors. J. Carle et al did a good survey on energy efficient area monitoring of sensor networks [3], and pointed out that the best method for conserving energy is to turn off as many sensors as possible, while still keeping the system functioning. An analytical model was proposed in [2] to analyse network capacity and data delivery delay, against the sensor dynamics in on/off modes. Work in [14] is to determine on/off modes of sensors, such that the required amount of data is delivered to the BS and lifetime of the network is maximized. The proposed algorithm achieves 0.73 of the maximal lifetime. The on/off scheduling was studied in target (point) coverage in wireless sensor networks [4]. The problem is to find the maximum number of subsets of sensors (a sensor can appear in several subsets), such that each subset can sufficiently cover all targets in the region. However, the network aspect, i.e., how to forward the sensed data to the BS, was not considered in [4]. Moreover, the proposed heuristic in [4] is applicable to only homogeneous networks where sensors have uniform transmission/surveillance range and initial energy reserves, while the optimal solution proposed in this work can be applied to heterogeneous networks.

Integrated solutions: Most of existing studies addressed only one aspect, i.e., either energy efficient routing or on/off scheduling. We have proposed the integrated solutions which combine the both techniques to maximize the lifetime of sensor-target surveillance networks. The proposed integrated solutions are optimal in the scenarios where a target is required to be watched by one sensor [9] or *k* sensors [10].

Our contribution: All our prior work [9] [10] assumes that a sensor is able to watch only one target at a time, which greatly limits applications of the optimal solutions. In fact, most of current sensors, e.g., acoustic, magnet and temperature sensors, can watch/monitor all the targets as long as the distance between the sensor and the target is less than the sensor's surveillance range. This paper studies a general maximal lifetime problem where each sensor can watch all the targets within the surveillance range which matches real applications of sensor-target surveillance networks. We propose an optimal solution to the problem.

III. Problem Specification

We first introduce the following notations. Note that $S(i)$ may overlap with $S(j)$ for $i \neq j$, and $T(i)$ may overlap with $T(j)$ for $i \neq j$.

Tab. 1. Notations.

B, S, T	base station, set of sensors ($n=\|S\|$), set of targets ($m=\|T\|$).
$S(j)$	set of sensors that are able to watch target j, $j=1,\ldots,m$.
$T(i)$	set of targets that are within the surveillance range of sensor i, $i=1,\ldots,n$.
$N(i)$	set of neighbors of sensor i, $i=1,\ldots,n$.
E_i	initial energy reserve of sensor i, $i=1,\ldots,n$.
d_{ij}	distance between sensor i and j, $i, j=1,2,\ldots,n, B$.
R	data rate generated from sensors while watching targets.
e^S, e^T, e^R	energy required for sensing, transmitting, receiving one unit data.

We assume that a sensor is able to watch all targets within its surveillance range and each target should be watched by at least one sensor at any time. Each sensor has an initial energy reserve, a fixed surveillance range and an adjustable transmission range. In a transmission from s_i to s_j, s_i adjusts its transmission range to exactly reach s_j to save energy. Transmitting one unit data from s_i to s_j costs energy $e^T(d_{ij})^\alpha$, where α is the signal decline factor. We assume the positions of targets, sensors and the BS are static and are given in prior.

The Maximal Lifetime problem of Sensor-Target Surveillance (**MLSTS** for short) is, for given S and T, to schedule the sensors to watch the targets and route the sensed data to the BS, such that the lifetime of the surveillance network is maximized. The lifetime of the network is the length of time until there exists a target, say j, such that all sensors in $S(j)$ run out of their energy or the sensed data cannot be forwarded back to the BS due to the disconnection of the network.

Each sensor in the network is either in the active mode for sensing/forwarding data or in the sleep mode. The lifetime of the surveillance network can be divided into a sequence of sessions, such that each sensor sticks to the same mode within a session. In each session, a set of sensors are scheduled to watch the targets and forward sensed data to the BS. Other sensors that have no sensing/forwarding tasks in this session go to sleep to save energy. Some active sensors can sense and forward data simultaneously. In the next session, another set of sensors are scheduled to work in similar way. Some sensors may work continuously for multiple sessions. The solution of MLSTS is to determine these sessions, each of which specifies which sensor is scheduled to watch which target and how the sensed data is forwarded to the BS.

