Maximizing lifetime of event-unobservable wireless sensor networks

Kemal Bicakci a,⁎, Hakan Gultekin b,1, Bulent Tavli c,2, Ibrahim Ethem Bagci a,3

⁎ Corresponding author. Tel.: +90 312 292 4262; fax: +90 312 292 4180.
E-mail addresses: bicakci@etu.edu.tr (K. Bicakci), hgultekin@etu.edu.tr (H. Gultekin), btavli@etu.edu.tr (B. Tavli), iebagci@etu.edu.tr (I.E. Bagci).
1 Tel.: +90 312 292 4275; fax: +90 312 292 4091.
2 Tel.: +90 312 292 4074; fax: +90 312 292 4180.
3 Tel.: +90 312 292 4000x5045; fax: +90 312 292 4180.

A R T I C L E   I N F O

Article history:
Received 30 April 2010
Received in revised form 18 January 2011
Accepted 18 January 2011
Available online 26 January 2011

Keywords:
Wireless sensor networks
Contextual privacy
Transactional confidentiality
Privacy
Location privacy
Event-unobservability
Linear Programming

A B S T R A C T

In wireless sensor networks (WSNs) contextual information such as the information regarding whether, when, and where the data is collected cannot be protected using only traditional measures (e.g., encryption). Contextual information can be protected against global eavesdroppers by periodic packet transmission combined with dummy traffic filtering at proxy nodes. In this paper, through a Linear Programming (LP) framework, we analyze lifetime limits of WSNs preserving event-unobservability with different proxy assignment methodologies. We show that to maximize the network lifetime data flow should pass through multiple proxies that are organized as a general directed graph rather than as a tree.

© 2011 Elsevier B.V. All rights reserved.

1. Introduction

In WSNs, sensor nodes convey their data to a base station possibly relaying through multiple intermediate sensor nodes [1]. Security and privacy in WSNs deployed in harsh and hostile environments are of paramount importance. Protecting integrity and confidentiality of sensor data content—a well-studied problem in the literature [2,3]—is generally achieved using cryptographic tools. On the other hand, contextual information such as whether, when, and where the data is collected cannot be secured only by traditional methods. This so called “event-unobservability” problem is usually addressed by inserting redundant packets into the network. By this way, attackers cannot distinguish real packets and, thus, obtain no useful information by traffic monitoring.

The amount of overhead to be carried in the network to safeguard against traffic analysis depends on the threat model (i.e., what capabilities attackers have). The possibility of the existence of a global eavesdropper who can monitor the entire network traffic brings a greater challenge for resource-constraint sensor nodes because a high-assurance solution should depend on a periodic collection method in which every sensor node periodically send encrypted packets regardless of whether there is real data to send or not [4]. Periodic transmission together with encryption hides the source of real packets against external attackers who do not hold the decryption key.

Battery power is a limited resource and communication is the dominant factor of energy dissipation in WSNs. Therefore, to increase network lifetime, there is a need for a more communication-efficient technique which reduces the overhead of the periodic collection scheme mentioned above. The basic idea of more advanced solutions referred in the literature as proxy-filtering techniques is that all sensor nodes again periodically generate packets but some of nodes assigned as proxies aggregate all incoming packets into a single packet [5].

Several proxy-filtering techniques were proposed in the past but their impact on the network lifetime has yet to be explored. In this study, through an LP framework, we make the first attempt in the literature to investigate the practical limits on the network lifetime of WSNs using proxy filtering techniques in idealized conditions. We investigate various different proxy assignment strategies and different deployment scenarios. Our investigation leads us to put forward a new filtering idea called OFS (Optimal Filtering Scheme) and enables us to maximize the network lifetime of wireless sensor networks which preserve event-unobservability against global eavesdroppers.

The rest of our paper is organized as follows: Section 2 provides the preliminaries for our paper. Section 3 introduces the LP formulation to investigate performance bounds of different proxy filtering techniques. Section 4 covers the experimental analysis we conduct to investigate various aspects of the problem. Section 5 discusses the results of this
2. Preliminaries

2.1. Linear Programming

Linear Programming (LP) is a method for finding the optimal value of a linear objective function under a set of linear constraints. The type of optimality is specified as either maximization or minimization. All variables in an LP problem take continuous and real values. The model of LP problems could be expressed generically as in Table 1 in which X represents the set of variables, C and B are vectors of coefficients and A is a matrix of coefficients.

LP is a powerful tool to solve optimization problems in WSNs. For instance LP can be used to obtain the amount of flow between the nodes that maximizes the network lifetime. A simple example illustrating the application of LP in finding the optimal flow distribution in a simple WSN composed of one base station and two sensor nodes is given in [6].

2.2. Proxy filtering

Protecting event-unobservability against global eavesdroppers is more challenging than dealing with local adversaries. To counteract this more challenging threat, all proposed solutions depend on inserting dummy traffic into the network to hide real event messages [4]. If all sensor nodes in the network periodically transmit packets, the traffic pattern in the network remains always the same and if the traffic in the network always has the same pattern, any traffic analysis technique can easily be defeated [7].

Periodically sending encrypted packets regardless of whether there is real data to send or not is a powerful but communication-inefficient technique to hide real event messages [4]. To prevent explosion of network traffic, Yang et al. [5] proposed two mechanisms named as PFS (Proxy-based Filtering Scheme) and TFS (Tree-based Filtering Scheme).

The basic idea of PFS is that a subset of sensors in the network selected as proxies collect and drop dummy messages before they reach the base station so that the problem of high communication cost of periodic packet transmission is mitigated. If one of the incoming packets to a proxy corresponds to a real event, proxy’s outgoing packet carries that information; otherwise the outgoing packet is a dummy one. In another view, a proxy acts like a base station, it collects packets from other sensors but since it does not have a direct connection to the outside world, the information it collects should be sent to the base station again in a periodic fashion to preserve privacy properties. Hence the data is hidden in a controlled fashion to an outsider so that any useful information cannot be extracted from the traffic patterns observed in the network data flow (i.e., a global eavesdropper cannot determine whether an event-triggered activity is happening in the network and cannot single out any particular node as the source node). According to Theorem 1 in [5], there is an information-theoretic guarantee that event-unobservability is preserved if PFS scheme is employed.

TFS, the second scheme proposed by Yang et al. [5], allows filtering at multiple proxies. In TFS, proxies form a tree rooted at the base station with each proxy having a parent node and possibly multiple child nodes. Parent nodes aggregate traffic originated by child nodes and child leaf nodes aggregate data coming from ordinary sensors.

2.3. Proxy selection

Maximum achievable lifetime using proxy-based schemes depends both on the number of proxies as well as location of these proxies. Determining proxies (i.e. selecting P elements out of V nodes) is an NP-hard problem as discussed in [5]. Therefore, we adapt heuristics based on localized search to efficiently solve the proxy selection problem. Algorithm 1 formally describes this heuristics. We note that different types of definitions are possible for the lifetime in Algorithm 1. See Section 3.1 for the specific definition we use in our LP framework.

Algorithm 1. Proxy Selection Algorithm (PSA)

Input: A network topology with node set V and the desired number of proxies k.
Output: A set of proxies and the corresponding lifetime.
1: Initialize \( P = \emptyset \), \( \text{Lifetime}_b = 0 \), \( \text{Lifetime}_p = \infty \), \( PB = \emptyset \) and \( NB = \emptyset \)
2: \( \text{for } i = 1 \text{ to } k \text{ do} \)
3: \( \text{Select randomly node } j \in V \)
4: \( \text{Let } P = P \cup \{j\} \text{ and } P = P \cup \{j\} \)
5: \( \text{end for} \)
6: Solve PFS using sets P and \( \mathcal{P} \), calculate lifetime;
7: Update \( \text{Lifetime}_b = \text{Lifetime}_p \)
8: \( \text{for } (\forall i \in V) \text{ and } (\forall j \in P) \text{ do} \)
9: \( \text{Set } \mathcal{P} = \{P \cup \{j\}\} \text{ and } \mathcal{P} = \{P \cup \{j\}\} \)
10: Solve PFS using sets P and \( \mathcal{P} \), calculate lifetime;
11: \( \text{if } \text{Lifetime}_p > \text{Lifetime}_b \text{ then} \)
12: \( \text{Set } \text{Lifetime}_p = \text{Lifetime}_p, \text{PB = } \{i\} \text{ and } NB = \{j\} \)
13: \( \text{end if} \)
14: \( \text{end for} \)
15: \( \text{if } \text{Lifetime}_p > \text{Lifetime}_b \text{ then} \)
16: \( \text{Set } P = \{P \cup \text{PB}\} \text{ and } \mathcal{P} = V \setminus P \)
17: \( \text{Go to } 6 \)
18: else
19: \( \text{STOP} \)
20: \( \text{end if} \)

2.4. Threat model

In this paper we adopt the threat model given in [5]. Specifically, we assume that attackers are external, passive and global. More precisely stated, attackers cannot obtain encryption/decryption keys or control sensor nodes via some other means. They cannot also insert or block packets or trigger network events. However attackers can listen to all communication in the network, analyze the collected data and try to determine the location of each sensor node.

3. LP models for proxy filtering techniques

3.1. Network model

In our network model, each node periodically generates the same amount of data (amount of data generated by node-i in time t is \( s_i \)). The optimization problem in its general form is formulated as maximizing t (the minimum lifetime of the sensor nodes). We adopt the network lifetime definition given in [8–10] which is the time when the first sensor node exhausts all its battery power.

Let us define U as the set of all sensors including the base station. We will denote the set of all sensors except the base station with V and the set of all proxies with P. Note that, V and P are subsets of U. First constraint of the problem states that all flows in the network are non-negative (\( f_{ij} \) denotes the direct flow from node-i to node-j).

\[
f_{ij} \geq 0 \quad \forall i, j \in U
\]  

(1)

Second constraint is the flow balancing constraint when proxy nodes are not employed and thus no filtering is performed. Amount of
data flowing out of node-i is equal to data flow generated by node-i plus data flowing into node-i. This should hold for all nodes except the base station (Access Point—AP).

\[ \sum_{j \in U} f_j = s_t + \sum_{j \in V} f_j \quad \forall i \in V \quad (2) \]

Third constraint is the energy constraint. The cost associated with processing is ignored in our model hence each node consumes its battery having \( e_i \) amount of energy only for reception and transmission of flows. Energy consumed per unit amount (per bit) of flow transmission, \( E_{tx,ij} \), increases by \( \alpha \)-power of the inter-node distance (\( d_{ij} \)) but energy spent to receive unit amount (per bit) of flow, \( E_{rx} \), is constant (\( E_{rx} = E_{elec} \)). \( E_{tx,ij} = E_{elec} + \epsilon_{amp} d_{ij}^\alpha \) \[11]. Here, \( E_{elec} \) is the electronics energy; \( \epsilon_{amp} \) is the amplifier energy and \( \alpha \) is the path loss exponent.

\[ \left\{ \sum_{j \in V} E_{tx,ij} f_j \right\} + \left\{ \sum_{j \in U} E_{rx,j} f_j \right\} \leq e_i \quad \forall i \in V \quad (3) \]

Eqs. (1)–(3) define our model for the baseline scheme where no proxies are employed. If some of the nodes are assigned as proxies, flow balancing constraint—Eq. (2)—is no longer valid and should be revised as in Eq. (4). Energy constraint should also be changed accordingly as in Eq. (5).

\[ \sum_{j \in U} f_j = s_t + \sum_{j \in V} f_j \quad \forall i \in (V - P) \]

\[ \sum_{j \in V} g_j = \sum_{j \in V} g_j \quad \forall i \in (V - P) \quad (4) \]

\[ \sum_{j \in U} g_j = s_t + \sum_{j \in V} g_j \quad \forall i \in P \]

\[ \left\{ \sum_{j \in U} E_{tx,ij} (f_j + g_j) \right\} + \left\{ \sum_{j \in V} E_{tx,j} (f_j + g_j) \right\} \leq e_i \quad \forall i \in V \quad (5) \]

Similar to PFS scheme in [5],\(^4\) the constraint above—Eq. (4)—models filtering at only one proxy. Here, \( f \)-flows are generated by ordinary sensor nodes and they terminate either at the base station or at one of the proxies whereas \( g \)-flows are generated by proxies and cannot be filtered at other nodes including other proxies on their way to the base station. Note that ordinary sensor nodes can also relay flows originated from proxies (\( g \)-flows). We need different labels for \( f \)-flows and \( g \)-flows in Eq. (4) since flow balancing constraint should hold for ordinary sensor flows and proxy flows separately.