IV. Our Optimal Solution

The MLSTS problem can be solved in three steps. First, we compute the upper bound on the maximal lifetime of the network, a workload matrix and data flows of sensors. Second, we completely decompose the workload matrix into a sequence of schedule matrices to achieve the upper bound. Finally, we determine a sensor-target surveillance tree of each session which specifies the active sensors and the routes to forward data to the BS. We present our optimal solution step by step.

A. Find Maximal Lifetime

We use linear programming (LP) technique to find the maximal lifetime of the surveillance network. Let L denote the lifetime of the surveillance network. We introduce two variables:

x_{ij}: total time sensor i watching target j, $i \in S$, $j \in T$.

f_{ij}: amount of data transmitted from sensor i to sensor j (the receiver could be the BS).

The problem can be formulated as the following:

Objective: Max L

$$\text{s.t.} \quad \sum_{i \in S(j)} x_{ij} = L \qquad \forall j \in T; \qquad (1)$$

$$e^S R \sum_{j \in T(i)} x_{ij} + e^T \sum_{j \in N(i) \cup \{B\}} (d_{ij})^\alpha f_{ij} + e^R \sum_{j \in N(i)} f_{ji} \le E_i \qquad \forall i \in S; \qquad (2)$$

$$R \sum_{j \in T(i)} x_{ij} + \sum_{j \in N(i)} f_{ji} = \sum_{j \in N(i) \cup \{B\}} f_{ij} \qquad \forall i \in S; \qquad (3)$$

$$x_{ij} \ge 0, f_{ij} \ge 0 . \qquad (4)$$

Note that topology information that indicates which sensor is connected to which target/sensor is contained in $S(j)$, $T(i)$ and $N(i)$, i=1,…, n, j=1,…,m.

Eq. (1) specifies that for each target j in T, the total time that sensors watch it is equal to the lifetime of the network. That is, each target should be watched throughout the lifetime of the surveillance network. Ineq. (2) implies that the total energy cost of a sensor node shall not exceed its initial energy reserve. There are three components of energy cost of a sensor node: the cost for sensing data, the cost for transmitting data (which is dependent on the transmission distance), and the cost for receiving data. Eq. (3) is for flow conservation. It implies that for each sensor i in S, the total amount of data sensed and data received should be equal to the amount of data transmitted. Since all data flows are originated from targets and do not return to the targets, it will not lead to cycles in our solution. All data flows will eventually reach the BS.

The above formulation is a typical LP formulation, where x_{ij}, $1 \le i \le n$ and $1 \le j \le m$, and f_{ij}, $i,j=1,2,\ldots,n,B$, are real number variables and the objective is to maximize L. So the optimal results of x_{ij}, f_{ij}, and L can be computed in polynomial time.

However, L, obtained from the LP formulation (1)~(4), is the upper bound on the lifetime, and each x_{ij} specifies only the total time that sensor i should watch target j in order to achieve this upper bound L. Each f_{ij} specifies only the total amount of data transmitted from sensor i to sensor j or the BS. Our task is to find a schedule that specifies from what time up to what time which sensor watches which targets and through which route to pass the sensed data to the BS in each session. In the next two steps we will find the schedule and routes that will finally achieve the optimal lifetime L.

The values of x_{ij}, $1 \le i \le n$ and $1 \le j \le m$, obtained from the LP, can be represented as a matrix:

$$X_{n\times m} = \begin{bmatrix} x_{11}x_{12}\ldots x_{1m} \\ x_{21}x_{22}\ldots x_{2m} \\ \ldots\ldots \\ x_{n1}x_{n2}\ldots x_{nm} \end{bmatrix}_{n\times m} .$$

We call matrix $X_{n\times m}$ ***workload*** matrix, for it specifies the total length of time that a sensor watch a target. In the next step, we find the detailed schedule in each session for sensors watching targets based on the workload matrix.

B. Compute Schedule of Each Session

In each session of the surveillance lifetime, a set of sensors are scheduled to watch the targets. Since each sensor is able to watch all targets within its surveillance range, it is difficult to determine how many sensors are required in each session. Without loss of generality, we assume that each sensor can watch at most k, $k \ge 1$, targets within its surveillance range at a time. Note that this assumption is more general than that each sensor can watch all targets within its surveillance range which can be handled by setting k, $k \le m$, to the maximum node degree for watching targets.

Suppose there is no switching of watching targets in each session. That is, each target is continuously watched by the same sensor in the session. Schedule of each session can be represented as a matrix, where there are exactly one positive number in each column, representing each target should be watched by one sensor; and at most k positive numbers in each row, representing each sensor can watch at most k target at a time. The rest elements in the matrix are zeros. All the non-zero elements in the matrix have the same value, which is the time duration of this session. Now, our task becomes to decompose the workload matrix into a sequence of schedule matrices of sessions:

$$\begin{bmatrix} x_{11}x_{12}\ldots x_{1m} \\ x_{21}x_{22}\ldots x_{2m} \\ x_{31}x_{32}\ldots x_{3m} \\ \ldots\ldots \\ x_{n1}x_{n2}\ldots x_{nm} \end{bmatrix}_{n\times m} = \begin{bmatrix} 0c_1 0\ldots 0 \\ c_1 00\ldots 0 \\ 000\ldots c_1 \\ \ldots\ldots \\ 00c_1\ldots 0 \end{bmatrix} + \begin{bmatrix} 000\ldots 0 \\ 000\ldots. c_2 \\ 0c_2 0\ldots 0 \\ \ldots\ldots \\ c_2 0c_2\ldots 0 \end{bmatrix} + \ldots + \begin{bmatrix} c_t 00\ldots 0 \\ 00c_t\ldots c_t \\ 000\ldots 0 \\ \ldots\ldots \\ 0c_t 0\ldots 0 \end{bmatrix}$$
$$=P_1 + P_2 \ldots + P_q,$$

where c_i, i=1,2,…,q, is the length of time of session i, and q the total number of sessions. We call this sequence of session schedule matrices P_i, i=1,2,…,q, the ***schedule*** matrices. In schedule matrix P_i, all elements are either "0" or c_i, each column has exactly one non-zero element, and each row has at most k non-zero elements (it could be all "0", indicating the sensor has no sensing task in this session).

Since each sensor can watch at most k targets and all targets should be watched at any time, we have $kn \ge m$ (otherwise no feasible solutions). We first consider a simple case of $kn=m$, i.e., the number of targets is exactly k times of the number of sensors in the network, which implies that all sensors should work until the surveillance operation of the network terminates. Then, we extend the results to the general cases of $kn>m$.

1) A Simple Case $kn=m$

Let R_i, i=1,2,…,n, and C_j, j=1,2,…,m, denote the sum of elements in row i and the column j in the workload matrix, respectively. According to eq. (1), we have:

$$C_j = L, j=1,2,\ldots,m. \quad (5)$$

Since each sensor can watch at most k targets, we have:

$$R_i \le kL, i=1,2,\ldots,n. \quad (6)$$

Note that the sum of R_i equals to the sum of C_j. That is, $\sum_{i=1}^{n} R_i = \sum_{j=1}^{m} C_j = m\times L$. Since $kn=m$, we have:

$$\sum_{i=1}^{n} R_i = nkL . \quad (7)$$

Combining (6) and (7), we have:

$$R_i = kL, i=1,2,\ldots,n. \quad (8)$$

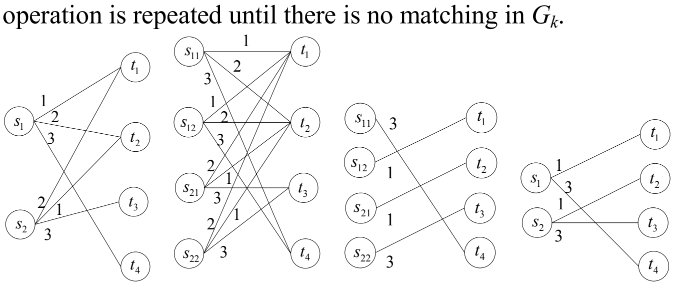
(5) and (8) give an important feature of the workload matrix when $kn=m$ that the sum of elements in each row is equal to kL and the sum of elements in each column is equal to L. This feature will guarantee the possibility of decomposing the workload matrix into schedule matrices in Theorem 1.