It is also possible to model TFS by extending the model for PFS. For instance with a two-level hierarchy, we can separate set \( P \) into two disjoint sets \( (P1 \cup P2 = P \text{ and } P1 \cap P2 = \phi) \) and the proxies in \( P1 \) and \( P2 \) (corresponding to level-1 and level-2 proxies, respectively) can generate two different types of flows (\( g \)-flows and \( h \)-flows, respectively). Eqs. (6)–(12) are used to model TFS with two-level hierarchy.

\[ \sum_{j \in U} f_j = s_t + \sum_{j \in V} f_j \quad \forall i \in (V - (P1 \cup P2)) \quad (6) \]

\[ \sum_{j \in U} g_j = \sum_{j \in V} g_j \quad \forall i \in P1 \quad (7) \]

\[ \sum_{j \in V} h_j = s_t + \sum_{j \in V} h_j \quad \forall i \in (V - (P1 \cup P2)) \quad (8) \]

\[ \sum_{j \in U} h_j = \sum_{j \in V} h_j \quad \forall i \in P2 \quad (9) \]

\[ \left\{ \sum_{j \in U} E_{tx,ij} (f_j + g_j + h_j) \right\} + \left\{ \sum_{j \in V} E_{tx,j} (f_j + g_j + h_j) \right\} \leq e_i \quad \forall i \in V \quad (10) \]

\[ \left\{ \sum_{j \in U} E_{tx,ij} (f_j + g_j + h_j) \right\} + \left\{ \sum_{j \in V} E_{tx,j} (f_j + g_j + h_j) \right\} \leq e_i \quad \forall i \in V \quad (12) \]

Eq. (6) states that all nodes except proxies and base station generate \( f \)-flows and these flows are conserved on these sensors (incoming flow plus self produced data must be equal to the outgoing flow). Eq. (7) states that \( f \)-flows can terminate at the base station or at a level-1 proxy but level-2 proxies should relay the \( f \)-flows. Eqs. (8) and (10) state that \( g \)-flows and \( h \)-flows are only generated by first level proxies and second level proxies, respectively. Eqs. (9) and (11) show that \( g \)-flows are relayed by ordinary sensor nodes and second-level proxies and \( h \)-flows are relayed by ordinary sensor nodes and first-level proxies. Eq. (12) is used to model the energy constraint when three distinct types of flows are of concern. Fig. 1 is helpful in understanding the operating policy of the TFS scheme. For instance, in a level-1 proxy denoted by \( P1 \) in Fig. 1(a) \( f \)-flows are terminated, \( g \)-flows and \( h \)-flows are conserved and self produced \( s \)-flows are also added to the outgoing flows in terms of \( g \)-flows. Similarly, \( f \)-flows, \( g \)-flows, and \( h \)-flows are conserved by an ordinary sensor and the amount of \( f \)-flows is increased by the addition of self produced \( s \)-flows.

We argue that the hierarchy imposed by TFS is restrictive since each proxy can perform a single type of aggregation. Hence, neither PFS nor TFS gives optimal results with respect to network lifetime. Eq. (13) defines the flow balancing constraint when proxies can be organized as a general directed graph rather than as a tree. In other words, each proxy can now filter packets from every other proxy as well as from ordinary sensor nodes. We call this strategy as Optimal Filtering Scheme (OFS) since the proof of Theorem 1 shows that this filtering strategy achieves optimality in terms of lifetime. The generalization also leads a simplification in the flow balancing constraint which can be expressed using a single type of flow as shown in Eq. (13). Fig. 1(b) illustrates the filtering made under this scheme. In Fig. 1(b) there is no

\(^4\) To be precise, this model is for a slightly different version of PFS since in the original PFS scheme, each sensor is assigned to a fixed default proxy.
flow type other than f-flows which terminate at the OFS proxy denoted by \( P \) and self produced \( s_j \)-flows are added to the outgoing flows of \( P \) in terms of f-flows.

\[
\sum_{j \in S} f_{ij} = s_i t + \sum_{j \notin S} f_{ij} \quad \forall i \in (V - P)
\]

\[
\sum_{j \in P} f_{ij} = s_i t \quad \forall i \in P
\]

(13)

As Theorem 1 shows, OFS scheme offers the longest lifetime when each sensor node acts as a proxy (i.e., the sum of incoming flows does not put a constraint in the flow balancing). Thus, an upper bound for network lifetime can easily be derived using Eq. (14).

\[
\sum_{j \in S} f_{ij} = s_i t \quad \forall i \in V
\]

(14)

Before proceeding with the theorem we first need to distinguish different instances of the problem. An instance of the problem is uniquely defined by the parameters; \( t_{\text{fl}}, t_{\text{PFS}}, t_{\text{OFS}} \), and \( s_i \). Let \( t_{\text{fl}} \), \( t_{\text{PFS}} \), and \( t_{\text{OFS}} \) denote the network lifetime of a WSN under the TFS, PFS and OFS schemes, respectively. Additionally, let \( t_{\text{fl}}^{\text{TFS}}, t_{\text{fl}}^{\text{PFS}} \), and \( t_{\text{fl}}^{\text{OFS}} \) denote the corresponding optimal values.

**Theorem 1.** For a given instance of the problem, the lifetime of the network is maximized by the OFS scheme. That is, \( t_{\text{fl}}^{\text{OFS}} \geq t_{\text{fl}} \) for all other available filtering schemes. Furthermore, \( t_{\text{fl}}^{\text{OFS}} \geq t_{\text{fl}}^{\text{TFS}} \geq t_{\text{fl}}^{\text{PFS}} \) for the same problem instance.