The basic idea of decomposing the workload matrix, $X_{n\times m}$, is to represent it as a bipartite graph $G(S \cup T, E)$, where one side are sensors $S=(s_1, s_2, \ldots, s_n)$ and the other are targets $T=(t_1, t_2, \ldots, t_m)$. For each non-zero element x_{ij} in $X_{n\times m}$, there is an edge from s_i to t_j and the weight of the edge is x_{ij}. Considering each schedule matrix of session i, each column has exactly one non-zero element that specifies each target should be watched by one sensor, and each row has exactly k non-zero elements which implies that each sensor should watch k targets to guarantee all targets are under surveillance. That is, k distinct

978-1-4244-4148-8/09/$25.00 ©2009
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2009 proceedings.

targets are watched by one sensor in each session, which can be represented as one sensor matching k targets in the bipartite graph G. Thus, the problem of finding a schedule matrix is equivalent to finding a ***k*-matchings** in G, i.e., one sensor matching k targets.

The k-matching algorithm proposed in [10] is applied to find a sequence of schedule matrices from the workload matrix. For completeness, we briefly introduce basic idea of the k-matching algorithm. It replaces each sensor node in G by k duplicate nodes. The links adjacent to the original sensor are adjacent to each of the duplicate nodes. In the new bipartite graph, denoted by G_k, one sensor (duplicate) matches exactly one target. Thus, the problem of finding a k-matching in G is transformed to the problem of finding a perfect matching in G_k. The perfect matching algorithm in [13] could be adopted and its time complexity is $O(\log|V|)$, where $|V|$ is the number of nodes in the bipartite graph. The k-matching algorithm works as follows. Each time G is converted to G_k, we find a perfect matching on G_k and merge the duplicate sensors to obtain a k-matching where one sensor matches exactly k targets. Each k-matching is corresponding to a schedule matrix. Let c_i be the smallest weight of edges in the k-matching. We update G by deducting c_i from the weight of the m edges in the k-matching and remove the edges whose weight becomes zero. This operation is repeated until there is no matching in G_k.



(a) G (b) G_2 (c) perfect matching in G_2 (d) 2-matching in G

Fig. 1 Decomposition of the workload matrix.

For example, suppose a workload matrix is $\begin{bmatrix} 1 & 2 & 0 & 3 \\ 2 & 1 & 3 & 0 \end{bmatrix}$ and k=2. The matrix is represented as a bipartite graph $G(S \cup T, E)$, where $S=\{s_1, s_2\}$ and $T=\{t_1, t_2, t_3, t_4\}$ (Fig. 1(a)). Then, targets s_1 and s_2 are replaced by duplicates s_{11}, s_{12} and s_{21}, s_{22}, respectively, resulting a new graph G_2 (Fig. 1(b)). A perfect matching is found in G_2 (Fig. 1(c)) and a 2-matching is obtained by merging the duplicates sensors (Fig. 1(d)). We obtain the schedule matrix $\begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$ based on the 2-matching in Fig. 1(d).

The follow theorems state that there exists a k-matching in every round of the decomposition and the number of decomposition rounds is bounded.

Theorem 1. In every round of decomposition, there exists a k-matching which is corresponding to the workload matrix computed from LP formulation (1)~(4).

Theorem 2. The number of decomposition rounds is bounded by the number of non-zero elements in $X_{n\times m}$.

Thus, the sensor-target watching schedule of each session can be computed when $kn=m$. In the next section, we will discuss the general cases of $kn>m$.

2) General Cases kn>m

To solve general cases $kn>m$, our basic idea is to transform the case to $kn=m$ by introducing some dummy targets into the network. That is to "fill" the workload matrix $X_{n\times m}$ with some dummy columns, such that the sum of elements in each column is equal to L and the sum of elements in each row is kL.

Let $Z_{n\times(kn-m)}$ be the dummy matrix, which has $(kn - m)$ columns. By appending the columns of the dummy matrix to the right hand side of $X_{n\times m}$, the resulting matrix, denoted by $W_{n\times kn}$, is in the form as:

$$W_{n\times kn}=\begin{bmatrix} x_{11}x_{12}\ldots x_{1m} & z_{11}z_{12}\ldots z_{1(kn-m)} \\ x_{21}x_{22}\ldots x_{2m} & z_{21}z_{22}\ldots z_{2(kn-m)} \\ \ldots\ldots & \ldots\ldots \\ x_{n1}x_{n2}\ldots x_{nm} & z_{n1}z_{n2}\ldots z_{n(kn-m)} \end{bmatrix}_{n\times kn}.$$

To make matrix $W_{n\times kn}$ have the features of (5) and (8), i.e., the sum of elements in each column is equal to L and the sum of elements in each row is kL, the dummy matrix $Z_{n\times(kn-m)}$ should satisfy the conditions:

- $\sum_{j=1}^{kn-m} z_{ij} = kL - R_i$, for $\forall$ i =1, 2, …, n. (9)
- $\sum_{i=1}^{n} z_{ij} = L$, for $\forall$ j = 1, 2, …, kn–m. (10)

The *FillMatrix* algorithm in [9] is applied to determine elements of $Z_{n\times(kn-m)}$. The algorithm is to greedily assign value to each element in $Z_{n\times(kn-m)}$ from top-left corner to bottom-right corner. Each time, it assigns the sum of the remaining undetermined elements of the row (or column), as much as possible, to the current element without violating conditions of (9) and (10). Let R_i^- and C_j^- record the sum of the remaining undetermined elements of row i and column j, respectively, for i=1,2,..,n and j=1,2,…,kn–m. Suppose we are going to determine z_{ij}, i.e., elements of z_{kl}, for k=1,…,i–1 and l=1,…,j–1, are already determined so far. If $C_j^- > R_i^-$, set $z_{ij}= R_i^-$ and the undetermined elements of row i are assigned to 0. If $R_i^- > C_j^-$, set $z_{ij}= C_j^-$ and the undetermined elements of column j are assigned to 0. If $R_i^- = C_j^-$, set $z_{ij}= R_i^-$ and all undetermined elements of row i and column j are assigned to 0. We can see that conditions (9) and (10) hold in the assignment process. Thus, general case $kn>m$ can be smoothly transformed to the case $kn=m$. The complete algorithm to decompose the workload matrix is as follows.

DecomposeMatrix Algorithm {
if $kn>m$
fill the matrix $X_{n\times m}$ to obtain $W_{n\times kn}$;
construct a bipartite graph G from $W_{n\times kn}$;
while there exist edges in G **do**
find k-Matching and corresponding schedule matrix P_i;
deduct P_i from $W_{n\times kn}$ and remove corresponding edges in G;
endwhile
Output $W_{n\times kn}=P_1+P_2+\ldots+P_q$;
}

978-1-4244-4148-8/09/$25.00 ©2009
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2009 proceedings.

Theorem 3. The time complexity of the *DecomposeMatrix* algorithm is $O(n\times m\times \log|2kn|)$.

Given a workload matrix $X_{n\times m}$, using the proposed algorithm, we can fill the matrix as $W_{n\times kn}$ and decompose $W_{n\times kn}$ into a sequence of schedule matrices:

$$W_{n\times kn} = P_1+P_2+\ldots+P_q. \quad (11)$$

Let $P_i^{'}$ denote the matrix which contains the first m columns in P_i (i.e., the information for the m valid targets by dropping the $kn–m$ dummy columns), i=1, 2, …, q. By removing the dummy columns in P_i, we have:

$$X_{n\times m}= P_1^{'} + P_2^{'} + ... + P_q^{'}. \quad (12)$$

The above discussions conclude that a workload matrix is decomposable to a sequence of schedule matrices such that each value of x_{ij}, $1 \le i \le n$, $1 \le j \le m$, can be actually met. In the next section, we will determine a sensor-target surveillance tree for each session, such that the maximal lifetime L can be finally achieved.

C. Determine Sensor-Target Surveillance Trees

We have obtained a sequence of schedule matrices. Each schedule matrix specifies the active sensors watching targets in the session. That is, the number of sessions is the number of schedule matrices. To allow the active sensors send their sensed data to the BS, we need to construct a sensor-target surveillance tree in which the root is the BS and all leaf nodes are the active sensors that are watching targets. Intermedial nodes of the tree are the sensors which forward data for others. The sensed data flow from active sensors to the BS along the tree.