**Proof.** Note that \( s_i \) is one of the parameters that define an instance of the problem. These parameters simply state that the amount of data generated by node \( i \) in time \( t \) is \( s_i t \). This means that the minimum amount of data generated and must be transferred to the base station at time \( t \) in any WSN is \( \sum_{i} s_i t \). However, if some incoming flow is conserved by any one of the nodes in addition to the self produced flow, this amount of data that must be transferred to the base station will be greater. From the energy constraints given in Eqs. (3), (5), and (12), we know that as the amount of flow increases in a WSN, the network lifetime will never be increased. It may reduce or stay the same. Since there are no constraints on the number of proxies under the OFS scheme all sensor nodes can be selected as a proxy. In such a case, no inflow is conserved. Hence, the resulting network lifetime after determining the optimal routing of flows will be identical to the upper bound of the lifetime. This proves the optimality of \( t_{\text{fl}}^{\text{OFS}} \).

Now let us prove the second statement of the theorem. We can separate the constraints of the mathematical formulation into two: One is the set of flow balance constraints and the other one is the set of energy constraints. Flow balance constraints determine the operating rules of the WSN like TFS or OFS schemes, whereas the energy constraints determine the network lifetime. Any solution for the PFS scheme is feasible (satisfy the energy constraints) for the TFS scheme and any solution for the TFS scheme is feasible for the OFS scheme. However, the reverse statement is not valid in most of the cases. This statement is a direct consequence of the reduction in the amount of flow when proxies are used. Then, we can claim that if any solution for the PFS is feasible for the TFS, then the optimal solution is also feasible with the same network lifetime. However, there may be some other solutions for the TFS scheme, since some of the flows are filtered under this scheme, with higher lifetime values. This is not possible for the PFS scheme; otherwise this contradicts with the fact that the current solution is optimal. A similar argument can be done between the TFS and OFS schemes. Hence, \( t_{\text{fl}}^{\text{OFS}} \geq t_{\text{fl}}^{\text{TFS}} \geq t_{\text{fl}}^{\text{PFS}} \).

We note that the optimality of OFS scheme is valid if delays due to proxy filtering are not taken into consideration. We investigate this issue in Section 3.3.

Here the resulting formulation for OFS expressed with Eqs. (1), (3), and (14) is an LP formulation which can be solved efficiently by commercial solvers like CPLEX. However, this formulation can only be used to obtain an upper bound on the lifetime. This is because; the formulation presented above may lead to loops of flows that do not reach to the base station. An example may be helpful in understanding: as a solution to the above formulation one may get \( f_{ij}, f_{jk}, f_{ki} > 0 \). This situation is very similar to the subtours in the well known problem of Travelling Salesman Problem (TSP). The only constraint that makes the TSP an NP-Hard problem is the exponential number of subtour elimination constraints (the interested reader can find a detailed analysis of TSP problems in [12]). In our case, one can eliminate such loops by defining new binary variables \( x_{ij} \) to show if node \( i \) sends a flow to node \( j \). Let \( S \) be a proper subset of \( V \). Then we need to insert the following constraints to the formulation above:

\[
\sum_{i} \sum_{j \in S} f_{ij} \geq 1, \quad \text{for all proper subsets } S, |V| / 2 \geq |S| \geq 2
\]

\[
f_{ij} \leq x_{ij} M, \quad \forall i \in V \text{ and } \forall j \in U
\]

\[
x_{ij} \in \{0, 1\}, \quad \forall i \in V \text{ and } \forall j \in U.
\]

In this formulation \( M \) denotes a very large number used to guarantee that the flow amount between nodes \( i \) and \( j \) is not constrained, a frequently used method in Integer Programming. After inserting these constraints the resulting formulation becomes a Mixed Integer Program (MIP) which is harder than Linear Programs in most of the cases. Additionally, the first constraint above must be written for all proper subsets of \( V \). Hence, the number of constraints increases exponentially as the number of sensors in the system increases. These properties bring a great complexity for the solution of the problem and increases the computation time drastically. One way to overcome this is to use an iterative algorithm which starts with the LP version of the problem. If there are no loops in the solution, then this solution is optimal otherwise only the constraints that will prevent the loops formed in the solution of the LP are inserted and the resulting formulation is solved once again. This procedure is repeated until a solution with no loops is obtained. This is the most common method used in solving the TSPs. Although this methodology helps dealing with exponential number of constraints it may not reduce the computation time. Such an analysis is beyond the scope of the current

Fig. 1. In and outflows from the sensors and the proxies under TFS and OFS schemes. Disks without labeling denote ordinary sensor nodes. Disk labeled with \( P \) denotes OFS proxy. Disks labeled with \( P_1 \) and \( P_2 \) denote level 1 TFS proxy and level 2 TFS proxy, respectively.
study and finding an upper bound is enough for the purposes of our work.

3.2. Proxy selection

The PSA algorithm presented as Algorithm 1 in Section 2.3 determines a set of proxies that maximizes the lifetime, given the number of proxies. In order to determine the optimal number of proxies, we need to run the PSA algorithm for all alternatives \((k = 1, 2, ..., |V|)\) and select the one that maximizes the lifetime. PSA algorithm starts with an initial selection of proxies which can be a random selection. Then all the nodes in the proxy set are swapped with all the nodes that are not proxy one by one and the lifetime values are evaluated for each swap. The swap operation that maximizes the lifetime most is accepted and the proxy set is updated. This procedure is continued until we find a set of proxies for which we cannot improve the lifetime by a swap operation. Note that, this procedure has no guarantee of providing the optimal solution. However, we conducted a computational study in which we compared the lifetime corresponding to our localized search algorithm with the lifetime obtained with brute force method and observed that proxies selected by the localized search algorithm provided the optimal results in all cases. Hence, we can conclude that PSA algorithm is very efficient in terms of both the solution quality and computational time and can be used as a solution procedure in our further analysis.

3.3. Delay analysis

Due to periodic collection property, once a real event is sensed, its information cannot be delivered by sensor nodes immediately. If sensor nodes generate data with a period of \(T\), then the average delay due to periodic collection is \(T/2\). Furthermore, lifetime improvement using proxy nodes is not always free because collecting packets from sensor nodes and delaying them until the next proxy transmission time may bring extra requirements or present additional delays.

The amount of additional delay due to proxy filtering depends on the level of synchronization that exists between sensor nodes and proxies. Broadly speaking, there are two opposing cases here:

1. Perfect Synchronization: Proxy node transmits its periodic packet immediately after receiving all incoming packets for that period. The sensor nodes are synchronized thus all transmissions happen at the same time (we assume queuing delays, delays due to encryption/decryption and packet collisions can be ignored and there is no bandwidth limitation).