From LP formulation (1)~(4) in section IV-A, we have obtained a data flow f_{ij} from any sensor i to sensor j, including the BS. To forward data to the BS, each sensor, say i, needs to follow its outgoing flow f_{ij} in order to achieve the maximal lifetime L. Suppose sensor i has l downstream nodes, denoted by s_1, s_2, …, s_l, to forward its data to the BS (i.e., f_{i1}, f_{i2}, …, f_{il} have non-zero values). Since there is no ordering of data flow f_{i1}, f_{i2}, …, f_{il}, we simply let sensor i pass its outgoing data first to s_1 until flow f_{i1} is saturated, then switch to s_2 until the value of f_{i2} is met, …, and finally it pass the last flow f_{il} to s_l. The outgoing data of sensor i include its own sensed data and the data it helps others to forward to the BS, as shown in the left hand side of eq. (3). By following the data flow obtained from the LP formulation, the optimal routes, in terms of energy efficiency, can be determined and thus the maximal lifetime L is achieved. The process will be illustrated by a concrete example in section V.

Theorem 4. The time complexity and space complexity of the proposed optimal solution are $O(n\times m)^{3.5}$ and $O(m^2n^4)$, respectively.

Note that computation of the optimal schedule is an *one-off* operation at the system initialization stage. Once the BS disseminates the schedule to all sensors, each sensor operates according to the schedule and no additional computation on the schedule is required. When the network starts operation, each sensor will watch targets, turn off to sleep, receive and forward data according to its schedule and the corresponding flows. Note that some sensors may work continuously for multiple sessions. There is no need to synchronize sensors to switch target watching at the end of session. Each sensor operates according to its own schedule *independently* from the others. A sensor can watch a set of targets continuously until it is time to switch to another set of targets or go to sleep.

The sensors perform their own schedule based on their local clocks which may drift away from each other from time to time. To ensure a target will be watched by another sensor continuously before the current one switches to other targets or goes to sleep, clocks of the sensors need to be synchronized. There are some clock synchronization protocols, e.g., [8] and [12], available for wireless sensor networks. When scheduling sensors to watch targets, the system can add a small buffer-period (in the order of milliseconds depending on the clock error) in the front and at the end of a working session to ensure that a target will be watched continuously at sensor switching. Note that compared with the duration of a working session the buffer-period is several orders of magnitude smaller.

V. A NUMERIC EXAMPLE

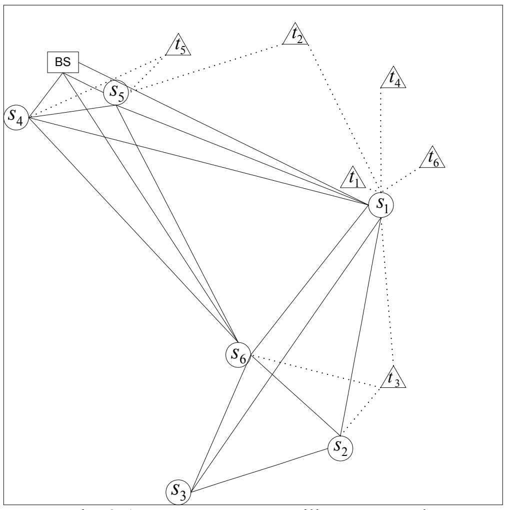


Fig. 2 A sensor-target surveillance network.

Tab. 2 Initial energy reserves of sensors.

s_1	s_2	s_3	s_4	s_5	s_6
58.638	34.914	39.74	82.662	44.434	17.364

We randomly place the BS, 6 sensors (circles) and 6 targets (triangles) in a 10×10 two-dimensional region as shown in Fig. 2. The surveillance range of sensors is set to 0.4×10, and the maximum transmission range is set to 0.8×10. There is a dashed edge between a sensor and a target if the target is within the surveillance range of the sensor. There is an arc from sensor s_i to s_j if s_j is within the maximum transmission range of s_i (in this example, the maximum transmission ranges for all sensors are uniform and thus arcs are replaced by solid edges). The initial energy reserves of sensors are random numbers generated in [0, 100] with the mean value 50, as shown in Tab. 2. To simulate the energy consumed on different tasks, we set e^T = 0.12, e^R = 0.1. These values are in proportional to the actual power consumption for transmitting and receiving data, respectively, as pointed out in [11]. Experiments in [11] further showed that energy cost of sensing data, such as monitoring temperature and humidity, is comparable to the energy cost of receiving data. Thus, we set $e^S = e^R = 0.1$ and the sensing data rate R = 1. The signal decline factor α is set to 2.

978-1-4244-4148-8/09/$25.00 ©2009
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2009 proceedings.

First, the linear programming, described in section IV-A, is utilized to compute the maximal lifetime L, in terms of hours, workload matrix $X_{6\times6}$ and data flows f_{ij} that achieve L: L=4.601441hrs,

$$X_{6\times6}=\begin{bmatrix} 4.601441 & 0 & 0.074291 & 4.601441 & 0 & 4.601441 \\ 0 & 0 & 4.3625 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.930141 & 0 \\ 0 & 4.601441 & 0 & 0 & 2.6713 & 0 \\ 0 & 0 & 0.16465 & 0 & 0 & 0 \end{bmatrix},$$

$$(f_{ij})_{6\times7}=\begin{bmatrix} 0 & 0 & 0 & 0 & 14.348814 & 0 & 0 \\ 0.26983 & 0 & 1.85107 & 0 & 0 & 2.2416 & 0 \\ 0.20037 & 0 & 0 & 0 & 0 & 1.6507 & 0 \\ 0 & 0 & 0 & 0 & 0.16465 & 0 & 13.9361514 \\ 0 & 0 & 0 & 12.259509 & 0 & 0 & 13.583646 \\ 0 & 0 & 0 & 0 & 4.05695 & 0 & 0 \end{bmatrix}.$$

Second, we run the *DecomposeMatrix* algorithm in section IV-B to decompose the workload matrix into 4 schedule matrices as below. k equals to 5 since s_1 can watch five targets in Fig. 2. That is, each sensor can watch at most 5 targets in each session. It is equivalent to that each sensor can watch all targets within its surveillance range. We can see that the whole lifetime of the network can be divided into 4 sessions, each of which lasts 0.074291hr, 1.6912hrs, 2.6713hrs and 0.16465hr, respectively.

$$X_{6\times6}=P_1+P_2+P_3+P_4=$$

$$\begin{bmatrix} 0.074291 & 0 & 0.074291 & 0.074291 & 0 & 0.074291 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.074291 & 0 \\ 0 & 0.074291 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}+\begin{bmatrix} 1.6912 & 0 & 0 & 1.6912 & 0 & 1.6912 \\ 0 & 0 & 1.6912 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.6912 & 0 \\ 0 & 1.6912 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$+\begin{bmatrix} 2.6713 & 0 & 0 & 2.6713 & 0 & 2.6713 \\ 0 & 0 & 2.6713 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2.6713 & 0 & 0 & 2.6713 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}+\begin{bmatrix} 0.16465 & 0 & 0 & 0.16465 & 0 & 0.16465 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.16465 & 0 \\ 0 & 0.16465 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.16465 & 0 & 0 & 0 \end{bmatrix}.$$

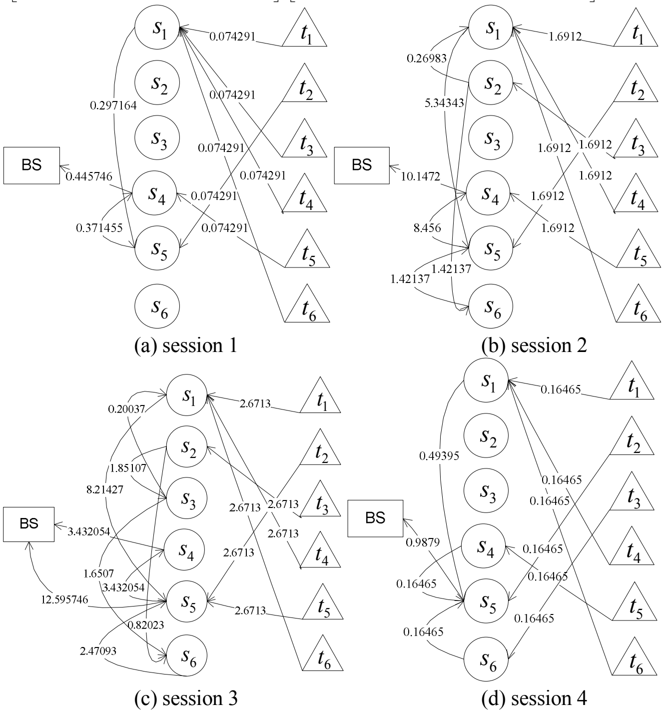


Fig. 3 Sensor-target surveillance trees of 4 sessions.