2. No Synchronization: If the transmission time of the proxy is set without considering when sensor nodes send their data and if the transmission period of proxies is also equal to \(T\), then the average additional delay due to proxy filtering is \(T/2\) per proxy. In this case, the delay associated with proxy nodes increases by number of filtering levels.

4. Experimental analysis

This section is devoted to the presentation of the methodology and the parameters used in our experimental analysis. In order to compare different filtering schemes with each other and observe the effects of different parameters on the results we conducted an extensive computational study. Environmental parameters used in this study are presented in Table 2. Since there are no closed form solutions for LP problems [6], we use GAMS optimization tool [14] together with CPLEX 9 solver ([14]) to obtain results numerically. CPLEX is one of the most widely used solvers for Linear Programs and guarantees optimality of the provided solutions [15]. To illustrate the logic behind our LP formulation, we first consider a simple linear topology with a base station and five nodes. Fig. 2 illustrates such a network and how flows are distributed between nodes when different LP models are in use. The distance between adjacent nodes is fixed and chosen as 95 m. Since we are interested only in lifetime ratios, battery energy and transmission period do not have an impact on our results. Path loss exponent is taken as 2 and energy parameters are chosen as \(E_{\text{elec}} = 50 \mu J\) and \(E_{\text{amp}} = 100 \mu J\) [8,11]. Flow values in Fig. 2 are calculated by solving the LP with constraints defined by Eqs. (1), (2), and (3) given in Section 3 for the baseline scheme and with constraints defined by Eqs. (1), (4), and (5) for PFS. Constraints defined by Eqs. (1) and (6)-(12) are used for two-level TFS and constraints defined by Eqs. (1), (3), and (14) are used for OFS. The lifetime of baseline scheme is normalized to unity and relative lifetime ratios are calculated as ratios of lifetime values obtained with proxy-based schemes to the lifetime value in the baseline scheme.

For PFS, we find the optimal number of proxies by running Algorithm 1 and we see that lifetime values with one proxy and two proxies are exactly the same. In Fig. 2, the result with two proxies is illustrated. For TFS, after executing a tailored form of Algorithm 1 we see that network lifetime is maximized when there is exactly one first-level proxy and one second-level proxy as illustrated in the figure. For OFS, lifetime values when all nodes are proxy and when all nodes except the farthest node to the base station are proxy are exactly the same. The latter option is illustrated in the figure. Note that not only determining the number of proxies but also selection of these proxies among sensor nodes is performed using Algorithm 1.

As a result of this experimental study, we observe that PFS performs better than the baseline scheme and lifetime improvement is 127%. We also see that the network lifetime with OFS scheme is 86% more than the lifetime obtained with PFS and 68% more than the lifetime obtained with TFS with two-level filtering. The longest lifetime is achieved when OFS is preferred; therefore, for the sake of simplicity from now on we exclude TFS in our analysis and concentrate on the other three models.

The results in Fig. 2 are obtained when each node has an infinite transmission range. As a second study (results are presented in Fig. 3) we investigate the change in lifetime ratios in the same topology when transmission range of each node cannot exceed a certain limit. More formally this can be expressed as follows:

\[
f_{ij} = 0, \quad g_{ij} = 0 \quad \forall i \in V, \forall j \in U : \left| d_{ij} \right| > d_{\text{max}}
\]  

(15)
As seen in Fig. 3, the gain with PFS and OFS slightly increases as transmission range decreases.

In the third experimental setting, we use an 80-node disk topology with an area of $25,920 \pi = \approx 81,400$ m$^2$. Base station is deployed at the center and the other nodes are randomly distributed across the disk. Energy parameters are again chosen as $E_{\text{elec}} = 50$ nJ and $\varepsilon_{\text{amp}} = 100$ pJ [8]. In order to minimize the effects of randomness, we take 10 replications for each parameter setting and obtain the averages. In these runs, we use Algorithm 1 introduced in Section 2.3 to determine the number of proxies and to select proxies among sensor nodes.

Our purpose here is threefold. First, to make a comparison between PFS and OFS schemes with respect to network lifetime improvement they can offer in a disk topology. Second purpose is to investigate the effect of number of proxies on the network lifetime. Third, we want to evaluate the influence of path loss exponent value to achievable normalized lifetimes.

In Fig. 4 relative lifetime ratios show the ratio of lifetime values obtained with proxy-based schemes to the lifetime value in the baseline scheme without proxy. Note that the lifetime obtained with the baseline scheme without proxy for $\alpha = 2$ is 1516 times the lifetime obtained with $\alpha = 4$. We observe that normalized lifetimes obtained by OFS are almost the same as the lifetimes with PFS for small number of proxies. Other major results of this experimental setting are itemized as follows:

1. In PFS scheme, there is a threshold value for number of proxies such that after this threshold normalized lifetime starts to decrease. On its limit when each sensor becomes a proxy, the optimization problem reduces to the baseline scheme and there is not any gain in using proxies. On the other hand, lifetime with OFS scheme shows a non-decreasing behavior although a fast saturation is observed and improvement becomes non-significant as number of proxies increases (lifetime in all-proxy case is only 2.45% larger than lifetime if only 10 out of 80 nodes are chosen as proxy when $\alpha = 2$).
2. Optimal number of proxies in PFS depends on the path loss exponent (18 for $\alpha = 2$ and 8 for $\alpha = 4$).

As seen in Fig. 3, the gain with PFS and OFS slightly increases as transmission range decreases.

In the third experimental setting, we use an 80-node disk topology with an area of 25,920 m$^2$ ($\approx 81,400$ m$^2$). Base station is deployed at the center and the other nodes are randomly distributed across the disk. Energy parameters are again chosen as $E_{\text{elec}} = 50$ nJ and $\varepsilon_{\text{amp}} = 100$ pJ [8]. In order to minimize the effects of randomness, we take 10 replications for each parameter setting and obtain the averages. In these runs, we use Algorithm 1 introduced in Section 2.3 to determine the number of proxies and to select proxies among sensor nodes.

Our purpose here is threefold. First, to make a comparison between PFS and OFS schemes with respect to network lifetime improvement they can offer in a disk topology. Second purpose is to investigate the effect of number of proxies on the network lifetime. Third, we want to evaluate the influence of path loss exponent value to achievable normalized lifetimes.