Finally, the sensor-target surveillance trees are determined based on the above schedule matrices and the data flows matrix $(f_{ij})_{6\times7}$. The surveillance trees of 4 sessions are shown in Fig. 3. We can see that the sensor-target surveillance trees satisfy the data flow constraints and the maximal lifetime L is achieved.

VI. CONCLUSION

We have proposed an optimal solution to maximize the lifetime of sensor-target surveillance networks. The proposed solution was illustrated by a numeric example.

ACKNOWLEDGMENT

This work is supported in part by grants from Research Grants Council of Hong Kong [Project No. CityU114307 and HKBU211009], CityU 7002359 and FRG/08-09/II-41. P.-J. Wan is supported in part by NSF under grant CNS-0831831.

REFERENCES

[1] A. Bari, A. Jaekel, S. Bandyopadhyay, "Clustering Strategies for Improving the Lifetime of Two-tiered Sensor Networks," *Computer Communications*, vol. 31, no. 14, pp. 3451-3459, 2008.

[2] C.F. Chiasserini, M Garetto, "Modeling the Performance of Wireless Sensor Networks," *INFOCOM'04*, 2004.

[3] J. Carle and D. Simplot-Ryl, "Energy-Efficient Area Monitoring for Sensor Networks," *IEEE Computer*, vol. 37, no. 2, pp. 40-46, 2004.

[4] M. Cardei, M. Thai, Y. Li, W. Wu, "Energy-efficient Target Coverage in Wireless Sensor Networks," *INFOCOM'05*, 2005.

[5] W. Heinzelman, A. Chandrakasan, and H. Balakrishna, "Energy-efficient Communication Protocol for Wireless Microsensor Networks," *HICSS'00*, Jan 2000.

[6] Y.T. Hou, Y. Shi, J. Pan, S.F. Midkiff, "Maximizing the Lifetime of Wireless Sensor Networks through Optimal Single-Session Flow Routing," *IEEE Transactions on Mobile Computing*, vol. 5, no. 9, pp. 1255-1266, 2006.

[7] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed Diffusion: a scalable and robust communication paradigm for sensor networks," *MOBICOM'00*, 2000.

[8] A. Krohn, M. Beiql, C. Decker, T. Riedel, "Syncob: Collaborative Time Synchronization in Wireless Sensor Networks," *INSS'07*, 2007.

[9] H. Liu, X. Jia, P. Wan, C.-W. Yi, S. Makki, and N. Pissinou, "Maximizing Lifetime of Sensor Surveillance Systems," *IEEE/ACM Transactions on Networking*, vol. 15, no. 2, pp. 334-345, 2007.

[10] H. Liu, P. Wan, and X. Jia, "Maximal Lifetime Scheduling for Sensor Surveillance Systems with *K* Sensors to 1 Target," *IEEE Transactions on Parallel & Distributed Systems*, vol. 17, no. 12, 2006.

[11] A. Savvides, C.C. Han, M. Srivastava, "Dynamic Fine-grained Localization in Ad-Hoc Networks of Sensors," *MOBICOM'01*, 2001.

[12] O. Simeone, U. Spagnolini, Y. Bar-Ness, S.H. Strogatz, "Distributed Synchronization in Wireless Networks," *IEEE Signal Processing Magazine*, pp. 81-97, Sept 2008.

[13] T. Uno, "A Fast Algorithm for Enumerating Bipartite Perfect Matchings," *LNCS 2223*, pp. 367-379, 2001.

[14] Y. Wu, S. Fahmy, N.B. Shroff, "Energy Efficient Sleep/Wake Scheduling for Multi-hop Sensor Networks: Non-convexity and Approximation Algorithm," *INFOCOM'07*, pp. 1568-1576, 2007.

[15] Y. Wu, S. Fahmy, N.B. Shroff, "On the Construction of a Maximum-Lifetime Data Gathering Tree in Sensor Networks: NP-Completeness and Approximation Algorithm," *INFOCOM'08*, 2008.

[16] Y. Yu, B. Krishnamachari, V.K. Prasanna, "Data Gathering with Tunable Compression in Sensor Networks," *IEEE Transactions on Parallel & Distributed Systems*, vol. 19, no. 2, pp. 276-287, 2008.