In Fig. 4 relative lifetime ratios show the ratio of lifetime values obtained with proxy-based schemes to the lifetime value in the baseline scheme without proxy. Note that the lifetime obtained with the baseline scheme without proxy for $\alpha = 2$ is 1516 times the lifetime obtained with $\alpha = 4$. We observe that normalized lifetimes obtained by OFS are almost the same as the lifetimes with PFS for small number of proxies. Other major results of this experimental setting are itemized as follows:

1. In PFS scheme, there is a threshold value for number of proxies such that after this threshold normalized lifetime starts to decrease. On its limit when each sensor becomes a proxy, the optimization problem reduces to the baseline scheme and there is not any gain in using proxies. On the other hand, lifetime with OFS scheme shows a non-decreasing behavior although a fast saturation is observed and improvement becomes non-significant as number of proxies increases (lifetime in all-proxy case is only 2.45% larger than lifetime if only 10 out of 80 nodes are chosen as proxy when $\alpha = 2$).
2. Optimal number of proxies in PFS depends on the path loss exponent (18 for $\alpha = 2$ and 8 for $\alpha = 4$).
3. Lifetime improvement achievable with filtering is more significant in environments having a high path loss exponent which is more typical in indoor environments.

In another experimental setting, in order to investigate the effects of node density to the gain obtainable in using PFS and OFS, we change the size of disk area while keeping the parameters and number of nodes the same. Fig. 5 reveals that in high density networks, there is no gain in using either OFS or PFS.

In our final experimental setting, we investigate the effects of number of nodes on the lifetime gain that can be obtained. For this purpose, we use a linear topology and increase the number of nodes of the one-dimensional network while keeping the same inter-node distances. We prefer one-dimensional topology in this case because the algorithmic complexity of Algorithm 1 we use to select optimum proxy locations puts a limit on number of nodes in a two-dimensional topology. Fig. 6 reveals that as the number of nodes and average distance between the ordinary nodes and the base station increase, there is no significant increase in the lifetime gain using the PFS scheme. On the other hand, the relative lifetime obtained with OFS scheme increases linearly with the increasing number of nodes in the network.

5. Discussion

Having conducted the experimental analysis and presented the results in the previous section, in this section we discuss the results. The lifetime obtained with OFS is superior to the lifetimes obtained with other schemes because OFS carries the least amount of data within the network without sacrificing the event-unobservability criteria. Note that PFS and TFS also preserve the event-unobservability criteria, however, they achieve the goal of unobservability by transporting more than enough data hence they have comparatively lower lifetimes.

Theorem 1 proved that the longest lifetime is achieved when OFS is preferred and the results of the first experimental setting illustrated in Fig. 2 are consistent with this theorem.

We note that lifetime gain relative to baseline scheme increases as maximum transmission range requirements become more stringent. The main reason for such a behavior arises from the fact that the baseline scheme distributes flows in parts and a small percentage of the data is transferred directly to the base station. However, the limitations on the maximum transmission ranges prevent the baseline scheme to use far reaching links, which results in a reduction in the lifetime. On the other hand, neither PFS nor OFS use longer links as much as the baseline scheme does; hence, the effects of transmission range limitations are more severe on the baseline scheme. In consequence, the lifetimes of PFS and OFS normalized with the lifetime of the baseline scheme are larger for more constrained networks. Nevertheless, absolute lifetimes of all the schemes (baseline, OFS, and PFS) decrease as the constraints get more restrictive.

The lifetime gain that can be achieved by preferring OFS over PFS is significantly lower in disk topologies as compared to linear topologies, especially, when small number of proxies is used. As a result, the overhead and extra complexity multilevel proxy filtering brings may prohibit its use, especially, in disk topologies where the gain is less significant.

Under the OFS scheme, the optimal solution is attained by selecting all sensors as proxies. However, this may have disadvantages in terms of creating loops and especially delaying the flows. Then one can ask whether it is possible to attain a high level of lifetime (even optimal) without losing much in terms of delays and loops. Our analysis suggests that this is possible since we show that most of the advantages of the OFS scheme in terms of network lifetime can be attained by using relatively small number of proxies, which means less delay and a small number of loops. As a result, the decision maker can give the decision on the number of proxies considering all these advantages and disadvantages. On the other hand, for the PFS scheme, the maximum of the relative lifetime ratio corresponds to a small value for the number of proxies in the WSN. This solution is most likely to be selected by the decision maker.

Since receive energy becomes more dominant than transmission energy in high density networks in which inter-node distances are small, packet relaying becomes inefficient and not preferable. In such networks, the optimal lifetime is achieved when most sensor nodes send their data directly to the base station and multi-hop transmission rarely happens. In such networks, proxy filtering brings minimal gain as expected. Note that when the distance between a source and a destination is very low then the energy dissipation is mostly on the electronic circuitry (\(E_{\text{elec}}\)), which is equal to the reception energy dissipation (\(E_{\text{rec}}\)). If the source directly transmits to the base station the total energy dissipation of the network on one bit of data is only \(1 \times E_{\text{elec}}\) (base station energy dissipation is not taken into account). However, if the source relays its data via an intermediate node (e.g., a proxy node) then the total energy dissipation becomes \(3 \times E_{\text{elec}}\) (transmission by the source, reception by the proxy, and transmission by the proxy to the base station—each results in \(E_{\text{elec}}\) amount of energy dissipation).

On the other hand, our results also show that when the network area exceeds a threshold value and network becomes sparse enough, increasing the network area even further does not affect the normalized lifetimes obtained either with PFS or OFS significantly.

With respect to the number of nodes in the network, the scalability of PFS scheme is not as successful as the scalability of OFS scheme. As the network gets larger by the addition of more nodes the amount of generated data (consisting of real and dummy traffic) inflates. OFS successfully eliminates the dummy traffic not required for the event-unobservability without being relayed to unnecessarily farther distances unlike PFS, which cannot get rid of dummy traffic as much as the OFS scheme. We argue that in a linear topology number of nodes and hop distance are most important parameters to decide whether a single level filtering strategy is adequate or multiple filtering alternatives need to be explored in order to achieve a long-life unobservable WSN. We expect a similar trend also in disk topologies for large networks but as previously mentioned computational complexity of Algorithm 1 prevents us to increase further the number of nodes in a disk network.

6. Related and complementary work

Although contextual privacy is not a topic studied as extensively as content privacy in the context of wireless sensor networks, there is still considerable amount of previous work particularly in recent years. In our literature search, we notice that the terminology might be confusing because different authors use different terms for the same concepts. For example, “transactional confidentiality” [16] and “contextual privacy” [17] are essentially the same thing. Contextual...
privacy in wireless sensor networks is “concerned with protecting the context associated with the measurement and transmission of sensed data” [17].

Contextual information has many aspects. Location privacy of event source may be the most popular and studied aspect but certainly not the only one. For instance, in the generic monitoring application (panda-hunter game) introduced in [17], the hunter in the role of adversary may be interested in any information about the panda, not only its location. Therefore even the information regarding whether and when a panda is detected by the network should be protected from adversaries. On the extreme, it is argued that information of the frequency band used in wireless communication could be sensitive. This is because an attacker can use this information to figure out the hardware platform used and exploit the vulnerabilities of the software running on that platform [16].

A further classification for location privacy is also needed. Source Location Privacy deals with hiding the source location. Concealing the location of sink nodes may also be a desirable property since by deducing the geographic location of the base station, an attacker can conduct a physical attack to destroy the sink and render the whole network inoperable [18]. In addition, since the final destination for every event-triggered packet is the sink, its location may be a valuable piece of information to start tracking packets in the path towards the source event. Since protecting sink location privacy and carrier frequency information are more difficult problems than others, we prefer to use the term “event-unobservability” [7] to cover and concentrate on the contextual privacy aspects regarding when, whether and where an event is sensed.

Type and complexity of countermeasures for contextual privacy should take into account the capabilities of attackers. Most of the time, the assumption is that the attacker is an external entity who does not hold the cryptographic key to decrypt data packets. As a matter of fact, there is really not much to do to protect contextual privacy against insiders. Broadly, external attackers can be either a local or a global eavesdropper. While a local eavesdropper can detect transmission of messages within a certain range, a global eavesdropper can listen to the entire network. By rate monitoring and time correlation analysis, collecting communication in the network reveals sink location [18] and whether and when an event happens. Source location can be determined by triangulation using multiple antennas. Even with a single transceiver, an adversary can determine the distance of a sensor node transmitting at a fixed rate by calculating the message reception rate [16].

Starting from the sink location, a local eavesdropper can trace messages toward the message source. Privacy-aware routing strategies aim to maximize source location privacy by increasing trace-back time [19]. For instance, in phantom routing the delivery of every message experiences a directed walk phase to direct the message to a phantom source. In the second and final phase, the packet is sent from the phantom source to the sink either in a single path or by flooding [16].

Contextual privacy can also be enhanced with a transmit range limitation technique [20]. This novel technique is also against local adversaries who can hear only in a local area. This time, the local area does not cover the location of the sink and the source.

Contextual privacy cannot be protected against global eavesdroppers by new routing algorithms or by limiting the transmission range. To counteract this more challenging threat, as we have already mentioned all proposed solutions depend on inserting dummy traffic into the network to hide real event messages [4]. There is an information-theoretic guarantee that event-unobservability is preserved if each node in the network acts as a source and transmits fixed-length random-looking encrypted periodic messages regardless of whether there is a real data to send or not [5]. Alternatively, reducing number of fake sources was proposed to reduce energy consumption and provide lower levels of privacy [4]. Event-triggered fake source transmission can only protect source location privacy. On the other hand, “whether” and “when” aspects of contextual privacy can only be protected if total number of sources either they are fake or real is always fixed.

Similar to source location privacy, random areas of high communication activity can be helpful to deceive an adversary about the location of the sink [18]. To the best of our knowledge, there is no prior study that aims to provide unconditional security guarantees for hiding the sink location. This is an interesting and challenging future work.

Table 3 provides summarized information for the comparison of contextual privacy enhancing techniques discussed so far. Since periodic collection is the only but an inefficient countermeasure against global eavesdroppers, there are a number of improvements to this baseline scheme which are introduced in Table 4. Before discussing more sophisticated techniques, we should be aware that increasing collection period is a simple yet effective technique to minimize traffic and prolong network lifetime if it is applicable for the sensor network application and if an increase in event reporting delay is tolerable.

As we have already discussed, an efficient method to achieve longer network lifetimes is to combine periodic collection with filtering [5]. Proxy nodes can filter dummy messages intelligently by

### Table 3
Comparison of contextual privacy enhancing techniques.

<table>
<thead>
<tr>
<th>Where</th>
<th>Source location</th>
<th>Sink location</th>
<th>When/whether</th>
<th>Carrier freq.</th>
<th>Energy efficiency</th>
<th>Event delivery latency</th>
<th>Implementation complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Periodic collection [4]</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Low</td>
<td>Adjustable</td>
<td>Very low</td>
</tr>
<tr>
<td>Privacy-aware routing [17–19]</td>
<td>Local</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>High</td>
<td>High</td>
<td>High</td>
</tr>
<tr>
<td>Fake sources/sinks [4]</td>
<td>Probabilistic</td>
<td>Probabilistic</td>
<td>No,probabilistic</td>
<td>No</td>
<td>High</td>
<td>Low/high</td>
<td>High</td>
</tr>
<tr>
<td>Transmit range limitation [20]</td>
<td>Local</td>
<td>Local</td>
<td>Local</td>
<td>Local</td>
<td>High</td>
<td>Low</td>
<td>Low</td>
</tr>
</tbody>
</table>
sending a single aggregate packet in each period. This packet is again a dummy one if all incoming packets in the period are dummy ones. If one or more incoming packets carry real-event information, the aggregate packet is structured accordingly. The filtering technique can be implemented either in a single level or in multiple levels.

Periodic collection together with proxy filtering may bring two different kinds of event reporting delay. The first delay is due to periodic collection from real sources and equal to half of collection period on average if any additional delay minimization technique is not implemented. The second delay is caused by proxy filtering. The packet may be delayed further on each proxy on its way to the base station. Fitted Probabilistic Source Rate is a technique targeting the first source of the delay [7]. The basic idea is to send dummy traffic having an exponential distribution generated from a secret seed rather than at a constant rate and send real packets as fast as possible hiding in that exponential distribution.

There are paths than one from source nodes to the base station. These paths use different proxies as relay nodes and have different amount of delays considering the packet transmission schedule of each proxy. Fastest Path Discovery is an algorithm that determines the fastest of these paths and therefore minimizes the second cause of delay due to proxy filtering [13]. We argue that another proxy delay minimization technique can be built by fine-grained synchronization and scheduling of packet transmissions. Design of alternative techniques addressing proxy delays is an interesting open problem.

Up to now, we review previous work directly targeting contextual privacy problem in wireless sensor networks. We finish this section by presenting two broader research areas related to contextual privacy problem.

One of the well-known techniques to prolong lifetime of wireless sensor networks is data aggregation. For instance in a network composed of temperature reading sensors, aggregator nodes can aggregate the data coming from sensors into a single stream to reduce the total volume to be sent to the base station. However, the level of data reduction possible is strictly application dependent (e.g., is it appropriate to send only the arithmetic average of temperature readings?). While data aggregation has some similar features with proxy filtering discussed in this paper, there are subtle differences between LP formulations that can model these problems [21].

In the proxy filtering technique, there is one security problem which does not exist in the baseline scheme. If the sensor data is encrypted end-to-end between the source and the sink in a traditional way, then in order to distinguish real packets from dummy ones, proxy node should have a reach to the decryption key; which has security implications. The alternative is to use a homomorphic encryption scheme [22] by which the proxy nodes have the capability to perform certain algebraic operations on the data without decrypting it. We believe that homomorphic encryption is a valuable cryptographic tool for privacy protection purposes.

7. Conclusion

In this study we analyzed the energy dissipation and network lifetime characteristics of methods for preserving event-unobservability in wireless sensor networks through novel LP formulations. Hence, we introduced a systematic methodology of analyzing such mechanisms under widely accepted network models (e.g., lifetime definition, energy dissipation model, and network topologies) [8,9]. Therefore, both the LP framework and the analysis performed by using this framework are novel technical contributions to the literature. We are not aware of any existing work attempted such an analysis. Any service designed for wireless sensor networks (including security services) must adhere to the general expectations from wireless sensor networks, one of which is energy efficiency and network lifetime optimization. Hence, our study closes the gap between the provided service (proxy filtering service) and the performance metric (network lifetime) through the developed framework (LP model that captures both energy dissipation and proxy filtering).

We particularly investigated the effects of event-unobservability methods on the network lifetime which is defined as the time when the first sensor node runs out of energy. We considered a global adversary (an adversary that can listen to entire network traffic) model. We used an LP framework to characterize the network dynamics and energy dissipation trends in a practical yet idealized setting. In this LP framework, we modeled PFS (Proxy based Filtering Scheme) and TFS (Tree based Filtering Scheme) schemes proposed earlier for preserving unobservability against a global eavesdropper. We have proposed and modeled a new scheme called OFS (Optimal Filtering Scheme). We have shown the optimality of OFS scheme in terms of network lifetime both mathematically and through computational studies.

PFS scheme uses a single level proxy architecture, where data packets pass through a single proxy at most. Both ordinary nodes and proxy nodes always generate data at a constant rate. In case of a proxy node: if no data is received dummy packets are created, if multiple data packets received they are aggregated into a single packet, and all the received dummy packets are discarded. In TFS scheme, proxies are organized in a tree structure; proxies at higher levels in the tree aggregate the packets coming from lower level proxies. To maximize the overall network lifetime, in OFS scheme data flow can pass through multiple proxies organized as a general directed graph and no restrictions are imposed on the type and level of filtering possible in each proxy node. The results of the experimental study reveal that PFS is sufficient for small-size WSNs. As network size grows the superiority of OFS to PFS becomes more significant. As a rule of thumb hop distance is the most useful parameter for making decisions regarding number of required filtering levels.

We conclude our paper by calling attention to three open problems two of which were already mentioned. First of these problems is modeling and analyzing sink location privacy problem and derive corresponding lifetime bounds. A second promising future work is the design and implementation of new proxy delay minimization techniques providing more efficiency and/or less complexity. The last but not the least, in our research we have experienced that more efficient optimum proxy selection algorithms are highly desirable for large scale optimization problems in wireless sensor networks.

References


Kemal Bicakci is currently an associate professor at the Department of Computer Engineering, TOBB University of Economics and Technology. He has obtained his Ph.D. degree from Middle East Technical University, Ankara, Turkey in 2003. Between 2004 and 2006, he was a postdoc researcher in Vrije Universiteit Amsterdam working with Prof. Tanenbaum in EU FP6 project named SecurE-Justice. His previous research experience includes several NSF funded security projects in which he participated as a research assistant during his M.S. studies in University of Southern California, Los Angeles, USA.

Beginning with a BS graduation project in Hacettepe University, Kemal Bicakci now has over 15 years research experience on security and related topics. His research interests include information security, applied cryptography, wireless and sensor networks, health informatics, human-computer interaction and usability.

Hakan Gültekin is an Assistant Professor at TOBB University of Economics and Technology in Ankara, Turkey. He received his B.S., M.S. and Ph.D. degrees in Industrial Engineering from Bilkent University in Turkey. His research interests include scheduling, optimization modeling and exact and heuristic algorithm development especially for the problems arising in modern manufacturing systems and wireless sensor networks.

Ibrahim Ethem Bagci is a Teaching Assistant and pursuing an M.S. degree in Computer Engineering at the TOBB University of Economics and Technology, Ankara, Turkey. He received the B.S. degree in Computer Engineering in 2008 from the TOBB University of Economics and Technology, Ankara, Turkey. His research interests lie in the areas of information security and privacy, wireless sensor networks, linear programming and computer networks.

Bulent Tavli is an associate professor of electrical and electronics engineering at TOBB University of Economics and Technology, Ankara, Turkey. He received the B.S. and M.S. degrees in electrical and electronics engineering in 1996 from Middle East Technical University, Ankara, Turkey and in 1998 from Bilkent University, Ankara, Turkey. He received the M.S. and Ph.D. degrees in electrical and computer engineering in 2001 and 2005 from the University of Rochester, Rochester, NY, USA. His research interest areas are wireless communications and networking, ad hoc and sensor networks, signal and image processing, and linear programming.